

US-Bestseller
in deutscher Übersetzung



Laura Lemay
Denise Tyler

HTML 4

**Professioneller
Schritt für Schritt-Einstieg
ins Web-Publishing**

**Jedes Kapitel
mit Übungen**

**Aktuellste Infos zu XHTML,
DHTML, Stylesheets,
Multimedia und mehr**



Komplettes Buch
als HTML-Datei,
HTML-Editoren,
Web-Browser,
45-Tage-Testversion
von Frontpage 2000

SAMS



HTML

in 21 Tagen

Laura Lemay

Denise Tyler

Deutsche Übersetzung:

Marcus Linke

HTML

in 21 Tagen



Markt+Technik Verlag

Bitte beachten Sie: Der originalen Printversion liegt eine CD-ROM bei.
In der vorliegenden elektronischen Version ist die Lieferung einer CD-ROM nicht enthalten.
Alle Hinweise und alle Verweise auf die CD-ROM sind ungültig.

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Ein Titeldatensatz für diese Publikation ist bei
Der Deutschen Bibliothek erhältlich.

Die Informationen in diesem Produkt werden ohne Rücksicht auf einen
eventuellen Patentschutz veröffentlicht.
Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.
Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter
Sorgfalt vorgegangen.
Trotzdem können Fehler nicht vollständig ausgeschlossen werden.
Verlag, Herausgeber und Autoren können für fehlerhafte Angaben
und deren Folgen weder eine juristische Verantwortung noch
irgendeine Haftung übernehmen.
Für Verbesserungsvorschläge und Hinweise auf Fehler sind Verlag und
Herausgeber dankbar.

Autorisierte Übersetzung der amerikanischen Originalausgabe:
HTML in 21 Days © 2000 by Sams Publishing

Alle Rechte vorbehalten, auch die der fotomechanischen
Wiedergabe und der Speicherung in elektronischen Medien.
Die gewerbliche Nutzung der in diesem Produkt gezeigten
Modelle und Arbeiten ist nicht zulässig.

Fast alle Hardware- und Software-Bezeichnungen, die in diesem Buch
erwähnt werden, sind gleichzeitig auch eingetragene Warenzeichen
oder sollten als solche betrachtet werden.

Umwelthinweis:
Dieses Buch wurde auf chlorfrei gebleichtem Papier gedruckt.
Die Einschrumpffolie – zum Schutz vor Verschmutzung – ist aus
umweltverträglichem und recyclingfähigem PE-Material.

10 9 8 7 6 5 4 3 2 1

04 03 02 01 00

ISBN 3-8272-5784-0

© 2000 by Markt+Technik Verlag,
ein Imprint der Pearson Education Deutschland GmbH.
Martin-Kollar-Straße 10–12, D–81829 München/Germany
Alle Rechte vorbehalten
Übersetzung: Marcus Linke
Lektorat: Jürgen Bergmoser, jbergmoser@pearson.de
Herstellung: Claudia Bäurle, cbaeurl@pearson.de
Satz: reemers publishing services gmbh, Krefeld
Einbandgestaltung: ■■■
Druck und Verarbeitung: Bercker, Kvelaer
Printed in Germany

Inhaltsverzeichnis

Tag 1	Die Welt des World Wide Web	13
	Was ist das World Wide Web?	14
	Web-Browser	24
	Web-Server	28
	Uniform Resource Locator (URL)	29
	Zusammenfassung	30
	Workshop	30
Tag 2	Vorbereitungen	35
	Anatomie einer Web-Site	37
	Was wollen Sie im Web tun?	39
	Legen Sie Ihre Ziele fest	41
	Unterteilen Sie Ihre Inhalte in Hauptthemen.	42
	Ideen zur Organisation und Navigation.	44
	Ein Storyboard für Ihre Web-Site	54
	Zusammenfassung	57
	Workshop	58
Tag 3	Eine Einführung in HTML.	61
	Was HTML ist ... und was nicht	62
	Eine kurze Geschichte zu den HTML-Tags	66
	Mit XHTML 1.0 für die Zukunft gerüstet sein	68
	So sehen HTML-Dateien aus	69
	Übung 3.1: Erstellen Sie Ihre erste HTML-Seite	70
	Übung 3.2: Sehen Sie sich das Resultat an.	72
	Programme, die Ihnen beim Schreiben von HTML helfen	74
	Zusammenfassung	75
	Workshop	75



Tag 4	Beginnen wir mit den Grundlagen	79
	Strukturierung von HTML-Code	80
	Der Titel	83
	Überschriften	84
	Absätze	87
	Listen, Listen und nochmals Listen	88
	Kommentare	102
	Übung 4.1: Erzeugen eines richtigen HTML-Dokuments . . .	103
	Zusammenfassung	104
	Workshop	106
Tag 5	Alles über Verknüpfungen	111
	Alles über das <a>-Tag	112
	Verknüpfungen erzeugen	112
	Übung 5.1: Verbindung zweier Seiten	114
	Verknüpfen lokaler Dokumente: relative und absolute Pfad-Namen	118
	Verbindungen mit anderen Dokumenten im Web	122
	Übung 5.2: Verbinden Sie Ihre Caesar-Seiten mit dem Web . .	123
	Übung 5.3: Erstellen eines Verknüpfungs-Menüs	125
	Bestimmte Stellen in einem Dokument erreichen	128
	Übung 5.4: Abschnitte zwischen zwei Dokumenten verbinden	130
	Anatomie einer URL	136
	Verschiedene URL-Arten	139
	Zusammenfassung	144
	Workshop	145
Tag 6	Mehr HTML	151
	Zeichen-Gestaltung	152
	Vorformatierter Text	159
	Horizontale Linien	162
	Attribute der <hr />-Kennung	163
	Zeilenumbrüche	167
	Adressen	169
	Zitate	170
	Sonderzeichen	171

	Textausrichtung	175
	Schriftarten und -grade	177
	Das gefürchtete <blink>	180
	<nobr> und <wbr>	181
	Übung 6.1: Erstellen Sie eine richtige HTML-Seite	181
	Zusammenfassung	194
	Workshop	196
	Übungen	198
Tag 7	Die Verwendung von Abbildungen, Farben und Hintergründen	199
	Abbildungen im Web	200
	Interne Abbildungen in HTML: die -Kennung	202
	Eine Alternative zu Abbildungen hinzufügen	203
	Übung 7.1: Bilder hinzufügen	203
	Abbildungen und Text	207
	Den Abstand um die Abbildungen einstellen	215
	Abbildungen und Verknüpfungen	217
	Übung 7.2: Navigationssymbole	218
	Externe Abbildungen einsetzen	222
	Übung 7.3: Verbindungen mit externen GIF- und JPEG-Dateien	223
	Andere nette Tricks mit Bildern	226
	Verwendung von Farben	228
	Tipps für eine bessere Nutzung von Abbildungen	234
	Zusammenfassung	237
	Workshop	238
Tag 8	Animierte Grafiken erstellen	243
	Was versteht man unter einer Animation?	244
	Wann und wo Animation anwenden?	246
	Erstellen transparenter GIF-Dateien	247
	Das transparente GIF	248
	Programme, die Ihnen beim Erstellen animierter GIFs helfen	249
	Übung 8.1: Erstellen einer Animation	251
	Verringerung der Animationsgröße	253
	Übung 8.2: Eine Animation optimieren	255



	Werkzeuge, die Ihnen beim Erstellen Ihrer Bilder helfen	257
	Nützliche Software-Features	258
	Übung 8.3: Erstellen der Frames für die Ball-Animation. . . .	259
	Zusammenfassung	264
	Workshop	265
Tag 9	Imagemaps	267
	Was ist ein Abbildungsplan?	268
	Erzeugen Sie serverseitige Abbildungspläne	271
	Übung 9.1: Eine anklickbare Jukebox	280
	Erstellen clientseitiger Abbildungspläne	285
	Übung 9.2: Noch ein Besuch bei der anklickbaren Jukebox . .	288
	Erzeugen von Webseiten, die beide Typen von Abbildungsplänen unterstützen	290
	Zusammenfassung	291
	Workshop	292
Tag 10	XHTML und Style-Sheets	295
	Was ist XHTML und warum es verwenden?	296
	Entwurf von Style-Sheets	299
	Wissenswertes über Style-Sheets	301
	Besuchen wir wieder den »Bookworm Bookshop«	303
	Style-Sheet-Ansätze	306
	Übung 10.1: Erstellen und Einbinden eines externen Style Sheet	308
	Übung 10.2: Erstellen eines eingebetteten Style Sheet	311
	Was bedeutet Cascading?	314
	Die wesentlichen Style-Sheet-Eigenschaften und -Werte. . . .	314
	Übung 10.3: Randabstand und Padding auf einer Seite anwenden	317
	Übung 10.4: Hintergründe und Farben mit CSS auf Elemente angewandt	321
	Übung 10.5: Umrahmung von Elementen mit CSS.	323
	Eigenschaften zur Textausrichtung in CSS	328
	Übung 10.6: Anwendung von Schriftart- und Texteigenschaften mit CSS	335
	Workshop	341

Tag 11	Tabellen	345
	Eine Bemerkung zur Definition von Tabellen.	346
	Einfache Tabellen gestalten	347
	Tabellenkomponenten	347
	Übung 11.1: Erzeugen Sie eine einfache Tabelle.	353
	Die Ausrichtung von Tabellen und Zellen	357
	Ausrichtung von farbigen Tabellen und Zellen	364
	Ausrichtung des Tabelleninhalts	368
	Zellen, die sich über mehrere Zeilen oder Spalten erstrecken	373
	Übung 11.2. Eine Tabelle mit Kundendienst-Spezifikationen	376
	Tabellenverbesserungen für Fortgeschrittene	383
	Zusammenfassung	392
Tag 12	Frames und verknüpfte Fenster	399
	Frames und wie sie unterstützt werden	400
	Mit verknüpften Fenstern arbeiten	402
	Übung 12.1 Arbeiten mit Fenstern	402
	Frames anwenden	409
	Frame-Ränder verändern	416
	Komplexe Framesets gestalten	419
	Übung 12.2: Erstellen Sie den Seiteninhalt für Ihr Frameset.	419
	Übung 12.3: Reihen und Spalten kombinieren	425
	Übung 12.5: <frameset>-Kennungen ineinander verschachteln.	427
	Übung 12.5: Benannte Frames und Hyperlinks anwenden.	428
	Übung 12.6: Dokumente mit individuellen Frames verknüpfen	429
	Übung 12.7: Fügen Sie den NOFRAMES-Inhalt hinzu.	430
	Floating Frames	432
	Zusammenfassung	436
	Workshop	437
Tag 13	Multimedia: Einbinden von Sounds, Videos und mehr	441
	Präsentieren von Sound und Video	443
	Die altbewährte Methode: Linking	444
	Übung 13.1: Erstellen eines Multimedia-Familienalbums	446
	Einbetten von Sound-und Video-Dateien	454

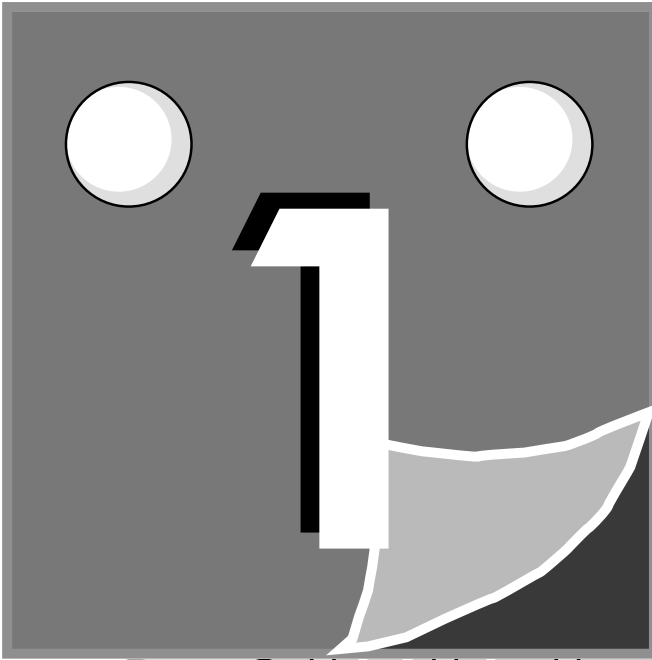


	Übung 13.2: Einbetten eines QuickTime-Movies	467
	Sound- und Videodateitypen	472
	Plugins und Player	474
	Zusammenfassung	479
	Workshop	480
Tag 14	Definieren von Formularen	483
	Verstehen von Formularen und ihrer Funktionen.	485
	Übung 14.1: Erstellen eines einfachen Formulars, das einen Namen und ein Passwort akzeptiert.	486
	Die wichtigsten Elemente eines Formulars	492
	Übung 14.2: Erstellen eines Formulars mit verschiedenen Eingabefeldern	503
	Hinzufügen von Extras	517
	Übung 14.3: Erstellen eines Formulars mit erweiterten Formularfeldern	521
	Planen Sie Ihre Formulare	526
	Zusammenfassung	526
	Workshop	528
Tag 15	Dynamic HTML	531
	Was genau ist denn Dynamic HTML?	533
	Willkommen in der Wirklichkeit: Browser-DHTML- Techniken im tatsächlichen Gebrauch	552
	Übung 15.1: Erstellen aufklappbarer Menüs	559
	Übung 15.2: Das Tic-Tac-Toe-Spiel	571
	Fahren wir mit Ihrer DHTML-Ausbildung fort	601
	Zusammenfassung	602
	Workshop	602
Tag 16	Schreiben und Gestalten von Webseiten: Tipps	605
	Die Anwendung von HTML-Erweiterungen	606
	Schreiben für Online-Leser	610
	Gestaltung und Seitenlayout	616
	Verknüpfungen anwenden	619
	Bilder verwenden	627
	Andere gute Angewohnheiten und Tipps	632

	Zusammenfassung	637
	Workshop	638
Tag 17	Beispiele für gutes und schlechtes Web-Design	643
	Beispiel eins: ein Firmenprofil	644
	Beispiel zwei: eine Multimedia-Enzyklopädie	652
	Beispiel drei: ein Einkaufskatalog	658
	Zusammenfassung	670
	Workshop	671
Tag 18	Willkommen in der Wirklichkeit: Web-Design für den tatsächlichen Gebrauch	675
	Wie sieht denn der tatsächliche Gebrauch überhaupt aus? . . .	676
	Wie gut kennt sich der Anwender aus?	678
	Erstellen Sie eine Liste mit Dateien, die der Anwender möglicherweise gebrauchen könnte	683
	Finden Sie die Präferenzen der Anwender heraus	684
	HTML 4.0 oder XHTML 1.0?	685
	HTML 4.0 und XHTML 1.0 Transitional	686
	HTML 4.0 und XHTML 1.0 Framesets	690
	Zusammenfassung	697
	Workshop	697
Tag 19	Alles online verfügbar machen	701
	Web-Server?	702
	Einen Server finden	705
	Organisieren und Installieren Ihrer HTML-Dateien	707
	Wie lautet meine URL?	713
	Testen, testen und nochmals testen	715
	Fehler finden	715
	Zusammenfassung	718
	Workshop	718
Tag 20	Lassen Sie die Leute wissen, dass Sie online sind	723
	Registrieren Sie Ihre Webseiten und werben Sie dafür	724
	Site-Indizes und Suchmaschinen	729
	Kündigen Sie Ihre Site über das Usenet an	737
	Web-Ringe	738



	Visitenkarten, Briefköpfe und Broschüren	740
	Finden Sie heraus, wer Ihre Webseiten betrachtet	740
	Zusammenfassung	742
Tag 21	Testen, Überarbeiten und Verwalten von Web-Sites	745
	Integritätstests	746
	Übung 21.1: Überprüfen einer Beispielseite	752
	Verwendungstests.	761
	Seiten hinzufügen und diese aktualisieren	762
	Zusammenfassung	765
	Workshop	766
	Stichwortverzeichnis	769



Die Welt des World Wide Web

**Woche
1**

Auch eine Reise über Tausende von Kilometern beginnt mit einem einzelnen Schritt, und hier befinden Sie sich nun in Kapitel 1, dem ersten Tag einer Reise, die Ihnen zeigen wird, wie die Hypertext Markup Language (HTML) verwendet wird, um Dokumente für das World Wide Web (WWW) zu produzieren.

In diesem Kapitel lernen Sie die folgenden Grundlagen kennen:

- ▶ Was das World Wide Web ist und weshalb es so unglaublich toll ist
- ▶ Web-Browser: Wie sie funktionieren sowie eine kleine Auswahl der wichtigsten Browser
- ▶ Was ein Web-Server ist und warum Sie einen brauchen
- ▶ Was ein Uniform Resource Locator (URL) ist

Wenn Sie bereits einige Zeit mit der Erkundung des Web verbracht haben, dürften Sie bereits mit einem Großteil dieses Kapitels vertraut sein, vielleicht sogar mit allem. Falls dem so ist, gehen Sie auf alle Fälle weiter zum nächsten Kapitel, wo Sie eine Übersicht anderer allgemeiner Dinge finden, die Sie berücksichtigen sollten, wenn Sie Ihre eigenen Webseiten gestalten und organisieren wollen.

Was ist das World Wide Web?

Ich habe einen Freund, der die Dinge gerne durch eine Fülle bedeutsamer Wörter beschreibt, die in einer langen Kette miteinander verbunden sind, sodass man einige Minuten braucht, um herauszubekommen, was er gerade gesagt hat.

An seiner Stelle würde ich das World Wide Web wie folgt beschreiben: als ein globales, interaktives, dynamisches, plattformübergreifendes, verteiltes, grafisches Hypertext-Informationssystem, das im Internet betrieben wird. Ui! Auch wenn Sie jedes einzelne dieser Wörter verstehen und wie sie zusammengehören, ergibt das wohl nicht viel Sinn.

So, nun wollen wir jedes einzelne dieser Wörter unter dem Aspekt betrachten, dass das Web als Publikationsmedium benutzt werden soll.

Das Web ist ein Hypertext-Informationssystem

Wenn Sie jemals irgendein Online-Hilfesystem benutzt haben, dann sind Sie bereits mit Hypertext, dem grundlegenden Konzept des World Wide Web, vertraut.

Die Idee hinter Hypertext ist, dass Sie, anstatt den Text in einer starren, fortlaufenden Form (wie in einem Buch) zu lesen, einfach von einem Punkt an einen anderen sprin-

gen können, dort weitere Informationen finden, zurückgehen, zu anderen Themen springen und so durch den Text navigieren, abhängig davon, was Sie gerade interessiert.



Hypertext ermöglicht es Ihnen, Text und visuelle Informationen auf nicht lineare Weise zu lesen und durch sie zu navigieren, indem sie das auswählen, was Sie als nächstes wissen wollen.

Online-Hilfe-Systeme oder Help-Stacks wie Microsoft *Windows' Help*, Sun Microsystems' *Answerbook* oder *HyperCard* auf dem Macintosh verwenden Hypertext, um Informationen zu präsentieren. Um weitere Informationen zu einem Thema zu erhalten, klicken Sie einfach auf das Thema. Es erscheint ein neuer Bildschirm mit weiteren Informationen (oder ein zusätzliches Fenster oder eine Dialogbox, jedenfalls macht das Programm eine Verzweigung). Vielleicht gibt es hier Verknüpfungen, die Sie zu anderen Themen führen, und auch hier wiederum Verknüpfungen, die Sie noch weiter von Ihrem ursprünglichen Thema wegführen. Abbildung 1.1 stellt die Arbeitsweise eines solchen Systems in einem einfachen Diagramm dar.

Nun stellen Sie sich vor, dass Ihr Online-Hilfe-System verknüpft ist mit dem Online-Hilfe-System eines Ihrer anderen Programme. Als Beispiel: Die Hilfe Ihres Zeichenprogramms ist verknüpft mit der Hilfe Ihres Textprogramms. Die Hilfe Ihres Textprogramms wiederum ist verknüpft mit einem Wörterbuch, wo Sie jeden Begriff nachschlagen können, den Sie nicht verstehen. Das Wörterbuch ist verbunden mit einem globalen Index von Zeitschriftenartikeln, der es Ihnen ermöglicht, die jüngsten Informationen zu dem Thema zu erhalten; der Artikelindex wiederum ist verknüpft mit Informationen über die Autoren jener Artikel und ein paar Bildern von deren Kindern (siehe Abbildung 1.2).

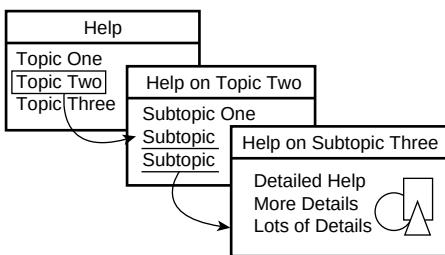


Abbildung 1.1:
Ein einfaches Online-Hilfe-System

Wenn nun aber die Informationen nicht viel Plattenplatz benötigten und frei verfügbar wären und Sie sie einigermaßen schnell und zu jeder Zeit bekommen könntent, dann wäre die Sache wohl schon interessanter. Tatsächlich könnte das Informationssystem schließlich viel interessanter werden als die Software, die Sie zunächst gekauft hatten.

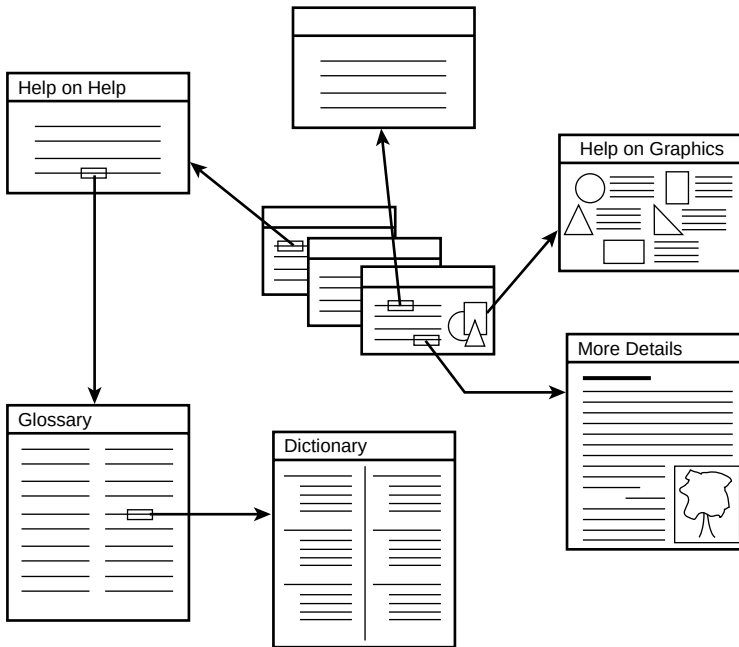


Abbildung 1.2:
Ein komplexeres
Online-Hilfe-System

Dies ist genau das, was das World Wide Web ausmacht: Mehr Informationen als Sie in einem Leben verarbeiten können, auf vielerlei Weise miteinander verbunden, sind Ihnen da draußen im Netz zugänglich, wann immer Sie mögen. Das Netz ist groß und tief, und man kann sich leicht darin verlieren. Doch es birgt auch eine Menge Spaß.



Die Web-Technologien bieten viele Möglichkeiten in Bezug auf die Präsentation und Organisation von Information. Die Geschäftswelt hat natürlich Notiz davon genommen und viele große und mittlere Unternehmen wie auch Organisationen profitieren von der Web-Technologie: Web-Technologien werden zum Beispiel in den Bereichen Projektmanagement, Materialbestellungen und Veröffentlichung von Firmeninformationen im Netz eingesetzt. Die Firmendokumente befinden sich auf einem privaten, sicheren Web-Server in einem Intranet. So können Firmen die Vorteile der Technologien nutzen, die das World Wide Web anbietet, während Informationen firmenintern bleiben.

Das Web ist grafisch aufgebaut und leicht zu durchqueren

Früher musste man im Internet fast ausschließlich mit Text arbeiten. Sie mussten getippte Befehle eingeben und *obskure Werkzeuge* verwenden, um zu den verschiedenen Angeboten im Internet zu gelangen. Obwohl schon viele interessante und auf-

regende Informationen im Web zu finden waren, waren sie oft nicht besonders attraktiv anzusehen. Dann erschien der erste Grafik-Web-Browser »Mosaic«, der nun das Anzeigen von Text und Grafiken in Farbe auf demselben Bildschirm ermöglichte, was wohl das Beste am ganzen Web ist. Wohl gemerkt, wurde das Web dadurch auch so bekannt und beliebt. Heutzutage ermöglichen Web-Browser das Einbinden von Grafiken, Sound und Video in Texten, sowie immer fortschrittlicheres Arbeiten mit Multimedia und eingebetteten Anwendungen.



Ein *Browser* wird benutzt, um durch Webseiten und andere Informationen auf dem World Wide Web zu navigieren und sie zu betrachten.

Was noch viel wichtiger ist: die Navigation mit Interfaces ist bei all diesen Anwendungsmöglichkeiten ganz einfach. Springen Sie einfach ein Link nach dem anderen, eine Seite nach der anderen in Sites und Servern an.



Wenn das Web nun so viel mehr enthält als nur Text, warum bleibe ich dabei, das Web ein Hypertext-System zu nennen? Nun ja, wenn Sie technisch absolut korrekt sein wollen, ist das Web kein Hypertext-System – es ist ein HyperMedia-System. Andererseits jedoch könnte man argumentieren, dass das Web als textbasiertes System begonnen habe und sein Inhalt überwiegend noch immer textlastig sei, mit einigen zur Hervorhebung hinzugefügten Medien. Ich bevorzuge den Begriff *Hypertext*.

Das Web ist plattformübergreifend

Wenn Sie aufs Internet zugreifen können, können Sie auf das World Wide Web zugreifen. Es ist egal, ob Sie einen billigen PC, einen schicken, teuren, grafikfähigen Arbeitsplatzrechner benutzen oder einen Großrechner, der viele Millionen Dollar kostet. Sie können eine einfache Modemverbindung einsetzen, einen kleinen 14-Zoll-Schwarzweiß-Monitor oder auch ein vollfarbiges super-Gamma-korrigiertes grafisch beschleunigtes 21-Zoll-Display-System. Wenn Sie meinen, die Windows-Darstellung sehe besser aus als die von Macintosh oder umgekehrt (oder ob Sie glauben, sowohl Windows- als auch Mac-Nutzer seien Banausen), dann ist das ebenfalls egal. Das World Wide Web ist nicht beschränkt auf eine bestimmte Maschine oder auf eine Firma. Das Web ist vollkommen plattformübergreifend.



Plattformübergreifend bedeutet, dass Sie mit jeder beliebigen Computer-Hardware auf Web-Informationen zugreifen, und jedes beliebige zu dieser Hardware passende Betriebssystem und Display verwenden können.



An der Vorstellung, dass das Web plattformübergreifend ist – was es eigentlich sein sollte – können nur Puristen festhalten. Die Wirklichkeit ändert sich allerdings sehr schnell. Mit der Einführung von neuen Möglichkeiten, Technologien und Medienarten verliert das Web einiges vom plattformübergreifenden Ansatz. Sobald Web-Autoren diese neuen Features nutzen, beschränken sie willentlich das potentielle Publikum für den Inhalt ihrer Sites. So kann zum Beispiel eine Site, die um ein Java-Programm herum aufgebaut ist, im Wesentlichen nicht im Nur-Text-Modus verwendet werden. Ganz ähnlich verhält es sich mit bestimmten Erweiterungen (Plug-Ins) für den Netscape Navigator, die nur für eine Plattform erhältlich sind (entweder nur für Windows, nur für Mac oder nur für Unix). Durch die Verwendung einer dieser Erweiterungen wird dieser Bereich Ihrer Site unzugänglich für Anwender, die entweder einen ganz anderen Browser verwenden oder aber den Netscape Browser auf der falschen Plattform.

Sie greifen auf das Web mit Hilfe einer Applikation zu, die *Browser* genannt wird, wie zum Beispiel den Netscape Navigator oder den Microsoft Internet Explorer. Es gibt eine Fülle von Browsern für die meisten beliebten Plattformen. Wenn Sie erst einmal einen Browser und eine Verbindung zum Internet hergestellt haben, haben Sie es geschafft. Sie sind im Web. (Ich erkläre später in diesem Kapitel näher, was ein Browser tatsächlich macht.)

Das Web ist auf zahlreiche Orte verteilt

Informationen benötigen eine ziemliche Menge Platz, besonders wenn man Multimedia-Fähigkeiten wie Bilder, Klänge und Video darin aufnimmt. Um all die Informationen zu speichern, die das Web anbietet, würden Sie eine unermessliche Menge Plattenplatz benötigen, und die Verwaltung wäre nahezu unmöglich.

Stellen Sie sich vor, Sie wären daran interessiert, mehr Informationen über Alpaka-Schafe herauszufinden, aber Sie würden aufgefordert, die CD-ROM Nr. 456 ALP bis ALR in Ihr Laufwerk einzulegen, wenn Sie in Ihrer Online-Enzyklopädie nachschlagen. Es könnte eine Weile dauern, bis Sie die CD gefunden hätten!

Das Web kann so viele Informationen erfolgreich anbieten, weil die Informationen global verteilt sind, über Tausende verschiedener Systeme hinweg, von denen jedes den Platz für eben jene Informationen beisteuert, die es selbst publiziert. Sie, als Nutzer dieser Informationen, bewegen sich zu jenem System, um die Informationen anzusehen, und wenn Sie damit fertig sind, gehen Sie irgendwo anders hin, und Ihr System »gewinnt« gewissermaßen den Plattenplatz, den die Original-Informationen benötigten. Sie müssen nichts installieren und keine Disketten wechseln. Sie müssen mit Ihrem Browser lediglich zu den entsprechenden Informationen navigieren.



Eine *Web-Site* ist ein Ort im Web, der irgendwelche Informationen veröffentlicht. Wenn Sie sich eine Webseite anschauen, dann verbindet sich Ihr Browser mit dieser Web-Site, um diese Informationen zu bekommen.

Jede Website und jede Informationseinheit auf dieser Site hat eine eindeutige Adresse. Diese Adresse wird Uniform Resource Locator, oder URL, genannt. Wenn jemand Sie bittet, eine Site unter <http://www.coolsite.com/> zu besuchen, dann hat diese Person Ihnen die URL angegeben. Sie können Ihren Browser einsetzen, um die URL einzugeben (mit dem Befehl **ÖFFNEN**, manchmal auch **URL ÖFFNEN** oder **GEHE ZU** genannt, oder Sie können ihn einfach kopieren und einfügen).



Ein *Uniform Resource Locator (URL)* ist ein hinweis zu einem spezifischen Teil von Information auf dem Internet.



URL kann man unterschiedlich aussprechen: indem man buchstabiert (»U_R_L«) oder indem man es als ein Wort ausspricht, »url«, ganz wie Sie es vorziehen.

Sie werden später in diesem Kapitel noch mehr über URLs erfahren.

Das Web ist dynamisch

Da die Informationen im Web auf dem System abgelegt werden, das sie publiziert hat, können die Leute, die sie veröffentlicht haben, die Informationen jederzeit sehr schnell aktualisieren.

Wenn Sie diese Informationen durchgehen, brauchen Sie keine neue Version des Hilfe-Systems zu installieren, ein anderes Buch (»2. Auflage«) zu kaufen oder den technischen Support anzurufen, um aktualisierte Informationen zu erlangen. Rufen Sie einfach Ihren Browser auf, und sehen Sie nach, was es dort gibt.

Wenn Sie Informationen im Web veröffentlichen, können Sie sicherstellen, dass die Informationen, die Sie anbieten, stets aktuell sind. Sie müssen nicht viel Zeit damit verbringen, Neufassungen Ihrer Dokumente als Ergänzungslieferungen oder Änderungsmittelungen zu verbreiten. Es gibt keine Materialkosten. Sie müssen keine Angebote zu Auflagenhöhen oder Druckqualitäten einholen. Farbe ist kostenlos. Und Sie erhalten keine Anrufe von unglücklichen Kunden, die eine Fassung des Buchs haben, die schon vor vier Jahren überholt war.

Sehen Sie sich zum Beispiel die Entwicklung des Web-Server mit dem Namen Apache an. Apache wird von einer Kerngruppe von Freiwilligen entwickelt und getestet, hat viele der Merkmale der größeren kommerziellen Server, und seine Benutzung ist kos-

tenlos. Die Apache Web-Site unter <http://www.apache.org/> ist die zentrale Stelle für Informationen über die Apache-Software, Dokumentation und die Server-Software selbst (Abbildung 1.3 zeigt ihre Homepage). Da die Site jederzeit aktualisiert werden kann, können Neufassungen schnell und einfach verteilt werden. Änderungen und ausgeübte Probleme der Dokumentation, die online vorliegt, können direkt an den Dateien vorgenommen werden. Und neue Informationen und Nachrichten können fast sofort veröffentlicht werden.



Abbildung 1.3:
Die Apache Web-Site



Die Bilder in diesem Buch sind normalerweise einem Windows Browser entnommen (meistens Microsoft Internet Explorer). Dafür gibt es nur einen Grund: ich habe dieses Buch hauptsächlich auf einem Windows PC geschrieben. Wenn Sie ein Macintosh oder Unix System verwenden, fühlen Sie sich bitte nicht ausgeschlossen. Wie ich vorhin schon erwähnt habe, ist das Phantastische am Web, dass Sie, egal welche Plattform Sie verwenden, überall die gleiche Information sehen können. Ignorieren Sie also die Schaltflächen und Fensterleisten und konzentrieren Sie sich auf den Fensterinhalt.

Für einige Sites ist es wichtig, dass man sie jederzeit und vor allem schnell aktualisieren kann. Die Abbildung 1.4 zeigt die Homepage der *Nando-Times*, eine Online-Zeitung, die 24 Stunden am Tag aktualisiert wird, um die neuesten Nachrichten noch zum Zeitpunkt ihres Geschehens zu reflektieren. Da die Site immer aktiv und erreichbar

bar ist, hat sie eine Unmittelbarkeit, die weder von gedruckten Zeitungen noch von den meisten Fernsehnachrichten nachvollzogen werden kann. Besuchen Sie *The Nando Times* unter <http://www.nando.net/nt/nando.cgi>.



Abbildung 1.4:
Die Nando Times

Web-Browser können auf viele Formen von Internet-Informationen zugreifen

Wenn Sie bereits eines der unzähligen Bücher über die Verwendung des Internet gelesen haben, dann wissen Sie sicherlich über Dutzende von unterschiedlichen Weisen Bescheid, wie man im Netz an Informationen gelangen kann: FTP, Gopher, Usenet-News, WAIS-Datenbanken, Telnet und E-Mail. Bevor das Web so populär wurde, wie es jetzt ist, musste man, um an diese unterschiedlichen Arten von Informationen zu kommen, für jede einzelne ein anderes Werkzeug benutzen, von denen jedes installiert werden musste und andere Befehle verwendete. Obwohl alle diese Auswahlmöglichkeiten einen wunderbaren Markt für Bücher mit Gebrauchsanweisungen fürs Internet darstellten, war es nicht einfach, sie anzuwenden.

Das Web macht damit Schluss. Obwohl das Web selbst sein eigenes Informationssystem mit seinem eigenen Internet-Protokoll darstellt (HTTP, das Hypertext Transfer Protocol), können Web-Browser auch Dateien von anderen Internet-Diensten lesen. Und, was noch besser ist, Sie können Verknüpfungen zu Informationen auf diesen Systemen genauso herstellen, wie Sie sie zu Informationen auf Webseiten herstellen. Es ist alles nahtlos miteinander verbunden und über eine einzige Applikation verfügbar.

Um mit Ihrem Browser zu unterschiedlichen Informationen im Internet zu springen, benutzen Sie verschiedene Arten von URLs. Die meisten URLs fangen mit `http:` an, was auf eine Datei auf einer tatsächlichen Web-Site hinweist. Um FTP aus dem Web heraus zu benutzen, verwenden Sie eine URL, die ungefähr so aussieht: `ftp://name_des_systems/verzeichnis/dateiname`

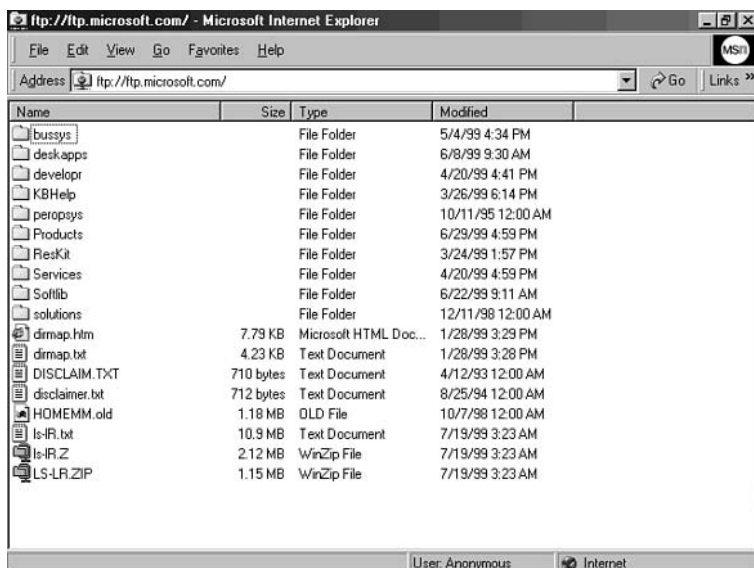


Abbildung 1.5:
Eine Liste mit
Dateien und Ver-
zeichnissen, die
auf der FTP-Site
von Microsoft
erhältlich ist.

Gopher Server sind heutzutage nicht mehr so gängig, wie sie es vor ein paar Jahren einmal waren. Wenn Sie aber Zugriff zu Gopher-Informationen von einem Web-Browser benötigen, verwenden Sie eine URL, die in etwa so aussieht: `gopher://name_of_gopher_server/`.

Sie werden noch mehr über die unterschiedlichen Arten von URLs in Kapitel 4 lesen.

Das Web ist interaktiv

Interaktionsfähigkeit ist die Fähigkeit, mit dem Web-Server »zu sprechen«. Traditionellere Medien wie Fernsehen sind nicht wirklich interaktiv; Sie sitzen nur da und sehen sich die Shows an, die für Sie abgespielt werden. Sie haben nicht viel Kontrolle darüber, was Sie zu sehen bekommen, sondern können höchstens einen anderen Kanal wählen. Fügen Sie WebTV-Möglichkeiten hinzu, und Fernsehen wird zu einer ganz neuen Erfahrung.

Das Web ist von Natur aus interaktiv; schon die Art, wie Sie eine Verbindung auswählen und andere Seiten mit Informationen bekommen, ist eine Form von Interaktivität. Zusätzlich jedoch zu dieser einfachen Interaktivität ermöglicht Ihnen das Web, sich mit dem Herausgeber der Seiten und mit anderen Lesern der Seiten zu verständigen.

Es können zum Beispiel Seiten gestaltet werden, die interaktive Formulare enthalten, die von den Lesern ausgefüllt werden können. Formulare können Texteingabefelder enthalten oder auch »Options-Felder« (sog. *radio buttons*) oder einfache Menüs mit Wahlmöglichkeiten. Wenn das Formular »eingesandt« wird, wird die von Ihnen eingetippte Information zum ursprünglichen Server zurückgeschickt.

Die Abbildung 1.6 zeigt das Beispiel des Surrealist Census-Formulares (Sie werden dieses Formular etwas später im Buch erstellen):

The image shows a screenshot of a web browser window titled "The Surrealist Census - Microsoft Internet Explorer". The browser's menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". The address bar shows "Links" and "MSN". The main content area of the browser displays a web page titled "The Surrealist Census". The page text reads: "Welcome to the Surrealist Census. Please fill out the following form to the best of your abilities." followed by "Use **Submit** To submit your results." Below this is a horizontal line. The form fields include: "Name:" followed by a text input box; "Sex:" followed by three radio button options: "Male", "Female", and "Null"; and "Contains (Select all that Apply):" followed by a list of checkboxes: "Vitreous Humor", "Fish", "Propylene Glycol", "SVGA Support", "Angst", and "Catalytic Converter". The browser's status bar at the bottom shows "Done" and "My Computer".

Abbildung 1.6:
Das Surrealist
Census-
Formular

Als Herausgeber dieser Informationen auf dem Web können Sie diese Formulare für viele verschiedene Zwecke einsetzen, z.B.:

- ▶ Um ein Feedback über Ihre Seiten zu bekommen.
- ▶ Um Informationen über Ihre Leser zu bekommen (Übersichts-, Wahl-, demografische oder andere Arten von Daten).
- ▶ Um online Bestellformulare für Produkte oder Dienste anzubieten, die auf dem Web verfügbar sind.

- Um Gästebücher oder Konferenz-Systeme anzubieten, die es Ihren Lesern ermöglichen, ihre eigenen Informationen auf Ihren Seiten einzugeben. Diese Arten von Systemen erlauben es Ihren Lesern, sich nicht nur mit Ihnen, sondern auch mit anderen Lesern Ihrer Seiten zu verständigen.

Zuzüglich zu Formularen, die einige der populärsten Arten der Interaktivität im Web bereitstellen, bieten fortgeschrittene Merkmale der Web-Entwicklung sogar noch mehr Interaktivität. So ermöglichen es Technologien wie Java und Shockwave, dass Sie ganze Programme und Spiele in Ihre Webseiten integrieren. Sie können Software im Web laufen lassen, die es Ihren Lesern ermöglicht, sich in Echtzeit miteinander zu unterhalten (sog. chat-sessions). Und Entwicklungen in 3-D-Welten ermöglichen es Ihnen und Ihren Lesern, das Web so zu erforschen, als ob Sie durch dreidimensionale Räume wanderten und andere Leute trafen. Das Web entwickelt sich zunehmend zu einem Medium, durch das man Leute in der ganzen Welt erreichen und sich mit ihnen verständigen kann, und ist weniger ein Medium für Leute, die passiv dasitzen und Informationen verdauen (und so zu »Net-Säcken« werden).

Web-Browser

Um auf das World Wide Web zuzugreifen, benutzen Sie etwas, das *Browser* genannt wird. Browser werden bisweilen auch *Clients* genannt, weil sie Informationen von einem *Server* beziehen.

Es gibt eine große Auswahl von Web-Browsern für fast jede Plattform sowohl für Systeme, die auf einer grafischen Benutzeroberfläche basieren (Mac, Windows), als auch für rein textorientierte Systeme, mit denen Unix-Verbindungen angewählt werden. Die meisten Browser sind Freeware (also kostenlos) oder Shareware (vor dem Kauf auszuprobieren) oder haben eine großzügige Art der Lizenzvergabe (Netscape erlaubt es Ihnen, ihren Browser für eine Weile auszuprobieren, danach wird von Ihnen erwartet, dass Sie ihn kaufen). Obwohl diese beiden Browser die Hauptkonkurrenten auf dem Markt sind, sind sie trotzdem nicht die einzigen Browser im Web. Alles, was Sie normalerweise tun müssen, um einen Browser zu bekommen, ist, ihn aus dem Netz herunterzuladen (obwohl das Downloaden in der Regel einfacher ist, wenn man bereits einen Browser hat – eine Art Henne-Ei-Situation).

Wenn Sie Ihre Internet-Verbindung durch einen kommerziellen Online-Service wie America Online oder CompuServe herstellen, haben Sie vielleicht mehrere Browser zur Verfügung, unter denen Sie auswählen können; probieren Sie einfach ein paar aus und finden Sie selber heraus, welcher Ihnen am besten gefällt.

Die gängigsten Browser für das World Wide Web sind Microsoft Internet Explorer (manchmal auch nur Internet Explorer oder »IE« genannt) und Netscape Navigator (oft nur mit Netscape bezeichnet). Obwohl Netscape den Löwenanteil am Markt hat, ist es

nicht der einzige Browser im Web. Dies wird später ein wichtiger Punkt werden, wenn Sie lernen, Webseiten zu entwerfen, und die unterschiedlichen Fähigkeiten der unterschiedlichen Browser herausfinden. Wenn Sie davon ausgehen, dass Internet Explorer und Netscape die einzigen Browser auf dem Web sind, und Ihre Seiten dementsprechend entwerfen, werden Sie das Publikum einschränken, das Sie mit den Informationen erreichen können, die Sie präsentieren wollen.



Entwickeln Sie Ihr Angebot für einen speziellen Browser, wie zum Beispiel Netscape Navigator oder Internet Explorer dann, wenn Sie ein ganz bestimmtes Publikum ansprechen möchten, das die entsprechende Browser-Software verwendet. Für Unternehmen, die mit Intranets arbeiten, ist dies eine gängige Methode. In diesem Fall kann man davon ausgehen, dass alle Benutzer desselben Unternehmens den gleichen Browser verwenden. Es besteht die Möglichkeit, die Web-Komponente des Intranets so zu gestalten, dass die speziellen Fähigkeiten des entsprechenden Browsers genutzt werden können.

Was macht ein Browser?

Ein Browser hat zweierlei Aufgaben: Gibt man ihm einen Verweis auf eine Information im Netz (die URL), muss er in der Lage sein, auf diese Information zuzugreifen oder in Abhängigkeit vom Inhalt dieses Verweises damit irgendwie umzugehen. Das bedeutet, dass er unter Verwendung des HTTP-Protokolls mit dem Server kommunizieren muss, um Hypertext-Web-Dokumente anzufordern. Da das Web auch Informationen verwalten kann, die auf FTP- und Gopher-Servern abgelegt sind, in News-Postings, E-Mails und so weiter, muss der Browser ebenso die Sprache der dazu gehörenden Werkzeuge sprechen.

Was der Browser jedoch am häufigsten macht, ist Web-Dokumente zu formatieren und abzubilden. Jede »Seite«, die Sie aus dem Web laden, ist ein einzelnes Dokument, geschrieben in einer Sprache namens »HTML« (*Hypertext Markup Language*), das den Text des Dokuments enthält, seine Struktur, jegliche Verknüpfungen mit anderen Dokumenten sowie Abbildungen und andere Medien. (Sie werden HTML an den Tagen 3 bis 6 erlernen, da Sie darüber Bescheid wissen müssen, bevor Sie Ihre eigenen Webseiten schreiben können). Verschiedenartige Browser werden dieselbe Datei unterschiedlich formatieren und darstellen, was von den Fähigkeiten des Systems und den Voreingabe-Optionen für das Layout des Browser abhängt.

Dokumente vom Web zu beziehen und sie dann für Ihr System zu formatieren, sind die zwei Kernbestandteile der Funktionalität Ihres Browsers. Sie können jedoch, je nachdem welcher Browser mit welchen Fähigkeiten Ihnen zur Verfügung steht, auch dazu in der Lage sein, Multimedia-Dateien abzuspielen, Java-Applets anzuschauen und mit ihnen zu interagieren, Ihre Post zu lesen oder andere erweiterte Funktionen zu benutzen, die ein bestimmter Browser anbietet.

Eine Übersicht über die gängigsten Browser

Dieser Abschnitt beschreibt ein paar der am meisten verbreiteten Browser, die im Web, während dieses Buch geschrieben wurde, eingesetzt werden. Dies sind keineswegs alle verfügbaren Browser, und wenn der Browser, den Sie benutzen, nicht dabei ist, glauben Sie nicht, nun einen von diesen verwenden zu müssen. Welchen Browser auch immer Sie haben, er ist in Ordnung, solange er tut was Sie wollen.

Die Browser, die in diesem Abschnitt vorgestellt werden, können nur dann verwendet werden, wenn Sie eine direkte Internet-Verbindung oder eine einwählbare SLIP- oder PPP-Internet-Verbindung haben. Wie Sie Ihre Maschine mit dem Internet verbinden, geht über den Rahmen dieses Buchs hinaus, doch es gibt genügend Bücher, die Ihnen dabei helfen.

Wenn Ihre Verbindung zum Internet über einen kommerziellen Online-Service (wie AOL, CompuServe oder MSN) hergestellt wird, dann haben Sie möglicherweise die Wahl zwischen verschiedenen Browsern, einschließlich denen in diesem Abschnitt und denjenigen, die Ihr Anbieter liefert.

Wenn die einzige Verbindung, die Sie zum Internet haben, die textbasierte Einwahl in Unix (oder ein anderes System) ist, so sind Sie darauf beschränkt, textbasierte Browser wie Lynx zu benutzen. Sie werden nicht in der Lage sein, Dokumente in Farbe oder mit Grafik online anzusehen (obwohl Sie sie gewöhnlich auf Ihr System herunterladen und sie dort betrachten können).

Microsoft Internet Explorer

Der Browser von Microsoft namens Internet Explorer, der meist einfach Explorer genannt wird, läuft unter Windows 3.1, Windows 95/98, Windows NT, Macintosh und UNIX. Er kann von der Microsoft Web-Site kostenlos heruntergeladen werden (<http://www.microsoft.com/ie/>). Man braucht keine Lizenzgebühr zu bezahlen. Wenn Sie Windows 98 verwenden, dann ist der Internet Explorer 4 schon im System integriert. Sie können, wenn Sie möchten, den Netscape Navigator installieren und verwenden, aber wenn Sie nur den Internet Explorer verwenden wollen, brauchen Sie nicht mehr zu tun.

Bis jetzt war Microsoft der einzige Entwickler eines Browsers, der es geschafft hat, mit der Entwicklungsgeschwindigkeit von Netscape Schritt zu halten, wobei Microsoft viele der Merkmale des Netscape Navigator unterstützt und einige eigene hinzufügt. Zusätzlich dazu hat Microsoft einige bemerkenswerte Vereinbarungen mit kommerziellen Online-Diensten getroffen, was bedeutet, dass sein Anteil am Browser-Markt beträchtlich gestiegen ist. Dies ist eine direkte Herausforderung an Netscape in Bezug auf die Kontrolle des Browser-Marktes. Obwohl eine große Anzahl verschiedener Browser verfügbar ist, werden im Wesentlichen nur zwei verwendet: der von Netscape und der von Microsoft.

Für mehr Informationen über alle Versionen des Internet Explorer, besuchen Sie die Homepage unter <http://microsoft.com/windows/ie/default.htm>. Abbildung 1.7 zeigt den Internet Explorer, der gerade in Windows 98 verwendet wird.



Abbildung 1.7:
Microsoft Internet Explorer

Wie beim Netscape Navigator, arbeitet Microsoft sehr eifrig an der Version 6 des Browsers. Dies hängt mit der Entwicklung von Windows 2000 zusammen. Dieser Browser verspricht eine schnellere und bessere Integration als alle vorhergehenden Versionen.



Mit jeder neuen Browserversion haben Microsoft und Netscape versucht, modernere und schickere Features zu integrieren. Die aktuellen Versionen sind keine Ausnahme. Wie Sie an Tag 10 sehen werden, enthalten diese neuen Features Style Sheets, um genaueste Kontrollen der erscheinenden Dokumente zu ermöglichen. An Tag 15, werden Sie mehr über Dynamic HTML lernen, die alles, von der präzisen Layout-Kontrolle bis hin zum verbesserten Scripting, auf HTML-Seiten umfasst.

Netscape Navigator

Der heute mit Abstand beliebteste Browser auf dem Web ist der Netscape Navigator von der Netscape Communications Corporation. Der einfachste Weg, den Netscape Navigator zu erhalten, ist es sich den Netscape Communicator zu beschaffen. Neben dem Navigator enthält der Communicator noch ein E-Mail-Programm, einen News-

reader für den Zugriff auf Newsgroups, den Composer für die Bearbeitung von Webseiten und Netscape Conference. Netscape Communicator gibt es für Windows, Macintosh und für viele Versionen von UNIX. Dort läuft er dann unter X Windows.

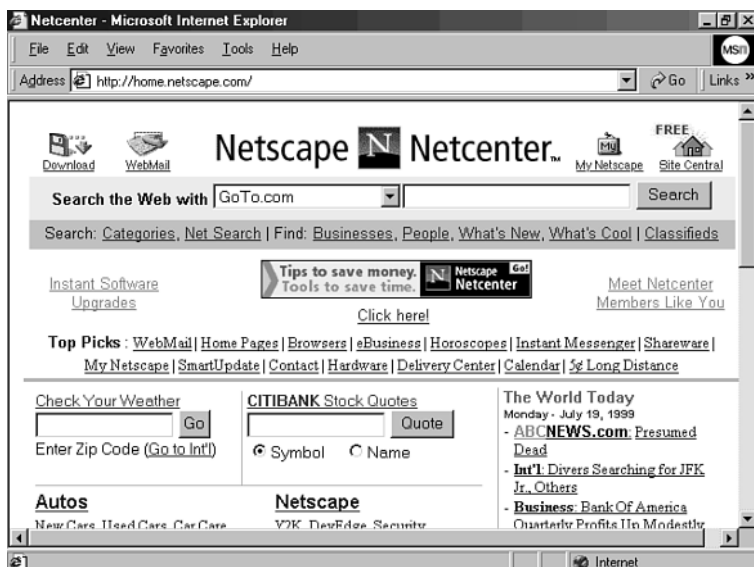


Abbildung 1.8:
Netscape (für
Windows 95/98)

Das Gute ist, dass Sie die Standard-Edition von Netscape Communicator kostenlos downloaden und benutzen können. Die aktuelle Version von Netscape ist die Version 4.7, die Sie sich auf der Netscape-Site unter <http://home.netscape.com/> downloaden oder in Ihrem Computerhandel kaufen können. Für eine nicht allzu hohe Zusatzgebühr erhalten Sie Netscape Communicator Professional. Sie erhalten dann noch zusätzliche Kalenderfunktionen, Zentralmanagementfunktionen und spezielle Features für IBM-Hostanwendungen.

Web-Server

Um Seiten auf dem Web zu betrachten und zu durchstöbern, brauchen Sie nur einen Web-Browser. Um Dokumente im Web zu veröffentlichen, brauchen Sie meistens einen Server.



Ein *Web-Server* ist ein Programm, das auf einem Computer läuft und dafür verantwortlich ist, auf die Anfragen der Web-Browser nach Dateien zu antworten. Sie benötigen einen Web-Server, um Dokumente im Web zu publizieren.

Wann immer Sie Ihren Browser auf ein Web-Dokument einstellen, baut dieser Browser eine Verbindung mit einem Server auf (unter Verwendung des HTTP-Protokolls). Der Server akzeptiert die Verbindung, schickt den Inhalt der Dateien, die angefordert wurden, und unterbricht dann die Verbindung. Daraufhin formatiert der Browser die Information, die er vom Server bekam.

Auf der Server-Seite können sich viele verschiedene Browser mit demselben Server verbinden, um dieselbe Information anzufordern. Der Web-Server ist verantwortlich dafür, dass alle diese Nachfragen ordnungsgemäß behandelt werden.

Web-Server tun mehr als nur Dateien zu deponieren. Es fällt ebenfalls in ihren Verantwortungsbereich, Formular-Inputs zu verarbeiten und Formulare und Browser mit anderen Programmen zu verbinden, wie zum Beispiel Datenbanken, die sich auf demselben Server befinden.

Genauso wie bei den Browsern gibt es viele unterschiedliche Server für viele unterschiedliche Plattformen, jeder mit vielen unterschiedlichen Merkmalen, wobei sich das Preisniveau von kostenlos bis zu sehr teuer bewegt. Für heute brauchen Sie nur zu wissen, wozu der Server dient; Sie werden mehr darüber am achten Tag im Kapitel »Live gehen im Web« erfahren.

Uniform Resource Locator (URL)

Eine URL (zu deutsch ungefähr: »einheitliche Quellenadresse«) ist ein Zeiger auf eine Informationsquelle im Web, sei es ein Web-Dokument, eine Datei in FTP oder Gopher, ein Posting im Usenet oder eine E-Mail-Adresse. Die URL bietet eine universelle, konsistente Methode für das Auffinden und Zugreifen auf Informationen, nicht unbedingt für Sie, sondern vor allem für Ihren Web-Browser. (Wenn sie für Sie wäre, so wäre sie in einem Format, das es erleichterte, sie sich zu merken.)

Neben dieser unmittelbaren Verwendung können Sie URLs auch benutzen, wenn Sie in einem Dokument eine Hypertext-Verbindung zu einem anderen Dokument herstellen. So sind URLs, egal wie Sie es betrachten, wichtig dafür, wie Sie und Ihr Browser sich im Web bewegen.

URLs enthalten Angaben darüber, wie an die Informationen heranzukommen ist: welches Protokoll verwendet wird (FTP, Gopher, HTTP), den Internet-Hostnamen, der zeigt, wo nachzusehen ist, (www.ncsa.uiuc.edu oder [ftp.apple.com](ftp://ftp.apple.com) oder netcom16.net-com.com und so weiter), und das Verzeichnis oder sonstige Stelle auf diesem System, um die Datei zu finden. Es gibt auch spezielle URLs für Tätigkeiten wie das Versenden von Post an andere Leute (genannt »mailto«-URLs) und die Benutzung des Telnet-Programms.

Sie werden im Kapitel 5 alles über URLs erfahren und was deren einzelne Teile eigentlich bedeuten.

Zusammenfassung

Um im Web publizieren zu können, müssen Sie zuerst die grundlegenden Konzepte verstehen, aus denen sich das Web aufbaut. In diesem Kapitel haben Sie drei Dinge gelernt. Erstens haben Sie etwas über die nützlichen Merkmale des Web zum Publizieren von Informationen erfahren. Zweitens haben Sie etwas über Web-Browser und -Server gelernt und etwas darüber, wie sie interagieren, um Webseiten hervorzubringen. Drittens haben Sie gelernt, was eine URL ist und warum diese so wichtig ist, wenn Sie das Web durchstöbern oder darin publizieren möchten.

Workshop

Jedem Kapitel in diesem Buch folgt ein Workshop, der Ihnen das Wiederholen der bereits durchgenommenen Themen erleichtert. Im ersten Teil dieses Workshops werden allgemeine Fragen zum Web gestellt. Danach werden Sie einige Fragen zum Web beantworten. Die Antworten zum Quiz finden Sie im nächsten Punkt. Am Ende eines jeden Kapitels finden Sie einige Übungen, mit deren Hilfe Sie sich die Informationen zum Web gut merken werden.

Fragen und Antworten

F Wem gehört das Web? Wer kontrolliert all diese Protokolle? Wer ist für all dies verantwortlich?

A *Kein einzelnes Wesen »besitzt« das World Wide Web. Bei der enormen Anzahl von unabhängigen Sites, die zur Information auf dem Web beitragen, ist es unmöglich, dass eine einzige Organisation Regeln oder Richtlinien bestimmt. Es gibt jedoch zwei Gruppen von Organisationen, die einen großen Einfluss darauf haben, wie das Web auszusehen hat und in welche Richtung es sich entwickelt. Das erste ist das World-Wide-Web-(W3-)Konsortium, das seinen Sitz beim MIT in den Vereinigten Staaten und bei INRIA in Europa hat. Das W3-Konsortium ist eine Gruppe von Individuen und Organisationen, die daran interessiert ist, die Sprachen und Protokolle, die das Web ausmachen (HTTP, HTML und so weiter) zu unterstützen und zu definieren. Es bietet auch Produkte an (Browser, Server und so weiter), die kostenlos jedermann zur Verfügung stehen. Dies ist die einzige Stelle, die die Möglichkeit hat,*

Standards für das World Wide Web zu setzen und Regeln dafür vorzugeben. Sie können die Homepage des Konsortiums unter <http://www.w3.org/> besuchen.

Die zweite Gruppe von Organisationen, die das Web beeinflussen, sind die Browser-Entwickler selbst, und zwar vor allem Netscape Communications Corporation und Microsoft. Der Wettbewerb um den beliebtesten und technisch fortschrittlichsten Browser auf dem Web ist zur Zeit heftig, wobei Netscape und Microsoft die Hauptwettbewerber sind. Obwohl beide Organisationen vorgeben, dass sie die Richtlinien vom W3-Konsortium unterstützen und sich daran halten, fügen beide auch ihre eigenen neuen Merkmale in die neuen Versionen ihrer Software ein – Merkmale, die oft zueinander und zur Arbeit, die das W3-Konsortium leistet, im Konflikt stehen.

Manchmal fühlt man sich wie inmitten einer Kriegszone, wenn man mit all den neuen und sich schnell verändernden Entwicklungen Schritt halten will, mit Netscape auf der einen Seite, Microsoft auf der anderen und dem W3, das versucht, zu vermitteln und den globalen thermonuklearen Krieg zu verhindern. Als Web-Designer stecken Sie in der Mitte fest, und Sie müssen sich entscheiden, welche Seite Sie unterstützen wollen, wenn überhaupt, und wie Sie mit dem rapiden Wechsel fertig werden wollen. Aber davon handelt der Rest dieses Buchs!

F Warum sollte jemand einen textbasierten Browser wie Lynx benutzen, wenn es doch grafische Browser gibt?

A Wenn Ihr Computer nicht direkt ans Internet angeschlossen ist, brauchen Sie ein Modem oder eine ISDN-Verbindung und einen Zugang, um eine Verbindung mit dem Internet aufbauen zu können. Diese Zugänge sind nicht eben billig. Auch mit einem sehr schnellen Modem, können Webseiten eine ganze Weile brauchen, bis sie geladen sind, besonders, wenn viele Grafiken bzw. Multimediaelemente darin enthalten sind.

Lynx ist die ideale Lösung für Leute, die entweder keine direkte Internet-Anbindung besitzen oder keine Zeit damit vergeuden mögen, das Web grafisch zu benutzen. Lynx ist schnell, und kann nahezu alles im Web verarbeiten.

F In einer Reihe von Magazinartikeln, die ich über das Web gelesen habe, wird CERN, das europäische Laboratorium für Elementarteilchenphysik, im Zusammenhang damit erwähnt, dass es eine beträchtliche Rolle in der Web-Entwicklung spielt. Sie haben es nicht erwähnt. Wo befindet es sich hinsichtlich der Web-Entwicklung?

A Das Web wurde beim CERN von Tim Berners-Lee erfunden, dies werden Sie inzwischen bestimmt schon aus all diesen Magazinartikeln herausgelesen haben. Und für mehrere Jahre war CERN das Zentrum für einen Großteil der

Entwicklungen im WWW. Ende 1995 hat CERN jedoch seinen Teil in der Entwicklung des WWW an das INRIA (Institut National pour la Recherche en Informatique et Automatique) in Frankreich weitergegeben. Das INRIA ist heute der europäische Ableger des W3-Konsortiums.

Quiz

1. Was macht eine Hypertext-Information so cool?
2. Benötigen Sie einen besonderen Computer, um Zugang zum Internet zu bekommen?
3. Was ist, außer einer Verbindung zum Internet, noch notwendig, um zu Webseiten und anderen Informationen im World Wide Web zu gelangen und diese anzusehen? Warum benötige ich diese Hilfsmittel?
4. Was ist eine URL?
5. Was muss bei Dokumentveröffentlichungen im Web beachtet werden?

Antworten

1. Eine Hypertext-Information ermöglicht Ihnen das Anspringen von einem Punkt zum nächsten, sodass Sie nicht mit einer linearen Textstruktur arbeiten müssen.
2. Sie benötigen keinen besonderen Computer. Sie können mit jedem Computer ins Internet gelangen, vom Low-End-Computer bis hin zum teuren UNIX-Workstations. Sie können jedes Betriebssystem und jeden Displaytyp verwenden. Das Web ist voll und ganz plattformübergreifend.
3. Sie benötigen einen Browser, um Webseiten anzusehen und um im Web navigieren zu können. Ein Browser lädt Web-Dokumente herunter, aber seine wichtigste Aufgabe ist das Formatieren, das Anzeigen und das Lesbarmachen von Web-Dokumenten auf Ihrem System.
4. Eine URL, oder Uniform Ressource Locator, ist eine Adresse, die auf ein spezielles Dokument oder auf Informationen im Internet verweist.
5. Meistens benötigen Sie einen Zugang zum Web-Server. Web-Server sind Programme in einer Web-Site, antworten auf Datenanfragen der Browser und senden die geforderten Seiten an viele verschiedene Browser. Web-Server bearbeiten auch Formular-Inputs und Datenbank-Integration.

Übungen

1. Surfen Sie zu jeder der in diesem Kapitel angegebenen URL (http:, ftp:, und gopher).
2. Um einen kleinen Überblick über die Vielzahl der verschiedenen Browser zu bekommen, besuchen Sie `BROWSERS.COM`, Teil der CNET Web-Site. Die URL für `BROWSERS.COM` ist `http://www.browsers.com`. Am Anfang sind dies für Sie zu viele Informationen – aber der Sinn der Übung ist ja nur, Ihnen die Vielfalt der Browser zu zeigen, die die vielen Features unterstützen. Sie werden auf diese URL öfter zurückkommen, wenn Sie mehr über HTML wissen. Da werden Sie nämlich über die neuesten gängigen Browserversionen und die Features, die sie unterstützen, bestens informiert sein.



Vorbereitungen

**Woche
1**

Wenn Sie ein Buch schreiben, einen Vortrag, einen Artikel oder auch nur ein Memo, so beginnen Sie normalerweise wohl nicht einfach mit dem ersten Satz und schreiben dann in einem Zuge durch bis zum Ende. Das gleiche gilt für die bildenden Künste – normalerweise fangen Sie nicht mit der linken oberen Ecke der Leinwand oder Seite an und arbeiten sich dann nach rechts unten vor.

Ein besserer Weg, ein Werk zu schreiben, zu malen oder zu gestalten, ist wohl, zuvor einige Planungen anzustellen – sich klarzumachen, was man vorhat und erreichen will, und eine allgemeine Idee oder einen groben Entwurf von der Struktur des Werkes zu haben, bevor man loslegt und daran arbeitet.

Genauso wie die eher traditionellen Formen der Kommunikation erfordert auch das Schreiben und Gestalten von Webseiten ein gewisses Maß an Planung und Nachdenken, bevor Sie beginnen, mit Text und Grafik um sich zu werfen und sie wild miteinander zu verknüpfen. Vielleicht sogar noch ein bisschen mehr, weil der Versuch, die Regeln traditionellen Schreibens und Gestaltens auf Online-Hypertext anzuwenden, oft zu Dokumenten führt, die entweder schwer zu verstehen und somit online weniger gut zu gebrauchen sind oder aber einfach die Vorzüge der Möglichkeiten, die Hypertext bietet, nicht ausnutzen. Schlecht organisierte Web-Dokumente sind außerdem schwer zu verändern oder zu erweitern.

In diesem Kapitel beschreibe ich einiges davon, worüber Sie nachdenken sollten, bevor Sie loslegen und beginnen, Ihre eigenen Webseiten zu entwickeln. Insbesondere werden Sie

- ▶ den Unterschied zwischen einer Web-Präsentation, einer Webseite und einem Web-Dokument kennen lernen;
- ▶ nachdenken über die Art der Informationen (den Inhalt), die Sie ins Web stellen wollen;
- ▶ die Ziele Ihrer Präsentation festlegen;
- ▶ Ihre Inhalte in Hauptpunkten organisieren;
- ▶ mit einer allgemeinen Struktur von Seiten und Themen vertraut werden.

Sobald Sie eine allgemeine Vorstellung davon haben, wie Sie Ihre Webseiten strukturieren, sind Sie so weit, morgen dann tatsächlich mit dem Schreiben und Gestalten solcher Seiten zu beginnen: in Kapitel 3. Wenn Sie darauf brennen, loszulegen, üben Sie Geduld! Es wird noch mehr als genug HTML in den nächsten paar Tagen zu lernen geben.

Anatomie einer Web-Site

Lassen Sie uns zunächst einen Blick auf die einfache Terminologie werfen, die wir das ganze Buch hindurch verwenden werden. Sie müssen wissen, was die folgenden Begriffe bedeuten und wie sie auf die Arbeit, die Sie für das Web machen, anzuwenden sind:

- ▶ Web-Site, besteht aus einer oder mehreren Webseiten
- ▶ Web Server: ein Computer im Internet oder Intranet, der eine oder mehrere Web-Sites gespeichert hat
- ▶ Webseiten; einzelnes Element einer Web-Site, auf der Platte in einer einzigen Datei gespeichert
- ▶ Homepage; Einstiegsseite einer Web-Site, die zu zusätzlichen Seiten in derselben Web-Site verweisen kann

Eine Web-Site besteht aus einer oder mehreren Webseiten, die Text und Grafik enthalten und in sinnvoller Weise miteinander verknüpft sind. In ihrer Gesamtheit umfasst sie den Kern aus Informationen und erzeugt einen insgesamt zusammenhängenden Eindruck (vgl. Abbildung 2.1).



Eine Web-Site fasst eine oder mehrere Webseiten zusammen.

Eine Web-Präsentation wird auf einem Web-Server gespeichert, welcher der tatsächliche Rechner im Web ist, der die Präsentation enthält. Einige Leute setzen die Begriffe »Web-Präsentation« und »Web-Site« gleich; ich halte sie gerne getrennt, da eine einzige Web-Site viele verschiedene Präsentationen mit vielen unterschiedlichen Absichten, die von verschiedenen Leuten entwickelt wurden, enthalten kann. In der ersten Woche dieses Buchs werden Sie die Kunst der Web-Präsentation erlernen; später werden Sie dann herausfinden, wie Sie Ihre Präsentation auf eine tatsächliche Web-Site publizieren können.



Ein Web-Server ist ein Computer im Internet, der eine oder mehrere Web-Sites enthält.

Eine Webseite ist auf dieselbe Weise ein individuelles Element einer Präsentation, wie eine Seite ein einzelnes Element eines Buchs oder einer Zeitung ist (obwohl, im Unterschied zu Papierseiten, Webseiten von beliebiger Länge sein können). Webseiten werden bisweilen auch *Web-Dokumente* genannt. Beide Ausdrücke beziehen sich auf

denselben Sachverhalt: Eine Webseite ist eine einzelne Datei mit einem eindeutigen Datei-Namen, die von einem Server heruntergeladen und dann von einem Web-Browser formatiert wird.

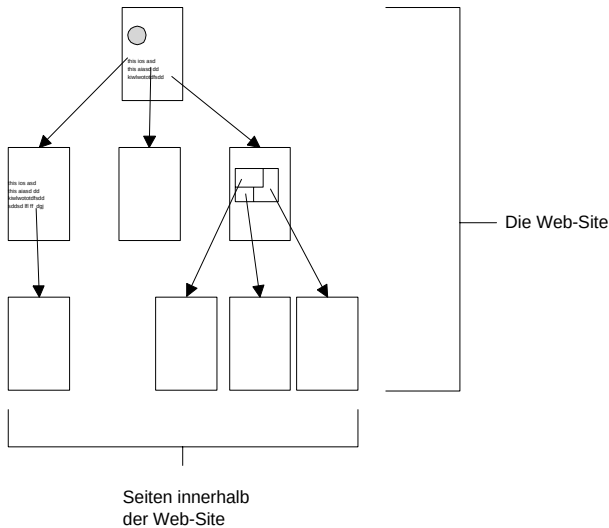


Abbildung 2.1:
Web-Site und Webseiten



Eine *Webseite* ist ein einzelnes Element einer Web-Präsentation, das sich in einer einzigen Datei befindet.

Die Ausdrücke Web-Präsentation, -Site und -seite sind noch ziemlich einfach zu verstehen, aber der Ausdruck »Homepage« ist ein wenig problematischer, da er verschiedene Bedeutungen haben kann.

Wenn Sie das Web durchstöbern, bezieht sich der Begriff Homepage gewöhnlich auf die Webseite, die geladen wird, wenn Sie Ihren Browser starten oder wenn Sie den »Home«-Button betätigen. Jeder Browser hat per Voreinstellung seine eigene Homepage, welche oft mit der Seite für die Firma übereinstimmt, die den Browser entwickelt hat. (Zum Beispiel befindet sich die Netscape Homepage auf der Netscape Web-Site und die Internet Explorer-Homepage auf der Microsoft Web-Site.

Innerhalb Ihres Browser können Sie die Voreinstellung so einstellen, dass jede beliebige Homepage beim Start Ihres Browsers aufgerufen wird – was eine verbreitete Taktik ist, die ich bereits bei vielen Leuten gesehen habe, wodurch sie eine einfache Seite mit Verknüpfungen zu anderen interessanten Orten oder Seiten herstellen, die sie oft besuchen.

Wenn Sie jedoch selber in Webseiten publizieren, hat der Ausdruck Homepage eine ganz andere Bedeutung. Die *Homepage* ist der Startpunkt Ihrer Gruppe von Webseiten. Sie ist der Ort, an dem die meisten Leute beginnen werden, Ihre Web-Präsentation durchzugehen, und die erste Seite, die Ihre Leser sehen werden (siehe Abbildung 2.2).

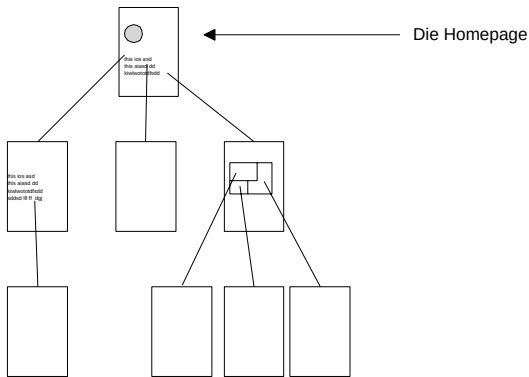


Abbildung 2.2:
Die Homepage



Die *Homepage* ist der Eingangs- oder Startpunkt für den Rest Ihrer Web-Site.

Was wollen Sie im Web tun?

Das mag wie eine alberne Frage erscheinen. Sie hätten dieses Buch nicht gekauft, wenn Sie nicht schon einige Ideen gehabt hätten, was Sie ins Web stellen wollen, oder Sie haben zumindest eine vage Idee, aber noch nichts Konkretes. Vielleicht haben Sie gerade den Job bekommen, Ihre Firma ins Web zu bringen und jemand hat Ihnen dieses Buch gereicht und gesagt: »Hier, das hilft bestimmt.« Möglicherweise haben Sie bereits ein paar andere Webseiten gesehen und halten diese für ziemlich gut gemacht und möchten gerne selbst etwas Ähnliches erstellen.

Was Sie ins Web stellen wollen, ist das, worauf ich mich in diesem Buch als *Ihre Inhalte* beziehe. Inhalte ist ein allgemeiner Ausdruck, der sich auf Text, auf Grafik, auf Medien, auf interaktive Formulare oder sonstwas beziehen kann. Wenn Sie jemandem erzählen, worum es in Ihren Webseiten geht, würden Sie Ihre Inhalte beschreiben.



Ihre *Inhalte* sind die Daten, die Sie ins Web bringen. Informationen, Erzählungen, Bilder, Kunst, Programme, Humor, Diagramme, Spiele – all dies sind Inhalte.

Welche Art von Inhalten können Sie ins Web stellen? So ziemlich alles, was Sie möchten. Hier eine Auswahl einiger Inhalte, die gegenwärtig im Web recht populär sind:

- ▶ *Persönliche Informationen.* Sie können alles mitteilen, was irgend jemand jemals wissen wollen könnte, über Sie und wie fabelhaft Sie sind – Ihre Hobbies, Ihre Ein- und Ansichten, Ihr Bild, Dinge, die Sie gemacht haben und so weiter.
- ▶ *Hobbies oder besondere Interessen.* Ein Web-Dokument könnte Informationen enthalten zu einem bestimmten Punkt oder Hobby oder irgend etwas, an dem Sie interessiert sind; beispielsweise Musik, StarTrek, Motorräder, Kultfilme, halluzinogene Pilze, antike Tintenflaschen oder angekündigte Jazz-Konzerte in Ihrer Stadt.
- ▶ *Publikationen.* Zeitungen, Zeitschriften und andere Publikationen eignen sich besonders gut fürs Web, und sie haben den Vorteil, direkter und schneller aktualisierbar zu sein als ihre gedruckten Gegenstücke.
- ▶ *Firmenprofile.* Es könnten Informationen darüber angeboten werden, was eine Firma tut, wo sie ihren Sitz hat, offene Stellen, Datenblätter, Presseveröffentlichungen, Marketing-Konzepte, Ansprechpartner; solch eine Seite könnte sogar Demo-Software präsentieren, wenn es das ist, was die Firma herstellt.
- ▶ *Online-Dokumentation.* Der Begriff »Online-Dokumentation« kann sich auf alles beziehen, von einer Kurzübersicht über vollständige Referenzdokumentationen bis hin zu interaktiven Übungen und Trainingsteilen. Und er muss sich auch nicht unbedingt auf eine Produktdokumentation beziehen; jede Tätigkeit (das Wechseln des Öls in Ihrem Auto, das Backen eines Soufflés, das Malen von Landschaftsbildern in Öl, das Lernen von HTML) könnte in einer Online-Dokumentation beschrieben werden.
- ▶ *Verkaufskataloge.* Wenn Ihre Firma bestimmte Produkte zum Verkauf anbietet, ist das Verfügbarmachen Ihrer Listen ein schneller und einfacher Weg, Ihre Kunden wissen zu lassen, was Sie anzubieten haben und was es kostet – und wenn die Preise verändert werden, können Sie ganz einfach Ihre Webseiten aktualisieren.
- ▶ *Wahl- und Meinungsforschung.* Interaktivität und Formulare im Web erlauben es Ihnen, Rückmeldungen zu nahezu jedem Thema von Ihren Lesern zu erhalten, einschließlich Meinungsumfragen, Sammeln von Vorschlägen, Meinungsäußerungen zu Ihren Webseiten oder Ihren Produkten und so weiter.
- ▶ *Online-Läden.* Statt nur die Rolle eines Kataloges zu übernehmen, kann das Web auch dafür verwendet werden, um Artikel an Kunden zu verkaufen. Diese Sites stellen meist einen »Einkaufswagen« zur Verfügung, in dem die Benutzer, während sie den Katalog durchstöbern, Waren legen bzw. daraus entfernen können. Am Schluss werden eine Kreditkartennummer und Angaben zur Lieferung erfragt, um die Bestellung zu vervollständigen.

- ▶ *Meinungsumfragen.* Interaktivität und Formulare im Web ermöglichen es Ihnen, zu beinahe jedem Thema ein Feedback von Ihren Lesern zu erhalten: Meinungsumfragen, Eingabefelder für Vorschläge, Anmerkungen zu Ihrer Webseite usw.
- ▶ *Alles andere, was einem in den Sinn kommt.* Hypertext-Literatur, Online-Spiele, Medien-Archive, gemeinschaftliche Kunst ... einfach alles!

Das Web ist nur durch das begrenzt, was Sie mit ihm machen wollen. Und wenn Sie etwas darin anstellen wollen, das nicht in dieser Liste steht oder sehr außergewöhnlich scheint, dann ist das schon ein ausgezeichnete Anlass, es zu versuchen. Die interessantesten Webseiten sind solche, die die gedachten Grenzen des Web überschreiten.

Falls Sie nun wirklich keine Idee haben, was ins Web zu stellen wäre, dann glauben Sie nicht, dass Sie hier nun aufhören, dieses Buch beiseite legen und sich erst etwas einfallen lassen müssen, bevor Sie weitermachen. Möglicherweise kommen Ihnen beim Durchlesen dieses Buchs ein paar Ideen (und das Buch wird nützlich sein, selbst wenn Sie keine Ideen haben). Ich persönlich finde, dass der beste Weg, auf neue Ideen zu kommen, ein Nachmittag ist, an dem Sie das Web durchstöbern und erkunden, was andere Leute so gemacht haben.

Legen Sie Ihre Ziele fest

Was sollen die Leute durch Ihre Präsentation erreichen können? Suchen sie nach bestimmten Informationen darüber, wie etwas zu machen ist? Sollen sie eine Seite nach der anderen lesen, erst weiterblättern, wenn sie die Seite durchhaben, auf der sie gerade sind? Sollen sie einfach auf Ihrer Homepage beginnen und dann ziellos herumstreunen, um Ihre »Welt« zu erkunden, bis sie genug haben und woanders hingehen?

Stellen Sie zur Übung einmal eine Liste zusammen, mit Zielen, die Ihre Leser bei Ihren Webseiten haben könnten. Je klarer die Ziele eingegrenzt sind, desto besser.

Nehmen wir zum Beispiel an, Sie würden eine Webseite erstellen, die die Firma beschreibt, in der Sie arbeiten. Der eine oder andere, der diese Präsentation liest, möchte vielleicht etwas über offene Stellen wissen. Andere möchten erfahren, wo Ihr Firmensitz ist. Wieder andere könnten gehört haben, dass Ihre Firma technische Dokumentensammlungen über das Netz verfügbar macht, und wollen sich nun die aktuellste Version eines bestimmten Dokuments holen. All dies wären zulässige Ziele, und Sie sollten jedes einzelne notieren.

Für die Web-Präsentation eines Einkaufskatalogs bräuchten Sie nur wenige Vorgaben: Sie erlauben Ihren Lesern, die einzelnen Sachen, die Sie zum Verkauf anbieten, nach Namen oder Preis durchzusehen, und, wenn sie dann mit dem Durchblättern fertig sind, können sie eine bestimmte Sache bestellen.

Bei einer persönlichen Homepage oder einer für spezielle Interessen mögen Sie nur ein einziges Ziel haben: Ihren Lesern zu gestatten, die Informationen, die Sie anbieten, durchzusehen und zu erkunden.

Ihre Ziele brauchen nicht allzu hochtrabend sein («diese Web-Präsentation bringt alles über den Weltfrieden»); sie müssen auch für Außenstehende nicht besonders viel Sinn ergeben. Doch eine Zusammenstellung von Vorgaben für Ihr Web-Dokument ist das Rüstzeug, mit dem Sie Ihre Webseiten so gestalten, organisieren und schreiben, dass Sie diese Ziele erreichen, und sie hilft Ihnen, der Versuchung zu widerstehen, Ihre Inhalte durch zusätzliche Informationen unverständlich zu machen.

Wenn Sie Webseiten für jemand anderen entwerfen – wenn Sie zum Beispiel eine Web-Site für Ihre Firma entwerfen oder wenn Sie als Berater angestellt worden sind – dann ist einer der wichtigsten Teile der Information, die Sie haben sollten, bevor Sie auch nur mit einer einzigen Seite anfangen, welche Ziele Ihr Auftraggeber mit der Site verfolgen will. Die Ideen, die Sie für die Präsentation haben, können von denen anderer Leute abweichen, und auf diese Weise müssen Sie vielleicht eine ganze Menge Ihrer Arbeit wegwerfen.

Unterteilen Sie Ihre Inhalte in Hauptthemen

Versuchen Sie nun, mit Ihren Zielen im Hinterkopf, die Inhalte, die Sie haben, in Hauptthemen oder Abschnitte zu unterteilen, wobei Sie verwandte Informationen unter einem einzelnen Stichwort zusammenfassen. Bisweilen werden die Ziele, die Ihnen im vorherigen Abschnitt eingefallen sind, und Ihre Themenliste eng miteinander verwandt sein. Wenn Sie beispielsweise die Webseite für einen Bücherladen zusammenstellen, passt das Ziel, Bücher bestellen zu können, gut in eine Rubrik, die passenderweise »Bestellen von Büchern« genannt wird.

Sie brauchen an dieser Stelle der Entwicklung noch nicht besonders genau zu sein; Ihr Ziel ist hier, bloß zu versuchen, eine Idee davon zu bekommen, was Sie im Besonderen auf Ihrer Webseite beschreiben wollen. Sie können später alles noch besser organisieren, wenn Sie die Seiten wirklich erstellen.

Nehmen wir zum Beispiel an, Sie gestalteten eine Web-Präsentation darüber, wie man sein Auto pflegt. Das ist ein leichtes Beispiel, da die Pflege aus einer konkreten Folge von Schritten besteht, die gut in Themenüberschriften passen. In diesem Beispiel könnten Ihre Themen die folgenden sein:

- ▶ Wechseln von Öl und Ölfilter
- ▶ Prüfen und Einstellen des Motors
- ▶ Prüfen und Einstellen der Ventile

- ▶ Prüfen und Ersetzen der Zündkerzen
- ▶ Prüfen der Flüssigkeiten, Gurte und Schläuche

Kümmern Sie sich nicht um die Reihenfolge der Schritte oder darum, wie Sie Ihre Leser von einem Abschnitt zum anderen führen. Stellen Sie einfach all die Dinge zusammen, die Sie in Ihrer Web-Präsentation beschreiben möchten.

Wie sieht es nun mit einem weniger tätigkeitsbezogenen Beispiel aus? Nehmen wir an, Sie wollen eine Gruppe von Webseiten über eine spezielle Rock-Band erstellen, weil Sie ein großer Fan und sich sicher sind, dass es noch andere Fans gibt, die von Ihrem detaillierten Wissen über die Band profitieren könnten. Ihre Stichwörter mögen sein:

- ▶ Die Geschichte der Band
- ▶ Biographien von jedem Bandmitglied
- ▶ Eine Diskographie von all den Alben und Singles, die die Band herausgebracht hat
- ▶ Ausgewählte Liedertexte
- ▶ Abbildungen von Platten-Covers
- ▶ Informationen über angekündigte Konzerte und künftige Produktionen

Sie können so viele Themen zusammenstellen wie Sie wollen, aber versuchen Sie, jedes Thema sinnvoll und kurz zu halten; falls eines zu umfangreich erscheint, versuchen Sie, es in Unterthemen aufzuteilen. Wenn Sie zu viele kleine Bereiche haben, versuchen Sie, diese unter einem allgemeineren Thema zusammenzufassen. Zum Beispiel, wenn Sie eine Online-Enzyklopädie über giftige Pflanzen erstellen würden, so wäre es gewiss zuviel des Guten, für jede einzelne Pflanze einen eigenen Eintrag zu machen; Sie könnten ganz einfach jeden Pflanzennamen unter einem Buchstaben des Alphabets einordnen (A, B, C und so weiter) und jeden Buchstaben als einen Abschnitt benutzen. Das setzt natürlich voraus, dass Ihre Leser die Informationen in Ihrer Enzyklopädie auch alphabetisch suchen. Falls sie Informationen über giftige Pflanzen mit einer anderen Methode finden wollen, dann müssten Sie sich eine andere Abschnittsaufteilung einfallen lassen.

Ihr Ziel ist es also, mehrere Abschnitte zu haben, die ungefähr den gleichen Umfang besitzen, und verwandte Informationen, die Sie präsentieren wollen, zusammenzufassen.

Ideen zur Organisation und Navigation

An dieser Stelle sollten Sie nun eine Vorstellung davon haben, worüber Sie sprechen wollen, sowie eine Liste der Themen. Der nächste Schritt besteht nun darin, mit der Strukturierung des Stoffes zu beginnen. Doch bevor Sie das tun, vergegenwärtigen Sie sich einige »Standard«-Strukturen, die in anderen Hilfe-Systemen und Online-Tools verwendet werden. Dieser Abschnitt beschreibt einige dieser Strukturierungen, ihre unterschiedlichen Möglichkeiten sowie einige wichtige Überlegungen, einschließlich

- ▶ der Art von Information, mit der die jeweilige Struktur gut funktioniert;
- ▶ wie die Leser ihren Weg durch den Inhalt jeder Struktur finden, um zu bekommen, wonach sie suchen;
- ▶ wie man sicherstellen kann, dass die Leser herauskriegen können, wo sie sich gerade in Ihren Informationen (Inhalten) befinden, und den Weg zu einer bestimmten Stelle zurückfinden.

Überlegen Sie, während Sie diesen Abschnitt lesen, wie Ihre Informationen in eine dieser Formen passen könnten oder wie Sie diese Formen kombinieren, um eine neue Struktur für Ihre Web-Präsentation zu entwerfen.



Viele der Ideen, die ich in diesem Abschnitt beschreibe, sind einem Buch von William K. Horton mit dem Titel »*Designing and Writing Online Documentation*« (John Wiley & Sons, 1994) entnommen. Obwohl Hortons Buch in erster Linie für technische Autoren und Entwickler geschrieben wurde, die insbesondere mit Online-Hilfe-Systemen arbeiten, ist es ein tolles Buch über die Strukturierung von Dokumenten und zur Arbeit mit Hypertext-Informationen im Allgemeinen. Falls Sie damit beginnen, viel Arbeit ins Web zu stecken, sollten Sie sich dieses Buch besorgen; es bietet eine Menge Einblicke über das hinaus, was ich anzubieten habe.

Hierarchien

Wohl der einfachste und logischste Weg, Ihre Web-Dokumente zu strukturieren, ist eine hierarchische oder Menü-orientierte Art und Weise, dargestellt in Abbildung 2.3.

Hierarchien und Menüs empfehlen sich besonders gut für Online- und Hypertext-Informationen. Die meisten Online-Hilfe-Systeme zum Beispiel sind hierarchisch aufgebaut. Sie beginnen mit einer Liste oder einem Menü von Hauptpunkten; das Auswählen eines von ihnen führt Sie zu einer Liste von Unterthemen, die Sie dann zur Darstellung eines bestimmten Themas führen. Unterschiedliche Hilfe-Systeme haben natürlich verschiedene Stufen, aber die meisten folgen dieser einfachen Struktur.

In einer hierarchischen Organisation ist es für Ihre Leser leicht, ihre Position innerhalb der Struktur zu erkennen. Die Auswahlmöglichkeiten sind, sich nach oben zu allgemeineren Angaben zu bewegen oder nach unten zu spezifischeren Informationen. Das Anbieten einer Verbindung zur obersten Stufe ermöglicht Ihren Lesern zudem, schnell und einfach zu einer bekannten Position zurückzukehren.

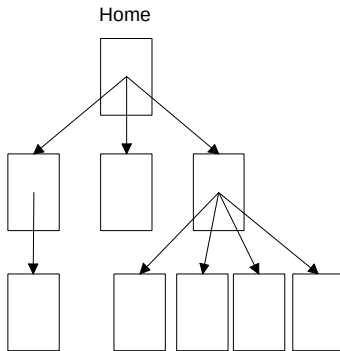


Abbildung 2.3:
Hierarchische Organisation

In Hierarchien bietet die Homepage den allgemeinsten Überblick der ihr untergeordneten Inhalte und definiert die Hauptverbindungen zu den Seiten weiter unten in der Hierarchie.

Zum Beispiel könnte eine Web-Präsentation über Gartenbau eine Homepage mit jenen Abschnitten haben, die in Abbildung 2.4 gezeigt werden.

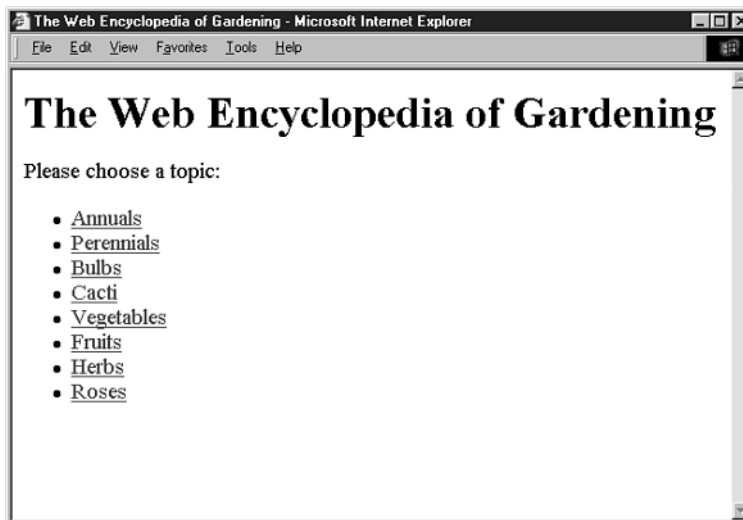


Abbildung 2.4:
Die Gartenbau-
Homepage mit
hierarchischer
Struktur

Wenn Sie »Fruits« (Früchte) auswählen, werden Sie mit einer Seite über Früchte weiter »unten« in der Hierarchie verbunden (vgl. Abbildung 2.5). Von dort können Sie zur Homepage zurückgehen oder eine andere Verknüpfung auswählen und weiter hinuntergehen zu spezifischeren Informationen über die einzelnen Früchte.

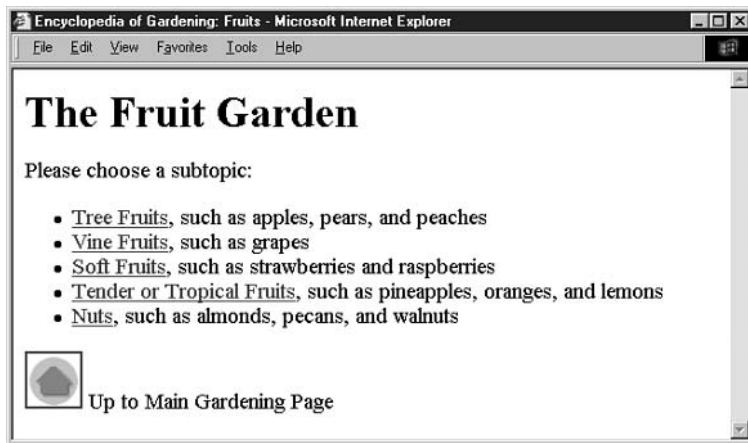


Abbildung 2.5:
Die Früchte-Seite

Das Auswählen von »Soft Fruits« (weiche Früchte) bringt Sie wieder weiter zu einer Menü-artigen Seite, auf der Sie noch mehr Auswahlmöglichkeiten haben (vgl. Abbildung 2.6). Von dort können Sie zu »Fruits« nach oben gehen, zurück zur Homepage oder hinunter zu einer der Auswahlmöglichkeiten dieses Menüs.

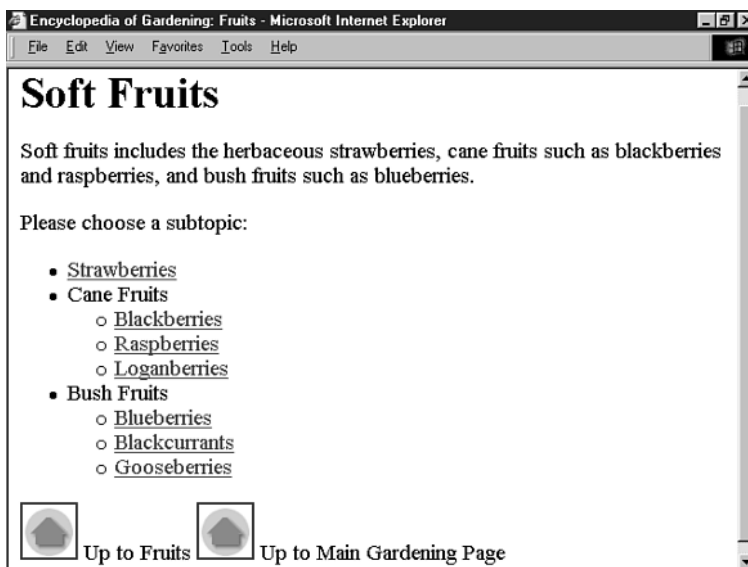


Abbildung 2.6:
Soft Fruits

Beachten Sie, dass jede Seite eine gleichartige Schnittstelle besitzt (hoch, runter, zurück zum Index) und dass Sie jedes Mal eine begrenzte Auswahl an Möglichkeiten für die grundlegende Navigation haben. Hierarchien sind ausreichend strukturiert, sodass die Chancen, sich zu verirren, minimal sind.

Vermeiden Sie jedoch zu viele Stufen und zu viele Auswahlmöglichkeiten, da Sie Ihre Leser leicht verärgern können. Zu viele Menü-Seiten haben das »Voice-Mail-Syndrom« zur Folge: Nachdem Sie in zu vielen Menü-Seiten auswählen mussten, haben Sie vergessen, was Sie eigentlich wollten, und sind zu verärgert, um sich noch weiter darum zu kümmern. Versuchen Sie, Ihre Hierarchie nicht mehr als zwei oder drei Stufen tief werden zu lassen; wenn nötig, fassen Sie auf den Seiten mit den Verzweigungen in der Hierarchie Informationen zusammen.

Linear

Ein anderer Weg, Ihre Dokumente zu organisieren, ist die Verwendung einer linearen oder sequentiellen Struktur, die den gedruckten Dokumenten ähnlich ist. In einer linearen Struktur, illustriert in Abbildung 2.7, ist die Homepage der Titel oder die Einleitung, und jede Seite folgt dieser Struktur dann sequentiell. In einer streng linearen Struktur gibt es Verknüpfungen, die von einer Seite zu einer anderen führen, üblicherweise vorwärts und zurück. Sie könnten auch eine Verbindung zur Homepage einfügen, die Sie schnell zurück zur ersten Seite bringt.

Home

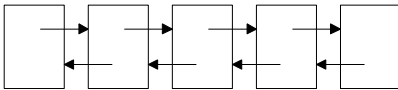


Abbildung 2.7:
Lineare Organisation

Der jeweilige Zusammenhang ist generell leicht herauszufinden, einfach weil es so wenig Stellen gibt, wohin man gehen kann.

Eine lineare Organisation ist sehr streng und beschränkt sowohl die Freiheit, die Ihre Leser beim Erkunden der Seiten haben, als auch Ihre eigene, die Sie selbst bei der Präsentation Ihrer Informationen haben. Lineare Strukturen sind gut, um Material online anzubieten, das auch offline eine sehr lineare Struktur besitzt (wie etwa Kurzgeschichten, Schritt-für-Schritt-Anleitungen oder computerunterstützte Übungen), oder wenn Sie Ihre Leser ausdrücklich davon abhalten wollen, in Ihren Seiten herumzuspringen.

Nehmen Sie zum Beispiel an, Sie wollten jemandem beibringen, wie man Käse macht. Die Käseherstellung ist ein komplexer Prozess, der mehrere Schritte erfordert, die in einer ganz bestimmten Reihenfolge erledigt werden müssen.

Diesen Prozess mit Hilfe von Webseiten zu beschreiben, legt von vornherein eine lineare Struktur ziemlich nahe. Beim Navigieren in mehreren Webseiten zu diesem The-

ma würden Sie mit der Homepage beginnen, die eine Zusammenfassung enthalten könnte oder einen Überblick der Schritte bietet, die folgen. Dann führt die Benutzung der Verknüpfung »vorwärts« zum ersten Schritt, »Die richtige Milch besorgen«, zum nächsten Schritt, »Setzen und Eindicken der Milch«, den ganzen Weg bis zum letzten Schritt, »Haltbarmachen und Reifung des Käses«. Wenn Sie etwas nochmal sehen wollen, könnten Sie die Verknüpfung für »zurück« benutzen. Da der Prozess selbst linear ist, gibt es wenig Anlass für Verbindungen, die aus dem Hauptfluss ausbrechen, oder Verknüpfungen, die verschiedene Schritte des Prozesses miteinander verbinden.

Linear mit Alternativen

Sie können die Starrheit einer linearen Struktur abschwächen, indem Sie den Lesern erlauben, vom Hauptpfad abzuweichen. Beispielsweise könnten Sie eine lineare Struktur haben, die Alternativen besitzt, die an einer Stelle des Hauptweges verzweigen (vgl. Abbildung 2.8). Die Verzweigungen können dann an einer Stelle weiter unten wieder auf den Hauptweg zurückführen, oder sie können auf einem separaten Weg weiter hinunterführen, bis sie ein »Ende« erreichen.

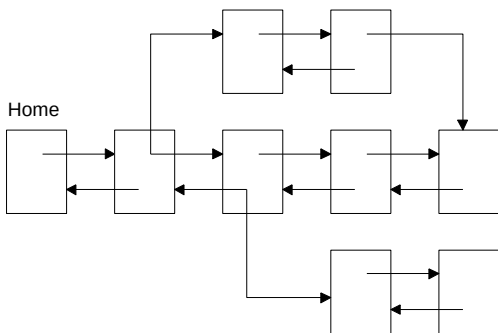


Abbildung 2.8:
Linear mit Alternativen

Lassen Sie uns zum Beispiel sagen, Sie hätten eine Installationsprozedur für ein Software-Paket, die in den meisten Schritten ähnlich ist, mit Ausnahme eines Schrittes, der unterschiedliche Prozeduren benutzt, je nachdem, auf welchem System die Software installiert werden soll. An dieser Stelle der linearen Installation könnten Sie verzweigen, um jedes System abzudecken, wie es in Abbildung 2.9 gezeigt wird.

Nach dem systemspezifischen Teil der Installation könnten Sie dann zum Originalzweig zurückführen und mit der allgemeinen Installation weitermachen.

Zusätzlich zum Verzweigen in linearen Strukturen könnten Sie auch Verbindungen anbieten, die es den Lesern erlauben, in der Abfolge vorwärts- oder zurückzuspringen, wenn sie einen bestimmten Schritt noch einmal sehen wollen oder wenn sie einen bestimmten Inhalt bereits verstehen (vgl. Abbildung 2.10).

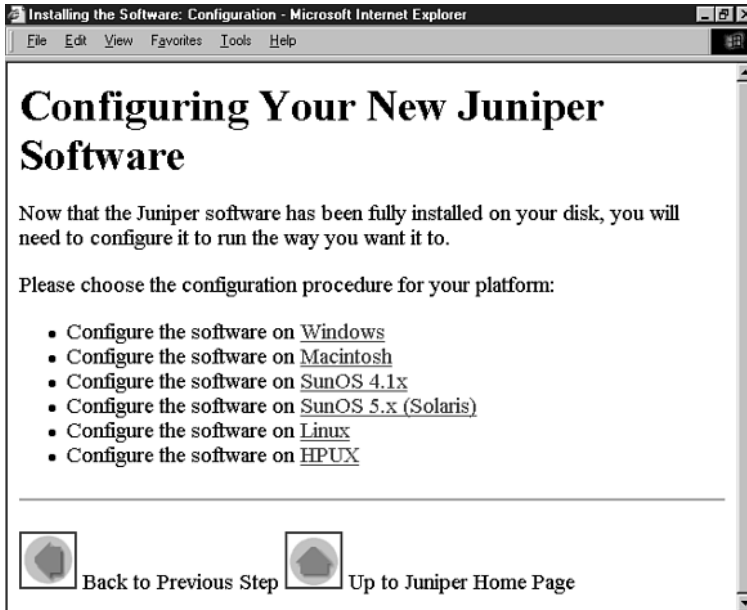


Abbildung 2.9:
Unterschiedliche
Schritte auf
verschiedenen
Systemen

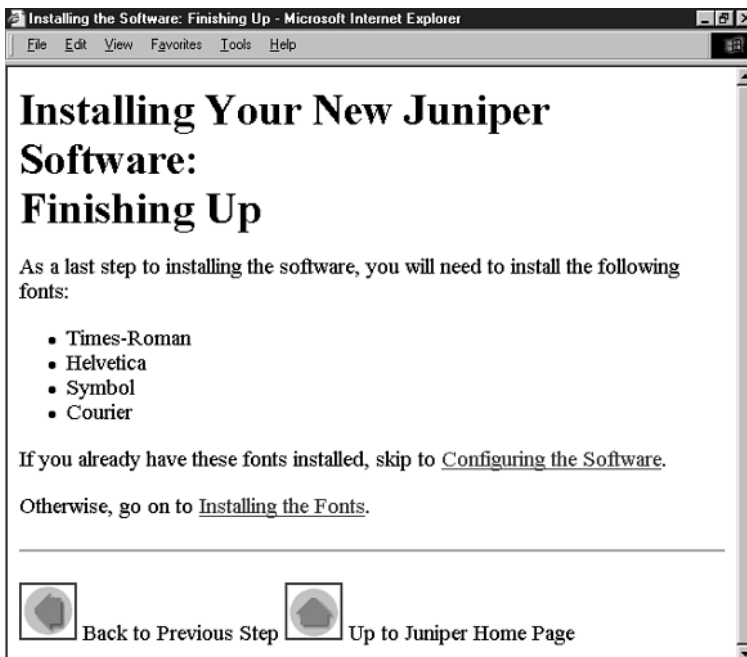


Abbildung 2.10:
Vorwärts- oder
zurückspringen

Kombination von linear und hierarchisch

Eine beliebte Form, Dokumente im Web zu organisieren, ist eine Kombination aus linearer und hierarchischer Struktur, dargestellt in Abbildung 2.11. Diese Struktur taucht meist auf, wenn sehr strukturierte aber lineare Dokumente online verfügbar gemacht werden; die beliebten FAQ-Dateien (*Frequently Asked and Answered Questions*, häufig gestellte und beantwortete Fragen) verwenden diese Struktur.

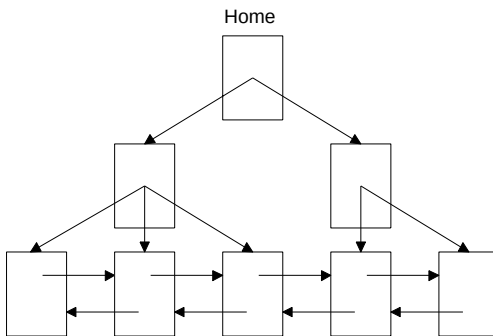


Abbildung 2.11:
Kombination von linear und hierarchisch

Die Kombination von linearen und hierarchischen Dokumenten funktioniert gut, solange es ausreichend viele Anhaltspunkte bezüglich des Zusammenhangs gibt. Da man sich entweder hoch und runter **oder** vorwärts und rückwärts bewegen kann, ist es leicht, seine inhaltliche Position in der Hierarchie aus den Augen zu verlieren, wenn man die hierarchischen Grenzen kreuzt, indem man sich vorwärts und rückwärts bewegt.

Nehmen wir zum Beispiel an, Sie würden das Shakespeare-Stück *Macbeth* auf verschiedenen Webseiten online verfügbar machen. Zusätzlich zur einfachen linearen Struktur des Stückes könnten Sie ein hierarchisches Inhaltsverzeichnis erstellen und Zusammenfassungen jeden Aktes mit passenden Stellen des Textes verknüpfen, ähnlich dem, das in Abbildung 2.12 dargestellt ist.

Da dies eine sowohl lineare als auch hierarchische Struktur ist, bieten Sie auf jeder Seite des Scripts Verbindungen an, um vorwärts, rückwärts, zurück zum Anfang und hoch zu gehen. Doch was bedeutet in diesem Zusammenhang, »hoch« zu gehen?

Wenn Sie zu dieser Seite gerade von einer Akt-Zusammenfassung heruntergekommen sind, ist der Zusammenhang verständlich. »Hoch« bedeutet, zurück zu der Zusammenfassung zu gehen, von der Sie gerade kamen.

Doch angenommen, Sie kamen von einer Zusammenfassung, sind dann vorwärts gegangen, haben die Grenze zwischen zwei Akten überschritten (sagen wir, vom ersten Akt zum zweiten). Was bedeutet »hoch« jetzt?

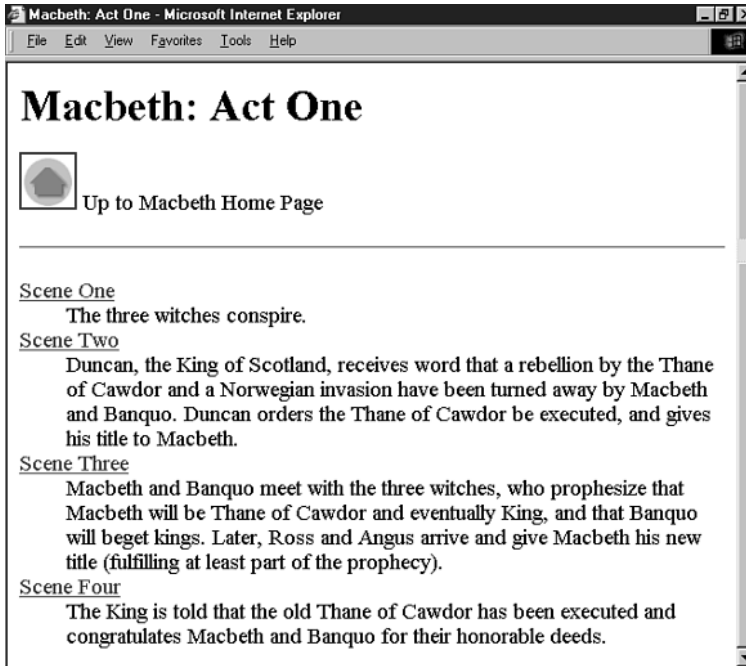


Abbildung 2.12:
Die Macbeth-
Hierarchie

Der Umstand, dass Sie sich zu einer Seite hoch bewegen, die Sie zuvor womöglich noch gar nicht gesehen haben, wirkt desorientierend, gemessen an dem, was die Leser von einer Hierarchie erwarten. Von hoch und runter wird erwartet, dass sie konsequent gleichmäßig funktionieren.

Zwei mögliche Lösungen wären vorstellbar:

- ▶ Erlauben Sie keine »vorwärts«- und »zurück«-Verknüpfungen über hierarchische Grenzen hinweg. In diesem Fall müssten Sie sich, um vom ersten zum zweiten Akt in »Macbeth« weiterzulesen, in der Hierarchie hoch bewegen und dann hinunter zum zweiten Akt.
- ▶ Bieten Sie mehr Hinweise im Verknüpfungstext. Anstatt einfach nur »hoch« oder ein Bildchen für die Verbindung, die in der Hierarchie hoch führt, zu verwenden, fügen Sie eine Beschreibung ein, wohin man sich hoch bewegt.

Verwoben

Als verwoben bezeichnet man eine Reihe von Dokumenten mit wenig oder gar keiner richtigen Gesamtstruktur. Das einzige, was die einzelnen Seiten zusammenbindet, ist eine Verknüpfung (vgl. Abbildung 2.13). Die Leser gelangen von Dokument zu Dokument, indem sie den Verknüpfungen folgen.

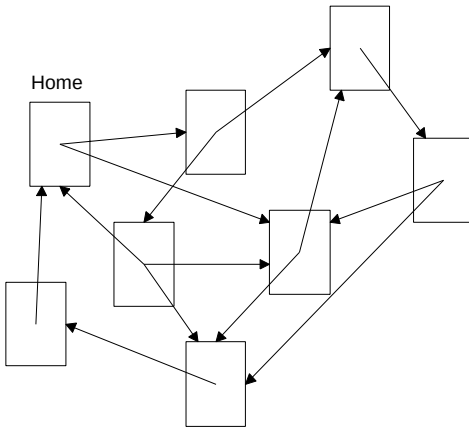


Abbildung 2.13:
Eine verwobene Struktur

Verwobene Strukturen tendieren dazu, fließend zu sein und den Lesern zu erlauben, uneingeschränkt durch den Inhalt zu wandern. Verwobene Strukturen eignen sich hervorragend für Inhalte, die sich so dahinschlingeln sollen oder beziehungslos sind, oder für Inhalte, die zum Herumschmökern anregen sollen. Das World Wide Web selbst ist natürlich eine gigantische verwobene Struktur.

Ein Beispiel für Inhalte, die in einer verwobenen Struktur organisiert sind, könnten virtuelle »Räume« sein, die durch Webseiten geschaffen wurden. Wenn Sie schon mal ein altes Text-Abenteuerspiel wie »Zork« oder »Dungeon« gespielt oder einen MUD (Multi-User-Dungeon) benutzt haben, so sind Sie mit dieser Art von »Umwelt« bereits vertraut.

Im Zusammenhang mit einer Web-Präsentation ist die Welt so organisiert, dass jede Seite als bestimmter Ort angesehen wird (und üblicherweise eine Beschreibung dieses Orts enthält). Von diesem Ort aus können Sie sich in verschiedene Richtungen begeben, um die Umgebung zu erkunden, genauso wie Sie sich in einem Gebäude der wirklichen Welt von Zimmer zu Zimmer bewegen würden (und sich genauso schnell verlaufen würden). Zum Beispiel könnte die einleitende Homepage so ähnlich aussehen wie in Abbildung 2.14 dargestellt.

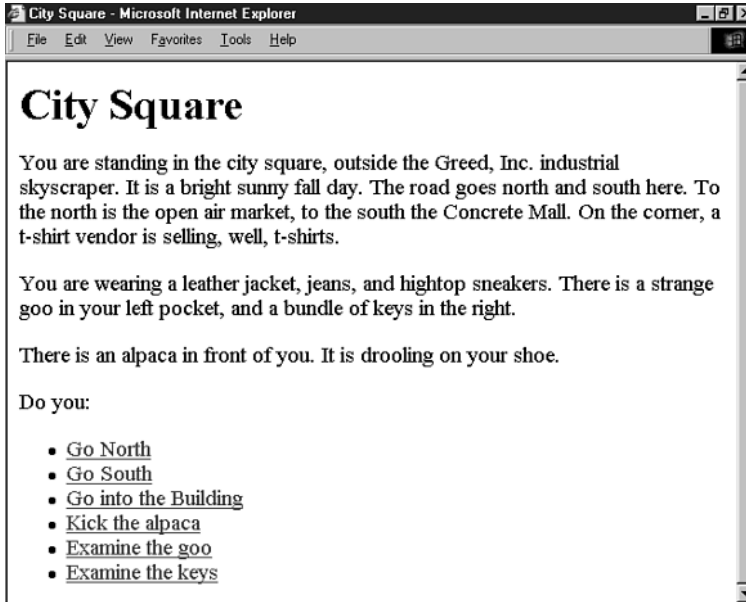


Abbildung 2.14:
Die Homepage
einer virtuellen
Welt im Web

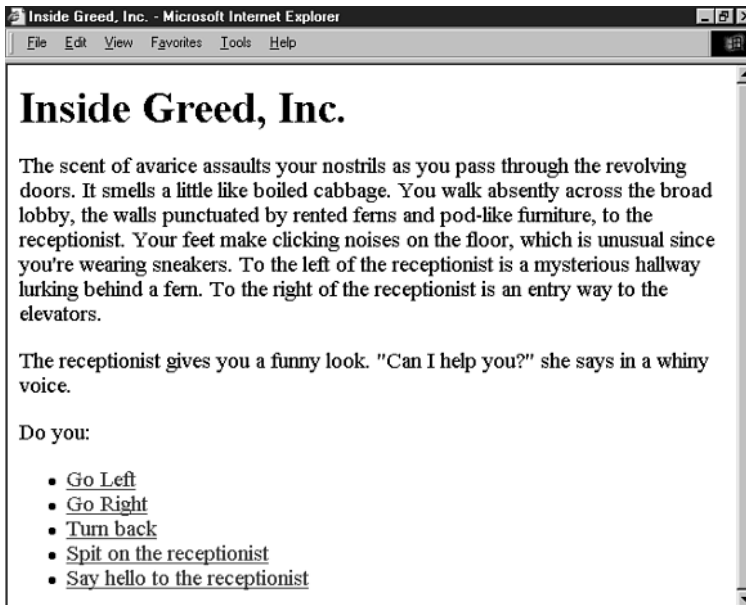


Abbildung 2.15:
Eine weitere Seite
im Web

Von dieser Seite aus können Sie eine der Verbindungen erkunden, sagen wir mal, indem Sie in das Gebäude gehen, dies wird Sie dann zum Inhalt der Abbildung 2.15 führen.

Jeder Raum hat Verbindungen mit jedem »angrenzenden« Raum der Umgebung. Indem Sie diesen Verbindungen folgen, können Sie die Räume dieser Welt erkunden.

Das Problem bei verwobenen Strukturen besteht darin, dass es sehr leicht ist, sich darin zu verirren – gerade so wie in der »Welt«, die Sie in dem Beispiel kennen gelernt haben. Ohne eine Gesamtstruktur des Inhalts ist es schwer, die Beziehung herauszukriegen zwischen dem, wo Sie gerade sind, und dem, wohin Sie gehen, sowie häufig auch dem, wo Sie bereits gewesen sind. Der Zusammenhang ist schwer zu erkennen, und oft ist die einzige Möglichkeit, Ihren Weg aus einer so verwobenen Struktur zurückzufinden, Ihre einzelnen Schritte zurückzuverfolgen. Verwobene Strukturen können außerordentlich desorientierend wirken und immens frustrierend sein, wenn Ihre Leser ein bestimmtes Ziel im Sinn haben.

Um das Problem der Desorientierung zu lösen, können Sie Hinweise auf jeder Seite verwenden. Zwei Ideen:

- ▶ Bieten Sie einen Ausweg an. »Zurück zur Homepage« ist eine ausgezeichnete Verknüpfung.
- ▶ Fügen Sie einen Plan der Gesamtstruktur auf jeder Seite ein, mit einem Hinweis »Sie sind hier« irgendwo auf der Karte. Es muss nicht unbedingt eine visuelle, gezeichnete Karte sein, doch das Anbieten irgendeiner Form von Zusammenhang bringt schon eine Menge, um Ihre Leser vor dem Verirren zu schützen.

Ein Storyboard für Ihre Web-Site

Der nächste Schritt in der Planung Ihrer Web-Site ist, festzulegen, welche Inhalte auf welche Seite kommen, und sich ein paar einfache Verbindungen für die Navigation zwischen diesen Seiten einfallen zu lassen.

Wenn Sie eine der im vorherigen Abschnitt beschriebenen Strukturen verwenden, mag sich ein Großteil der Organisation schon aus der Struktur ergeben, sodass dieser Abschnitt für Sie sehr einfach sein wird. Wenn Sie indes unterschiedliche Arten von Strukturen miteinander kombinieren wollen oder wenn Sie eine ganze Menge Inhalte haben, die auf ausgetüftelte Weise miteinander verbunden werden müssen, wird es später, wenn Sie jede einzelne Seite entwickeln und verknüpfen, sehr hilfreich sein, wenn Sie sich zuvor einmal hinsetzen und sich einen genauen Plan davon machen, was wohin kommen soll.

Was ist ein Storyboard, und weshalb brauche ich es?

Das Erstellen eines Storyboard ist ein dem Filmemachen entliehenes Konzept, wo jede Szene und jede einzelne Kameraeinstellung skizziert und in eben der Reihenfolge festgehalten wird, in der sie im Film auftaucht. Storyboarding bietet eine übersichtliche Struktur und einen Plan zu dem Film. Der Filmregisseur und seine Mitarbeiter erhalten eine präzise Information über die passende Stelle der Bildaufnahmen im Film.



Das *Storyboard*, ein vom Filmemachen entliehener Ausdruck; es ist der Prozess, der einen ungefähren Überblick darüber verschafft und eine Skizze davon anlegt, wie Ihre Präsentation aussehen soll, bevor Sie überhaupt mit dem Schreiben der Seiten anfangen. Das Erstellen eines Storyboard hilft Ihnen dabei, Ihre gesamte Präsentation und deren Aussehen nach ihrer Fertigstellung zu visualisieren.

Das Storyboard-Konzept funktioniert auch ziemlich gut bei der Entwicklung von Webseiten. Das Storyboard stellt einen groben Allgemeinüberblick darüber dar, wie die Web-Präsentation aussehen wird, wenn sie fertig ist. Sie wissen dann, welche Themen auf welche Seiten kommen, Sie kennen die Hauptverbindungslinien und haben konzeptionelle Überlegungen angestellt, welche Art von Grafik Sie verwenden wollen und wo sie eingesetzt werden soll. Mit dieser Darstellung können Sie alle Seiten nacheinander entwickeln, ohne sich jedesmal genau erinnern zu müssen, wohin die jeweilige Seite innerhalb der Gesamtpräsentation nun gehört und wie ihre häufig komplexen Beziehungen zu anderen Seiten aussehen.

Im Fall von wirklich umfangreichen Dokumenten erlaubt eine solche Darstellung, dass verschiedene Leute unterschiedliche Teile derselben Web-Präsentation entwickeln. Mit einem klaren Storyboard können Sie Mehrfacharbeit minimieren und die Menge des Wissens um die Zusammenhänge reduzieren, die sich jede einzelne Person merken muss.

Für kleinere oder einfachere Web-Präsentationen oder Präsentationen mit einer einfachen logischen Struktur mag das Erstellen eines Storyboard unnötig sein. Doch bei großen und komplexeren Projekten kann die Existenz eines Storyboard eine Menge Zeit und Frustration einsparen helfen. Wenn Sie nicht sämtliche Teile Ihrer Inhalte und deren jeweilige Wechselbeziehungen im Kopf haben können, erwägen Sie ein Storyboard.

Wie sieht nun ein Storyboard für eine Web-Präsentation aus? Es kann einfach ein Stapel Papier sein, wobei jedes einzelne Blatt eine Seite repräsentiert. Es enthält die Liste der Themen, die auf den entsprechenden Seiten behandelt werden, und einige Gedanken zu den Verbindungen, die auf den Seiten enthalten sind. Ich habe auch Storyboards für sehr komplexe Hypertext-Systeme gesehen, die ein ziemlich großes

Schwarzes Brett, Karteikarten und Fäden erforderten. Auf jede Karteikarte war ein Thema geschrieben, und die Verbindungen wurden repräsentiert durch an Nadeln befestigte Fäden, die von Karte zu Karte führten (vgl. Abbildung 2.16).

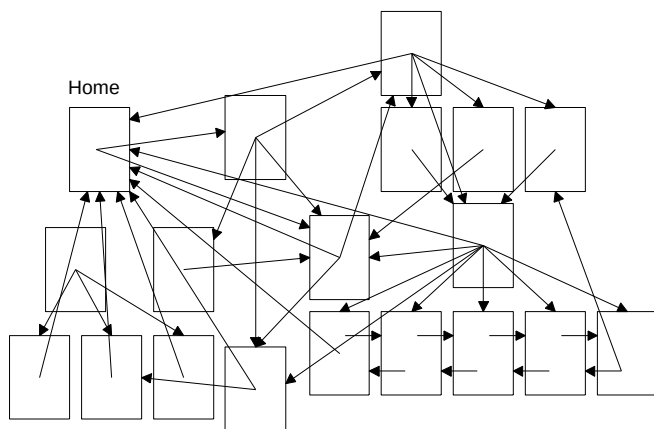


Abbildung 2.16:
Ein komplexes Storyboard

Das Entscheidende an einem Storyboard ist, dass es Ihre Webseiten auf eine Weise organisiert, die für Sie zweckmäßig ist. Wenn Sie Karteikarten und Fäden lieber mögen, arbeiten Sie damit. Wenn es mit einem einfachen Abriss der Struktur auf Papier oder im Computer besser geht, verwenden Sie dies.

Hinweise zum Erstellen eines Storyboard

Einige Überlegungen, wenn Sie Ihr Storyboard anlegen:

- ▶ Welche Themen kommen auf welche Seite?
- ▶ Eine einfache Faustregel ist, jedes Thema durch eine einzelne Seite zu repräsentieren. Doch wenn Sie eine große Anzahl von Themen haben, können ihre Pflege und ihre Verknüpfungen zu einer beängstigenden Aufgabe werden. Erwägen Sie stattdessen die Zusammenfassung kleiner verwandter Themen auf einer einzigen Seite. Doch übertreiben Sie es nicht und packen Sie nicht alles auf eine Seite; Ihre Leser müssen Ihr Dokument über das Netz laden. Es ist besser, einige Seiten mittlerer Größe zu haben (sagen wir, im Umfang von zwei bis zehn Seiten in Ihrem Textverarbeitungsprogramm), als eine gigantische Seite oder Hunderte von klitzekleinen Seiten.
- ▶ Was sind die vornehmlichen Arten der Navigation zwischen den Seiten?
- ▶ Welche Verbindungen brauchen Sie für Ihre Leser, um von Seite zu Seite zu kommen? Dies sind die Hauptverbindungen in Ihrem Dokument, die Ihren Lesern er-

lauben, die Ziele zu erreichen, die Sie im ersten Arbeitsschritt definiert haben. Die Verknüpfungen für hoch, runter, vorwärts, rückwärts und zur Homepage fallen alle unter die Kategorie grundlegender Navigation.

- ▶ Welche alternativen Arten der Navigation möchten Sie anbieten?
- ▶ Zusätzlich zu den einfachen Navigationsverbindungen enthalten manche Web-Präsentationen parallel zum Hauptinhalt ergänzende Informationen, wie etwa ein Begriffs-Glossar, einen alphabetischen Index der Konzepte oder eine Seite mit Dank-sagungen. Denken Sie an solche zusätzlichen Formen von Information, wenn Sie Ihren Plan machen, und überlegen Sie, wie Sie sie mit dem Hauptinhalt verbinden können.
- ▶ Was packen Sie auf Ihre Homepage?
- ▶ Da die Homepage der Startplatz für den Rest der Informationen in Ihrer Präsentation ist, überlegen Sie, welche Art von Information Sie auf die Homepage packen. Eine allgemeine Zusammenfassung dessen, was kommt? Eine Liste mit Verbindungen zu den anderen Themen?
- ▶ Vergewärtigen Sie sich Ihre Ziele!
- ▶ Wenn Sie den Rahmenplan für Ihre Web-Präsentation gestalten, behalten Sie Ihre Ziele im Kopf, und stellen Sie sicher, dass Sie Ihr Anliegen nicht durch zusätzliche Informationen oder Inhalte unverständlich machen.



Es gibt viele Utilities und Software-Pakete, die Sie bei der Erstellung von Storyboards unterstützen können. An erster Stelle stehen hier Pakete für das Site-Management. Diese Programme helfen Ihnen bei der Verwaltung der Links der Site, zeigen eine grafische Repräsentation der Beziehungen zwischen den Dokumenten der Site, erlauben es sogar Dokumente zu verschieben und aktualisieren dabei automatisch alle relevanten Links in und auf diese Dokumente.

Zusammenfassung

Das Gestalten einer Web-Präsentation kann, wie der Entwurf einer Buchgliederung, eines Gebäudegrundrisses oder eines Gemäldes, mitunter ein komplexer und komplizierter Prozess sein. Wenn Sie vor dem Beginn einen Plan haben, kann Ihnen das helfen, die Einzelheiten übersichtlich zu halten, und es hilft Ihnen, das endgültige Produkt mit weniger Fehlstarts zu entwickeln. In diesem Kapitel haben Sie gelernt, wie Sie einen einfachen Plan und Strukturen für die Erstellung eines Satzes von Webseiten zusammenstellen, und

- ▶ zu entscheiden, welche Art von Inhalt Sie präsentieren
- ▶ einige Ziele für diesen Inhalt festzulegen
- ▶ sich auf eine Gruppe von Themen festzulegen
- ▶ die Präsentation zu organisieren und ein Drehbuch zu schreiben

Mit diesem Plan vor sich können Sie nun mit den nächsten paar Kapiteln weitermachen und die Einzelheiten darüber kennen lernen, wie einzelne Webseiten geschrieben, Verbindungen zwischen ihnen hergestellt und Grafiken sowie andere Medien eingefügt werden, um die Präsentation für Ihr Publikum zu verbessern.

Workshop

Im ersten Teil des Workshops werden einige Fragen aufgelistet, die man sich beim Planen einer Webseite öfters stellt. Zu jeder Frage gibt es natürlich auch eine Antwort. Danach haben sie die Gelegenheit, einige Quiz-Fragen selbst zu beantworten. Wenn Sie Probleme beim Finden einer passenden Antwort haben, sehen Sie im nächsten Absatz nach, dort finden Sie die Lösungen. In der Übung werden Sie ermutigt, selbst einige Ideen zu Ihrer eigenen Webseite zu bekommen.

Fragen und Antworten

- F Das sieht alles nach ziemlich viel Arbeit aus. Alles, was ich will, ist, etwas Einfaches zu machen, und Sie erzählen mir, ich müsse Ziele haben und Themen und Storyboards.**
- A** *Wenn Sie etwas Einfaches machen, dann werden Sie nicht allzuviel, wenn überhaupt etwas, von dem berücksichtigen müssen, was ich in diesem Kapitel empfohlen habe. Aber sobald Sie einmal über zwei oder drei oder noch mehr querverbundene Seiten sprechen, hilft es wirklich, wenn Sie einen Plan haben, bevor Sie anfangen. Wenn Sie einfach loslegen, werden Sie feststellen, dass es eine beängstigende Aufgabe ist, alles zusammenhängend im Kopf haben zu müssen. Und das Resultat könnte nicht das sein, das Sie erwartet haben; es macht es schwer für die Leute, an die Informationen aus Ihrer Präsentation zu kommen, die sie haben wollen, und schwierig für Sie, alles so zu reorganisieren, dass es sinnvoll ist. Einen Plan zu haben, bevor man anfängt, tut nicht weh und könnte, auf Dauer gesehen, viel Zeit sparen.*

F Sie haben in diesem Kapitel eine Menge über das Organisieren von Themen und Seiten gesprochen, aber nichts über die Gestaltung und das Layout der einzelnen Seiten gesagt.

A *Ich bespreche das später in diesem Buch, nachdem Sie mehr darüber wissen, was HTML (die Sprache, in der Webseiten geschrieben werden) in Bezug auf Layout beherrscht und was nicht. Es gibt ein ganzes Kapitel und mehr über Seiten-Layout und Gestaltung am sechsten Tag.*

F Was ist, wenn ich nun keine der grundlegenden Strukturen mag, über die Sie in diesem Kapitel gesprochen haben?

A *Die Gestaltung liegt ganz bei Ihnen. Solange Ihre Leser finden können, was sie finden wollen, oder tun, was Sie möchten, dass sie es tun können, gibt es keine Regel, die besagt, man muss eine Hierarchie verwenden oder eine lineare Struktur. Ich präsentiere diese Strukturen nur als mögliche Ideen zum Organisieren Ihrer Webseiten.*

Quiz

1. Wie würden Sie ganz kurz die Begriffe Web-Site, Web-Server und Webseiten erklären?
2. Was bedeutet der Begriff Homepage im Sinne des Web-Publishing?
3. Was ist der nächste Schritt zum Kreieren einer Homepage, wenn Sie schon eine bestimmte Vorstellung haben?
4. Es gibt ein Link, das, egal welche Navigationsstruktur Sie auf Ihrer Web-Site verwenden, auf jeder Ihrer Webseiten erscheint. Warum?
5. Was ist der Zweck eines Storyboard?

Antworten

1. Eine Web-Site besteht aus einer oder mehreren Webseiten, die sinnvoll miteinander verknüpft sind. Ein Web-Server ist ein Rechner, der die Web-Site bereithält. Webseiten sind einzelne Elemente einer Web-Site. Man könnte dies auch mit einer Seite in einem Buch vergleichen.
2. Eine Homepage ist im Sinne des Web Publishing der Ausgangspunkt zu den restlichen Seiten Ihrer Web-Site (Startseite oder Inhaltsverzeichnis)
3. Nachdem Sie eine Vorstellung haben, wie Ihre Web-Site in etwa aussehen soll, sollten Sie versuchen den Inhalt nach Themen oder in Absätzen zu gliedern.

4. Sie sollten ein Link zu Ihrer Homepage auf jeder Seite Ihrer Web-Site integrieren. So können Benutzer immer wieder zurückfinden, wenn sie sich verirrt haben.
5. Ein Storyboard liefert einen Gesamtumriss Ihrer Web-Site, der zeigt wie diese aussehen soll, wenn sie fertig ist. So können Sie Ihre Webseiten leichter organisieren. Storybards lohnen sich bei größeren Web-Sites am meisten.

Übungen

1. Erstellen Sie als Erstes eine Liste mit verschiedenen Ideen für Ihre Webseite, die Ihre Besucher interessieren würden. Je präziser diese Vorstellungen, desto besser.
2. Nachdem Sie diese Liste erstellt haben, besuchen Sie im Internet die Web-Sites, die sich mit ähnlichen Themen befassen, die Sie in Ihrer Webseite veröffentlichen wollen. Wenn Sie diese Sites genauer betrachten, achten Sie darauf, ob sie leicht zu navigieren sind und einen guten Inhalt haben. Erstellen Sie anschließend eine Liste der Punkte, die Ihnen an diesen Sites gefallen. Was würden Sie an Ihrer Web-Site besser machen?



Eine Einführung in HTML

**Woche
1**

Nach dem gestrigen Tag mit einer Menge Text zum Lesen und Konzepten zum Verdauen fragen Sie sich wahrscheinlich, wann Sie denn nun endlich eine richtige Webseite schreiben werden. Schließlich haben Sie ja deshalb dieses Buch gekauft. Willkommen am zweiten Tag! Heute erfahren Sie, wie man Webseiten gestaltet, und lernen HTML kennen, die Sprache, in der Web-Dokumente geschrieben werden, und erfahren im einzelnen das Folgende:

- ▶ Was HTML ist und warum Sie sie benutzen müssen
- ▶ Was bei der Gestaltung von HTML-Dokumenten möglich ist und was nicht
- ▶ HTML-Kennungen (»Tags«): Was ist das, und wie werden sie benutzt?

Was HTML ist ... und was nicht

Bevor Sie damit loslegen, tatsächlich etwas in HTML zu schreiben, sollten Sie wissen, was HTML ist, was HTML kann, und besonders, was HTML nicht kann. HTML steht für *Hypertext Markup Language* (Hypertext-Auszeichnungssprache). HTML basiert auf SGML (die *Standard Generalized Markup Language*), bei der es sich um ein viel umfassenderes System zur Bearbeitung von Dokumenten handelt. Um HTML-Seiten zu schreiben, brauchen Sie nicht viel über SGML zu wissen, es ist jedoch hilfreich, wenn Sie verstehen, dass eine der Haupteigenschaften von SGML darin besteht, dass sie die generelle Struktur des Inhalts innerhalb eines Dokuments beschreibt, nicht jedoch die endgültige Darstellungsweise des Inhalts auf der Seite oder auf dem Bildschirm. Dies könnte neu für Sie sein, wenn Sie daran gewöhnt sind, mit grafischen Editoren (Editoren, die nach dem WYSIWYG-Prinzip arbeiten = What You See Is What You Get, das heißt, Sie bekommen das, was Sie sehen) zu arbeiten.

HTML beschreibt die Struktur einer Seite

HTML ist, kraft ihres SGML-Erbes, eine Sprache, welche die Struktur eines Dokuments beschreibt und nicht seine tatsächliche Gestaltung. Die Theorie dahinter ist, dass die meisten Dokumente gemeinsame Elemente besitzen – zum Beispiel Titel, Absätze oder Listen –, und wenn Sie einige solcher Elemente, die ein Dokument beinhaltet, bevor Sie mit dem Schreiben beginnen, definieren, können Sie diese Teile des Dokuments mit ihren jeweiligen Namen bezeichnen (vgl. Abbildung 3.1).

Wenn Sie schon einmal mit Textverarbeitungsprogrammen gearbeitet haben, die so genannte »Dokumentvorlagen« (*style sheets*) (wie Microsoft Word) oder Absatz-Kataloge (wie FrameMaker) verwenden, dann haben Sie bereits etwas Ähnliches gemacht; jeder Abschnitt des Textes richtet sich nach einer der »Vorlagen« (*styles*), die bereits definiert wurden, bevor Sie mit der Arbeit beginnen.

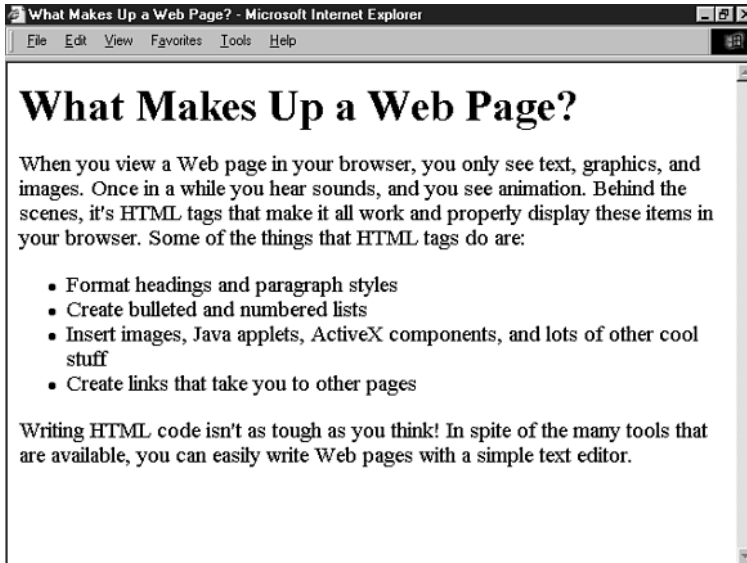


Abbildung 3.1:
Elemente eines
Dokuments

HTML beschreibt eine Reihe von gemeinsamen Webseiten-Formaten: Überschriften, Absätze, Listen und Tabellen. Sie beschreibt auch Schriftarten wie Fettdruck und Code-Beispiele. Jedes Element hat einen Namen und ist in einem HTML-Tag (»tag«, *Kennung*) enthalten. Wenn Sie eine Webseite in HTML erstellen, dann bezeichnen Sie die unterschiedlichen Elemente Ihrer Seite mit diesen Kennungen, die zum Beispiel so aussehen können: »dies ist eine Überschrift« oder »dies ist ein Listenpunkt.« Es ist so, als arbeiteten Sie für eine Zeitung oder ein Magazin, wo Sie zwar schreiben, aber jemand anderer sich um das Layout kümmert; Sie erklären dann vielleicht dem Layouter, dass diese Zeile den Titel darstellt, jene eine Bildüberschrift und eine weitere eine Überschrift. Genauso ist es mit HTML.

HTML beschreibt das Seiten-Layout nicht

Wenn Sie mit einem Textverarbeitungs- oder Seiten-Layoutprogramm arbeiten, beinhalten die Formate nicht nur die Seiteninformationen, vielmehr beschreiben sie auch die Schriftarten, die Größe, die Einrückung, Unterstreichungen usw. Wenn Sie einen Text also als Überschrift definieren möchten, können Sie das Überschriftsformat darauf anwenden, und das Programm wird den Absatz automatisch für Sie im korrekten Stil formatieren.

HTML geht nicht so weit. Im Großen und Ganzen sagt HTML nichts darüber aus, wie eine Seite aussieht, wenn sie betrachtet wird. Alles, was die HTML-Kennungen angeben, ist, dass es sich beim jeweiligen Element um eine Überschrift oder eine Liste han-

delt – sie sagen nichts darüber aus, wie diese Überschrift oder Liste denn nun formatiert werden soll. Also ist es genauso wie im Beispiel des Magazinartikels und des Setzers, der Ihren Artikel formatiert; es ist der Job des Setzers, darüber zu entscheiden, wie groß die Überschrift sein und in welcher Schriftart sie gesetzt werden sollte – das einzige, worum Sie sich kümmern müssen, ist, zu markieren, welcher Absatz als Überschrift dienen soll.



Auch wenn HTML nicht viel über das Layout einer Seite sagt, ermöglichen Ihnen Cascading Style Sheets (CSS) eine verbesserte Kontrolle über das Format, als es bei HTML-Tags der Fall ist. In HTML 4 gibt es viele Veränderungen, die das Verwenden von CSS-Tags begünstigen. Sie werden mehr über CSS am Tag 4 und am Tag 10 lernen.

Web-Browser dienen nicht nur der Abwicklung der Netzwerkfunktionen wie dem Laden von Dokumenten aus dem Netz, sie formatieren auch HTML-Dokumente. Wenn Sie ein HTML-Dokument mit einem Browser wie Netscape oder Lynx laden, liest und/oder parst (parse = Zergliederung in die Bestandteile und damit verbundene Kontrolle der Syntax) der Browser die HTML-Informationen und stellt den Text und die Abbildungen auf dem Bildschirm formatiert dar. Der Browser erstellt aus den Namen der Seitenelemente die tatsächlichen Formate auf dem Bildschirm; Überschriften könnten dann zum Beispiel in einer größeren Schrift als der restliche Text auf der Seite erscheinen. Der Browser packt den Text auch dementsprechend zusammen, sodass er in die gegenwärtige Fensterbreite hineinpasst.

Verschiedene Browser, die auf verschiedenen Plattformen operieren, können für dieselben Seitenelemente unterschiedliche Formatkonvertierungen benutzen. Einige Browser verwenden vielleicht andere Schriftarten als die übrigen. So könnte ein Browser zum Beispiel Kursivschrift auch kursiv darstellen, während ein anderer sie als invertierten oder unterstrichenen Text auf Systemen darstellt, die keine kursive Darstellung unterstützen. Oder er könnte eine Überschrift mit Großbuchstaben darstellen, anstatt eine größere Schrift einzusetzen. Für Sie als Webseiten-Designer bedeutet das, dass die Seiten, die Sie mit HTML gestalten, auf unterschiedlichen Systemen und mit unterschiedlichen Browsern ganz anders aussehen können. Die tatsächlichen Informationen und Verknüpfungen innerhalb der Seiten werden noch vorhanden sein, aber ihre Darstellung auf dem Bildschirm wird sich ändern. Sie können eine Webseite so entwerfen, dass sie auf Ihrem Computersystem perfekt aussieht, wenn man sie jedoch auf einem anderem System betrachtet, kann sie vollkommen anders aussehen (vielleicht sogar unlesbar sein).

Warum dies so ist

Wenn Sie gewohnt sind, auf Papier zu schreiben und zu entwerfen, dann wird Ihnen dieses Konzept fast pervers vorkommen. Keine Kontrolle über das Seiten-Layout? Das ganze Design kann sich ändern, je nachdem, wo die Seite betrachtet wird? Das ist ja schrecklich! Warum in aller Welt sollte es wohl so funktionieren?

Erinnern Sie sich noch daran, dass ich im ersten Kapitel erwähnte, dass eine der erfreulichen Eigenschaften des Web die ist, dass es plattformübergreifend ist und dass Webseiten auf jedem beliebigen Computersystem, mit jeder beliebigen Bildschirmgröße und beliebiger Grafikfähigkeit dargestellt werden können? Wenn Sie möchten, dass Ihre Seiten von jedermann in der Welt gelesen werden können, dann können Sie sich nicht darauf verlassen, dass Ihre Leser das gleiche Rechnersystem, einen genauso großen Bildschirm, die gleiche Anzahl von Farben oder dieselbe Schriftart wie Sie haben. Das Web zieht alle diese Unterschiede in Betracht und erlaubt es allen Browsern und Rechnersystemen, sich auf einer gemeinsamen Ebene zu treffen.

Das Web, als Design-Medium, ist keine neue Art von Papier. Das Web ist ein völlig neues Medium mit neuartigen Beschränkungen und Absichten, die sich vom Umgang mit Papier sehr unterscheiden. Die wichtigsten Regeln beim Design von Webseiten, die ich Ihnen in diesem Buch näher bringen möchte, sind folgende:

Was Sie tun sollten	... und was nicht
Kreieren Sie Ihre Seiten so, dass sie in den meisten Web-Browsern funktionieren.	Orientieren Sie sich beim Kreieren Ihrer Webseiten auf keinen Fall nach ihrem Aussehen im Computersystem und im Browser.
Achten Sie auf klaren, gut strukturierten Inhalt, der leicht lesbar und verständlich ist.	

Durch das ganze Buch hindurch werde ich Ihnen immer wieder Beispiele von HTML-Code und wie dieser aussieht, wenn er dargestellt wird, zeigen. In vielen Beispielen werden ich Ihnen im Vergleich zeigen, wie der Code in zwei sehr unterschiedlichen Browsern aussieht: unter Netscape, dem wohl momentan beliebtesten Browser auf dem Markt, und unter Lynx, einem Browser, der auf rein textbasierten Terminals läuft und der nicht so beliebt, aber immer noch weithin gebräuchlich ist. Durch diese Beispiele werden Sie einen Eindruck davon bekommen, wie unterschiedlich ein und dieselbe Seite in verschiedenen Browsern aussehen kann.



Obwohl die Regel, das Design nicht über das Erscheinungsbild sondern über die Struktur zu entwickeln, der richtige Weg ist, um gutes HTML zu erstellen, werden Sie überrascht sein, wenn Sie durch das Web surfen. Beim Gros der Web-Sites scheint es so, als seien sie mit Überlegungen zum Er-

scheinungsbild entworfen worden – meistens dem Erscheinungsbild in einem bestimmten Browser, wie zum Beispiel dem Netscape Navigator oder Microsoft Internet Explorer. Lassen Sie sich von diesen Designs nicht beeinflussen. Wenn Sie sich an die Regeln, die ich vorschlage, halten, werden Ihre Webseiten und Web-Sites am Ende die insgesamt erfolgreicher sein – ganz einfach deswegen, weil mehr Leute diese auf einfache Weise lesen und verwenden können.

HTML ist eine Auszeichnungssprache

HTML ist eine *Auszeichnungs-* bzw. *Kennungssprache* (»markup language«). In einer Kennungssprache zu schreiben bedeutet, dass Sie mit dem Text Ihrer Seite anfangen und dann spezielle Kennungszeichen um Worte und Absätze setzen. Wenn Sie bereits mit anderen Kennungssprachen wie troff oder LaTeX oder sogar mit älteren DOS-basierten Textverarbeitungsprogrammen gearbeitet haben, bei denen Sie spezielle Codes für solche Befehle wie »Fettdruck« einsetzen mussten, dann wird Ihnen dies nicht fremd vorkommen.

HTML hat einige Kennungen (tags) definiert, die Sie benutzen können. Sie können allerdings keine eigenen Tags erfinden, um neue Darstellungsweisen oder Merkmale zu erschaffen. Und, um die Sache noch richtig schwierig zu machen, es unterstützen unterschiedliche Browser unterschiedliche Arten von Tags. Zum besseren Verständnis werfen Sie einen kurzen Blick auf die HTML-Geschichte

Eine kurze Geschichte zu den HTML-Tags

Gewissermaßen der »Basiswortschatz« von HTML-Tags wird als HTML 2.0 bezeichnet. HTML 2.0 ist der gegenwärtige Standard für HTML (es gibt eine schriftliche Festlegung davon, die vom W3-Konsortium entwickelt wurde und aufrechterhalten wird) und beschreibt die Tags, die die Browser verstehen müssen. In den nächsten paar Kapiteln werden Sie vor allem etwas über HTML-2.0-Tags lernen, die Sie überall anwenden können.

Die HTML-Spezifikation wurde ungefähr 1996 entwickelt. Verschiedene Software-Hersteller, unter anderem IBM, Microsoft, Netscape Communications Corporation, Novell, Softquad, Spyglass und Sun Microsystems schlossen sich dem W3-Consortium an, um Spezifikationen zu entwickeln. So wurden HTML 3.2 Features wie Tabellen, Applets und textumfließende Bilder hinzugefügt. HTML 3.2 war voll abwärtskompatibel zu Standard- HTML 2.0.



Die in HTML 3.2 integrierten Verbesserungen werden später in diesem Buch behandelt. Am Tag 11 werden Sie mehr über Tabellen lernen. Am Tag 13 wird Ihnen gezeigt, wie sie Java Applets verwenden.

HTML 4.0, das 1997 auf dem Markt erschien, enthielt viel neue Features, die eine bessere Kontrolle als bei HTML 2.0 und 3.2 beim Designen der Seiten ermöglichten. Wie bei HTML 2.0 und 3.2 unterstützt das W3-Consortium Standard-HTML 4.0. Während Internet Explorer 4 und Netscape Navigator 4 beide die meisten HTML 4.0 Features unterstützen, werden Benutzer mit älteren Browsern HTML 4.0-Features wie Cascading Style Sheets und Dynamic HTML nicht sehen können.



Cascading Style Sheets und Dynamic HTML sind zusätzliche Web-Technologien, die in Verbindung mit HTML arbeiten und Ihnen zusätzliche Kontrolle über das Erscheinungsbild Ihrer Webseite geben. Style-Sheets werden detaillierter am Tag 10 besprochen. Werfen Sie einen Blick auf Tag 15, um eine kurze Information über die Fähigkeiten von Dynamic HTML zu bekommen.

Framesets (die in Netscape 2.0 integriert wurden) und Floating frames (die im Internet Explorer 3.0 eingeführt wurden) wurden offizieller Teil der HTML 4.0-Spezifikationen. Framesets werden am Tag 12 detaillierter besprochen. Man kann auch Verbesserungen im Bereich der Tabellenformatierung und Wiedergabe feststellen. Die größte Änderung in HTML 4.0 war jedoch die hohe Integration von Style Sheets.



Wenn Sie an der Entwicklung von HTML und den genauen Vorgängen im W3-Consortium interessiert sind, besuchen Sie die folgenden Seiten auf der Site des W3C: <http://www.w3.org/pub/www/MarkUp/>.

Zusätzlich zu den Tags, die in den verschiedenen HTML-Versionen definiert werden, implementieren einzelne Browser-Unternehmen browserspezifische Ausweitungen zu HTML. Microsoft und Netscape sind besonders an der Erstellung von Erweiterungen beteiligt und sie bieten viele neue Features an, die nur von ihren Browsern interpretiert werden.

Sind Sie schon verwirrt? Dann stehen Sie nicht alleine da. Sogar Web-Designer mit jahrelanger Erfahrung und Hunderten von publizierten Seiten kämpfen mit dem Problem, welche Tags sie auswählen sollen, um ein Optimum zwischen weitestgehender Unterstützung eines Designs (mit HTML 2.0) und der Möglichkeit, mehr Flexibilität im Layout zu erzielen, dafür aber weniger Übereinstimmung zwischen den Browsern zu haben (mit HTML 3.2- und 2.0 Level-Tags). Alles dies im Blick zu behalten, kann einen sehr verwirren. Im weiteren Verlauf des Buches werde ich Sie, sobald ich ein neues HTML-Tag einführe, darauf hinweisen, zu welcher HTML-Version es gehört, wie weit es unterstützt wird und wie es bei den verschiedenen Browsern am besten eingesetzt werden kann.

Später in diesem Buch werden Sie auch Hinweise dazu bekommen, wie Sie mit den unterschiedlichen HTML-Tags umzugehen haben, sodass Ihre Seiten in allen Arten von Browsern gut lesbar sind und gut aussehen.

Mit XHTML 1.0 für die Zukunft gerüstet sein

Das Internet besteht nicht mehr nur aus Soft- und Hardware. Mit WebTV haben Sie Zugriff auf das Internet. Ein Grund mehr, zum Stubenhocker zu werden. Personal Information Manager und Palmtop-Computer ermöglichen Ihnen den Zugriff auf das Internet während Sie unterwegs sind. Im zweiten Jahrtausend ist das Internet eine der wichtigsten Massenkommunikations- und Informationsplattformen.

Viele der neuen Technologien stellen trotz allem ein Problem für die älteren HTML-Spezifikationen dar. Sie haben nicht dieselbe Arbeitsgeschwindigkeit wie ein Desktop Computer und können deshalb nicht so fehlertolerant mit dem Code umgehen, wie das Browser bei HTML tun. Die Entwickler der HTML-Spezifikationen versuchten, dieses Problem in den Griff zu bekommen und die Reduzierungen von HTML wurden offensichtlich. Wir drehen, ändern und verzerren HTML weit hinter seinen Fähigkeiten. Das Ergebnis ist, dass es wahrscheinlich kein HTML 5 geben wird.

Die Zukunft des Internets verlangt eine Sprache, die leichter zu erweitern und einfacher zu handhaben als HTML ist. Eine neue Richtung wird eingeschlagen: die Verwendung von XML (Extensible Markup Language), eine Untergruppe von SGML, die das erstellen eigener Tags ermöglicht. Und da kommt auch schon XHTML 1.0 ins Spiel.

XHTML 1.0 ist in XML geschrieben und ist der neue Standard schlechthin, der Web-Designern die Zukunft erleichtern wird. Wenn Sie sich nun wundern, ob Sie gerade ein Buch über HTML 4 und nicht über XHTML 1.0 in der Hand halten, kann ich Sie beruhigen:

Was das Technische angeht, so sind XHTML 1.0 und HTML 4 sehr ähnlich. Die Tags und Attribute sind virtuell gesehen dieselben, aber es gibt ein paar ganz einfache Regeln, die man für die XHTML 1.0-Spezifikationen beachten muss. Ich werde Ihnen in diesem Buch immer wieder Tipps geben, wie man mit den verschiedenen HTML-Tags arbeiten muss, um sicherzugehen, dass Ihre Seiten lesbar sind, und in allen Browsern gut aussehen.

So sehen HTML-Dateien aus

In HTML geschriebene Seiten bestehen aus schlichtem Text (ASCII), was bedeutet, dass sie keine plattform- oder programmspezifischen Informationen enthalten. Deshalb können auch einige Editoren, die Text unterstützen (eigentlich sollte das jeder Editor tun – mehr dazu ist weiter hinten in diesem Kapitel zu finden) diese Seiten lesen. HTML-Dateien bestehen aus folgenden Elementen:

- ▶ den Text der Seite selbst,
- ▶ HTML-Tags, welche die Elemente der Seite, Strukturen, Formatierung und Hypertext-Verknüpfungen mit anderen Dokumenten oder eingefügten Medien kennzeichnen.

Die meisten HTML-Tags sehen ungefähr so aus:

```
<thetaname> affected text </thetaname>
```

Der Name des Tag ist in spitzen Klammern (<>) eingeschlossen. HTML-Tags haben im Allgemeinen eine Anfangs- und eine Ende-Kennung, womit sie den Text umgeben, den sie beeinflussen. Die Anfangs-Kennung »schaltet« ein Merkmal ein (etwa Überschriften, Fettdruck und so weiter), und die Ende-Kennung schaltet es wieder aus. Ende-Kennungen enthalten generell den Namen der Tags, eingeleitet durch einen Schrägstrich (/). Das einleitende Tag (zum Beispiel <p> für Paragraphen) und das abschließende Tag (zum Beispiel </p> für Paragraphen) sind zusammen das, was man offiziell als HTML-Elemente bezeichnet.



HTML-Tags (HTML-Kennungen) sind die Dinge innerhalb von spitzen Klammern (<>), die die Merkmale oder Elemente einer Seite anzeigen.



Eine Warnung vorweg: Achten Sie auf den Unterschied zwischen dem normalen Schrägstrich (Slash, »/«), wie er oben erwähnt wurde und dem inversen Schrägstrich (Backslash, »\«), wie er in Pfadangaben unter DOS und Windows verwendet wird (z.B. C:\Windows etc.). Wenn Sie in einer HTML-Anweisung versehentlich den Backslash anstelle des Slash verwenden, erkennt der Browser die schließenden Tags nicht.

Doch nicht alle HTML-Tags haben einen Beginn und ein Ende. Manche Tags sind leer, und wieder andere Tags sind »container« (Behälter), die weitere Informationen und Text innerhalb der Klammern enthalten. Bei XHTML jedoch ist es erforderlich, dass alle Tags geschlossen werden bzw. ein Ende haben.

Sie werden diese Tags im Verlauf des Buchs kennen lernen.

Ein weiterer Unterschied zwischen HTML 4 und XHTML 1.0 ist die Verwendung von Tags und Attributen in Kleinbuchstaben.

Alle HTML-Tags berücksichtigen keine Groß- und Kleinschreibung; das bedeutet, Sie können sie in Groß- oder Kleinbuchstaben schreiben oder in irgendeiner Mischung. Also ist `<html>` das gleiche wie `<HtMl>` und das wiederum das gleiche wie `<HTML>`. Das ist bei XHTML 1.0, wo alle Tags und Attribute in Kleinbuchstaben geschrieben werden, nicht der Fall. Damit Sie sich schon einmal daran gewöhnen können, werden alle Tag- und Attributnamen aus den Beispielen in diesem Buch klein geschrieben.

Übung 3.1: Erstellen Sie Ihre erste HTML-Seite

Sie haben jetzt gesehen, wie HTML aussieht – nun sind Sie an der Reihe. Lassen Sie uns mit einem recht einfachen Beispiel beginnen, sodass Sie ein Grundgefühl für HTML bekommen.

Um ein HTML-Dokument zu schreiben, brauchen Sie keinen Web-Server, Web-Provider und noch nicht mal eine Verbindung zum Web selbst. Alles, was Sie wirklich brauchen, ist etwas, womit Sie Ihre HTML-Dateien schreiben können, und noch mindestens einen Browser, mit dem Sie sie anschauen können. Sie können ganze Reihen von Webseiten schreiben, verknüpfen und testen, ohne jemals mit einem Netzwerk in Berührung zu kommen. Tatsächlich ist es das, was wir im Hauptteil dieses Buchs unternehmen werden – ich werde später darüber reden, wie Sie alles im Web publizieren, sodass andere es auch sehen können.

Zuerst brauchen Sie einen Texteditor. Ein Text-Editor ist ein Programm, das Dateien im ASCII-Format erzeugen kann. ASCII-Format ist einfach reiner Text, ohne Schriftformatierung oder Spezialzeichen. Unter Windows sind Notepad, Microsoft Write bzw. WordPad und DOS edit gute, einfache Texteditoren (diese kommen kostenlos mit dem System!). Shareware Text-Editors sind für verschiedene Betriebssysteme, inklusive DOS, Windows 3.1, Windows 95/98, Windows NT, Macintosh und LINUX erhältlich. Wenn Sie <http://www.download.com> im Web-Browser eingeben und Text-Editors als Such-Item eingeben, werden Sie viele Informationen erhalten, die Sie downloaden können.

Wenn Sie nur ein Textverarbeitungsprogramm wie Microsoft Word haben, brauchen Sie keine Panik zu bekommen. Sie können Seiten in Textverarbeitungsprogrammen genauso gut wie in Texteditoren schreiben, obwohl dies ein wenig komplizierter sein kann. Wenn Sie den Befehl »Speichern« oder »Speichern unter« geben, sehen Sie ein Menü mit verschiedenen Formaten, unter denen Sie die Datei speichern können. Eines davon sollte »Nur Text«, »Text mit Zeilenumbrüchen« oder »DOS- Text« sein. Alle diese Optionen speichern Ihre Datei als reinen ASCII-Text, genauso, wie es auch ein

Text-Editor täte. Wenn Sie bei HTML-Dateien die Wahl zwischen DOS-Text und reinem Text haben, wählen Sie DOS-Text, und benutzen Sie auch die Zeilenumbruch-Option, wenn möglich.



Wenn Sie ein Textverarbeitungsprogramm für Ihre HTML-Entwicklung verwenden, seien Sie sehr vorsichtig. Viele neue Textverarbeitungsprogramme beinhalten einen Modus für HTML oder XML Mechanismen, um HTML-Code zu schreiben. Das Textverarbeitungsprogramm kann Ihnen das HTML-Kodieren auf einmal ungewollt abnehmen oder Sie mysteriöserweise, ohne Vorwarnung, in diesen Modus umschalten. Dies kann ungewöhnliche Resultate zur Folge haben. Wenn Sie Probleme mit einem Textverarbeitungsprogramm haben, probieren Sie es einfach mit einem Texteditor, und sehen Sie, ob das besser ist.

Wie sieht es nun aber mit der Flut der kostenlosen und kommerziellen HTML-Editoren aus, die behaupten, Ihnen dabei zu helfen, HTML einfacher zu schreiben? Die meisten von ihnen sind einfache Texteditoren mit einigen Buttons, die die Kennungen (Tags) für Sie einsetzen. Wenn Sie bereits einen besitzen, können Sie ihn ruhig benutzen. Wenn Sie jedoch einen ausgefallenen Editor haben, der von sich behauptet, alle HTML-Befehle für Sie zu verstecken und alle Arbeit für Sie zu erledigen, so legen Sie ihn mal für ein paar Tage beiseite, und verwenden Sie für eine kleine Weile einen rein textbasierten Editor.

Starten Sie den Texteditor, und schreiben Sie folgenden Code. Sie müssen an dieser Stelle noch nicht verstehen, was das alles bedeutet; Sie lernen das später in diesem Kapitel. Dies ist nur ein einfaches Beispiel, um zu beginnen:

```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>
<head>
<title>My Sample HTML page</title>
</head>
<body>
<h1>This is an HTML Page</h1>
</body>
</html>
```



In diesem Beispiel, sowie auch in anderen Beispielen in diesem Buch, werden die HTML-Tags dunkler als der übrige Text geschrieben, damit sie die Tags leicht erkennen können. Wenn Sie Ihre eigenen HTML-Dateien erstellen, wird der ganze Text in einer einheitlichen Farbe erscheinen (es sei denn, Sie verwenden einen speziellen HTML-Editor, der Tags in bestimmten Farben erscheinen lässt).

Nachdem Sie Ihre HTML-Datei nun erzeugt haben, speichern Sie sie auf der Platte – und denken Sie daran, wenn Sie ein Textverarbeitungssystem benutzen, sie als reinen ASCII-Text zu speichern. Wenn Sie einen Namen für die Datei auswählen, gibt es zwei Regeln, die Sie befolgen müssen:

- ▶ Der Datei-Name sollte die Endung `.html` haben (oder `.htm` auf DOS- oder Windows-Systemen, die nur Extensionen mit drei Stellen haben); beispielsweise `Mein-Beispiel.html` oder `Text.html` oder `Index.htm`. Die meiste Web-Server-Software wird Dateien mit diesen Extensionen erwarten, also sollten Sie sich schon jetzt daran gewöhnen, sie zu benutzen.
- ▶ Benutzen Sie kurze, einfache Namen. Fügen Sie keine Leerstellen oder Sonderzeichen (Merkmale, hervorgehobene Zeichen) ein – es reicht vollkommen, Buchstaben und Ziffern zu verwenden.

Übung 3.2: Sehen Sie sich das Resultat an

Nun, da Sie eine HTML-Datei haben, starten Sie Ihren Web-Browser. Sie müssen jetzt nicht mit dem Netzwerk verbunden sein, da Sie ja keine Dokumente auf irgendeinem anderen System öffnen wollen (obwohl Ihr Browser nach einer Verbindung verlangen könnte; oft wird er sich aber damit abfinden, keine Verbindung zu haben, und weiter arbeiten).



Abbildung 3.2:
Eine Beispiel
HTML-Datei

Danach, wieder in Ihrem Browser, wählen Sie AKTUALISIEREN oder sein Äquivalent (NEU LADEN für Internet Explorer). Der Browser wird nun die neue Version Ihrer Datei lesen, und voilà, Sie können editieren und ansehen und editieren und ansehen, bis Sie alles richtig hingekriegt haben.

Wenn Sie den tatsächlichen HTML-Text in Ihrem Browser zu sehen bekommen und nicht den Inhalt der Abbildung 3.2, dann versichern Sie sich, dass Ihre HTML-Datei eine `.html`- oder `.htm`-Extension hat. Diese Datei-Extension ist wichtig, denn sie zeigt Ihrem Browser an, dass es sich um eine HTML-Datei handelt.

Wenn die Dinge absolut nicht stimmen – wenn Sie einen leeren Bildschirm zu sehen bekommen oder seltsame Zeichen sehen –, dann stimmt etwas mit Ihrer Originaldatei nicht. Wenn Sie ein Textverarbeitungsprogramm benutzt haben, um Ihre Dateien zu editieren, versuchen Sie zur Abwechslung, Ihre gespeicherte HTML-Datei in einem rein textbasierten Editor zu öffnen (hier werden NotePad oder SimpleText wieder gute Dienste leisten). Wenn der Text-Editor es nicht lesen kann oder das Resultat vermurkst ist, dann haben Sie die Originaldatei nicht im richtigen Format gespeichert. Gehen Sie zu Ihrem Originaleditor zurück, und versuchen Sie noch einmal, die Datei als »Nur Text« zu speichern, und probieren Sie sie dann noch mal in Ihrem Browser aus, bis Sie es richtig hinkriegen.

Ein Hinweis zum Formatieren

Wenn ein HTML-Dokument von einem Browser untersucht wird, werden alle Formatierungen, die Sie womöglich von Hand gemacht haben – wie etwa zusätzliche Leerzeichen, Tabulatoren, Zeilenumbrüche –, ignoriert. Das einzige, das ein HTML-Dokument formatiert, ist eine HTML-Kennung (HTML-Tag). Wenn Sie Stunden damit verbracht haben, eine Text-Datei so zu editieren, dass sie optimal formatierte Absätze hat und Spalten mit Zahlen, aber keinerlei HTML-Kennungen eingefügt haben, dann wird der Text, wenn Sie das Dokument in einen HTML-Browser laden, insgesamt in einem einzigen Absatz zusammenfließen, und all Ihre Arbeit wird vergeblich gewesen sein.



Es gibt eine Ausnahme dieser Regel: eine Kennung namens `<pre>`. Sie erfahren morgen in Kapitel 5 mehr über diesen Tag.

Der Vorteil des Ignorierens aller so genannten »white spaces« (Leerzeichen, Tabulatoren, Zeilenumbrüche) besteht darin, dass Sie Ihre Tags einfügen können, wo immer Sie mögen.

Die folgenden Beispiele führen alle zum gleichen Ergebnis. (Probieren Sie es!)

```
<h1>If music be the food of love, play on.</h1>
<h1>
If music be the food of love, play on.
</h1>
<h1>
If music be the food of love, play on.      </h1>
<h1>  If music  be  the  food of love,
play  on. </h1>
```

Programme, die Ihnen beim Schreiben von HTML helfen

Sie denken vielleicht, dass das Herumgetue mit den Tags eine wahre Pein ist, besonders dann, wenn Sie das kurze Beispiel nicht gleich beim ersten Mal richtig zum Laufen bekommen haben. (Ärgern Sie sich deshalb nicht; ich habe das Beispiel auch nicht gleich beim ersten Mal hingekriegt, und ich habe es immerhin entworfen.) Sie müssen an all die Tags denken. Und Sie müssen sie richtig eingeben und auch beenden. Was für ein Aufwand!

Was ist mit WYSIWYG-Editoren? Das Problem ist, dass es kein echtes WYSIWYG gibt (WYSIWYG: *What You See Is What You Get*, »Was Sie sehen, ist, was Sie kriegen«, na ja, fast...), wenn Sie mit HTML arbeiten, denn WYG (»Was Sie kriegen«) hängt stark vom Browser ab, der benutzt wird, um Ihr Dokument zu lesen. Wenn Sie sich also dessen bewusst sind, dass die Ergebnisse Ihrer Arbeit von Editor zu Editor variieren können, dann können Ihnen WYSIWYG-Editoren bei der Erstellung von HTML-Dateien viel Zeit sparen. Wenn Sie jedoch professionelle Web-Entwicklung betreiben und viele erweiterte Merkmale anwenden, sind WYSISWYG-Editoren oft unzureichend und Sie müssen sowieso »unter die Haube« gehen, um mit dem HTML-Code zu spielen. Auch wenn Sie beabsichtigen, einen WYSIWYG-Editor für den Hauptteil Ihrer Arbeit einzusetzen, empfehle ich Ihnen, dass Sie für ein paar Tage Geduld üben und diese Beispiele in Texteditoren ausprobieren, damit Sie ein Gefühl dafür bekommen, was HTML wirklich ist, bevor Sie sich dazu entschließen, zu einem Editor überzuwechseln, der die Kennungen versteckt.

Neben den HTML-Editoren gibt es auch Konverter, welche die Dateien vieler populärer Textverarbeitungsprogramme einlesen und sie nach HTML konvertieren. Mit einem Satz von Vorlagen (*templates*) könnten Sie Ihre Dokumente komplett mit dem Programm schreiben, das Sie üblicherweise verwenden, und das Ergebnis dann konvertieren.

In vielen Fällen können Konverter außerordentlich hilfreich sein, insbesondere, wenn Sie bereits vorhandene Dokumente möglichst schnell ins Web stellen wollen. Gleichwohl haben Konverter oft dieselben Probleme wie WYSIWYG-Editoren: Die Ergebnisse unterscheiden sich von Browser zu Browser, und viele der neueren und erweiterten Merkmale sind mit ihnen nicht verfügbar. Die existierenden Konverter-Programme sind ziemlich eingeschränkt, nicht unbedingt in ihren eigenen Möglichkeiten, sondern vor allem durch die Einschränkungen von HTML selbst. Auch ein großer Aufwand an einfallsreichen Konvertierungen bringt HTML nicht dazu, etwas zu tun, was es nicht kann. Wenn es eine bestimmte Fähigkeit in HTML nicht gibt, kann auch kein Konverter daran etwas ändern (aber er kann mit Ihren HTML-Dateien seltsame Dinge anstellen und Ihnen damit mehr Arbeit bereiten, als wenn Sie die Formatierung selbst vorgenommen hätten).

Wenn Sie bereits über einen einfachen HTML-Editor verfügen, scheuen Sie sich nicht, diesen für die Beispiele in diesem Buch zu verwenden. Wenn Sie nur einen Text-Editor besitzen, ist das auch kein Problem; Sie müssen nur ein bisschen mehr tippen.

Zusammenfassung

Versuchen Sie, die Quelle Ihrer bevorzugten Webseite anzusehen. Sie sollten schon die Methode wiedererkennen können, mit der Webseiten erstellt werden. Ebenfalls sollte Ihnen die Organisation der Seite bereits etwas sagen. Sie sollten schon langsam ein Gefühl für die in HTML verwendeten Tags bekommen. Sie können viel über HTML lernen, indem Sie den angezeigten Text mit der Quelle des Textes vergleichen.

Im heutigen Kapitel haben Sie einige grundlegende Dinge über HTML und das Definieren eines HTML-Dokuments als Webseite gelernt. Sie haben ein wenig über die HTML-Geschichte und die Gründe für die mehrfache Änderung der HTML-Spezifikationen seit Beginn von HTML kennen gelernt. Mittels Seitenstrukturierender Tags, einem Titel und einer Kopfzeile haben Sie Ihre erste Webseite erstellt. War doch alles nicht so schlimm, oder? Morgen werden Sie Ihr Wissen erweitern und mehr über das Hinzufügen von Überschriften, Text und Listen zu Ihren Seiten lernen.

Workshop

Da Sie jetzt schon ein gewisses Grundwissen über HTML verfügen, und einen Vorgehensschmack auf das Erstellen von einfachen Webseiten bekommen haben, wird Ihnen dieser Workshop zu weiteren Erkenntnissen verhelfen. Sie werden ein paar Fragen und Antworten über das HTML-Formatieren finden, gefolgt von einem kurzen Quiz und Antworten zur HTML. In den darauf folgenden Übungen werden Sie einen etwas schwierigeren Code in Ihrem Browser untersuchen.

Fragen und Antworten

F Kann ich Text in HTML so formatieren, wie ich es möchte?

- A** Sie können Strings formatieren indem sie zum Beispiel ein oder zwei Wörter fettgedruckt erscheinen lassen. Tags in HTML 3.2 (der Vorgänger von HTML 4.0) ermöglichten Ihnen eine Änderung der Schriftgröße und der Farbe des Texts (für Besucher deren Browser die Tags unterstützten – Netscape und Microsoft Internet Explorer miteinbegriffen), aber diese Tags wurden durch CSS-Formatierung in HTML 4.0 ersetzt. Formatierungstricks werden Sie am Tag 6 lernen.

F Ich benutze Windows. Mein Textverarbeitungsprogramm ermöglicht es mir nicht, außer .txt, eine Textdatei mit Erweiterung zu speichern. Wenn ich `index.html` eingebe, speichert mein Textverarbeitungsprogramm die Datei als `index.html.txt`. Was soll ich tun?

A *Sie können Ihre Dateien nach dem Speichern umbenennen, sodass Sie die Erweiterung `html` oder `htm` haben. Wenn Sie allerdings viele Dateien bearbeiten müssen, könnte das Ganze etwas ermüdend werden. Ziehen Sie doch die Verwendung eines Text-Editors oder eines HTML-Editors für Ihre Webseiten in Betracht.*

Quiz

1. Für was steht HTML?
2. Was ist die Hauptfunktion von HTML?
3. Warum kontrolliert HTML nicht das Seitenlayout?
4. Welche HTML-Version liefert den kleinsten gemeinsamen Nenner für HTML-Tags?
5. Was ist die Grundstruktur eines HTML-Tags?

Antworten

1. HTML steht für Hypertext Markup Language.
2. HTML definiert das Aussehen einer Webseite (Überschriften, Absätze, Listen, Tabellen, Aussehen von Buchstaben und mehr)
3. HTML kontrolliert das Seitenlayout nicht, da es plattformübergreifend ist. HTML untersucht die Unterschiede zahlreicher Plattformen, und stellt alle Browser und Computersysteme auf eine Ebene.
4. Der kleinste gemeinsame Nenner aller HTML-Versionen ist HTML 2.0, der älteste HTML-Standard. Es handelt sich hierbei um die Tags, die alle Browser unterstützen müssen. HTML 2.0-Tags können überall verwendet werden.
5. Die meisten HTML-Tags haben normalerweise ein einleitendes und ein abschließendes Tag, die einen Text umschließen. Die Tags werden in Klammern gesetzt (`<>`). Das einleitende Tag leitet ein Feature ein, während das abschließende Tag, vor dem ein Slash (`/`) steht, das Feature beendet.

Übungen

1. Bevor Sie eine etwas anspruchsvollere Webseite erstellen, sollten Sie sich ansehen, wie eine solche Webseite aussieht. Gott sei Dank, gibt es genügend Quellenmaterial, das Sie genauer untersuchen können. Jede Seite, die in Ihren Browser geladen wird, ist in HTML-Format geschrieben (Sie sehen fast nie die Codes in Ihrem Browser, sondern immer nur das Endergebnis).
2. Die meisten Web-Browser bieten Ihnen die Möglichkeit die HTML-Quelle einer Webseite anzuzeigen. Wenn Sie zum Beispiel den Internet Explorer 4.0 verwenden, suchen Sie die Webseite auf, deren Quelltext Sie gerne sehen möchten. Wählen Sie ANSICHT | QUELLTEXT ANZEIGEN. Der Quellcode wird in einem Textfenster angezeigt. Wählen Sie im Netscape Navigator ANSICHT | SEITENQUELLTEXT.

**Tipp**

In manchen Browsern können Sie die Quelle der Webseite nicht direkt sehen, aber Sie können die entsprechende Seite als Datei der lokalen Festplatte speichern. Sie müssen meistens noch das Format wählen, in dem der Quellcode abgespeichert werden soll – zum Beispiel Text, Postscript oder HTML. Speichern Sie die entsprechende Seite als HTML. Anschließend öffnen Sie diese Datei in einem Text-Editor oder Textverarbeitungsprogramm, um die HTML-Quelle zu sehen.

Versuchen Sie, eine typische Homepage zu besuchen und ihre Quellen anzusehen. Abbildung 3.3 zum Beispiel zeigt die Homepage von Alta Vista, einer bekannten Suchmaschine unter <http://www.altavista.com/>.

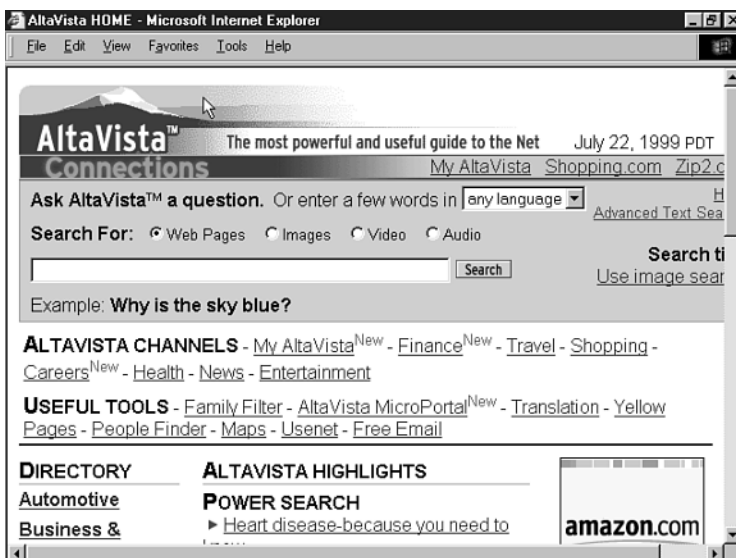
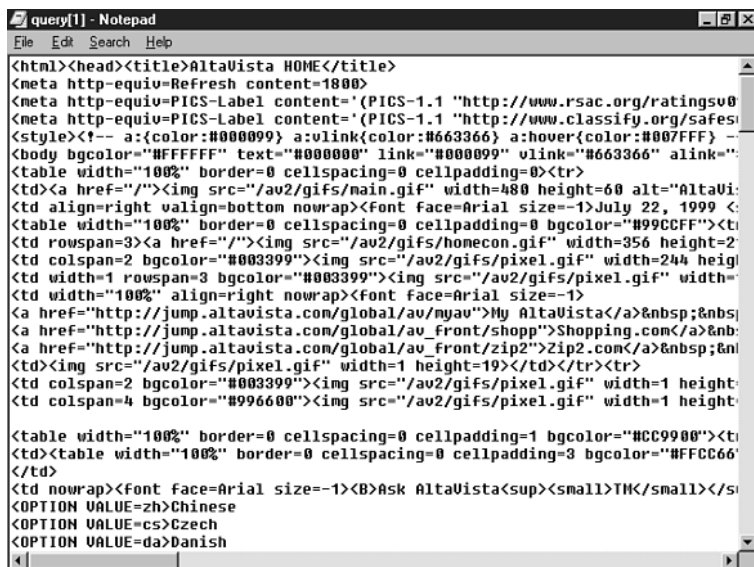


Abbildung 3.3:
Alta Vista Home-
page

Der HTML-Code dieser Seite sieht etwa so aus, wie in Abbildung 3.4 dargestellt.



```

query[1] - Notepad
File Edit Search Help
<html><head><title>AltaVista HOME</title>
<meta http-equiv=Refresh content=1800>
<meta http-equiv=PICS-Label content=(PICS-1.1 "http://www.rsac.org/ratingsv0
<meta http-equiv=PICS-Label content=(PICS-1.1 "http://www.classify.org/safes
<style><!-- a:{color:#000099} a:vlink{color:#663366} a:hover{color:#007FFF} --
<body bgcolor="#FFFFFF" text="#000000" link="#000099" vlink="#663366" alink="
<table width="100%" border=0 cellspacing=0 cellpadding=0><tr>
<td><a href="/"><font face=Arial size=-1>July 22, 1999 <
<table width="100%" border=0 cellspacing=0 cellpadding=0 bgcolor="#99CCFF"><t
<td rowspan=3><a href="/"><font face=Arial size=-1>
<a href="http://jump.altavista.com/global/av/myav">My AltaVista</a>&nbsp;&nbsp;&nbsp;
<a href="http://jump.altavista.com/global/av_front/shopp">Shopping.com</a>&nbsp;&nb
<a href="http://jump.altavista.com/global/av_front/zip2">Zip2.com</a>&nbsp;&nbsp;&nbsp;
<td></td></tr><tr>
<td colspan=2 bgcolor="#003399"><t
<td><table width="100%" border=0 cellspacing=0 cellpadding=3 bgcolor="#FFCC66"
</td>
<td nowrap><font face=Arial size=-1><B>Ask AltaVista<sup><small>TM</small></s
<OPTION VALUE=zh>Chinese
<OPTION VALUE=cs>Czech
<OPTION VALUE=da>Danish
  
```

Abbildung 3.4:
Ein wenig HTML-
Code

3. Versuchen Sie sich den Quelltext Ihrer Lieblings-Webseiten anzusehen. Normalerweise sollten Sie schon jetzt einige Ähnlichkeiten in Bezug auf die Organisation der Seiten erkennen können. Außerdem sollten Sie langsam ein Gefühl dafür entwickeln, welche Art von Tags HTML benutzt. Wenn Sie den angezeigten Text mit den Quellen für diesen Text vergleichen, können Sie viel über HTML lernen.



Einfache HTML- Tags

**Woche
1**

Gestern haben Sie einiges über das World Wide Web erfahren, wie man Web-Sites organisiert und plant und warum man zum Erstellen einer Webseite HTML braucht. Heute werden wir die HTML-Tags eingehender behandeln. Sie werden lernen, wie man Überschriften, Paragraphen und verschiedene Listen in die Webseite eingibt. Dieses Kapitel behandelt außer den HTML-Tags noch folgende Themen:

- Tags für eine allgemeine Seitenstruktur: `<html>`, `<head>` und `<body>`
- Tags für Titel, Überschriften und Paragraphen: `<title>`, `<h1>...<h6>` und `<p>`
- Tags für Kommentare: `<!--...-->`
- Tags für Listen: `<ol>`, `<ul>`, `<li>`, `<dt>` und `<dd>`

Strukturierung von HTML-Code

HTML definiert drei Tags, die benutzt werden, um die Gesamtstruktur eines Dokuments zu beschreiben und um einige einfache »Header«-Informationen (ein Begriff aus dem E-Mail-Bereich; hier wird mit »Header« gewissermaßen der »Briefumschlag« bezeichnet) anzubieten. Diese drei Tags – `<html>`, `<head>`, `<body>` – identifizieren Ihre Seite in Browsern oder HTML-Werkzeugen. Wenn man sich an die genaue HTML-Definition hält, dann sind diese Kennungen nicht zwingend. Wenn Ihre Seite sie nicht enthält, können die Browser sie normalerweise immer noch lesen. Diese Tags jedoch sind zwingend erforderliche Elemente in XHTML 1.0. In Zukunft wird es Werkzeuge und Browser geben, die diese Tags brauchen werden. Deshalb sollten Sie sich von vornherein angewöhnen, diese Struktur-Kennungen in Ihr Dokument einzufügen.



Obwohl es sich um keinen Seitenstruktur-Tag handelt, braucht man für die vorgeschlagene XHTML 1.0 Spezifikation noch eines: die DTD. Die erste Zeile einer jeden Seite muss einen DOCTYPE-Kennzeichner beinhalten, der die XHTML 1.0 Version, nach der sich Ihre Seite richtet, definiert. Das heißt also, dass Sie eine Document Type Definition (DTD) brauchen, die die Spezifikation definiert, gefolgt von den `<html>`, `<head>` und `<body>` Tags. Im folgendem Beispiel erscheint das XHTML 1.0 Strict Dokument vor den Seitenstruktur-Tags:

```
<!doctype html public "-//w3c//dtd/xhtml 1.0 transitional//en"
    "http://www.w3.org/tr/xhtml1/dtd/strict.dtd">
<html>
<head>
<title>Page Title.</title>
</head>
<body>
```

...Ihre Seite...

```
</body>  
</html>
```

Insgesamt gibt es drei verschiedene Arten von HTML 4.0 Dokumenten, die in den XHTML 1.0 Spezifikationen angegeben sind: Strict, Transitional und Frameset. Am Tag 10 erfahren Sie mehr über das DOCTYPE-Tag und am Tag 18 mehr über die Unterschiede von Strict-, Transitional- und Frameset-Dokumenten.

```
<html>
```

Die erste Struktur-Kennung in jedem HTML-Dokument ist das `<html>`-Tag, welches anzeigt, dass der Inhalt dieser Datei in HTML geschrieben wurde. In der XHTML 1.0 Spezifikation sollte das `<html>` Tag nach dem DOCTYPE-Kennzeichner kommen (siehe »Hinweis«). Sehen Sie hierzu folgendes Beispiel.

Der gesamte Text und alle HTML-Befehle in Ihrem HTML-Dokument sollten wie hier zwischen der einleitenden und abschließenden `<html>`-Kennung stehen:

```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"  
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">  
<html>  
...Ihre Seite...  
</html>
```

```
<head>
```

Das `<head>`-Tag bestimmt, dass die Zeilen zwischen der einleitenden und der abschließenden Kennung als Dateivorspann angesehen werden. Generell gibt es nur wenige Kennungen, die in den `<head>`-Teil der Seite gehören (vornehmlich der unten beschriebene Dokumententitel). Sie sollten niemals Teile des Textes Ihrer Seite in den `<head>`-Teil schreiben.

Hier ist ein typisches Beispiel dafür, wie Sie das `<head>`-Tag richtig benutzen (über `<title>` erfahren Sie nachher mehr):

```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"  
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">  
<html>  
<head>  
<title>This is the Title.</title>  
</head>  
...Ihre Seite...  
</html>
```

```
<body>
```

Der Rest Ihres HTML-Dokuments, einschließlich des gesamten Textes und anderen Inhalts (Verknüpfungen, Bilder und so weiter), ist in ein `<body>`-Tag eingeschlossen. In Kombination mit den `<html>`- und `<head>`-Kennungen sieht das dann so aus:

```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>
<head>
<title>This is the Title. It will be explined later on</title>
</head>
<body>
...Ihre Seite...
</body>
</html>
```

Sie werden bemerken, dass jede HTML-Kennung verschachtelt ist, die beiden `<body>`- und `</body>`-Tags gehören zwischen die beiden `<html>`-Kennungen; das gleiche gilt für die beiden `<head>`-Kennungen. Alle HTML-Tags funktionieren auf diese Weise, indem sie individuelle verschachtelte Abschnitte von Text bilden. Sie sollten darauf achten, Kennungen niemals zu überlappen, das heißt, *vermeiden* Sie Folgendes zu tun:

```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>
<head>
<body>
</head>
</body>
</html>
```

Vergewissern Sie sich, wenn Sie eine HTML-Kennung abschließen, dass Sie jeweils die Kennung abschließen, die zuletzt geöffnet wurde (Sie werden später noch mehr darüber erfahren).



In HTML 4.0 und früheren Versionen gibt es einige Tags, die man nicht zwingend schließen muss. In anderen Tags sind schließende Tags gänzlich verboten. In der XHTML 1.0 Spezifikation müssen alle Tags geschlossen werden. Warum, wie und wo wird in Kapitel 10 erklärt. Die in diesem Buch aufgezeigten Beispiele zeigen Ihnen wie man Tags korrekt abschließt. Somit können auch ältere Browser die geschlossenen Tags in XHTML 1.0 interpretieren.

Der Titel

Jede HTML-Seite braucht einen Titel, der anzeigt, was sie beschreibt. Der Titel wird von den Lesezeichen Ihres Browser, von Hotlist-Programmen sowie von anderen Programmen, die Webseiten katalogisieren, benutzt. Um einer Seite einen Titel zu geben, benutzen Sie die HTML-`<title>`-Kennung.



Der Titel zeigt an, wovon Ihre Webseite handelt, und dient als Bezugspunkt der Seite für Lesezeichen- oder Hotlist-Eintragungen. Titel werden in vielen Browsern mit grafischer Benutzeroberfläche in deren Titelleiste angezeigt (das ist zum Beispiel beim Netscape Navigator und beim Microsoft Internet Explorer der Fall).

`<title>` gehört immer in den Dokumenten-Header (also zwischen die `<head>`-Kennungen) und beschreibt den Inhalt der Seite wie hier:

```
<html>
<head>
<title> Der Loewe, die Hexe und der Kleiderschrank </title>
</head>
<body>
... your page ...
</body>
</html>
```

So können Abbildungen nur einen Titel auf Ihrer Seite haben und der Titel darf nur reinen Text enthalten; das heißt, es sollten keine anderen HTML-Tags innerhalb des Titels vorkommen.

Entscheiden Sie sich für einen Titel, der sowohl kurz ist als auch den Inhalt der Seite wiedergibt. Außerdem sollte man, auch ohne den Inhalt der Seite zu kennen, mit dem Titel etwas anfangen können. Wenn jemand durchs Web streift, einer zufälligen Verbindung folgt und auf dieser Seite landet oder wenn jemand Ihren Titel in der History-Liste vom Browser eines Freundes gefunden hat, würde man erkennen können, um was es auf dieser Seite geht? Sie beabsichtigen womöglich nicht, dass jemand diese Seite unabhängig von jenen Dokumenten benutzt, von denen aus Sie spezielle Verknüpfungen erzeugt haben. Da sich jedoch jeder zu jeder Zeit mit jeder beliebigen Seite verbinden kann, sollten Sie auf diese Konsequenzen vorbereitet sein und einen aufschlussreichen Titel wählen.

Außerdem können Sie in der Anzahl der Wörter eingeschränkt sein, weil viele Browser den Titel in eine Titelzeile des Bildschirmfensters schreiben. (Obwohl der Text innerhalb der `<title>`-Tags von beliebiger Länge sein darf, könnte er durch den Browser abgeschnitten werden, wenn er angezeigt wird.) Hier sind einige Beispiele für aufschlussreiche Titel:

```
<title>Poisonous Plants of North America</title>
<title>Image Editing: A Tutorial</title>
<title>Upcoming Cemetery Tours, Summer 1997</title>
<title>Installing The Software: Opening the CD Case</title>
<title>Laura Lemay es Awesome Home Page</title>
```

Und einige weniger gelungene Titel:

```
<title>Part Two</title>
<title>An Example</title>
<title>Nigel Franklin Hobbes</title>
<title>Minutes of the Second Meeting of the Fourth Conference of the
Committee for the Preservation of English Roses, Day Four, After Lunch</title>
```

Abbildung 4.1 zeigt Ihnen, wie ein Titel im Internet Explorer aussieht.



```
<title>Poisonous Plants of North America</title>
```

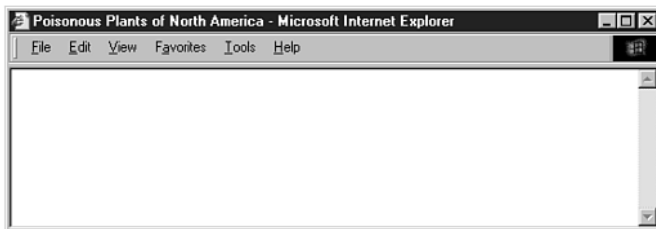


Abbildung 4.1:
Ausgabe im Internet
Explorer

Überschriften

Überschriften werden benutzt, um verschiedene Textabschnitte zu unterteilen, genauso, wie dieses Buch unterteilt ist. HTML definiert sechs Stufen von Überschriften. Die Tags der Überschriften sehen so aus:

```
<h1> Installation der Backup-Software </h1>
```

Die Ziffern bezeichnen die Stufe der Überschrift (<h1> bis <h6>). Die Überschriften selbst sind, wenn sie angezeigt werden, nicht nummeriert; sie werden entweder mit

größeren oder fetten Zeichen, zentriert oder unterstrichen oder in Großbuchstaben dargestellt – aber immer auf eine Weise, die sie vom normalen Text abhebt.

Stellen Sie sich Überschriften so vor, als seien sie die Stichwörter eines Themenabris- ses. Wenn der Text, den Sie schreiben, eine Struktur hat, benutzen Sie die Überschrif- ten, um diese Struktur zu verdeutlichen, wie es in den nächsten Code-Zeilen gezeigt wird. (Beachten Sie, dass ich die Überschriften in diesem Beispiel eingerückt habe, um die Hierarchie besser darzustellen. In Ihrem Dokument müssen sie nicht eingerückt sein, und vom Browser werden die Einrückungen ohnehin ignoriert.)

```
<h1>Mythology Through the Ages</h1>
<h2>Common Mythological Themes</h2>
<h2>Earliest Known Myths</h2>
<h2>Origins of Mythology</h2>
  <h3>Mesopotamian Mythology</h3>
  <h3>Egyptian Mythology</h3>
    <h4>The Story of Isis and Osiris</h4>
    <h4>Horus and Set: The Battle of Good vs. Evil</h4>
    <h4>The Twelve Hours of the Underworld</h4>
    <h4>The River Styx</h4>
  <h2>History in Myth</h2>
```

Beachten Sie, dass die Überschriften, anders als der Titel, von beliebiger Länge sein können, auch über mehrere Textzeilen hinweg (da Überschriften jedoch hervorgeho- ben dargestellt werden, können Zeilen über Zeilen hervorgehobenen Textes auf Ihre Leser ermüdend wirken).

Es ist eine verbreitete Praxis, die erste Stufe der Überschriften zu Beginn Ihres Doku- ments zu verwenden, um entweder den Titel zu duplizieren (der normalerweise woan- ders angezeigt wird) oder um eine kürzere Form des Titels vorzusehen. Wenn Sie zum Beispiel eine Seite haben, die Bestandteil eines langen Dokuments über das Zusam- menlegen von Bettwäsche ist und verschiedene Beispiele für das Falten von Bettlaken zeigt, dann könnte der Titel ungefähr so aussehen:

```
<title> Wie man Bettlaken zusammenlegt: Einige Beispiele </title>
```

Die oberste Überschrift könnte dann einfach lauten:

```
<h1> Beispiele </h1>
```

Benutzen Sie Überschriften nicht, um Fettschrift zu erzeugen oder um einzelne Teile Ihres Dokuments hervorzuheben. Obgleich es in Ihrem Browser ganz toll aussehen mag, wissen Sie nicht, wie es aussieht, wenn andere Leute einen anderen Browser verwenden, um Ihr Dokument zu lesen. Andere Browser nummerieren vielleicht die Überschriften oder formatieren sie auf eine unerwartete Weise. Außerdem können Programme, die Webseiten-Indexe erstellen, Ihre Überschriften herausziehen, um wichtige Teile der Seite anzuzeigen. Wenn Sie Überschriften zweckentfremden, kön- nen einige Suchprogramme seltsame Ergebnisse hervorbringen.

Abbildung 4.2 zeigt Überschriften in Internet Explorer



```
<h1>Mythology Through the Ages</h1>
<h2>Common Mythological Themes</h2>
<h2>Earliest Known Myths</h2>
<h2>Origins of Mythology</h2>
<h3>Mesopotamian Mythology</h3>
<h3>Egyptian Mythology</h3>
  <h4>The Story of Isis and Osiris</h4>
  <h4>Horus and Set: The Battle of Good vs. Evil</h4>
  <h4>The Twelve Hours of the Underworld</h4>
  <h4>The River Styx</h4>
<h2>History in Myth</h2>
```

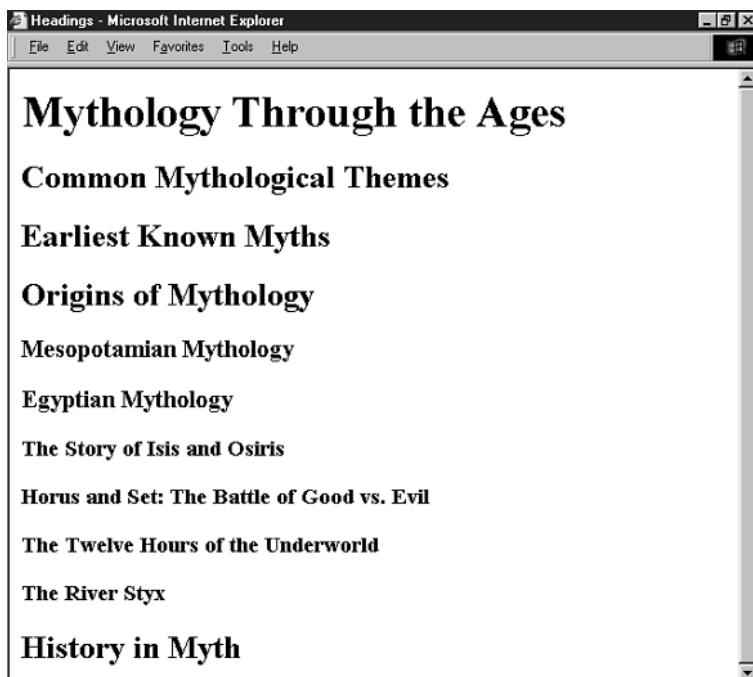


Abbildung 4.2:
Darstellung im
Internet Explorer

Absätze

Nachdem Sie nun also einen Seitentitel und unterschiedliche Überschriften haben, lassen Sie uns einige gewöhnliche Absätze in das Dokument einfügen.

Die erste Version von HTML spezifizierte die `<p>`-Kennung als einseitige Kennung. Es gab kein korrespondierendes `</p>`, und die `<p>`-Kennung wurde benutzt, um das Ende eines Absatzes zu kennzeichnen, nicht den Anfang. Somit sahen Absätze in der ersten Version von HTML folgendermaßen aus:

```
Slowly and deliberately, Enigern approached the mighty dragon.  
A rustle in the trees of the nearby forest distracted his attention  
for a brief moment, a near fatal mistake for the brave knight.<p>  
The dragon lunged at him, searing Enigern es armor with a rapid  
blast of fiery breath. Enigern fell to the ground as the dragon  
hovered over him. He quickly drew his sword and thrust it into the  
dragon es chest.<p>
```

Die meisten Browser, die zu Anfangszeiten des Web geschrieben wurden, unterstellten, dass Absätze auf diese Weise formatiert sind. Wenn sie auf eine `<p>`-Kennung stoßen, beginnen sie eine neue Zeile und fügen zusätzlichen Abstand zwischen der Zeile, die gerade beendet wurde, und der neuen Zeile ein.

In der Spezifikation HTML 4.0 (ebenfalls HTML 3.2 und 2.0) wurde im Zusammenhang mit den gängigsten Browsern das Absatz-Tag revidiert. In diesen HTML-Versionen werden die Absatz-Kennungen auf beiden Seiten gesetzt (`<p>...</p>`), wobei `<p>` den *Beginn* des Absatzes kennzeichnet. Somit ist die abschließende Kennung (`</p>`) optional. Die Sweater-Geschichte würde also in den aktuellen HTML-Versionen so aussehen:

```
<p>Slowly and deliberately, Enigern approached the mighty dragon.  
A rustle in the trees of the nearby forest distracted his attention  
for a brief moment, a near fatal mistake for the brave knight.</p>  
  
<p>The dragon lunged at him, searing Enigern es armor with a rapid  
blast of fiery breath. Enigern fell to the ground as the dragon  
hovered over him. He quickly drew his sword and thrust it into the  
dragon es chest.</p>
```

Es ist gut, wenn Sie es sich angewöhnen, `<p>` am Anfang eines Absatzes einzusetzen; dies wird dann wichtig werden, wenn Sie Text linksbündig, rechtsbündig oder zentriert ausrichten möchten. Ältere Browser werden diese Art der Absatzformatierung ohne Schwierigkeiten annehmen. Ob Sie dann die `</p>`-Kennung verwenden, bleibt Ihnen überlassen; sie weist Sie jedoch darauf hin, dass der Absatz hier zu Ende ist; vielleicht erscheint Ihnen dies aber auch überflüssig. Ich verwende das `</p>`-Tag jedenfalls im gesamten Buch.

Einige Leute verwenden zusätzliche `<p>`-Tags, um etwas mehr Platz um andere Tags herum zu schaffen und den Text auf der Seite etwas aufzulockern. Daher erneut die grundsätzliche Mahnung: Gestalten Sie den Inhalt, nicht die Erscheinungsweise! Jemand mit einem textbasierten Browser oder einem kleinen Bildschirm wird sich nicht allzu sehr um den zusätzlichen Leerraum kümmern, den Sie so sorgfältig eingefügt haben. Einige Browser werden vielleicht sogar mehrere `<p>`-Tags auf eine einzige verkürzen und so Ihre ganze mühsame Formatierung zunichte machen.

Abbildung 4.3 zeigt einen weiteren Absatz über Enigern und dem Drachen im Internet Explorer.



```
<p>The dragon fell to the ground, releasing an anguished cry and
seething in pain. The thrust of Enigern's sword proved fatal as the
dragon breathed its last breath. Now Enigern was free to
release Lady Aelfleada from her imprisonment in the dragon's lair. </p>
```

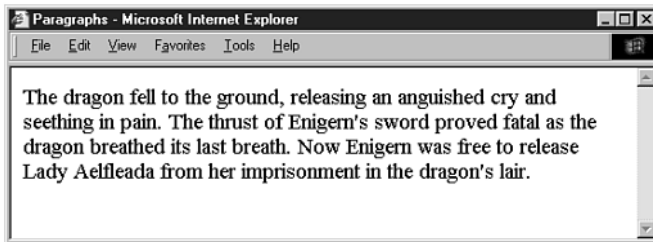


Abbildung 4.3:
Darstellung im Internet
Explorer

Listen, Listen und nochmals Listen

Abgesehen von Überschriften und Absätzen ist das wohl häufigste HTML-Element, mit dem Sie sich auseinandersetzen müssen, die Liste. Wenn Sie sich durch diesen Abschnitt gearbeitet haben, werden Sie nicht nur wissen, wie eine Liste erzeugt wird, sondern fünf verschiedene Arten von Listen erzeugen können, eine Liste für jede Gelegenheit!

HTML definiert drei Arten von Listen:

- ▶ nummerierte oder geordnete Listen, typischerweise mit Nummern eingeleitet,
- ▶ ungeordnete Listen, typischerweise mit einem dicken Punkt oder einem anderen Symbol eingeleitet,
- ▶ Glossar-Listen, in denen jeder Eintrag aus einem Stichwort und einer Definition dieses Stichworts besteht, meist so dargestellt, dass das Stichwort irgendwie hervorgehoben oder vom Text abgesetzt wird,



Sie werden einige alte Listen-Arten in der HTML 4.0 Spezifikation bemerken: menu lists (`<menu>`) und directory lists (`<dl>`). Diese zwei Listenarten werden nicht oft verwendet und die Unterstützung dieser Listen variiert von Browser zu Browser. Verwenden Sie lieber das `<ul>`-Tag (oder bulleted list).



Ein verworfenes Tag oder Attribut wird zwar noch unterstützt, wurde aber durch neuere Methoden ersetzt.



Wegen der Abwärtskompatibilität werden auch ältere Tags oder Attribute von Browsern immer noch unterstützt. Dennoch ist es von Nutzen, diese Tags und Attribute zu kennen, vor allem dann, wenn Sie davon ausgehen müssen, dass ein Teil Ihrer Web-Site Besucher HTML 3.2 Browser benutzt (wie zum Beispiel Netscape Navigator 3 oder aber auch Internet Explorer 3. Dies gilt natürlich auch für frühere Versionen.) Diese Tags und Attribute werden in der zukünftigen HTML Version nicht mehr zu finden sein. Sie sollten deshalb die neueren Methoden anwenden



Die Mehrheit der älteren Tags und Attribute in HTML 4.0 wurden zugunsten der Cascading Style Sheets (CSS) -Eigenschaften und -Werte verworfen. Am Tag 10 erfahren Sie mehr über CSS.

Listen-Tags

Alle Listen-Kennungen (List-Tags) haben gemeinsame Bestandteile:

- ▶ Die gesamte Liste ist umgeben von den passenden Einleitungs- und Abschluss-Kennungen für die jeweilige Listenart (zum Beispiel `<ol>` und `</ol>` oder `<menu>` und `</menu>`).
- ▶ Jedes Element in der Liste hat seine eigene Kennung: `<dt>` und `<dd>` für die Glossar-Listen und `<li>` für alle anderen Listen.



Die abschließenden Tags für `<dd>`, `<dt>` und `<li>` sind in HTML nicht zwingend notwendig. Wenn Sie mit XHTML arbeiten wollen, sollten Sie sich jetzt schon angewöhnen diese Tags abzuschließen: `</dd>`, `</dt>`, `</li>`.

Obgleich die Kennungen und die Listeneinträge in Ihrem HTML-Code beliebig angeordnet sein können, ziehe ich es vor, den Listen-Code so anzuordnen, dass die Listen-Kennungen jeweils in einer eigenen Zeile stehen und jeder neue Eintrag ebenfalls mit einer neuen Zeile beginnt. Das macht es leicht, sowohl die Liste selbst als auch die einzelnen Listeneinträge schnell zu erkennen. Mit anderen Worten, ich finde Anordnungen wie diese:

```
<p> Dantes Goettliche Komoedie besteht aus drei Buechern: </p>
<ul>
  <li> Hoelle </li>
  <li> Fegefeuer </li>
  <li> Paradies </li>
</ul>
```

leichter zu lesen als die folgende Anordnung, obwohl beides zur gleichen Darstellung im Browser führt:

```
<p>Dantes Goettliche Komoedie besteht aus drei Buechern:</p>
<ul><li>Hoelle<li>Fegefeuer<li>Paradies</ul>
```

Nummerierte Listen

Nummerierte Listen werden durch die `<ol>...</ol>`-Tags eingeschlossen (`ol` steht für *Ordered List*, geordnete Liste), und jeder Eintrag in der Liste beginnt mit der `<li>`-Kennung (`li` = *List Item*, Listen-Eintrag).



Auch die `<li>`-Kennung ist einseitig, das heißt, Sie brauchen keine abschließende Kennung anzugeben. Das Auftreten der nächsten `<li>`-Kennung (oder der abschließenden `</ol>`-Kennung) bezeichnet zugleich das Ende des Eintrags in der Liste. In XHTML 1.0 müssen Sie ein abschließendes `</li>` Tag verwenden.

Wenn der Browser eine geordnete Liste interpretiert, nummeriert er jeden der Einträge fortlaufend (und rückt sie häufig etwas ein). Sie brauchen die Nummerierung also nicht selber durchzuführen, und wenn Sie Einträge hinzufügen oder entfernen, nummeriert der Browser sie neu, sobald er das Dokument erneut lädt.



Geordnete Listen sind Listen, in denen jeder Punkt nummeriert ist.

Verwenden Sie nummerierte Listen nur, wenn Sie deutlich machen wollen, dass die Elemente geordnet sind, das heißt, dass sie in einer bestimmten Reihenfolge auftreten müssen. Wenn Sie nur einige Elemente in beliebiger Reihenfolge auflisten wollen, benutzen Sie stattdessen eine ungeordnete Liste.

Im folgenden Beispiel sehen Sie eine geordnete Liste, die Ihnen Schritt für Schritt zeigt, wie Sie ein neues Betriebssystem installieren müssen. Nachstehende Input- und Output-Angaben zeigen die geordnete Liste. Auf Bild 4.4 können Sie sehen, wie diese Liste im Internet Explorer aussieht.



```
<p>Installing Your New Operating System</p>
<ol>
<li>Insert the CD-ROM into your CD-ROM drive.</li>
<li>Choose RUN.</li>
<li>Enter the drive letter of your CD-ROM (example: D:\),
followed by SETUP.EXE.</li>
<li>Follow the prompts in the setup program.</li>
<li>Reboot your computer after all files are installed.</li>
<li>Cross your fingers.</li>
</ol>
```

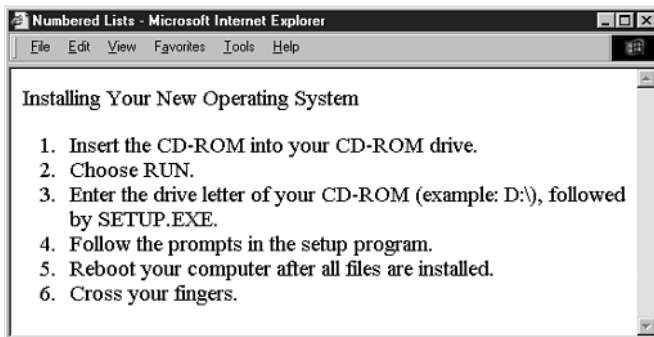


Abbildung 4.4:
Darstellung im Internet
Explorer

Anpassen geordneter Listen in HTML 3.2

HTML 3.2 stellt verschiedene Attribute für geordnete Listen zur Verfügung. Sie werden dazu verwendet, um zu sehen, wie der Browser die Liste wiedergibt. Diese Attribute ermöglichen Ihnen einige Features der geordneten Listen zu kontrollieren. Die geordneten Listen beinhalten ebenfalls das zu verwendende Nummerierungsschema, welches schon vorher festlegt, mit welcher Zahl begonnen werden soll (falls Sie nicht mit 1 beginnen wollen). In HTML 4.0 jedoch sind die, in diesem Abschnitt erwähnten Attribute, zugunsten von Style Sheet-Eigenschaften und -Werten verworfen. Um HTML 3.2 Browser zu unterstützen, werden Sie diese Attribute eventuell noch gebrauchen können



Attribute sind besondere Bestandteile von HTML-Tags, die entweder bestimmte Optionen oder zusätzliche Informationen zu den Tags selbst enthalten

Sie können bei geordneten Listen im Wesentlichen zwei Dinge anpassen: Zum einen wie diese nummeriert werden und zum anderen, bei welcher Zahl die Nummerierung beginnt. HTML 3.2 verfügt über das `type`-Attribut. Dieses Attribut kann einen der folgenden fünf Werte annehmen, der festlegt, welche Art der Nummerierung verwendet wird:

- ▶ »1«: Legt fest, dass arabische Standardziffern zur Nummerierung der Liste verwendet werden sollen (also 1, 2, 3, 4 usw.).
- ▶ »a«: Legt fest, dass Kleinbuchstaben zur Nummerierung der Liste verwendet werden sollen (sprich a, b, c, d usw.).
- ▶ »A«: Legt fest, dass Großbuchstaben zur Nummerierung der Liste verwendet werden sollen (A, B, C, D usw.).
- ▶ »i«: Legt fest, dass kleine römische Ziffern zur Nummerierung der Liste verwendet werden sollen (i, ii, iii, iv usw.).
- ▶ »I«: Legt fest, dass große römische Ziffern zur Nummerierung der Liste verwendet werden sollen (I, II, III, IV usw.).

Die Art der Nummerierung kann im `<ol>`-Tag wie folgt festgelegt werden: `<ol type="a">`. Standardmäßig wird `type="1"` verwendet.



Das angenehme an Web-Browsern ist, dass sie im Allgemeinen Attribute, die sie nicht verstehen, ignorieren. Wenn ein Browser zum Beispiel das `type`-Attribut des `<ol>`-Tag nicht unterstützt, wird er es einfach ignorieren, sobald er darauf stößt.

Nehmen Sie die folgende Liste als Beispiel:

```
<p>The Days of the Week in French:</p>
<ol>
<li>Lundi</li>
<li>Mardi</li>
<li>Mercredi</li>
<li>Jeudi</li>
<li>Vendredi</li>
<li>Samedi</li>
<li>Dimanche</li>
</ol>
```

Würden Sie, wie im Folgenden geschehen, das Attribut `type="I"` dem `<ol>`-Tag hinzufügen, dann würde die Liste in Netscape wie in Abbildung 4.9 dargestellt werden.



```
<p>The Days of the Week in French:</p>
<ol type="I">
<li>Lundi</li>
<li>Mardi</li>
<li>Mercredi</li>
<li>Jeudi</li>
<li>Vendredi</li>
<li>Samedi</li>
<li>Dimanche</li>
</ol>
```

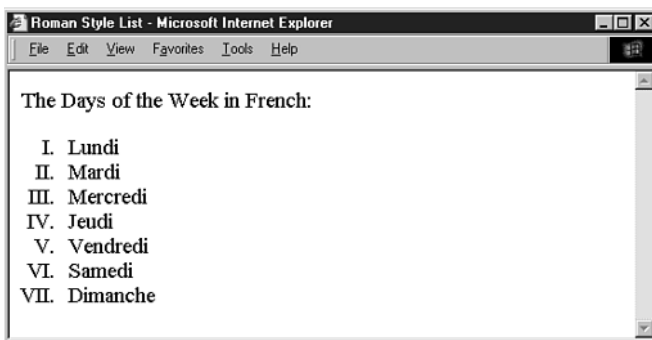


Abbildung 4.5:
Darstellung im Internet
Explorer

Das `type`-Attribut kann auch im `<li>`-Tag verwendet werden. Dadurch erreicht man einen Wechsel in der Art der Nummerierung innerhalb der Liste. Wird das `type`-Attribut mit dem `<li>`-Tag verwendet, so wirkt es sich auf den fraglichen Eintrag und alle darauffolgenden in der Liste aus.

Durch die Verwendung eines anderen Attributs – `start` – können Sie festlegen, mit welcher Zahl bzw. mit welchem Buchstaben die Nummerierung Ihrer Liste beginnt. Die Standard-Startposition ist natürlich die 1. `<ol start=4>` würde zum Beispiel dafür sorgen, dass die Liste bei Nummer 4 beginnt. Wohingegen `<ol type="a" start=3>` dafür sorgen würde, dass die Liste bei c beginnt und von dort im Alphabet fortfährt.

Sie können zum Beispiel die letzten 6 Monate des Jahres auflisten und bei der Nummerierung mit der römischen Zahl VII beginnen. In Bild 4.6 sehen Sie das Ergebnis.



```
<p>The Last Six Months of the Year:</p>
<ol type="i" start="7">
<li>July</li>
<li>August</li>
<li>September</li>
<li>October</li>
<li>November</li>
<li>December</li>
</ol>
```

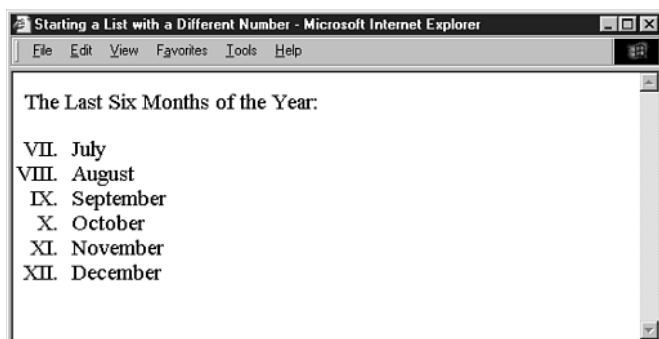


Abbildung 4.6:
Darstellung im Internet
Explorer

Wie auch mit dem `type`-Attribute können Sie an jedem Punkt einer Liste die Nummer eines Eintrags ändern. Dies erreichen Sie über das `value`-Attribut des `<li>`-Tag. Die Nummerierung wird beginnend mit dem angegebenen Wert bei dem betroffenen Eintrag neu gestartet.

Was wäre zum Beispiel wenn die letzten drei Einträge in der Liste der Zutaten von oben anstatt der Nummern 6, 7, 8 die Nummern 10, 11, 12 haben sollten? Sie können die Nummerierung bei dem Eintrag `Eggs` über das `value`-Attribut entsprechend zurücksetzen:

```
<p>Cheesecake ingredients:</p>
<ol type="i">
  <li>Quark Cheese</li>
  <li>Honey</li>
  <li>Cocoa</li>
  <li>Vanilla Extract</li>
  <li>Flour</li>
  <li value=10>Eggs</li>
  <li>Walnuts</li>
  <li>Margerine</li>
</ol>
```

Ungeordnete Listen

Ungeordnete Listen sind Listen, deren Elemente in beliebiger Reihenfolge erscheinen können. Ungeordnete Listen sehen in HTML genauso aus wie geordnete Listen, abgesehen davon, dass die Liste durch die `<ul>...</ul>`-Kennung anstelle von `<ol>` gekennzeichnet wird. Die einzelnen Einträge der Liste werden durch die `<li>`-Kennung unterteilt, genau wie in geordneten Listen, zum Beispiel so:

Browser formatieren ungeordnete Listen normalerweise mit einem dicken Punkt oder einer anderen symbolischen Markierung; Lynx verwendet das Sternchen (*, Asterisk).



Ungeordnete Listen sind Listen, bei denen die Listenpunkte mit einem dicken Punkt oder einer anderen symbolischen Markierung versehen sind.

Das folgende Beispiel zeigt eine ungeordnete Liste wie sie im Internet Explorer dargestellt wird (siehe Abbildung 4.7).



```
<p>Things I like to do in the morning:</p>
<ul>
<li>Drink a cup of coffee</li>
<li>Watch the sunrise</li>
<li>Listen to the birds sing</li>
<li>Hear the wind rustling through the trees</li>
<li>Curse the construction noises for spoiling the peaceful mood</li>
</ul>
```

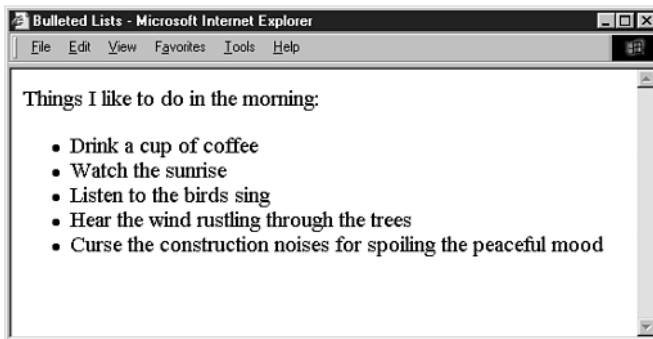


Abbildung 4.7:
Darstellung im Internet
Explorer

Anpassung ungeordneter Listen in HTML 3.2

Ungeordnete Listen können wie geordnete Listen mit HTML 3.2 Attributen (in HTML 4.0 verworfen) angepasst werden. Standardmäßig verwenden die meisten Browser (darunter auch Netscape und Internet Explorer) dicke Punkte, um die einzelnen Einträge ungeordneter Listen zu kennzeichnen; Text-Browser, wie zum Beispiel Lynx, verwenden im Allgemeinen ein Sternchen.

Über das `type`-Attribut des `<ul>`-Tag können Sie einige Browser dazu veranlassen, andere Arten von Blickfangpunkten zu verwenden, um die Einträge zu kennzeichnen. Nach der HTML-3.2-Spezifikation kann das `type`-Attribut drei verschiedene Werte annehmen:

- "`disc`": Mit dieser Einstellung wird ein dicker, ausgefüllter Punkt (Scheibe) dargestellt. Dieser Wert stellt im Allgemeinen den Standard dar.

- ▶ "square": In diesem Fall wird statt des Punktes ein ausgefülltes Quadrat verwendet.
- ▶ "circle": Im Vergleich zum Wert "disc", bei dem die meisten Browser einen gefüllten Kreis anzeigen, sollte dieser Wert in entsprechenden Browsern einen nicht gefüllten Kreis erzeugen.

Im nachfolgenden Eingabe/Ausgabe-Beispiel sehen Sie einen Vergleich der drei Typen, wie sie im Internet Explorer dargestellt werden (siehe Abbildung 4.8).



```
<ul type="disc">
<li>DAT - Digital Audio Tapes</li>
<li>CD - Compact Discs</li>
<li>Cassettes</li>
</ul>
<ul type="square">
<li>DAT - Digital Audio Tapes</li>
<li>CD - Compact Discs</li>
<li>Cassettes</li>
</ul>
<ul type="circle">
<li>DAT - Digital Audio Tapes</li>
<li>CD - Compact Discs</li>
<li>Cassettes</li>
</ul>
```

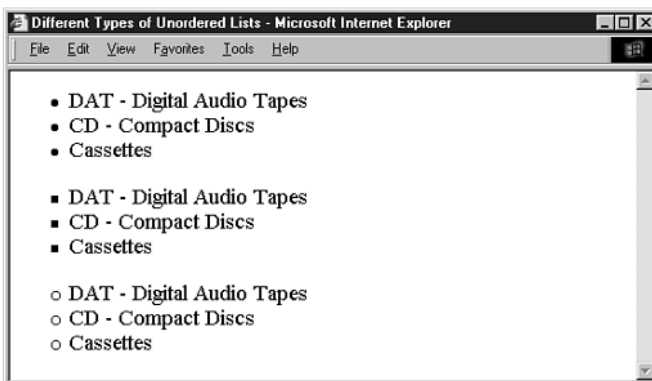


Abbildung 4.8:
Darstellung im Internet
Explorer

So wie Sie die Art der Nummerierung innerhalb einer geordneten Liste ändern können, können Sie die Art der Blickfangpunkte im Lauf einer Liste ändern, indem Sie das `type`-Attribut des `<li>`-Tag verwenden. Dieses Attribut ist in HTML 4.0 ebenfalls verworfen worden.

Glossar-Listen

Glossar-Listen, bisweilen auch Definitions-Listen genannt, unterscheiden sich ein wenig von den anderen Listen. Jedes Element einer Glossar-Liste besteht aus zwei Teilen:

- ▶ einem Stichwort und
- ▶ der Definition dieses Stichworts.

Jeder Teil der Glossar-Liste hat seine eigenen Kennungen: `<dt>` (»definition term«) für das Stichwort und `<dd>` (»definition definition«) für seine Definition. `<dt>` und `<dd>` sind beides einseitige Kennungen, und normalerweise tauchen sie paarweise auf, obwohl die meisten Browser auch mit einzelnen Stichwörtern oder Definitionen umgehen können. Die ganze Glossar-Liste wird durch die `<dl>...</dl>`-Kennung (»definition list«) gekennzeichnet.



Glossar-Listen sind Listen, bei denen jeder Listenpunkt aus zwei Teilen besteht: einem Ausdruck und seiner Definition. Deswegen werden Glossar-Listen manchmal auch Definitions-Listen genannt.

Hier folgt das Beispiel einer Glossar-Liste anhand von einigen Kräutern und darüber, wie sie wachsen:

```
<dl>
<dt>Basil</dt>
<dd>Annual. Can grow four feet high; the scent of its tiny white
flowers is heavenly.</dd>
<dt>Oregano</dt>
<dd>Perennial. Sends out underground runners and is difficult
to get rid of once established.</dd>
<dt>Coriander</dt>
<dd>Annual. Also called cilantro, coriander likes cooler
weather of spring and fall.</dd>
</dl>
```

Glossar-Listen werden natürlich nicht nur für Stichwörter und Definitionen verwendet. Sie können überall dort benutzt werden, wo eine eingerückte Liste benötigt wird. – Hier ein Beispiel:

```
<dl>
<dt>Macbeth</dt>
<dd>I'll go no more. I am afraid to think of
what I have done; look on't again I dare not.</dd>
<dt>Lady Macbeth</dt>
<dd>Infirm of purpose! Give me the daggers.
The sleeping and the dead are as but pictures. 'Tis the eye
of childhood that fears a painted devil. If he do bleed, I'll
gild the faces of the grooms withal, for it must seem their
guilt. (Exit. Knocking within)</dd>
<dt>Macbeth</dt>
<dd>Whence is that knocking? How is't wit me when
every noise apalls me? What hands are here? Ha! They pluck out
mine eyes! Will all Neptune es ocean wash this blood clean from
my hand? No. This my hand will rather the multitudinous seas
incarnadine, making the green one red. (Enter Lady Macbeth)</dd>
<dt>Lady Macbeth</dt>
<dd>My hands are of your color, but I shame to
wear a heart so white.</dd>
</dl>
```

Das folgende Beispiel zeigt, wie eine Glossar-Liste im Internet Explorer (Abbildung 4.9) dargestellt wird:



```
<dl>
<dt>Basil</dt>
<dd>Annual. Can grow four feet high; the scent
of its tiny white flowers is heavenly.</dd>
<dt>Oregano</dt>
<dd>Perennial. Sends out underground runners
and is difficult to get rid of once established.</dd>
<dt>Coriander</dt>
<dd>Annual. Also called cilantro, coriander
likes cooler weather of spring and fall.</dd>
</dl>
```

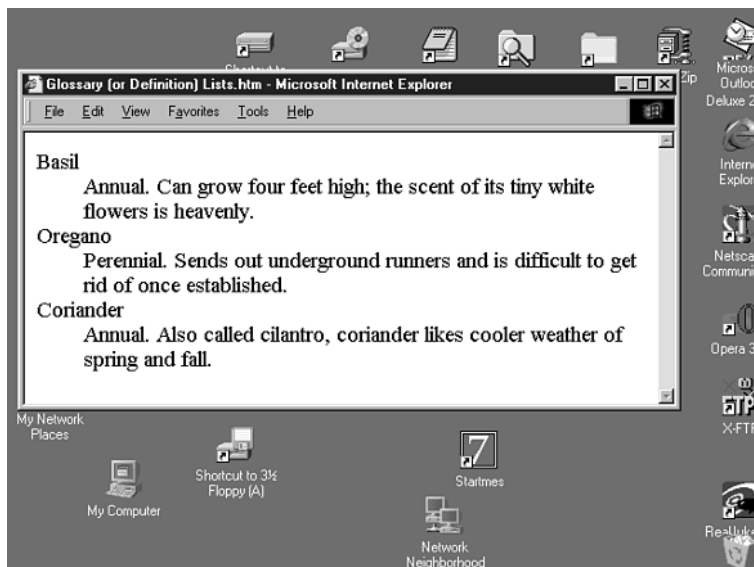


Abbildung 4.9:
Darstellung im
Internet Explorer

Verschachtelte Listen

Was passiert nun, wenn Sie eine Liste in eine andere Liste setzen? Das geht in Ordnung, soweit es HTML betrifft; fügen Sie einfach die komplette Listenstruktur als einen Eintrag in eine andere Liste ein. Die eingefügte Liste wird einfach zu einem weiteren Element der anderen Liste und vom Rest dieser Liste eingerückt abgesetzt. Derartige Listen eignen sich besonders gut für Menü-artige Einträge, bei denen eine Hierarchie abgebildet werden soll (beispielsweise in Inhaltsverzeichnissen), oder bei Entwürfen.

Das Einrücken von verschachtelten Listen im HTML-Code zeigt, wie sie sich im endgültigen Layout verhalten:

```
<ol>
  <ul>
    <li>WWW</li>
    <li>Organization</li>
    <li>Beginning HTML</li>
  </ul>
```



```
<li>What HTML is</li>
<li>How to Write HTML</li>
<li>Doc structure</li>
<li>Headings</li>
<li>Paragraphs</li>
<li>Comments</li>
</ul>
<li>Links</li>
<li>More HTML</li>
</ol>
```

Manche Browser formatieren verschachtelte geordnete und ungeordnete Listen abweichend von der sie umschließenden Liste; beispielsweise könnten sie ein anderes Symbol und nicht den Punkt für eine verschachtelte Liste verwenden oder die umschließende Liste mit Buchstaben anstelle von Zahlen nummerieren. Gehen Sie aber nicht davon aus, sondern beziehen Sie sich in Ihrem Text auf »Absatz 8, Abschnitt b«, da Sie nicht sicher sein können, wie die genaue Formatierung in der endgültigen Darstellung aussehen wird.

Das folgende Beispiel zeigt, wie eine verschachtelte Liste im Internet Explorer (Abbildung 4.10) dargestellt wird:



```
<h1>Peppers</h1>
<ul>
<li>Bell</li>
<li>Chile</li>
  <ul>
    <li>Serrano</li>
    <li>Jalapeno</li>
    <li>Habanero</li>
    <li>Anaheim</li>
  </ul>
<li>Szechuan</li>
<li>Cayenne</li>
</ul>
```

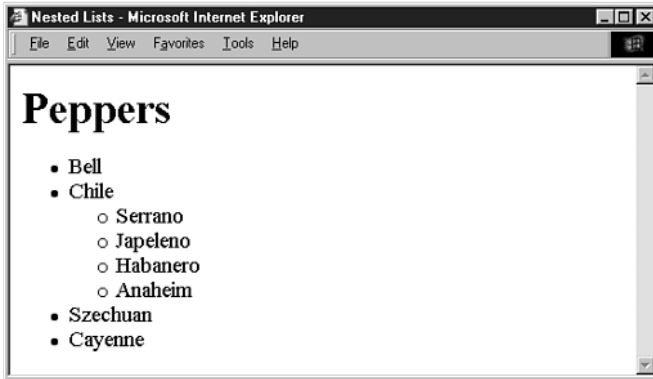


Abbildung 4.10:
Darstellung im Internet
Explorer

Kommentare

Sie können Kommentare in HTML-Dokumente aufnehmen, um das Dokument selbst zu beschreiben oder um irgendwelche Angaben zum Status des Dokuments zu machen; einige Quellcode-Kontroll-Programme können solche Statusangaben beispielsweise in Kommentare schreiben. Der Kommentartext wird ignoriert, wenn die HTML-Datei ausgewertet wird; Kommentare erscheinen auch nicht auf dem Bildschirm – deshalb sind sie ja Kommentare.

Kommentare sehen so aus:

```
<!-- Dies ist ein Kommentar -->
```

Jede Zeile sollte einzeln kommentiert sein, und es ist normalerweise von Vorteil, keine anderen HTML-Kennungen in die Kommentarzeilen einzufügen. (Obwohl dies nicht verboten ist, kommen viele Browser durcheinander, wenn sie in Kommentarzeilen auf HTML-Kennungen treffen, und zeigen sie dann trotzdem an.) Als Faustregel sollte man beachten, die Zeichen <, > oder -- innerhalb eines HTML-Kommentars nicht zu verwenden.

Hier sind einige Beispiele:

```
<!-- schreiben Sie diesen Abschnitt neu mit etwas weniger Humor -->
<!--Neil hat mir bei diesem Abschnitt geholfen -->
<!-- go tigers! -->
```

Übung 4.1: Erzeugen eines richtigen HTML-Dokuments

Mittlerweile sollten Sie schon genug wissen, um einfache HTML-Dokumente erzeugen zu können: Sie wissen, was HTML ist, Sie haben einige HTML-Tags kennen gelernt, und Sie haben sogar probiert, eine HTML-Datei mit dem Browser anzusehen. Sie haben allerdings noch keine Verknüpfungen verwendet, doch das lernen Sie im nächsten Kapitel.

Diese Übung zeigt Ihnen, wie eine HTML-Datei erzeugt wird, die die HTML-Tags, die Sie in diesem Kapitel kennen gelernt haben, benutzt. So bekommen Sie ein Gefühl dafür, wie sie aussehen, wenn sie am Bildschirm angezeigt werden, und welche typischen Fehler Sie dabei machen können. (Niemand bleibt davon verschont, und das ist der Grund, aus dem es oft hilfreich ist, einen HTML-Editor zu verwenden, der Ihnen das Schreiben der HTML-Tags abnimmt. Die Editoren vergessen weder die abschließenden Kennungen noch lassen sie den Schrägstrich weg oder verschreiben sich bei der Kennung selbst.)

Gut, erzeugen wir also ein einfaches Beispiel, verwenden Sie Ihren Editor dazu. Es muss alles in allem nicht viel aussagen; alles, was wir tun müssen, ist, die Struktur-Kennungen einzufügen, einen Titel, ein paar Überschriften und ein oder zwei Absätze. Hier ist das Beispiel:

```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>
<head>
<title>Company Profile, Camembert Incorporated</title>
</head>
<body>
<h1>Camembert Incorporated</h1>
<p>"Many es the long night I dreamed of cheese -- toasted, mostly."
-- Robert Louis Stevenson</p>
<h2>What We Do</h2>
<p>We make cheese. Lots of cheese; more than eight tons of cheese
a year.</p>
<h2>Why We Do It</h2>
<p>We are paid an awful lot of money by people who like cheese.
So we make more.</p>
<h2>Our Favorite Cheeses</h2>
<ul>
<li>Brie</li>
<li>Havarti</li>
<li>Camembert</li>
<li>Mozzarella</li>
</ul>
```

```
</body>
</html>
```

Speichern Sie es als HTML-Datei, laden Sie es in Ihren Browser, und schauen Sie sich an, was daraus geworden ist.

Wenn Sie Zugriff auf einen anderen Browser auf Ihrer oder einer anderen Plattform haben, empfehle ich dringend, dass Sie versuchen, die gleiche HTML-Datei auch mit dem anderen Browser zu betrachten, sodass Sie sehen, wie die verschiedenen Browser Ihre HTML-Seite darstellen. Bisweilen können diese Unterschiede Sie überraschen; Zeilen, die in einen Browser gut aussehen, erscheinen im anderen ziemlich seltsam.

Abbildung 4.11 zeigt das Käsefabrik-Beispiel im Internet Explorer

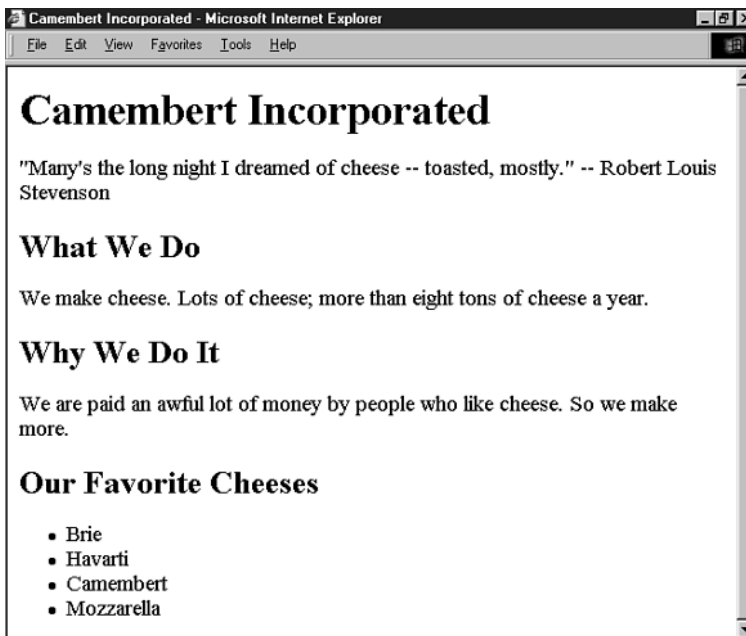


Abbildung 4.11:
Darstellung des
Käsefabrik-Bei-
spiels im Internet
Explorer

Zusammenfassung

HTML ist eine reine Text-Auszeichnungssprache, die verwendet wird, um Hypertext-Seiten im World Wide Web zu erstellen. Sie beschreibt die Struktur einer Seite, nicht ihr Erscheinungsbild.

In diesem Kapitel haben Sie gelernt, was HTML ist und wie Sie einfache HTML-Dateien schreiben und betrachten können. Sie haben darüber hinaus folgende HTML-Kennungen aus Tabelle 3.1 kennen gelernt.

Kennung	Attribute	Verwendungszweck
<code><html> ... </html></code>		Das gesamte HTML-Dokument
<code><head> ... </head></code>		Der Header oder Vorspann des HTML-Dokuments
<code><body> ... </body></code>		Der restliche Inhalt des HTML-Dokuments
<code><title> ... </title></code>		Der Titel des HTML-Dokuments
<code><h1> ... </h1></code>		Überschrift, Stufe 1
<code><h2> ... </h2></code>		Überschrift, Stufe 2
<code><h3> ... </h3></code>		Überschrift, Stufe 3
<code><h4> ... </h4></code>		Überschrift, Stufe 4
<code><h5> ... </h5></code>		Überschrift, Stufe 5
<code><h6> ... </h6></code>		Überschrift, Stufe 6
<code><p> ... </p></code>		Absatz
<code>...</code>		Eine geordnete (nummerierte) Liste. Die Listeneinträge beginnen jeweils mit <code></code> .
<code>...</code>		Individuelle Einträge in geordneten, ungeordneten, Menü- oder in Inhaltsverzeichnis-Listen. Das abschließende Tag ist in HTML fakultativ und in XHTML 1.0 obligatorisch.
	<code>type</code>	Legt das Schema der Nummerierung für die Liste fest.
	<code>start</code>	Legt fest, mit welcher Nummer die Liste beginnen soll.
	<code>compact</code>	Legt fest, dass ein kompaktes Format für die Liste verwendet werden soll.
<code>...</code>		Eine ungeordnete (mit Punkten oder anders markierte) Liste. Die Listeneinträge beginnen jeweils mit <code></code> .
	<code>type</code>	Legt die Art der Blickfangpunkte für die Liste fest.
	<code>compact</code>	Legt fest, dass ein kompaktes Format für die Liste verwendet werden soll.
<code><menu>...</menu></code>		Eine Menü-Liste (eine Liste mit kurzen Einträgen oder Absätzen).
	<code>compact</code>	Legt fest, dass ein kompaktes Format für die Liste verwendet werden soll.

Tabelle 4.1: Die HTML-Kennungen aus Kapitel 4

Kennung	Attribute	Verwendungszweck
<code><div>...</div></code>		Eine Liste mit besonders kurzen (1-2 Wörter) Einträgen. Verzeichnis-Listen werden in den meisten HTML-Dateien nicht oft benutzt.
	compact	Legt fest, dass ein kompaktes Format für die Liste verwendet werden soll.
<code></code>		Ein einzelner Listeneintrag in geordneten, ungeordneten, Menü- oder Verzeichnis-Listen.
	type	Legt innerhalb der Liste, beginnend mit dem aktuellen Listeneintrag, eine neue Nummerierungsart bzw. neue Blickfangpunkte fest. Funktioniert nur mit <code></code> - und <code></code> -Listen.
	value	Legt innerhalb einer <code></code> -Liste, beginnend mit dem aktuellen Eintrag, einen neuen Wert für die weitere Nummerierung fest.
<code><dl>...</dl></code>		Eine Glossar- oder Definitions-Liste. Einträge in der Liste haben zwei Bestandteile: den Ausdruck und seine Definition.
	compact	Legt fest, dass ein kompaktes Format für die Liste verwendet werden soll.
<code><dt></code>		Der Ausdruck einer Glossar-Liste.
<code><dd></code>		Die Definition einer Glossar-Liste.
<code><!-- ... --></code>		Kommentar.

Tabelle 4.1: Die HTML-Kennungen aus Kapitel 4

Workshop

In diesem Kapitel haben Sie viel gelernt und unser Workshop heute wird Ihnen dabei helfen, sich an die wichtigsten Punkte zu erinnern. Einige der Fragen, die im Laufe des Kapitels wahrscheinlich aufkamen, werden im folgenden Abschnitt behandelt.

F Ich habe bei vielen Webseiten bemerkt, dass die Struktur-Tags (`<html>`, `<head>`, `<body>`) oft nicht verwendet werden. Muss ich sie wirklich einfügen, auch wenn die Seiten ohne sie funktionieren?

A Die meisten Browser können HTML-Dateien ohne die Struktur-Tags verarbeiten. Wenn wir aber mit XHTML arbeiten, sind die Seitenstruktur-Tags wichtig und sie müssen obligatorisch auf Ihrer Seite erscheinen. Deshalb sollten Sie sich schon jetzt mit diesen Tags vertraut machen und sie anwenden. Das Ein-

fügen dieser Kennungen erlaubt es, die Dokumente auch von allgemeineren SGML-Tools einlesen zu lassen und in den Genuss der Möglichkeiten zukünftiger Browser zu kommen. Und es ist die »korrekte« Art, es zu tun, wenn Sie wollen, dass Ihre Dokumente dem eigentlichen HTML-Format entsprechen.

F Beim Ausdrucken meines Glossars musste ich feststellen, dass die Formatierung doch sehr seltsam ist. Die Begriffe sind mehr eingerückt als die Definitionen.

A Kann es sein, dass Sie das `<dd>`-Tag mit dem `<dt>`-Tag verwechselt haben? Das `<dt>`-Tag wird immer zuerst verwendet (Definitionsbegriff) und dann kommt erst das `<dd>`-Tag (Definition). Ich selbst verwechsle diese Tags andauernd. Es gibt eben zu viele D-Tags in Glossar-Listen.

F Ich habe HTML-Dateien gesehen, die `<li>` allein auf einer Seite und außerhalb einer Listenstruktur haben. Zum Beispiel: `<li> and then the duck said, »put it on my bill« </li>`.

A Die meisten Browser akzeptieren dieses Tag außerhalb des List-Tag und werden es entweder in einen einfachen Paragraphen oder in ein nicht eingerücktes Aufzählungszeichen konvertieren. Wenn man sich jedoch an die richtige HTML-Definition hält, darf man kein `<li>`-Tag außerhalb eines List-Tag verwenden. »Gute« HTML-Seiten sollten dies also nicht tun. Und weil Sie sicherlich danach streben, gutes HTML zu schreiben (oder?), sollten auch Sie Ihre Listen nicht auf diese Art und Weise schreiben. Geben Sie also immer Ihre Listen-Elemente innerhalb der Liste an, zu der sie auch gehören.

F Sie haben erwähnt, dass einige der Listen-Tags in HTML 4.0 verworfen werden. Was soll ich anstelle dieser Tags nehmen?

A In gewisser Weise, hängt das von den Besuchern Ihrer Site ab. Wenn sich zum Beispiel Ihre Webseiten im Intranet einer Gesellschaft befinden, wo Sie sicher sein können, dass jeder HTML 4.0 Browser verwendet, die Style Sheets (CSS) unterstützen, können Sie CSS-Eigenschaften und -Werte anstelle der verworfenen Tags verwenden. Stehen Ihre Webseiten nun im World Wide Wide, müssen Sie davon ausgehen, dass eine Vielzahl von Browsern und PC-Plattformen verwendet werden, um Zugriff auf Ihre Site zu haben. Hier könnte es von Vorteil sein, weiterhin mit den verworfenen Tags zu arbeiten, denn somit können Ihre Seiten in älteren Browsern präsentiert werden. Im Kapitel des 18. Tages werden Sie mehr über die Pros und Contras erfahren.

Quiz

1. Welche drei `<html>`-Tags werden verwendet, um die Gesamtstruktur einer Webseite zu beschreiben? Was genau wird von jedem der Tags beschrieben?
2. Wohin gehört das `<title>`-Tag und für was wird es verwendet?
3. Wie viele Überschriftenordnungen werden in HTML unterstützt? Wie lauten die jeweiligen Tags?
4. Warum ist es eine gute Idee immer ein einleitendes und ein abschließendes Tag zu verwenden, obwohl doch das abschließende Tag in HTML nicht zwingend ist?
5. Welche zwei Listenarten sind veraltet? Was kann man anstelle der verworfenen Listenarten verwenden?

Antworten

1. Das `<html>`-Tag gibt an, dass die Sprache des Dokuments HTML ist. Das `<head>`-Tag spezifiziert, dass die Zeilen innerhalb des Tag der Prolog für die restliche Datei ist. Das `<body>`-Tag enthält den Rest Ihrer HTML-Seite (Text, Links, Bilder usw.).
2. Das `<title>`-Tag wird verwendet, um den Titel einer Webseite in einer Browser-Lesezeichenliste (zum schnellen Finden von Daten), in einer Hot List (Liste bevorzugter Dateien oder Webseiten) oder in einem Programm, das Webseiten katalogisiert, anzuzeigen. Dieses Tag befindet sich immer innerhalb des `<head>`-Tags.
3. HTML unterstützt sechs Überschriftenordnungen. Die Tags sind hierbei `<h1>...</h1>` bis `<h6>...</h6>`.
4. Das abschließende `</p>`-Tag ist wichtig, wenn man den Text nach links oder rechts ausrichten will oder ihn zentrieren möchte. In XHTML 1.0 muss man mit abschließenden Tags arbeiten.
5. Die `<menu>`- und `<dir>`-Listarten wurden zugunsten von Merkmallisten oder ungeordneten Listen `<ul>` verworfen.

Übungen

1. Erstellen Sie, unter Zuhilfenahme der Camembert Incorporated Seiten, eine Seite, die die Themen beschreibt, die Sie auf Ihrer Web-Site haben möchten. Mit dieser Seite werden Sie im nächsten Kapitel lernen, wie man Links erstellt.
2. Erstellen Sie eine zweite Seite, die zusätzliche Informationen über eines der aufgelisteten Themen in Übung 1 liefert. Fügen Sie einige Unterpunkte in diese Seite mit ein (siehe Bild 4.2). Sollten Sie besonders abenteuerlustig sein, vervollständigen Sie den Seiteninhalt und erstellen Sie Listen, von denen Sie denken, dass Sie die Seite erweitern könnten. Diese Übung ist schon einmal ein kleiner Vorschmack von dem, was Sie im nächsten Kapitel erwartet.



Alles über Verknüpfungen

**Woche
1**

Nach dem Abschluss des letzten Kapitels haben Sie nun einige Seiten mit Überschriften, Texten und Listen darin. Das ist ja alles schön und gut, aber noch ziemlich langweilig. Der richtige Spaß beginnt erst, wenn Sie lernen, wie man Hypertext-Verknüpfungen herstellt und wie Sie all Ihre Dokumente mit dem Web verbinden, und in diesem Kapitel werden Sie genau das lernen.

Alles über das <a>-Tag

Wie verweise ich auf andere Seiten Ihrer lokalen Plattform und verwende dabei relative und absolute Pfadnamen?

- ▶ wie Sie mit URLs Verbindungen zu anderen Seiten im Web erzeugen,
- ▶ wie Sie Verknüpfungen und Anker einsetzen, um zu spezifischen Stellen innerhalb der Seiten zu führen,
- ▶ alles über URLs: die verschiedenen Bestandteile der URL und die Arten von URLs, die sie verwenden können.

Verknüpfungen erzeugen

Um eine Verknüpfung (eine Verbindung mit einem anderen Dokument) in HTML zu erzeugen, brauchen Sie zwei Dinge:

- ▶ den Namen der Datei (oder die URL der Datei), mit der Sie eine Verbindung herstellen wollen,
- ▶ einen Text, der als Fixpunkt dienen kann – das heißt den Text, der vom Browser optisch hervorgehoben wird und den Ihre Leser anwählen, um der Verbindung zu folgen.

Nur das zweite Element ist in Ihrem Dokument tatsächlich sichtbar. Wenn Ihre Leser den Text auswählen, der auf eine Verknüpfung deutet, dann verwendet der Browser das erste Element, um zu dem angegebenen Dokument zu »springen«.

Das Verknüpfungs-Tag): <a>

Um in einem HTML-Dokument eine Verknüpfung zu erzeugen, benutzen Sie die HTML-Verknüpfungs-Tags <a>.... Die <a>-Kennung wird häufig auch Anker-Kennung genannt, da sie auch verwendet werden kann, um Anker für Verknüpfungen

zu erzeugen. (Sie werden später in diesem Kapitel mehr über die Erzeugung von Anker lernen.) Die gebräuchlichste Verwendung des Verknüpfungs-Tag ist jedoch, Verbindungen mit anderen Dokumenten zu erzeugen.

Anders als die einfachen Tags, die Sie im letzten Kapitel kennen gelernt haben, besitzt die `<a>`-Kennung einige zusätzliche Merkmale: Die einleitende Kennung, `<a>` enthält sowohl den Namen der Kennung (»A«) als auch zusätzliche Angaben zu der Verknüpfung selbst. Die zusätzlichen Merkmale werden »Attribute« der Kennung genannt. (Sie haben Attribute zum ersten Mal in Kapitel 4 kennen gelernt, als das Thema Listen behandelt wurde.) Die einleitende Kennung enthält also nicht nur den Namen in spitzen Klammern, sondern sie sieht ungefähr so aus:

```
<a name="up" href="menu.html" title="the twelve caesars">
```

Die zusätzlichen Attribute (in diesem Fall `name`, `href` und `title`) beschreiben die Verbindung selbst. Das Attribut, das Sie wahrscheinlich am häufigsten verwenden werden, ist das `href`-Attribut, eine Kurzform von »Hypertext REFERENCE«. `href` wird verwendet, um den Namen oder die URL der Datei zu bezeichnen, auf die diese Verknüpfung weist.

Wie die meisten HTML-Tags hat auch das Verknüpfungs-Tag eine abschließende Kennung: `</a>`. Der gesamte Text, der sich zwischen der einleitenden und der abschließenden Kennung befindet, wird als Link angesehen und hervorgehoben, unterstrichen oder farbig (blau bzw. rot) dargestellt, wenn die Webseite angezeigt wird. Das ist der Text, den Sie oder Ihre Leser anklicken (oder bei Browsern, die keine Maus unterstützen, auswählen), um zu jenem Dokument zu springen, das durch das `href`-Attribut angegeben wurde.

Abbildung 5.1 zeigt die Teile einer typischen Verknüpfung mit der `<a>`-Kennung, einschließlich `href`, des Textes und der abschließenden Kennung:

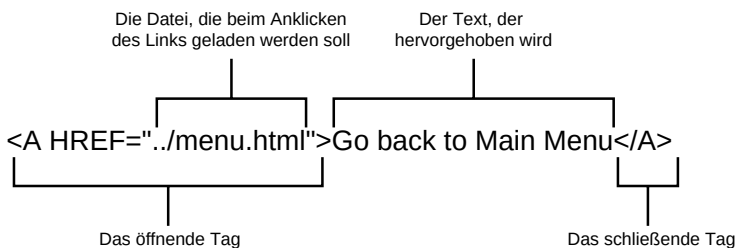


Abbildung 5.1:
Eine HTML-Verknüpfung mit der `<a>`-Kennung

Das folgende Beispiel zeigt eine einfache Verknüpfung und wie sie in Internet Explorer (Abbildung 5.2) aussieht.



Go back Main Menu



Abbildung 5.2:
Die Ausgabe im Internet
Explorer

Übung 5.1: Verbindung zweier Seiten

Lassen Sie uns nun durch ein recht einfaches Beispiel gehen, mit zwei HTML-Dokumenten auf Ihrer lokalen Platte. Sie brauchen Ihren Text-Editor und einen Web-Browser dafür, doch da beide Dokumente, mit denen Sie arbeiten, auf Ihrer lokalen Platte liegen, brauchen Sie nicht mit dem Netzwerk verbunden zu sein. (Haben Sie Geduld; im nächsten Abschnitt dieses Kapitels kommen wir auf das Netz zu sprechen.)

Zunächst erzeugen Sie zwei HTML-Dokumente und sichern sie in verschiedenen Dateien. Hier ist der Code der beiden HTML-Dateien, die ich mir für diesen Abschnitt ausgedacht und die ich `menu.html` und `feeding.html` genannt habe. Es macht wirklich nichts aus, wie Ihre Dokumente aussehen oder wie sie genannt werden, aber achten Sie darauf, Ihre eigenen Datei-Namen zu benutzen, wenn Sie mit diesem Beispiel fortfahren.

Die erste Datei heißt `menu.html` und sieht so aus:

```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>
<head>
<title>The Twelve Caesars</title>
</head>
<body>
<h1>"The Twelve Caesars" by Suetonius</h1>
<p>Seutonium (or Gaius Suetonius Tranquillus) was born circa A.D. 70
```

and died sometime after A.D. 130. He composed a history of the twelve Caesars from Julius to Domitian (died A.D. 96). His work was a significant contribution to the best-selling novel and television series "I, Claudius." Suetonius' work includes biographies of the following Roman emperors:</p>
</div>
<div data-bbox="104 212 356 467" data-label="Text">
<pre>

 Julius Caesar
 Augustus
 Tiberius
 Gaius (Caligula)
 Claudius
 Nero
 Galba
 Otho
 Vitellius
 Vespasian
 Titus
 Domitian

</body>
</html>
</pre>
</div>
<div data-bbox="104 478 889 532" data-label="Text">
<p>Die Liste der Menü-Punkte (»Julius Caesar, Augustus usw.) werden später zu Verknüpfungen mit anderen Dokumenten. Im Augenblick aber schreiben Sie sie als normalen Text; Sie verwandeln sie erst später in Verknüpfungen.</p>
</div>
<div data-bbox="104 540 516 558" data-label="Text">
<p>Die zweite Datei, claudius.html, sieht so aus:</p>
</div>
<div data-bbox="104 568 819 890" data-label="Text">
<pre>
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>
<head>
<title>The Twelve Caesars: Claudius</title>
</head>
<body>
<h2>Claudius Becomes Emperor</h2>
<p>Claudius became Emperor at the age of 50. Fearing the attack of Caligula es assassins, Claudius hid behind some curtains. After a guardsman discovered him, Claudius dropped to the floor, and then found himself declared Emperor.</p>
<h2>Claudius is Poisoned</h2>
<p>Most people think that Claudius was poisoned. Some think his wife Agrippina poisoned a dish of mushrooms (his favorite food). His death was revealed after arrangements had been made for her son, Nero, to succeed as Emperor.</p>
<p>Go back to Main Menu</p>
</body>
</html>
</pre>
</div>
<div data-bbox="829 917 889 937" data-label="Page-Footer">
<p>115</p>
</div>

Stellen Sie sicher, dass beide Dateien in demselben Verzeichnis oder Ordner gespeichert sind, und wenn Sie sie nicht `menu.html` und `claudius.html` genannt haben, merken Sie sich die Namen, denn Sie werden sie später brauchen.

Zuerst erzeugen Sie nun eine Verbindung von der Datei `menu` zur Datei `claudius`. Editieren Sie die Datei `menu.html`, und setzen Sie den Cursor auf die darauffolgende Zeile:

```
<li>claudius </li>
```

Verknüpfungs-Tags definieren nicht das Format des Textes selbst; lassen Sie also die Absatz-Kennungen in Ruhe, und fügen Sie das Verknüpfungs-Tag innerhalb des Absatzes ein. Als Erstes fügen Sie die Verknüpfungs-Kennungen (die `<a>`- und `</a>`-Kennungen) vor und hinter den Text ein, den Sie als Verbindung (Link) benutzen wollen:

```
<li> <a> Claudius </a> </li>
```

Nun ergänzen Sie den Namen der Datei, mit der Sie eine Verbindung herstellen wollen, als den `href`-Teil der einleitenden Kennung. Schließen Sie den Namen in Gänsefüßchen ein (mit geraden Strichen ["] nicht mit gerundeten [']), mit einem Gleichheitszeichen zwischen `href` und dem Namen. Beachten Sie, dass zwischen Groß- und Kleinschreibung unterschieden wird, und vergewissern Sie sich deshalb, den Datei-Namen genauso einzutippen, wie Sie ihn beim Abspeichern geschrieben haben (`Claudius.html` ist nicht das gleiche wie `claudius.html`; Groß- und Kleinschreibung beachten!). Hier habe ich `feeding.html` benutzt; wenn Sie einen anderen Datei-Namen verwendet haben, nehmen Sie diesen.

```
<li> <a href="claudius.html"> Claudius </a></li>
```

So, nun starten Sie Ihren Browser, wählen »Öffnen« (oder das passende Äquivalent) und öffnen die Datei `menu.html`. Der Absatz, den Sie für die Verknüpfung benutzt haben, sollte nun als Verbindung in einer anderen Farbe, unterstrichen oder sonstwie hervorgehoben angezeigt werden. Abbildung 5.3 zeigt, wie das in der Windows-98-Version des Internet Explorer aussieht.

Und wenn Sie nun auf die Verknüpfung klicken, sollte Ihr Browser die Datei `claudius.html` laden und diese so anzeigen, wie es in Abbildung 5.4 dargestellt wird.

Wenn Ihr Browser die Datei nicht finden kann, überprüfen Sie, ob der Name der Datei im `href`-Teil der Verknüpfungs-Kennung derselbe ist wie der Name der Datei auf Ihrer Platte und ob beide Dateien sich in demselben Verzeichnis befinden. Denken Sie daran, das Verknüpfungs-Tag mit der `</a>`-Kennung am Ende des Textes, der als Verbindung dient, abzuschließen. Und stellen Sie auch sicher, dass Sie Gänsefüßchen am Ende des Datei-Namens verwenden (es ist leicht, sie zu vergessen) und dass diese Gänsefüßchen von der gewöhnlichen, geraden Sorte sind. All diese Dinge können den Browser durcheinanderbringen und dazu führen, dass er die Datei nicht findet oder die Verknüpfung nicht richtig anzeigt.

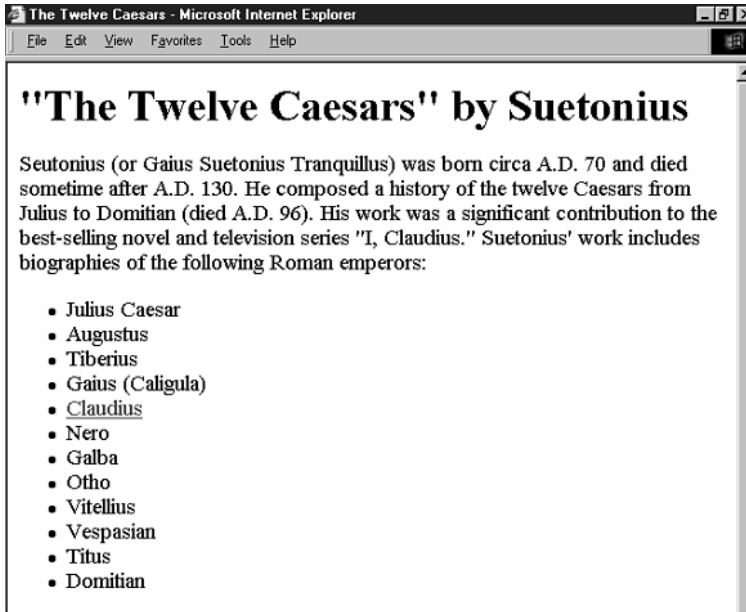


Abbildung 5.3:
Die menu.html-
Datei mit einer
Verknüpfung

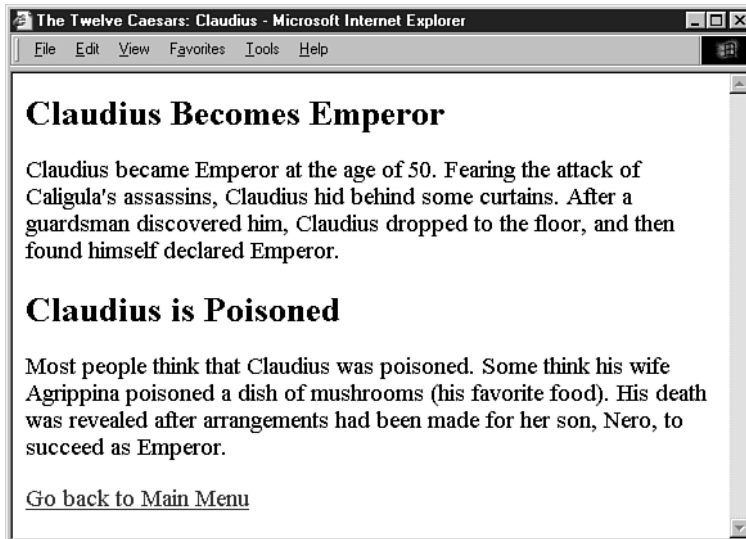


Abbildung 5.4:
Die claudius
.html-Seite



Lassen Sie sich von der Groß-/Kleinschreibung nicht verwirren. Die Tags in HTML sind unabhängig davon, ob sie groß oder klein geschrieben werden. Datei-Namen dagegen beziehen sich auf Dateien, die sich irgendwo auf einem Web-Server befinden. Da die meisten Web-Server unter einem Be-

triebssystem laufen, das bei Datei-Namen zwischen Groß- und Kleinschreibung unterscheidet (zum Beispiel Unix oder Windows NT), sollten Sie sicherstellen, dass die Schreibung der Datei-Namen in Ihren Links korrekt ist.

Nun lassen Sie uns eine Verbindung `feeding.html` zurück zu `menu.html` erzeugen. Es gibt einen Absatz am Ende des Dokuments `claudius.html`, der für diesen Zweck gedacht ist:

```
<p>Go back to Main Menu</p>
```

Ergänzen Sie die Verbindungs-Kennung mit der benötigten `href` in dieser Zeile wie hier, wobei `menu.html` die ursprüngliche Menü-Datei ist:

```
<p> <a href="menu.html">Go back to Main Menu</a> </p>
```



Wenn Sie Tags innerhalb anderer Tags verwenden, achten Sie darauf, dass die abschließende Kennung jenes Tag abschließt, das Sie zuletzt geöffnet haben. Das heißt, Sie machen es so:

```
<p> <a> ... </a> </p>
```

und nicht so:

```
<p> <a> ... </p> </a>
```

Manche Browser könnten durcheinander kommen, wenn sich Kennungen derart überschneiden, daher ist es am besten, immer darauf zu achten, dass das zuletzt geöffnete Tag als Erstes wieder geschlossen wird.

Wenn Sie nun die Claudius-Datei erneut laden, wird die Verbindung aktiv sein, und Sie können zwischen der `menu-` und `claudius-`Datei hin- und herspringen, indem Sie diese Verknüpfungen auswählen.

Verknüpfen lokaler Dokumente: relative und absolute Pfad-Namen

Das Beispiel aus dem vorhergehenden Abschnitt zeigt, wie man Dokumente miteinander verbindet, die in demselben Verzeichnis oder Ordner auf Ihrer lokalen Platte abgelegt sind (passenderweise *lokale* Dokumente genannt). Dieser Abschnitt spinnt diesen Faden weiter, indem wir Dokumente verbinden, die sich zwar auf der lokalen Platte befinden, aber in unterschiedlichen Verzeichnissen dieser Platte liegen können.



Ordner und Verzeichnisse beziehen sich zwar auf denselben Sachverhalt, werden aber, je nachdem, ob Sie sich auf einem Mac-, Windows-, DOS- oder Unix-System befinden, unterschiedlich genannt. Ich werde sie von jetzt ab Verzeichnisse nennen, um die Sache zu vereinfachen.

Wenn Sie den Pfad-Namen einer Datei, mit der Sie eine Verbindung herstellen wollen, als Datei-Namen, in Gänsefüßchen, angeben, sucht der Browser die angegebene Datei in demselben Verzeichnis, in dem sich auch die aktuelle Datei befindet, auch wenn der Browser von weither über das Netz danach sucht. Dies ist die einfachste Form eines relativen Pfad-Namens.

Relative Pfad-Namen können auch Verzeichnis-Namen enthalten, oder sie können auf den Pfad verweisen, den Sie benutzen würden, um sich zu der Datei zu bewegen, wenn Sie im aktuellen Verzeichnis oder Ordner starten würden. Ein Pfad-Name kann Bewegungsrichtungen enthalten, um beispielsweise zwei Verzeichnisebenen »hoch« zu gehen und dann zwei andere Verzeichnis-Ebenen »runter«, um die Datei zu erreichen.



Relative Pfad-Namen zeigen auf Dateien, indem sie vom Ort relativ zur aktuellen Datei ausgehen.

Um relative Pfad-Namen in Verknüpfungen anzugeben, verwenden Sie Pfad-Namen im Unix-Stil, unabhängig davon, mit welchem System Sie tatsächlich arbeiten. Das bedeutet, dass Verzeichnis- oder Ordner-Namen durch »normale« Schrägstriche (»/«) getrennt werden und dass Sie zwei Punkte benutzen, um sich auf das Verzeichnis zu beziehen, das über dem aktuellen liegt (»../«).

Tabelle 5.1 zeigt einige Beispiele für relative Pfad-Namen und was sie bedeuten.

Pfad-Name	Bedeutung
<code>href="datei.html"</code>	<code>datei.html</code> liegt im aktuellen Verzeichnis, also in der Verzeichnis-Ebene, in der man sich gerade befindet
<code>href="dateien/datei.html"</code>	<code>datei.html</code> liegt im Verzeichnis -Namens <code>dateien</code> (und das <code>dateien</code> -Verzeichnis liegt im aktuellen Verzeichnis)
<code>href="dateien/mehr/datei.html"</code>	<code>datei.html</code> liegt im <code>mehr</code> -Verzeichnis, das im <code>dateien</code> -Verzeichnis liegt, das wiederum im aktuellen Verzeichnis liegt
<code>href="../datei.html"</code>	<code>datei.html</code> liegt im Verzeichnis oberhalb des aktuellen Verzeichnisses, demselben Verzeichnis, in dem auch das aktuelle Verzeichnis enthalten ist
<code>href="../../dateien/datei.html"</code>	<code>datei.html</code> liegt zwei Verzeichnis-Ebenen oberhalb des aktuellen Verzeichnisses im Verzeichnis <code>dateien</code>

Tabelle 5.1: Relative Pfad-Namen

Wenn Sie Dateien auf einem Personalcomputer (Mac oder PC) verknüpfen und eine Verbindung zu einer Datei auf einem anderen Laufwerk herstellen wollen, verwenden Sie den Namen des Laufwerks einfach als einen weiteren Verzeichnis-Namen im relativen Pfad.

Auf dem Macintosh wird der Name des Laufwerks so benutzt, wie er auf der Platte selbst erscheint. Nehmen wir an, Sie haben eine Platte namens `Hard Disk 2` und Ihre HTML-Dateien sind in einem Ordner, der passenderweise `HTML-Dateien` heißt. Wenn Sie nun eine Verbindung zu einer Datei namens `jane.html` in einem Ordner namens `Public` auf einem Laufwerk namens `Jane es Mac` herstellen wollten, so würden Sie den folgenden relativen Pfad-Namen benutzen:

```
href="../../Jane es Mac/Public/jane.html"
```

Auf DOS-Systemen, Windows 95/98 oder Windows NT-Systemen werden die Laufwerke mit Buchstaben bezeichnet, genauso, wie Sie es erwarten würden, doch anstatt `C:`, `D:` und so weiter zu verwenden, ersetzen Sie den Doppelpunkt durch einen senkrechten Strich (`|`, ASCII-Zeichen Nr. 124) – der Doppelpunkt hat eine spezielle Bedeutung in einer Verknüpfung, und vergessen Sie auch nicht, wie bei Unix den Schrägstrich zu verwenden. Wenn also die aktuelle Datei in `C:\DATEIEN\HTML` liegt und Sie eine Verbindung herstellen wollen mit `D:\DATEIEN.NEU\HTML\MEHR\INDEX.HTM`, so wäre der relative Pfad-Name dieser:

```
href="../../d|/dateien.neu/html/mehr/index.htm"
```

In den meisten Fällen werden Sie den Namen einer Festplatte in relativen Pfad-Namen nicht benutzen, ich habe es hier jedoch der Vollständigkeit halber eingefügt. Meistens werden Sie, wenn es sich um dieselbe Präsentation handelt, Dateien verknüpfen, die einigermaßen dicht beieinander sind (nur ein oder zwei Verzeichnisse oder Ordner voneinander entfernt).

Absolute Pfad-Namen

Sie können die Verbindung mit einem anderen Dokument in Ihrem lokalen System auch durch einen absoluten Pfad-Namen bezeichnen. Während relative Pfad-Namen auf das Dokument, mit dem eine Verbindung hergestellt werden soll, durch eine Beschreibung seiner Relation zum aktuellen Dokument verweisen, zeigen absolute Pfad-Namen auf das Dokument, indem sie auf der obersten Ebene Ihrer Verzeichnis-Hierarchie beginnen und sich dann durch alle darunterliegenden Verzeichnisse arbeiten, bis sie die Datei erreicht haben.



Absolute Pfad-Namen weisen auf Dateien hin, indem sie den absoluten Ort innerhalb des Dateisystems angeben.

Absolute Pfad-Namen beginnen immer mit einem Schrägstrich (`»/«`), wodurch sie von relativen Pfad-Angaben unterschieden werden können. Auf den Schrägstrich folgen alle Verzeichnisse im Pfad von der obersten Ebene bis zu der Datei, mit der Sie eine Verbindung herstellen wollen. Tabelle 5.2 zeigt einige Beispiele von absoluten Pfad-Namen und was sie bedeuten.



Die »oberste Ebene« hat unterschiedliche Bedeutungen, je nachdem, wie Sie Ihre HTML-Dateien publizieren. Wenn Sie nur Dateien auf Ihrer lokalen Platte verbinden, wäre die »oberste Ebene« das Stammverzeichnis Ihres Dateisystems (`»/«` unter Unix oder der Laufwerks-Name auf einem Mac oder das Root-Verzeichnis auf einem PC). Wenn Sie Ihre Dateien mit einem Web-Server publizieren, dann kann die »oberste Ebene« zugleich das Stammverzeichnis Ihres Dateisystems sein oder auch nicht (im Allgemeinen ist das nicht der Fall). Über absolute Pfad-Namen und Web-Server erfahren Sie mehr am 19. Tag.

Die Tabelle 5.2 zeigt einige Beispiele absoluter Pfad-Namen und deren Bedeutung.

Pfad-Name	Bedeutung
<code>href="/u1/lemay/datei.html"</code>	<code>datei.html</code> liegt im Verzeichnis <code>/u1/lemay</code> (i.d.R. Unix)
<code>href="/d /dateien/html/datei.htm"</code>	<code>datei.htm</code> liegt auf dem Laufwerk D: im Verzeichnis <code>/dateien/html</code> (DOS-System)
<code>href="/Festplatte 1/HTML Dateien/datei.html"</code>	<code>datei.html</code> liegt auf Laufwerk Festplatte 1 im Ordner HTML Dateien (i.d.R. Macintosh)

Tabelle 5.2: Absolute Pfad-Namen

Sollten Sie relative oder absolute Pfad-Namen verwenden?

Um Ihre eigenen Seiten miteinander zu verknüpfen, sollten Sie in den meisten Fällen relative Pfad-Namen anstelle absoluter Pfad-Namen benutzen. Die Verwendung absoluter Pfad-Namen mag bei komplizierten Verbindungen von vielen Dokumenten leichter erscheinen, doch absolute Pfad-Namen sind nicht portabel, das heißt übertragbar. Wenn Sie Ihre Verknüpfungen als absolute Pfad-Namen angeben und Sie verschieben die Dateien auf Ihrem Laufwerk oder benennen ein Verzeichnis oder Laufwerk um, das in diesen Pfad-Namen aufgeführt ist, dann zerbrechen all Ihre Verbindungen, und Sie haben ordentlich damit zu tun, all Ihre HTML-Dateien zu editieren und zu korrigieren. Die Verwendung absoluter Pfad-Namen macht es auch sehr schwierig, Ihre Dateien auf einen Web-Server zu schieben, wenn Sie einmal beschließen, sie tatsächlich im Web verfügbar zu machen.

Die Verwendung relativer Pfad-Namen erlaubt es Ihnen, Ihre Dokumente auf Ihrem eigenen System umherzuschieben und sie auf ein anderes System, mit wenig oder gar keinen Veränderungen, zu verlagern, um die Verbindungen zu bewahren. Es ist viel leichter, HTML-Dokumente mit relativen Pfad-Namen zu pflegen, sodass die manchmal zusätzlich anfallende Arbeit, um sie einmal zu erzeugen, sich sehr schnell bezahlt macht.

Verbindungen mit anderen Dokumenten im Web

Sie haben nun eine ganze Menge eigener Dokumente miteinander verknüpft. An manchen Stellen in Ihren Dokumenten würden Sie jedoch gerne auf eine Seite verweisen, die sich irgendwo anders im Netz befindet, beispielsweise auf die Homepage der »The First Caesars«-Seite, von Dr. Ellis Knox, der an der Boise State Universität unterrichtet. Sie können die Verknüpfungs-Kennung ebenso benutzen, um Verbindungen zu anderen Dokumenten im Netz herzustellen, die ich entfernte (remote) Dokumente nenne.



Entfernte Seiten (Remote pages) sind Seiten, die sich im Web auf einem anderen System befinden als dem, mit dem Sie momentan arbeiten.

Der HTML-Code, den Sie benutzen, um Verbindungen zu anderen Seiten im Netz herzustellen, sieht genauso aus wie der Code, mit dem Sie lokale Verbindungen realisiert haben. Sie verwenden noch immer die `<a>`-Kennung mit einem `href`-Attribut und verwenden etwas Text, der auf Ihrer Webseite als Verknüpfung dient. Doch anstelle eines Datei-Namens oder -pfades in der `href` benutzen Sie die URL jener Seite im Web, wie Abbildung 5.5 zeigt.

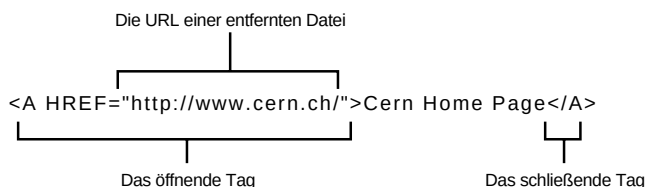


Abbildung 5.5:
Verknüpfung mit einer
remote-Datei

Übung 5.2: Verbinden Sie Ihre Caesar-Seiten mit dem Web

Kommen wir nun also auf die Dokumente über Caesar zurück, die Sie vorhin in diesem Kapitel miteinander verbunden haben. Die Datei `menu.html` enthält einige Verbindungen mit anderen lokalen Dokumenten, die Beschreibungen über die 12 römischen Kaiser liefern.

Lassen Sie uns nun annehmen, Sie wollten am Ende der Menü-Seite eine Verbindung hinzufügen, die zur »The First Caesar« Seite von Dr. Ellis Knox an der Boise State University verweist, dessen URL lautet: `http://history.idbsu.edu/westciv/julio-cl/index.html`

Geben Sie als Erstes den entsprechenden Text für das Link in Ihre Menü-Seite wie folgt ein:

```
<p><i>The First Caesars</i> page by Dr. Ellis Knox has more information on  
these Emperors.</p>
```

Was ist, wenn Sie die URL der Homepage »The First Caesar« (oder des Dokuments, zu dem Sie eine Verbindung herstellen wollen) nicht kennen, aber wissen, wie Sie sie erreichen, indem Sie den verschiedenen Verbindungen auf den Homepages verschiedener anderer Leute folgen? Kein Problem. Benutzen Sie Ihren Browser, um die Homepage des Dokuments zu finden, mit dem Sie eine Verbindung herstellen wollen. (Abbildung 5.6 zeigt, wie die Homepage aussehen könnte, falls sie existierte.)

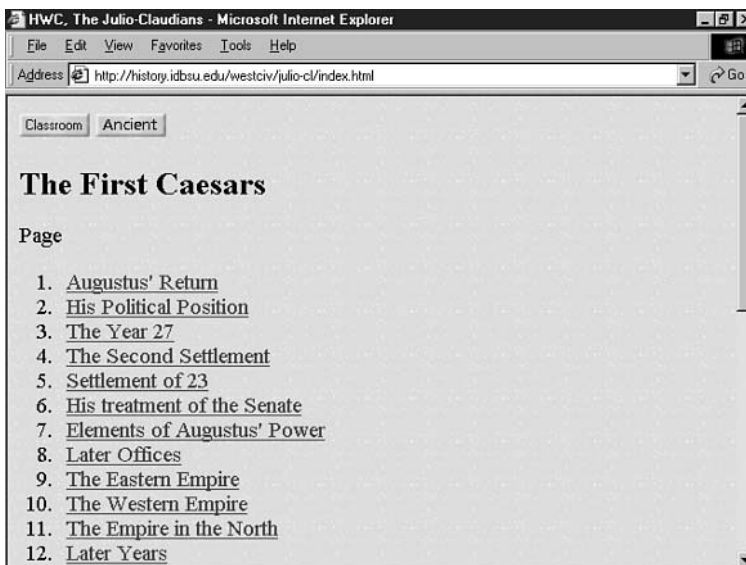


Abbildung 5.6:
Die Homepage
»The First
Caesars«



Sollte Ihr System so konfiguriert sein, dass keine Verbindung zum Netz besteht, dann sollten Sie die Verbindung nun herstellen, um das Beispiel nachvollziehen zu können.

Von den meisten Browsern wird die URL der Datei, die gerade angesehen wird, in einer kleinen Box oberhalb der Seite angezeigt. (Im Internet Explorer 4.0 oder 5.0 könnte diese Box verdeckt sein; Wählen Sie ANSICHT | SYMBOLLEISTEN | ADDRESSLEISTE, um die Box anzuzeigen,) Auf diese Art und Weise können sie einfach auf andere Seiten verweisen. Alles was Sie hierfür tun müssen, ist, Ihren Browser zu verwenden, um auf die Seite zu gelangen, zu der Sie den Link erstellen wollen, dort deren URL kopieren (copy) und diese dann in die HTML-Seite einfügen (paste). Keine Schreibarbeit!

Haben Sie nun einmal die URL (die URL der Seite steht in dem Textfeld oben auf der Seite, und Sie können sie normalerweise von dort kopieren), so können Sie jetzt die Verbindungs-Kennung in Ihrer Menü-Seite konstruieren und die benötigten URLs dort einfügen:

```
<p>"<a href="http://history.idbsu.edu/westciv/julio-cl/index.html"><i>The First  
Caesars</i></a>"  
page by Dr. Ellis Knox has more information on these Emperors.</p>
```

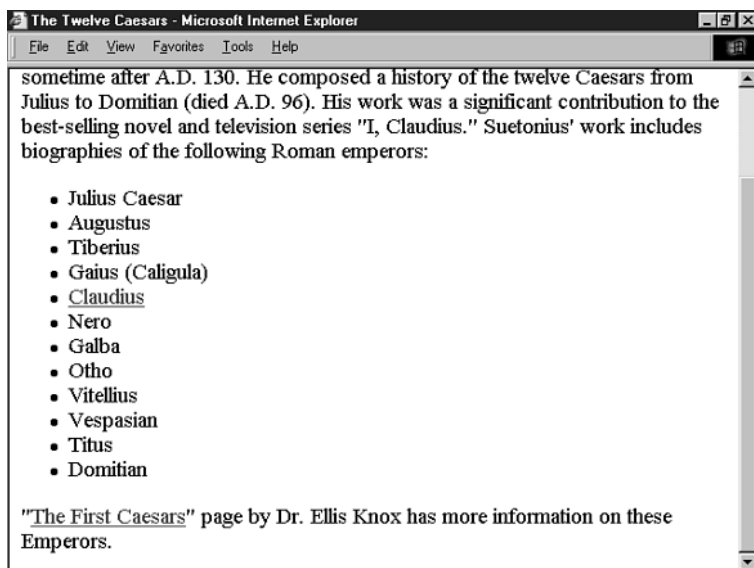


Abbildung 5.7:
Die Verknüpfung
mit »The First
Caesars«

Wenn Sie natürlich die URL der Seite, mit der Sie eine Verbindung herstellen wollen, bereits kennen, können Sie sie genauso gut direkt in den href-Teil der Verknüpfung eingeben, doch denken Sie daran, dass Ihr Browser nicht in der Lage sein wird, die

betreffende Datei zu finden, wenn Sie einen Schreibfehler machen. Die meisten URLs sind ein wenig zu komplex, um von normalen Menschen im Kopf behalten zu werden; ich bevorzuge, wann immer ich kann, das Kopieren und Einfügen (*copy & paste*), um die Möglichkeit, mich zu verschreiben, möglichst gering zu halten.

Abbildung 5.7 zeigt, wie die Datei `menu.html` mit der neuen Verknüpfung darin aussieht, wenn sie mit dem Internet Explorer angezeigt wird.

Übung 5.3: Erstellen eines Verknüpfungs-Menüs

Nachdem Sie bereits gelernt haben, wie Sie Listen erstellen, und im letzten Kapitel etwas über Verknüpfungen erfahren haben, sind Sie in der Lage, Verknüpfungs-Menüs zu erstellen. Verknüpfungs-Menüs sind Verknüpfungen auf Ihrer Webseite, die in Listenform oder in irgendeinem anderen knappen, leicht lesbaren und leicht verständlichen Format angeordnet sind. Verknüpfungs-Menüs eignen sich hervorragend für Seiten, die hierarchisch angeordnet sind, für Inhaltsverzeichnisse und für die Navigation zwischen verschiedenen Seiten. Webseiten, die nur aus Verknüpfungen bestehen, organisieren ihre Verknüpfungen oft in Menüform.



Verknüpfungs-Menüs (Link Menus) sind kurze Listen von Verknüpfungen auf Webseiten, die Ihren Lesern einen schnellen Überblick über die Möglichkeiten geben, wohin sie von der aktuellen Seite aus springen können.

Der Sinn eines Verknüpfungs-Menüs besteht darin, kurze, beschreibende Begriffe für die Verbindung zu benutzen, mit entweder gar keinem weiteren Text unmittelbar nach der Verknüpfung oder mit weiteren Beschreibungen, die der Verknüpfung folgen. Verknüpfungs-Menüs sehen in einem eingerückten oder ungeordneten Listenformat am besten aus; Sie können aber auch Glossar-Listen oder einfache Absätze verwenden. Verknüpfungs-Menüs ermöglichen Ihren Lesern, die Liste der Verbindungen schnell durchzusehen, und erleichtern ihnen, das zu finden, wonach sie suchen, etwas, das schwieriger sein könnte, wenn Sie Ihre Verbindungen einfach im Text der Seite verbergen.

In dieser Übung werden Sie eine Webseite für Buch-Erscheinungen erstellen. Diese Seite wird als Index für die Kritiken dienen, sodass das Verknüpfungs-Menü im Grunde genommen ein Menü der Buch-Namen ist.

Beginnen wir mit einer einfachen Seitenstruktur: eine Überschrift der ersten Ebene und ein wenig einfacher, erklärender Text:

```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>
```

```
<head>
<title>Really Honest Book Reviews</title>
</head>
<body>
<h1>Really Honest Book Reviews</h1>
<p>I read a lot of books about many different subjects. Though I'm not a
```

Nun fügen Sie die Liste hinzu, die Verknüpfungen enthalten wird, ohne die Verbindungs-Kennungen selbst. Es ist immer leichter, mit dem Verknüpfungs-Text zu beginnen und die richtigen Verbindungen später zu ergänzen. Für diese Liste verwenden wir die `<ul>`-Kennung, um eine eingerückte Liste einzelner Restaurants zu erzeugen. Sie könnten hier die `<menu>`-Kennung genausogut benutzen, nur die `<ol>`-Kennung wäre nicht angemessen, weil die Nummern andeuten würden, dass Sie die Bücher in irgendeiner Weise klassifizieren wollen. Hier ist also die HTML-Liste der Bücher; Abbildung 5.8 zeigt, wie die Seite im Moment im Internet Explorer aussieht.

```
<ul>
<li><i>The Rainbow Returns</i> by E. Smith</li>
<li><i>Seven Steps to Immeasurable Wealth</i> by R. U. Needy</li>
<li><i>The Food-Lovers Guide to Weight Loss</i> by L. Goode</li>
<li><i>The Silly Person es Guide to Seriousness</i> by M. Nott</li>
</ul>
</body>
</html>
```

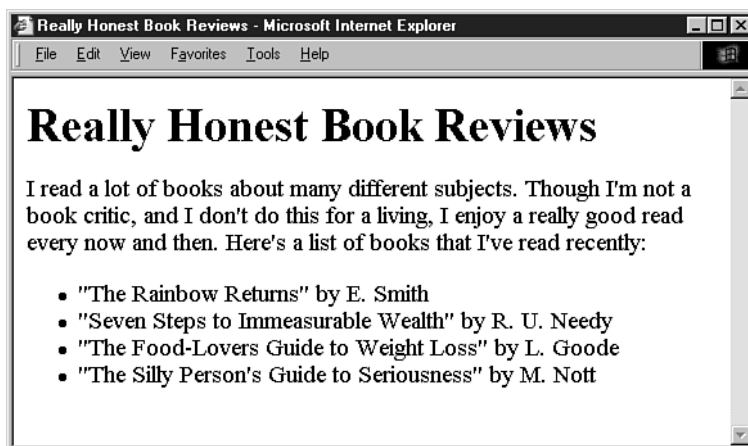


Abbildung 5.8:
Eine Liste ver-
schiedener
Bücher

Nun modifizieren Sie jeden Listeneintrag so, dass er die jeweils passende Verknüpfungs-Kennung enthält. Sie müssen die `<li>`-Kennung dort stehen lassen, da sie ja anzeigt, wo die Listeneinträge beginnen. Hier verbinden wir das Menü mit den Dateien auf dem lokalen Laufwerk im gleichen Verzeichnis, in dem auch diese Datei enthalten ist, wobei jede Datei die Besprechung eines bestimmten Buches enthält:

```
<ul>
  <li><a href="rainbow.html"><i>The Rainbow Returns</i> by E. Smith</a></li>
  <li><a href="wealth.html"><i>Seven Steps to Immeasurable Wealth</i> by R. U.
  Needy</a></li>
  <li><a href="food.html"><i>The Food-Lovers Guide to Weight Loss</i> by L.
  Goode</a></li>
  <li><a href="silly.html"><i>The Silly Person es Guide to Seriousness</i> by M.
  Nott</a></li>
</ul>
```

Das Menü der Bücher sieht gut aus, obwohl es ein wenig spärlich ist. Ihre Leser wissen weder, um welche Art von Buch es sich handelt (obgleich einige der Buch-Namen andeuten, welche Art von Thema behandelt wird), noch, ob die Besprechung gut oder schlecht ausfällt. Es wäre besser, einen kurzen Erklärungstext nach den Verknüpfungen einzufügen, um einen Hinweis darauf zu geben, was einen auf der anderen Seite der Verbindung erwartet.

```
<ul>
  <li><a href="rainbow.html"><i>The Rainbow Returns</i> by E. Smith</a>. A
  fantasy story set in biblical times. Slow at times, but interesting.</li>
  <li><a href="wealth.html"><i>Seven Steps to Immeasurable Wealth</i> by R. U.
  Needy</a>. I'm still poor, but I'm happy! And that es the whole point.</li>
  <li><a href="food.html"><i>The Food-Lovers Guide to Weight Loss</i> by L. Goode
  </a>. At last! A diet book with recipes that taste good!</li>
  <li><a href="silly.html"><i>The Silly Person es Guide to Seriousness</i> by M.
  Nott</a>. Come on ... who wants to be serious?</li>
</ul>
```

Die überarbeitete Liste sieht dann aus wie in Abbildung 5.9 gezeigt.

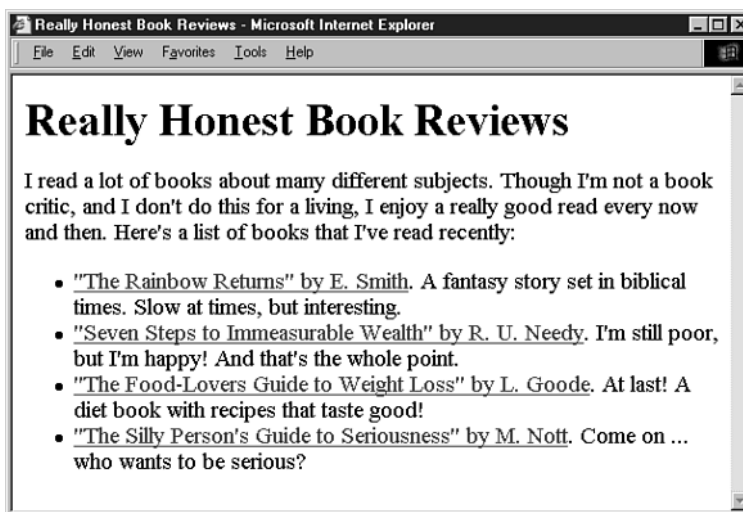


Abbildung 5.9:
Das fertige Ver-
knüpfungs-Menü

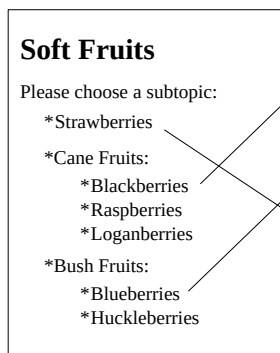
Wir werden ähnliche Menüs wie dieses noch durchs ganze Buch hindurch verwenden.

Bestimmte Stellen in einem Dokument erreichen

Die Verknüpfungen, die Sie bisher in diesem Kapitel erstellt haben, führten von einem Punkt auf einer Seite zu einer anderen Seite. Was ist nun aber, wenn Sie anstelle einer allgemeinen Verknüpfung mit einer anderen Seite eine Verbindung mit einer bestimmten Stelle innerhalb jener Seite herstellen wollen, beispielsweise zum vierten großen Abschnitt unten?

Sie können das mit HTML machen, indem Sie im zweiten Dokument einen *Anker* einfügen. Der Anker erschafft einen speziellen Bezugspunkt innerhalb der Seite, mit dem Sie sich verbinden können. Die Verknüpfung, die Sie auf der ersten Seite erschaffen, wird sowohl den Namen der Datei, zu der Sie sich verbinden, enthalten als auch den Namen des Ankers. Wenn Sie dieser Verbindung dann mit Ihrem Browser folgen, wird er das zweite Dokument öffnen (laden) und den dortigen Text bis zu der Position des Ankers durchrollen (Abbildung 5.10 zeigt ein Beispiel).

softfruits.html



berries.html



Abbildung 5.10:
Verknüpfungen und Anker



Anker (Anchors) sind spezielle Stellen innerhalb eines Dokuments, zu denen hin eine Verknüpfung geschaffen werden kann. Die Verknüpfungen können dann zu diesen speziellen Stellen und nicht nur zum Seitenanfang springen.

Sie können solche Verbindungen und Anker auch innerhalb derselben Seite so einsetzen, dass Sie, wenn Sie eine dieser Verknüpfungen auswählen, zu unterschiedlichen Stellen innerhalb derselben Seite springen können.

Anker und Verknüpfungen erzeugen

Einen Anker können Sie so ziemlich auf die gleiche Art erzeugen wie eine Verknüpfung, indem Sie nämlich die `<a>`-Kennung benutzen. Und wenn Sie sich schon gefragt haben, weshalb die Verknüpfungs-Kennung ein `<a>` enthält, anstelle eines `<l>` (für »link«), so wissen Sie es jetzt: Das »A« steht tatsächlich für »Anker« (*anchor*).

Als wir Verbindungen mit Hilfe der `<a>`-Kennung eingerichtet haben, bestanden die Tags aus zwei Teilen: dem `href`-Attribut in der einleitenden `<a>`-Kennung und jenem Text zwischen der einleitenden und der abschließenden Kennung, der als Verknüpfungs-Text diente.

Anker werden fast genauso erzeugt, mit der Ausnahme, dass Sie anstelle des `href`-Attributs in der `<a>`-Kennung das `name`-Attribut verwenden. Das `name`-Attribut besteht aus einem Kennwort (oder -wörtern), das gebraucht wird, um sich auf den Anker zu beziehen. Die Abbildung 5.11 zeigt, welche Teile der `<a>`-Kennung einen Anker anzeigen.

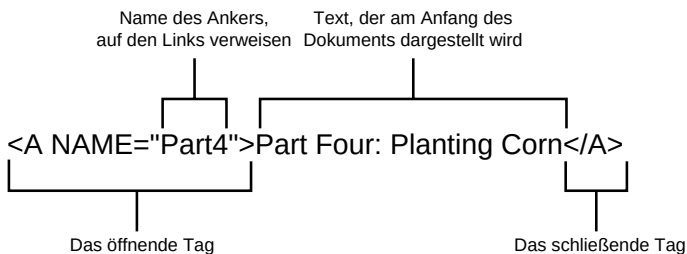


Abbildung 5.11:
Die `<a>`-Kennung und
die Anker

Anker können auch etwas Text zwischen der einleitenden und abschließenden `<a>`-Kennung enthalten, obwohl sie meistens nur zu einer Stelle mit einem einzigen Buchstaben führen. Dieser Text wird von manchen Browsern benutzt, wenn eine Verbindung zu diesem Anker ausgewählt wird; der Browser blättert durch das Dokument bis zu jenem Text, sodass dieser am oberen Rand des Bildschirms angezeigt wird, und er wird ihn auch optisch hervorheben.

Um zum Beispiel einen Anker im Seitenabschnitt, der `part4` genannt wird, zu setzen, könnten Sie einen Anker namens `part4` innerhalb der Überschrift einsetzen, und zwar folgendermaßen:

```
<h1><a name="part4">Part Four: Grapefruit from Heaven</a></h1>
```

Anders als Verknüpfungen werden Anker im fertigen Dokument nicht durch eine andere Schrift oder Farbe hervorgehoben. Ein Anker bleibt unsichtbar, bis Sie einer Verbindung folgen, die auf ihn verweist.

Um sich bei einer Verbindung auf einen Anker zu beziehen, verwenden Sie die gleiche Form einer Verknüpfung wie bei einer Verbindung mit dem Dokument als Ganzem,

mit dem Datei-Namen oder der URL des Dokuments im `href`-Attribut. Auf den Namen des Dokuments selbst folgt jedoch ein Nummern-Zeichen (`#`) und der Name des Ankers in genau der Form, wie er im `name`-Attribut des Ankers steht (einschließlich der gleichen Groß- und Kleinschreibung), so wie hier:

```
<a href="mybigdoc.html#part4">Go to Part 4</a>
```

Diese Verknüpfung veranlasst den Browser, aus dem Verzeichnis eine Ebene oberhalb des aktuellen Dokuments die Datei `mybigdoc.html` zu laden und darin dann bis zu dem Anker namens `"part4"` zu blättern, sodass der gewählte Text schließlich am oberen Rand des Bildschirms steht.

Übung 5.4: Abschnitte zwischen zwei Dokumenten verbinden

Lassen Sie uns ein Beispiel mit zwei Dokumenten durchspielen. Diese beiden Dokumente sind Teile einer Online-Referenz klassischer Musik, in der jede Webseite die Verweise für einen bestimmten Buchstaben des Alphabets enthält (`A.html`, `B.html` und so weiter). Die ganze Referenz hätte auch so organisiert werden können, dass jeder Abschnitt in einer eigenen Datei steht, doch dann wäre eine ziemlich große Menge von Dokumenten zu verwalten, und die Leser hätten eine ziemlich große Menge von Dokumenten zu laden, wenn sie durch die Referenz blättern. Es ist in diesem Fall also günstiger, die zusammengehörigen Abschnitte in buchstabierten Gruppen zusammenzufassen. (In Kapitel 11 gehe ich noch detaillierter auf die Gesichtspunkte für kurze und lange Dokumente ein.)

Die erste Seite, um die wir uns kümmern, ist jene für den Buchstaben »M«, deren erster Abschnitt in HTML so aussieht (Abbildung 5.12 zeigt die Darstellung auf dem Bildschirm):

```
<html>
<head>
<title>Classical Music: M</title>
</head>
<body>
<h1>M</h1>
<h2>Madrigals</h2>
<ul>
<li>William Byrd, <em>This Sweet and Merry Month of May</em>
<li>William Byrd, <em>Though Amaryllis Dance</em>
<li>Orlando Gibbons, <em>The Silver Swan</em>
<li>Roland de Lassus, <em>Mon Coeur se Recommande &agrave; vous</em>
<li>Claudio Monteverdi, <em>Lamento d'Arianna</em>
```

```
<li>Thomas Morley, <em>My Bonny Lass She Smileth</em>
<li>Thomas Weelkes, <em>Thule, the Period of Cosmography</em>
<li>John Wilbye, <em>Sweet Honey-Sucking Bees</em>
</ul>
<p>Secular vocal music in four, five and six parts, usually a capella.
15th-16th centuries.</p>
<p><em>See Also</em></p>
Byrd, Gibbons, Lassus, Monteverdi, Morley, Weelkes, Wilbye</p>
</body>
</html>
```

In der letzten Zeile (»Siehe auch:«) wäre es ganz praktisch, zwischen den Namen der Komponisten und ihren jeweiligen Abschnitten irgendwo anders in der Referenz Verbindungen herzustellen. Hätten Sie das Verfahren, das Sie bereits in diesem Kapitel kennen gelernt haben, gewählt, so würden Sie hier eine Verbindung zwischen Byrd und dem Dokument B.html erzeugen. Wenn Ihre Leser dann die Byrd-Verbindung auswählen, würde der Browser Sie an den Beginn der Bs führen. Die armen Leser müssten dann durch alle Bs blättern (und es gibt viele davon: Bach, Beethoven, Brahms, Bruckner), um zu Byrd zu kommen. Eine Menge Arbeit für ein System, das vorgibt, Informationen so zu verknüpfen, dass man sie schnell und leicht finden kann.

Was Sie brauchen, ist also die Möglichkeit, das Wort Byrd in M.html direkt mit dem Abschnitt Byrd in B.html zu verbinden. Hier folgt der für die Verknüpfung relevante Teil von B.html. (Ich habe all die Bs vor Byrd entfernt, um die Datei für dieses Beispiel kürzer zu machen. Nehmen Sie an, sie wären noch da.)

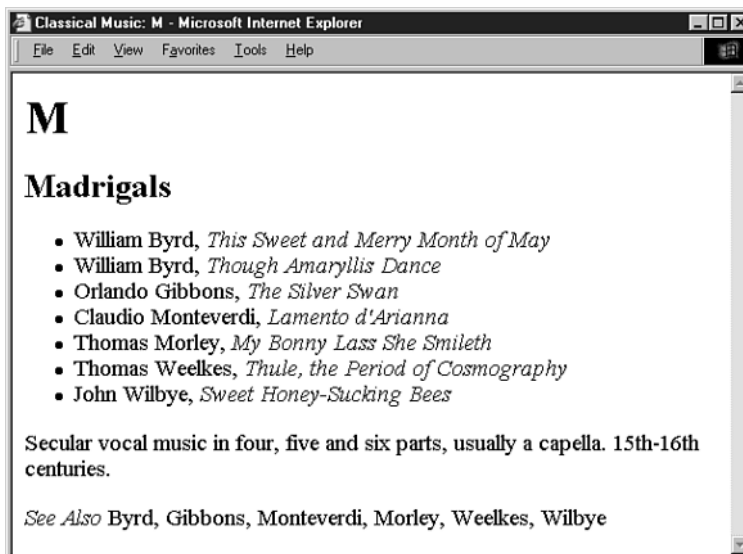


Abbildung 5.12:
Teil M des
Online-Musik-
Verzeichnisses



In diesem Beispiel werden Sie die Verwendung der `<em>`-Tags kennen lernen. Dieses Tag wird verwendet, um Text hervorzuheben. In Netscape und dem Internet Explorer wird die Hervorhebung durch kursive Stellen des Textes erreicht.

```
<html>
<head>
<title>Classical Music: B</title>
</head>
<body>
<h1>B</h1>
<!-- Ich habe alle B-Einträge vor byrd gelöscht, um das
Listing kürzer zu halten -->
<h2>Byrd, William, 1543-1623</h2>
<ul>
<li>Madrigals
<ul>
<li><em>This Sweet and Merry Month of May</em>
<li><em>Though Amaryllis Dance</em>
<li><em>Lullabye, My Sweet Little Baby</em>
</ul>
<li>Masses
<ul>
<li><em>Mass for Five Voices</em>
<li><em>Mass for Four Voices</em>
<li><em>Mass for Three Voices</em>
</ul>
<li>Motets
<ul>
<li><em>Ave verum corpus a 4</em>
</ul>
</ul>
<p><em>See Also</em>
Madrigals, Masses, Motets</p>
</body>
</html>
```

Was nun noch fehlt, ist ein Anker am Anfang des Textes zu Byrd. Sie können dann eine Verbindung zu diesem Anker von der »See also«-Datei (Siehe auch-Datei) unter M eingehen.

Wie im vorherigen Abschnitt beschrieben, brauchen Sie für einen Anker zwei Dinge: einen Anker-Namen und gegebenenfalls etwas Text, der vom Browser möglicherweise hervorgehoben wird, wenn das Dokument geladen wird. Das letztere ist einfach: Die Überschrift des Abschnitts eignet sich hervorragend dafür, da sie der Teil ist, zu dem Sie die Verknüpfung schaffen.

Für den Anker-Namen können Sie nehmen, was immer Sie mögen, aber jeder Anker in einem Dokument muss eindeutig identifizierbar sein. (Wenn Sie zwei oder mehr Anker mit dem Namen »Fritz« in demselben Dokument hätten, woher sollte der Browser dann wissen, an welche Stelle er springen soll, wenn eine Verbindung mit diesem Anker angewählt wird?) Ein kurzer und eindeutiger Anker-Name in unserem Beispiel wäre einfach »Byrd«, da es nur eine Stelle in der Datei gibt, wo »Byrd« auftauchen könnte, und die liegt hier vor.

Nun können Sie den Anker selbst in Ihre HTML-Datei schreiben. Fügen Sie die `<a>`-Kennung in die Byrd-Überschrift ein, doch seien Sie hier vorsichtig. Wäre es normaler Text innerhalb eines Absatzes, so würden Sie einfach die ganze Zeile mit `<a>` umschließen. Wenn Sie jedoch einen Anker zu einem großen Textabschnitt hinzufügen, der selbst in einem Element enthalten ist (wie eine Überschrift etwa, ein Absatz, eine Anschrift, ein Zitat usw.), schreiben Sie den Anker innerhalb dieses Elements.

Mit anderen Worten, machen Sie es so:

```
<h2> <a name="byrd"> Byrd, William, 1543 - 1623 </a> </h2>
```

Aber nicht so:

```
<a name="byrd"> <h2> Byrd, William, 1543 - 1623 </h2> </a>
```

Das zweite Beispiel könnte Ihren Browser durcheinander bringen: Handelt es sich hier um einen Anker, der genauso formatiert ist wie der Text davor und mit einem absonderlich platzierten Überschriften-Tag, oder ist es eine Überschrift, die zugleich auch ein Anker sein soll? Wenn Sie korrekten HTML-Code, mit dem Anker innerhalb der Überschrift, in Ihrer Datei erzeugen, vermeiden Sie solche Konfusionen.

Man vergisst das leicht, besonders wenn Sie so arbeiten wie ich und erst den Text schreiben und dann später die Verknüpfungen und Anker hinzufügen. Es erscheint ja auch sinnvoll, einfach alles mit einer `<a>`-Kennung zu umgeben. Denken Sie es sich so: Wenn Sie nur ein einzelnes Wort verknüpfen wollten und nicht das gesamte Element, würden Sie die `<a>`-Kennung ja auch innerhalb der `<h2>`-Kennung schreiben. Mit der ganzen Textzeile zu arbeiten, ist auch nichts anderes. Behalten Sie diese Regel im Hinterkopf und Sie werden seltener durcheinander kommen.

Sie haben nun also Ihren Anker zu der Überschrift hinzugefügt, und sein Name ist »Byrd«. Kehren Sie nun zu Ihrer Datei `M.html` zurück, in die Zeile mit dem »See Also«.

```
<p> <em> See Also </em>  
Byrd, Gibbons, Lassus, Monteverdi, Morley, Weelkes, Wilbye </p>
```

Sie erzeugen die Verknüpfung hier um das Wort `Byrd` ganz genauso, wie Sie auch jede andere Verknüpfung erzeugen würden. Doch wie lautet die URL? Wie Sie im vorherigen Abschnitt gelernt haben, sehen URLs von Ankern so aus:

SeitenName#AnkerName

Wenn Sie also eine Verknüpfung zur B.html-Seite selbst aufstellen würden, hätten Sie folgende href:

```
<a href="b.html">
```

Da Sie eine Verknüpfung zu einem Abschnitt innerhalb dieser Seite eingehen, müssen Sie den Anker-Namen hinzufügen, um auf diesen Abschnitt hinzuweisen, und das sieht so aus:

```
<a href="b.html#byrd">
```

Achten Sie auf das kleine b. Bei Anker-Namen (generell) und Verknüpfungen (betriebssystemabhängig) wird die Groß-/Kleinschreibung berücksichtigt; wenn Sie #byrd in die href schreiben, würde Ihr Browser den Anker im B-Dokument womöglich nicht finden. Achten Sie darauf, dass der Anker-Name, wie Sie ihn im name-Attribut benutzt haben, auch genauso in der Verknüpfung nach dem »#« steht.



Ein Fehler, der oft gemacht wird, ist, dass das Nummernsymbol (#) sowohl im Anker-Namen als auch in der darauf hinführenden Verknüpfung eingesetzt wird. Das Nummernsymbol wird ausschließlich dazu eingesetzt, die Seite und den Anker in der Verknüpfung auseinander zu halten; Anker-Namen sollten niemals Nummernsymbole enthalten.

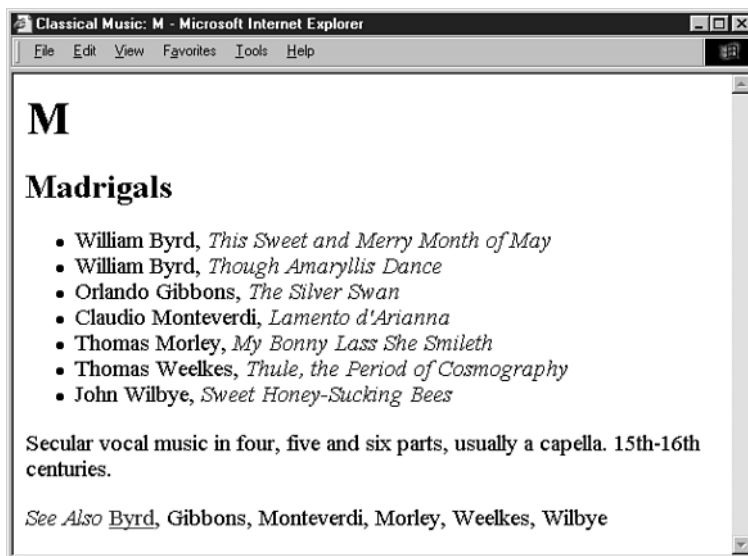


Abbildung 5.13:
Der Liederabschnitt mit der Verknüpfung

Mit dieser neuen Verknüpfung würde die Zeile »See Also« also so aussehen:

```
<p> <em> See Also </em>
<a href="b.html/#byrd"> Byrd </a>,
Gibbons, Lassus, Monteverdi, Morley, Weelkes, Wilbye </p>
```

Und Sie könnten nun auch weitermachen und Anker und Verknüpfungen in all den anderen Teilen der Online-Referenz für die übrigen Komponisten einfügen.

Mit all Ihren Verknüpfungen und Ankern sollten Sie nun alles testen. Abbildung 5.13 zeigt den Liederabschnitt mit der fertigen Verknüpfung zu Byrd.

Abbildung 5.14 zeigt, was erscheint, wenn Sie die Byrd-Verknüpfung anwählen.

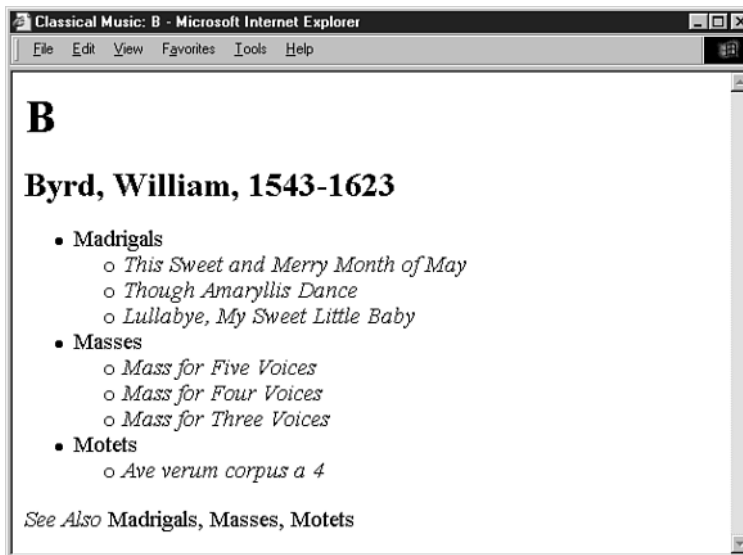


Abbildung 5.14:
Der Byrd-
Abschnitt

Verknüpfungen zu Ankern in demselben Dokument

Wie ist es nun, wenn Sie nur eine große Seite haben und zu Abschnitten innerhalb dieser Seite hin verknüpfen wollen? Sie können auch dazu Anker einsetzen, und bei größeren Seiten kann dies eine einfache Art und Weise sein, um zwischen Abschnitten hin- und herspringen zu können. Alles, was Sie tun müssen, um zu einem Abschnitt zu verknüpfen, ist, Ihren Anker wie gewohnt an jedem Abschnitt anzubringen. Wenn Sie dann zu diesem Anker hin verknüpfen, lassen Sie den Namen der Seite weg, fügen aber das Nummernsymbol und den Namen des Ankers ein. Wenn Sie also zu dem Anker mit dem Namen section5 hin verknüpfen würden, der sich auf derselben Seite wie die Verknüpfung befindet, würde es so aussehen:

Go to The Fifth Section

Wenn Sie den Seitennamen weglassen, nimmt der Browser an, dass Sie sich mit der aktuellen Seite verbinden, und rollt zum entsprechenden Abschnitt. Im Kapitel 6 werden Sie dieses Feature in Aktion sehen. In diesem Kapitel werden Sie eine komplette Webseite erstellen, die eine Inhaltstabelle am Seitenanfang enthält. Von dieser Inhaltstabelle aus, kann der Leser zu verschiedenen Sektionen der Webseite springen. Die Inhaltstabelle enthält Links zu jeder Sektions-Überschrift. Links, die sich am Ende einer jeden Sektion befinden, ermöglichen dem Leser zurück zur Inhaltstabelle oder zum Seitenanfang zu springen.

Anatomie einer URL

Bislang sind Ihnen URLs in diesem Buch zweimal begegnet – in Kapitel 1 als Teil der Einführung zum Web und im vorliegenden Kapitel, als Sie Verbindungen zu *remote*-Dokumenten hergestellt haben. Und wenn Sie das Web schon einmal durchstreift haben, so werden Ihnen URLs als etwas Selbstverständliches erscheinen. Sie könnten Ihren Streifzug ohne eine URL nicht beginnen.

Wie ich in Kapitel 1 bemerkte, sind URLs *Uniform Resource Locators* (in etwa: »gleichförmige Quellen-Auffinder«; nun ja ...). URLs sind so etwas wie die Hausnummern der Informationen im Internet. Die meiste Zeit über können Sie Ihre URLs herauskriegen, indem Sie sich einfach mit Ihrem Browser zu jener Information bewegen, zu der Sie eine Verbindung herstellen wollen, und dann die ganze Zeile mit allen Angaben aus dem Browser-Fenster in Ihre HTML-Datei kopieren. Doch es ist häufig nützlich, zu wissen, woraus eine URL überhaupt besteht und weshalb sie so lang und komplex ist. Zudem wird es, wenn Sie Ihre eigenen Informationen ins Web stellen, ganz hilfreich sein, wenn Sie ein wenig über URLs Bescheid wissen, sodass Sie den Leuten erzählen können, wo sich Ihre Webseite befindet.

In diesem Abschnitt lernen Sie die Bestandteile einer URL kennen und erfahren, wie Sie sie benutzen, um Informationen im Web zu finden. Außerdem lernen Sie die verschiedenen Arten von URLs kennen, die Sie benutzen können (*http*, *ftp*, *mailto* usw.).

Bestandteile der URLs

Die meisten URLs haben (im Wesentlichen) drei Bestandteile: das Protokoll, den Host-Namen und das Verzeichnis (vgl. Abbildung 5.15).

Das *Protokoll* bezeichnet die Art und Weise, mit der auf das Dokument zugegriffen wird, also den Protokolltyp oder das Programm, welches Ihr Browser benutzt, um an die Datei zu kommen. Wenn der Browser HTTP dafür verwendet, steht im Protokoll-

teil `http`. Wenn der Browser FTP benutzt, wäre es `ftp`. Wenn Sie Gopher benutzen, wäre es `gopher` und so weiter. Das Protokoll ist auf einen Informations-Server abgestimmt, der auf demselben System installiert sein muss, damit alles funktioniert. Sie können zum Beispiel keine FTP-URL auf einer Maschine benutzen, auf der keine FTP-Server-Software installiert ist.

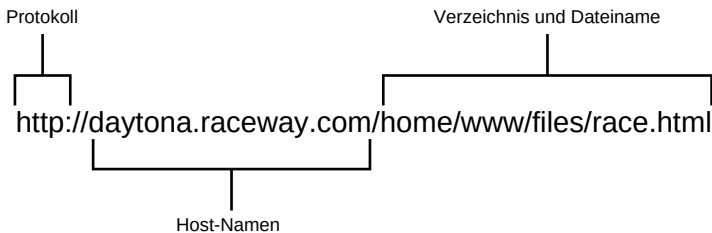


Abbildung 5.15:
URL-Bestandteile

Der *Host-Name* bezeichnet jenes System im Internet, auf dem die Informationen gespeichert sind, wie etwa `gopher.well.com`, `ftp.apple.com` oder `www.goethe.de`. Sie können denselben Host-Namen in verschiedenen URLs mit unterschiedlichen Protokollen haben:

```
http://mein.system.de
ftp://mein.system.de
gopher://mein.system.de
```

Dieselbe Maschine, drei verschiedene Informations-Server. Solange alle drei auf dem System installiert sind, gibt es keine Probleme. Der Teil der URL mit dem Host-Namen kann auch eine *Port-Nummer* enthalten. Die Port-Nummer veranlasst Ihren Browser, eine Verbindung mit dem jeweiligen Protokoll auf einem passenden Netzwerk-Port zu öffnen, der von dem Standard-Port für das entsprechende Protokoll abweicht. Lediglich dann, wenn der Server die Verarbeitung der Informationen auf einem solchen (abweichenden) Port installiert hat, benötigen Sie eine URL mit der entsprechenden Port-Nummer. (Dieses Thema wird am Tag 19 behandelt.)

Falls eine Port-Nummer nötig ist, folgt sie auf den Host-Namen, steht aber vor dem Verzeichnis, wie hier:

```
http://my-public-access-unix.com:1550/pub/file
```

Das *Verzeichnis* schließlich ist der Ort, an dem die Datei oder die sonstige Information auf dem Host gespeichert ist. Das Verzeichnis mag ein wirkliches Verzeichnis (directory) sein oder eine andere Art der Identifizierung, die das jeweilige Protokoll verwendet, um auf den Ort der Information zu verweisen. (Beispielsweise sind Gopher-Verzeichnisse keine richtigen Verzeichnisse.)

Sonder-Zeichen in URLs

Ein »Sonder-Zeichen« in einer URL ist alles, was nicht Groß- oder Kleinbuchstabe, Ziffer (0-9) oder eines der folgenden Zeichen ist: Dollar-Zeichen (»\$«), Bindestrich (»-«), Unterstrich (»_«), Punkt (».«) oder Plus (»+«). Alle anderen Zeichen müssen mit Hilfe besonderer URL-Ersatzzeichen angegeben werden, damit sie nicht versehentlich als Teil der URL interpretiert werden.

URL-Ersatzzeichen bestehen aus dem Prozent-Zeichen (%) und einem zwei Zeichen umfassenden hexadezimalen Symbol aus dem ISO-Latin-1-Zeichensatz (ein Superstandard von ASCII). So ist beispielsweise %20 das Leerzeichen, %3f ist das Fragezeichen und %2f der Schrägstrich.

Nehmen wir zum Beispiel an, Sie hätten ein Verzeichnis namens `All meine Dateien` (vermutlich auf einem Macintosh, weil Leerzeichen im Datei-Namen enthalten sind). Ihr erster Versuch mit einer URL, die diesen Namen enthält, könnte etwa so aussehen:

```
http://myhost.com/harddrive/All My Files/www/file.html
```

Wenn Sie diese URL in Gänsefüßchen in einer Verknüpfungs-Kennung verwenden, mag sie funktionieren (aber *nur*, wenn Sie sie in Gänsefüßchen schreiben). Da jedoch die Leerzeichen in einer URL als Spezial-Zeichen anzusehen sind, dürften manche Browser Probleme damit haben und den Pfad-Namen nicht richtig erkennen. Um mit allen Browsern voll kompatibel zu sein, ersetzen Sie die Leerzeichen durch %20:

```
http://myhost.com/harddrive/All My Files/www/file.html
```

In den meisten Fällen brauchen Sie keine Spezialzeichen in URLs einzusetzen, wenn Sie darauf acht geben, knappe Datei- und Verzeichnis-Namen zu verwenden, und nur alphanumerische Zeichen einsetzen. Denken Sie daran, wenn Sie Ihre eigenen Seiten schreiben.

HTML 4.0 und das <a>-Tag

HTML 4.0 enthält einige zusätzliche Attribute für das <a>-Tag, die nicht so bekannt sind:

- ▶ `tabindex` über dieses Attribut legen Sie die Position des Links innerhalb der Tabreihenfolge fest. Über die Tabreihenfolge wird die Reihenfolge bestimmt, in der die einzelnen Elemente beim Drücken der Tabulatortaste angesprungen werden.
- ▶ Unterstützung für Event-Handler, zum Beispiel solche, wie in der Netscape JavaScript-Umgebung und das Active Scripting Model von Microsoft.

Verschiedene URL-Arten

Es gibt viele Arten von URLs, die durch die Uniform-Resource-Locator-Spezifikation definiert werden. Dieser Abschnitt beschreibt einige der beliebtesten URLs und worauf zu achten ist, wenn sie benutzt werden.

HTTP

HTTP-URLs sind die am meisten verbreiteten URLs im World Wide Web. HTTP steht für »HyperText Transfer Protocol« und bezeichnet das Protokoll, das WWW-Server benutzen, um HTML-Dokumente über das Netz zu übertragen (URL: `ftp://info.cern.ch/pub/www/doc/http-spec.txt.Z`).

HTTP-URLs entsprechen der grundlegenden URL-Form:

```
http://www.foo.com/home/foo/
```

Wenn die URL mit einem Schrägstrich endet, wird der letzte Teil der URL als Verzeichnis-Name angesehen. Die Datei, die Sie bei Verwendung eines derartigen URL bekommen, ist die »Default«-Datei für das Verzeichnis, das für den HTTP-Server angegeben wurde, üblicherweise eine Datei namens `index.html`. (Wenn die Webseite, die Sie gestalten, die »oberste« Datei für all die Dateien in einem Verzeichnis ist, so wäre es gut, sie `index.html` zu nennen.)

Sie können den Datei-Namen genauso gut auch direkt in der URL angeben. In diesem Fall wird die Datei, die am Ende der URL angegeben ist, vom Browser geladen.

```
http://www.foo.com/home/foo/index.html  
http://www.foo.com/home/foo/homepag.html
```

Vermeiden Sie HTTP-URLs wie diese, wenn `foo` ein Verzeichnis ist:

```
http://www.foo.com/home/foo
```

Beachten Sie, dass `foo` ein Verzeichnis ist, sich aber kein Schrägstrich am Ende der URL befindet. Obgleich die meisten WWW-Server schlau genug sind, herauszukriegen, dass `foo` ein Verzeichnis und keine Datei ist, und Sie dann zur entsprechenden Datei »umleiten«, könnten doch manche älteren Server Probleme damit haben, diese URL auszuwerten. Achten Sie stets darauf, dass Sie ausdrücklich entweder auf ein Verzeichnis oder eine Datei verweisen und dass eine »Default«-Datei vorhanden ist, wenn Sie auf ein Verzeichnis verweisen.

Anonymous FTP

FTP-URLs werden benutzt, um auf Dateien, die sich auf FTP-Servern befinden, zu verweisen. Dies sind üblicherweise »anonymous ftp«-Server, also solche, bei denen Sie sich mit der Login-ID `anonymous` und Ihrer E-Mail-Adresse als Passwort anmelden können. FTP-URLs entsprechen ebenfalls der »Standard«-URL-Form, wie in den folgenden Beispielen gezeigt:

```
ftp://ftp.foo.com/home/foo
ftp://ftp.foo.com/home/foo/homepage.html
```

Da Sie mittels FTP sowohl eine Datei als auch einen Verzeichnis-Inhalt (directory-listing) erhalten können, ist hier die Vorschrift, wann Sie einen Schrägstrich am Ende der URL brauchen, nicht die gleiche. Die erste URL fordert das Inhaltsverzeichnis des Verzeichnisses `foo` an. Die zweite URL empfängt und interpretiert die Datei `homepage.html` aus dem `foo`-Verzeichnis.



Wenn Sie einen WWW-Browser benutzen, um sich auf FTP-Servern zu bewegen, ist dies häufig sehr viel langsamer als FTP selbst, da der Browser die Verbindung nicht aufrechterhält. Stattdessen öffnet er die Verbindung, sucht die Datei oder den Verzeichnis-Inhalt, zeigt dies an und beendet die FTP-Verbindung dann. Wenn Sie nun eine andere Datei aus diesem Inhaltsverzeichnis auswählen, wird der Browser mit der von Ihnen gewählten Datei eine neue FTP-URL konstruieren und mit dieser neuen URL eine weitere FTP-Verbindung aufbauen, die nächste Datei holen und die Verbindung wieder beenden. Daher sind FTP-URLs am besten geeignet, wenn Sie genau wissen, welche Datei Sie haben wollen, nicht aber, wenn Sie sich in einem FTP-Archiv umsehen wollen.

Beachten Sie: Obwohl Ihr Browser FTP zur Übertragung der Datei benutzt, können Sie eine HTML-Datei von dem Server bekommen, so als würde es sich um einen HTTP-Server handeln, und die Datei wird ordnungsgemäß interpretiert und dargestellt werden. Ein Web-Browser kümmert sich nicht darum, *wie* er eine Hypertext-Datei erhält. Solange er sie als HTML erkennen kann, entweder dadurch, dass der Server ihm mitteilt, dass es sich um eine HTML-Datei handelt (so wie bei HTTP – worüber Sie später noch mehr erfahren werden), oder durch die Datei-Endung (`.html` bzw. `.htm`), dann wird er die Datei als HTML-Datei auswerten und anzeigen. Erkennt er die Datei nicht als HTML-Datei, ist das aber auch kein großes Problem: Der Browser zeigt sie entweder an, wenn er weiß, um welche Art von Datei es sich handelt, und sofern er damit umgehen kann, oder er speichert sie einfach auf die Platte.

Non-anonymous FTP

Die FTP-URLs im vorherigen Abschnitt wurden alle für »anonymous ftp«-Server verwendet. Sie können aber auch eine FTP-URL für einen namentlichen Account auf einem FTP-Server angeben, wie Sie hier sehen können:

```
ftp://username:passwort@ftp.mein.system.de/home/foo/meineseite.html
```

Bei dieser Form der URL enthält der `username`-Teil Ihre Login-ID auf diesem Server, und `passwort` ist das Passwort für den Account (Zugang). Bedenken Sie bitte, dass kein Versuch unternommen wird, das Passwort im URL irgendwie zu verbergen, es wird im Klartext durch das Netz übertragen. Seien Sie also sehr vorsichtig, damit niemand Sie beobachtet, wenn Sie eine solche URL benutzen – und schreiben Sie sie in keine Verknüpfung, die jemand anderes finden kann!

Mailto

Die Mailto-URL wird verwendet, um elektronische Post (E-Mail) zu verschicken. Wenn der Browser Mailto-URLs unterstützt und eine Verknüpfung angewählt wird, die eine solche URL enthält, wird der Browser Sie nach dem Betreff und dem Text der elektronischen Nachricht fragen und die E-Mail dann, wenn Sie fertig sind, an die angegebene Adresse schicken.

Einige Browser unterstützen Mailto (noch) nicht und melden einen Fehler, wenn eine Verknüpfung angewählt wird, die eine Mailto-URL enthält.

Die Mailto-URL unterscheidet sich von der Standard-URL-Form. Er sieht ungefähr so aus:

```
mailto:internet_email_anschrift  
Zum Beispiel:  
mailto:lemay@netcom.com
```



Falls die E-Mail-Adresse ein Prozent-Zeichen (%) enthält, müssen Sie stattdessen das Ersatzzeichen `%25` benutzen. Prozent-Zeichen gelten bei URLs als Spezialzeichen.

Gopher

Gopher-URLs verwenden das Standard-URL-Format bis einschließlich des Host-Namens und benutzen dann spezielle Gopher-Protokolle, um den weiteren Weg zu einer bestimmten Datei zu finden. Bei Gopher bezeichnet das Verzeichnis nicht wie bei HTTP- und FTP-URLs einen Verzeichnis-Pfad-Namen und ist zu komplex für dieses Kapitel; falls Sie wirklich daran interessiert sind, lesen Sie die URL-Spezifikation.

Die meiste Zeit über werden Sie Gopher-URLs wahrscheinlich verwenden, um einfach auf einen Gopher-Server zu verweisen, was recht einfach ist; eine derartige URL sieht so aus:

```
Gopher://gopher.myhost.com/
```

Falls Sie wirklich auf eine bestimmte Datei auf einem Gopher-Server verweisen wollen, dürfte der beste Weg, die passende URL zu bekommen, wohl sein, sie nicht selbst per Hand zu konstruieren, sondern sich stattdessen mit Ihrem Browser zu der gewünschten Datei zu bewegen und dann die passende URL in Ihr HTML-Dokument zu kopieren.

Usenet

Usenet News-URLs können zwei Formen haben:

```
news:name_der_newsgroup
news:message_id
```

Die erste Form wird verwendet, um eine ganze »Newsgroup« (in manchen Netzen auch »Echo«, »Area«, »Board«, »Conference« genannt) zu lesen, wie etwa `comp.infosystems.www.providers` oder `alt.gothic`. Wenn Ihr Browser Usenet-News-URLs unterstützt (entweder direkt oder durch einen Newsreader), wird er Ihnen dann eine Liste der verfügbaren Nachrichten innerhalb der Newsgroup anzeigen.

Die zweite Form ermöglicht es Ihnen, eine bestimmte Nachricht zu lesen. Jede Nachricht (oder: »Artikel«) hat eine eindeutige Identifikation, die man *Message-ID* nennt und die normalerweise ungefähr so aussieht:

```
<lemayct76jq.cwg@netcom.com>
```

Um eine Message-ID in einer URL zu verwenden, entfernen Sie die spitzen Klammern und ergänzen den »news:«-Teil:

```
news:lemayCt76Jq.CwG@netcom.com
```

Bedenken Sie, dass News-Artikel nicht ewig existieren – sie »verfallen« nach einer gewissen Zeit und werden gelöscht. Daher kann eine Message-ID, die einmal gültig war, einige Zeit später ungültig werden. Wenn Sie eine dauerhafte Verknüpfung zu einem Nachrichtenartikel herstellen wollen, ist es am besten, den Artikel einfach in Ihre Web-Präsentation hineinzukopieren und ihn so wie eine normale Datei zu verknüpfen.

Beide Formen der URL unterstellen, dass Sie die Nachrichten auf einem NNTP-Server lesen, und beide können nur benutzt werden, wenn Sie irgendwo in einer Umgebungsvariablen oder einer Konfigurationsdatei einen NNTP-Server für Ihren Browser definiert haben. Insofern verwendet man News-URLs am besten, um einfach lokal bestimmte News-Artikel zu lesen, aber nicht unbedingt für Verknüpfungen in HTML-Seiten.



News-URLs werden – wie Mailto-URLs – nicht von allen Browsern unterstützt.

File-URLs

File-URLs sind dazu gedacht, auf Dateien, die sich auf einem lokalen Laufwerk befinden, zu verweisen, das heißt auf Dateien, die sich auf demselben System befinden, auf dem auch der Browser läuft. Bei lokalen Dateien haben File-URLs eine dieser beiden Formen: die erste mit einem leeren Host-Namen (sehen Sie die drei Schrägstriche anstelle der zwei?) oder mit dem Host-Namen `localhost`:

```
file:///dir1/dir2/datei
file://localhost/dir1/dir2/datei
```

Abhängig von Ihrem Browser wird mal die eine, mal die andere Form funktionieren.

File-URLs und FTP-URLs sind einander sehr ähnlich, und tatsächlich benutzt Ihr Browser, wenn der Host-Teil der URL nicht leer ist oder `localhost` lautet, FTP, um die angegebene Datei zu finden. Beide der folgenden URLs führen dazu, dass dieselbe Datei auf die gleiche Art geladen wird:

```
file://ein.system.de/pub/bla/eineseite.html
ftp://ein.system.de/pub/bla/eineseite.html
```

Die vermutlich beste Verwendung für File-URLs ist die Homepage (oder »Start-Seite«) Ihres Browser. In diesem Fall ist, da Sie sich dabei ja immer auf eine lokale Datei beziehen, eine File-URL sinnvoll.

Das Problem bei File-URLs besteht darin, dass sie sich auf lokale Dateien beziehen, wobei »lokal« bedeutet: auf demselben System wie der Browser, der die URL anspricht – *nicht* hingegen dasselbe System wie jenes, von dem das Dokument mit der File-URL darin geladen wurde! Wenn Sie File-URLs in Ihren Dokumenten für Verknüpfungen benutzen und dann jemand von sonst wo aus dem Netz Ihrem Dokument begegnet und versucht, dieser Verknüpfung zu folgen, so wird dessen Browser versuchen, die Datei auf *dessen* lokalem Laufwerk zu finden (was im Allgemeinen fehlschlägt). Da File-URLs den absoluten Pfad-Namen zur Datei benutzen, werden Sie außerdem, wenn Sie diese auf Ihrer Seite einsetzen, nicht dazu in der Lage sein, die Seite zu einem anderen Ort auf Ihrem oder einem anderen System zu verlegen.

Wenn Sie beabsichtigen, auf Dateien in demselben Dateisystem oder Verzeichnis wie das aktuelle Dokument zu verweisen, so benutzen Sie anstelle von File-URLs relative Pfad-Namen. Mit relativen Pfad-Namen für Ihre lokalen Dateien und anderen URLs für *remote*-Dateien gibt es wenige Gründe dafür, File-URLs überhaupt zu verwenden.

Zusammenfassung

In diesem Kapitel haben Sie alles Wesentliche über Links erfahren. Links verwandeln das Web von einer Ansammlung zusammenhangloser Dateien in ein riesiges vernetztes Informationssystem.

Um Verbindungen zwischen verschiedenen Dokumenten herzustellen, verwenden Sie die `<a>...</a>`-Kennung, welche Verknüpfungs-(Link-) bzw. Anker-(Anchor-)Tag genannt wird. Die Anker-Kennung hat verschiedene Attribute, mit denen sie anzeigt, welche Dateien verknüpft werden sollen (das `href`-Attribut), und mit denen Anker-Namen angezeigt werden (das `name`-Attribut).

Wenn Sie Dokumente, die alle auf Ihrer lokalen Platte gespeichert sind, miteinander verbinden, können Sie deren Pfad-Namen innerhalb des `href`-Attributs als relative oder absolute Pfad-Namen angeben. Für lokale Verknüpfungen werden relative Pfad-Namen bevorzugt, da sie es Ihnen erlauben, die lokalen Seiten einfacher zu einem anderen Verzeichnis oder auf ein anderes System zu bewegen. Wenn Sie absolute Pfad-Namen benutzen, werden Ihre Verknüpfungen unterbrochen, sobald Sie irgend etwas an dem fest angegebenen Pfad ändern.

Um die Verbindung zu einem Dokument im Web herzustellen (einem *remote*-Dokument), steht im `href`-Attribut die URL dieses Dokuments. Sie können die URL des Dokuments, zu dem Sie eine Verbindung herstellen wollen, einfach kopieren, indem Sie mit Ihrem bevorzugten Web-Browser zu jener Seite gehen und dann die URL aus dem Browser-Fenster an die gewünschte Stelle in Ihrer Verbindungs-Kennung kopieren (*Cut&Paste*).

Um Verknüpfungen zu einer spezifischen Stelle auf Ihrer Seite herzustellen, setzen Sie zuerst einen Anker an dem Punkt, zu dem Sie verknüpfen wollen, dann benutzen Sie die `<a>...</a>`-Kennung so wie bei einer normalen Verknüpfung, verwenden dann aber anstelle des `href`-Attributs das `name`-Attribut, um den Anker zu benennen. Dann können Sie direkt zu diesem Anker-Namen hin verknüpfen, indem Sie den Namen der Seite, das Nummersymbol (`#`) und den Anker-Namen einsetzen.

Schließlich werden URLs (Uniform Resource Locators) dazu benutzt, um auf Seiten, auf Dateien oder auf andere Informationen im Internet zu verweisen. Abhängig von der Art der Information können URLs aus verschiedenen Teilen bestehen, enthalten jedoch im Allgemeinen einen Protokolltyp und einen Standort (Location) oder eine Anschrift. URLs können verwendet werden, um auf viele Arten von Informationen zu verweisen, werden jedoch meistens für Web-Dokumente (`http`), FTP-Verzeichnisse oder Dateien (`ftp`), Informationen auf Gopher-Servern (`gopher`), E-Mail-Adressen (`mailto`) oder Usenet-News (`news`) benutzt.

Workshop

Glückwunsch! Im heutigen Kapitel haben Sie einiges gelernt! Nun ist es an der Zeit sich mit unserem Workshop zu beschäftigen. Hier werden viele Fragen über Links gestellt. Das Quiz wird Ihnen helfen, sich die wichtigsten Punkte dieses Kapitels noch einmal in Erinnerung zu rufen und in den heutigen Übungen werden Sie die gestern erstellten Seiten mit anderen verknüpfen.

Fragen und Antworten

F Meine Verknüpfungen werden überhaupt nicht in Blau oder Violett hervorgehoben. Sie sind nach wie vor einfacher Text.

A *Ist der Datei-Name vielleicht versehentlich in einem `href`-Attribut enthalten und nicht in einem `name`-Attribut? Haben Sie daran gedacht, die Anführungszeichen um den Datei-Namen, zu dem Sie verknüpfen, abzuschließen? Diese beiden Fehler können dazu führen, dass Verknüpfungen nicht als solche auftauchen.*

F Ich schreibe eine URL in eine Verknüpfung, und sie wird auch in meinem Browser hervorgehoben angezeigt, aber wenn ich sie anklicke, meldet der Browser: »Kann Seite nicht laden.« Wenn er die Seite nicht finden kann, weshalb hebt er dann den Text überhaupt hervor?

A *Der Browser hebt den Text innerhalb der Verknüpfungs-Kennung hervor, egal ob die Verbindung in Ordnung ist oder nicht. Tatsächlich kann man ja auch nicht mit dem Netz verbunden sein und trotzdem noch hervorgehobene Verknüpfungen auf der Seite haben, selbst wenn es keine Möglichkeit gibt, zu ihnen zu gelangen. Der einzige Weg, herauszubekommen, ob eine Verknüpfung in Ordnung ist, besteht darin, sie anzuwählen und zu versuchen, das Dokument zu laden, auf das sie zeigt.*

Weshalb nun der Browser die Seite nicht finden konnte? Überprüfen Sie, ob Sie mit dem Netzwerk verbunden sind und ob Sie die URL im Verknüpfungstag richtig geschrieben haben. Vergewissern Sie sich, dass Sie sowohl einleitende als auch abschließende Gänsefüßchen um den Datei-Namen gesetzt haben und dass Sie gerade Anführungszeichen (») benutzen. Wenn Sie einen Browser haben, der die Verknüpfungs-Zielorte in der Statusleiste anzeigt, wenn Sie die Maus über eine Verknüpfung ziehen, dann behalten Sie die Statusleiste im Auge, und vergleichen Sie, ob die URL, die angezeigt wird, die von Ihnen gewünschte ist.

Versuchen Sie schließlich, die URL direkt von Ihrem Browser aus zu öffnen. Wenn ein direktes Öffnen der Verbindung auch nicht funktioniert, dann könnte dies verschiedene Ursachen haben. Zwei häufige sind:

- ▶ Der Server ist überlastet oder nicht im Netz. Maschinen werden heruntergefahren, Netzwerk-Verbindungen ebenso. Wenn eine bestimmte URL bei Ihnen nicht funktioniert, könnte es sein, dass etwas mit der Maschine oder dem Netzwerk nicht stimmt. Vielleicht ist es auch ein sehr beliebtes System, und es versuchen zu viele Leute gleichzeitig darauf zuzugreifen. Versuchen Sie es einfach später noch mal oder außerhalb der Geschäftszeiten des Server. Wenn Sie die Leute kennen, die den Server betreiben, können Sie versuchen, ihnen eine E-Mail zu schicken oder sie anzurufen.
- ▶ Die URL selbst ist nicht in Ordnung. Manchmal werden URLs ungültig. Da eine URL eine Form eines absoluten Pfad-Namens ist, wird sie dann nicht mehr funktionieren, wenn die Datei, die sie bezeichnet, verschoben worden ist oder wenn der Name der Maschine oder eines Verzeichnisses geändert wurde. Versuchen Sie, Kontakt mit der Person oder dem System aufzunehmen, woher Sie die URL zuerst bekommen haben, und fragen Sie nach, ob es eine aktuellere Verbindung gibt.

F Kann ich jede URL in eine Verknüpfung schreiben?

- A** Aber sicher. Wenn Sie eine URL mit Ihrem Browser finden können, so können Sie sie auch in eine Verknüpfung schreiben. Beachten Sie jedoch, dass manche Browser URLs unterstützen, die von anderen nicht unterstützt werden. Beispielsweise ist Lynx ziemlich gut bei mailto-URLs (das sind URLs, die Ihnen erlauben, elektronische Post an eine E-Mail-Adresse zu schicken); wenn Sie eine mailto-URL mit Lynx anwählen, fragt Sie das Programm nach dem Betreff und dem Nachrichtentext; wenn Sie fertig sind, verschickt es die E-Mail.

Andere Browser hingegen können gegenwärtig noch keine mailto-URLs verarbeiten und behaupten, die Verknüpfung, die die mailto-URL enthält, sei ungültig. Die URL selbst mag in Ordnung sein, aber der Browser kann einfach nicht damit umgehen.

F Kann ich Bilder als Verknüpfungen einsetzen?

- A** Selbstverständlich. Sie werden das in Kapitel 7 erlernen.

F Sie haben nur zwei Attribute der <a>-Kennung beschrieben: href und name. Gibt es nicht noch andere?

A Doch. Die `<a>`-Kennung hat unterschiedliche Attribute wie `rel`, `rev`, `urn`, `methods` und `title`. Allerdings können die meisten dieser Attribute nur von Programmen verwendet werden, die automatisch Verbindungen zwischen Dokumenten erzeugen, oder von Browsern, die mit Verknüpfungen besser umgehen können als die meisten, die es heute gibt. Da 99 Prozent der Leser dieses Buchs sich nicht um derartige Verknüpfungen oder Browser kümmern, beschränke ich mich auf `href` und `name` und ignoriere die anderen Attribute.

F **Meine Verbindungen führen nicht zu meinen Ankern. Wenn ich eine Verknüpfung anwähle, wird irgendein anderes zufälliges Textstück angezeigt. Was geht da vor sich?**

A Geben Sie den Anker-Namen in der Verknüpfung nach dem Nummernzeichen (#) auf exakt dieselbe Art an, wie er im Anker selbst erscheint, mit all den Groß- und Kleinbuchstaben, Leerzeichen und anderen Zeichen? Bei Ankern ist die Groß-/Kleinschreibung wichtig, wenn also Ihr Browser den Anker-Namen mit einem exakten Vergleich nicht finden kann, versucht er vielleicht, irgendetwas anderes in dem Dokument zu nehmen, was ähnlich erscheint. Das hängt natürlich ganz vom Verhalten des Browser ab, doch wenn Ihre Verknüpfungen und Anker nicht funktionieren, so liegt das normalerweise daran, dass die Anker-Namen und die Anker selbst nicht zusammenpassen. Erinnern Sie sich auch daran, dass Anker-Namen keine Nummernzeichen enthalten – nur die darauf hinweisenden Verknüpfungen enthalten Nummernzeichen.

F **Es hört sich so an, als seien File-URLs nicht besonders nützlich. Wozu kann man sie denn überhaupt einsetzen?**

A Da fallen mir zwei Verwendungsmöglichkeiten ein. Die erste ist, dass Sie viele Anwendungen auf einem einzigen System haben (z.B. einem Unix-System), und Sie wollen diesen lokalen Anwendern (aber sonst niemandem) Zugang zu den Dateien auf diesem System verschaffen. Mit File-URLs können Sie direkt auf die Dateien auf dem lokalen System zeigen, sodass jeder auf diesem System zu ihnen gelangen kann. Leser außerhalb des Systems werden keinen direkten Zugang zur Platte haben und nicht an die Dateien gelangen können.

Ein zweiter guter Grund für die Verwendung von File-URLs ist, wenn Sie tatsächlich auf eine lokale Platte hinweisen wollen. Sie könnten zum Beispiel eine CD voller Informationen in HTML-Format erstellen und dann eine Verknüpfung von einer Webseite zu einer Datei auf der CD mit Hilfe einer File-URL herstellen. In diesem Fall lohnt es sich, eine File-URL einzusetzen, da Ihre Präsentation davon abhängt, dass Ihre Leser Zugang zu einer bestimmten Platte haben.

F Kann ich das Thema einer Mitteilung in einer Mailto-URL angeben?

- A** *Im Moment nicht. Entsprechend der gegenwärtigen Mailto-URL-Definition ist das einzige, was Sie in eine Mailto-URL einfügen können, die Adresse, an die Sie die E-Mail verschicken. Wenn Sie wirklich ein Hauptthema oder so etwas Ähnliches im Hauptteil der Nachricht benötigen, erwägen Sie, stattdessen ein Formular einzusetzen (über Formulare werden Sie mehr am Tag 12 lernen).*

Quiz

1. Welche zwei Dinge brauchen Sie, um ein Link in HTML zu erstellen?
2. Was ist ein relativer Pfad-Name? Warum ist es von Vorteil, diesen zu benutzen?
3. Was ist ein absoluter Pfad-Name?
4. Was ist ein Anker und für was wird er verwendet?
5. Welche URL gibt es, außer HTTP (»Webseite«)?

Antworten

1. Um ein Link in HTML erstellen zu können, benötigen Sie den Namen der URL der Datei oder der Seite, auf die Sie verweisen wollen. Außerdem benötigen Sie den Text, den die Leser dann lesen können, wenn Sie dem Link folgen.
2. Ein relativer Pfad-Name verweist auf eine Datei, die auf den Standort verweist, der in der Relation zur momentanen Datei steht. Relative Pfad-Namen sind übertragbar, was bedeutet, dass, wenn Sie Ihre Dateien auf der Festplatte verschieben oder einem Verzeichnis einen neuen Namen geben, die Links nur wenig oder gar keiner Veränderung bedürfen.
3. Ein absoluter Pfad-Name verweist auf eine Seite und startet dabei am Anfang einer Verzeichnis-Hierarchie. Von dort wird sich in der Verzeichnis-Hierarchie herunter gearbeitet, bis die Datei erreicht wird.
4. Ein Anker markiert eine Stelle, auf die Sie innerhalb eines Web-Dokuments verweisen können. Ein Link auf derselben oder einer anderen Seite, kann dann zu dieser spezifischen Stelle, anstatt zum Seitenanfang, springen.
5. Andere URL-Arten sind FTP-URLs (die sich auf Dateien in FTP beziehen), File-URLs (die sich auf eine Datei einer lokalen Festplatte bezieht), Mailto-URL (die für das Versenden von E-Mails verwendet wird), Gopher-URL (die sich auf Dateien eines Gopher-Servers bezieht) und Usenet-URL (die sich auf Diskussionsforen oder auf spezifische Artikel in einem Diskussionsforum bezieht).

Übungen

1. Erinnern Sie sich noch an die Themen-Liste, die Sie im letzten Kapitel erstellt haben? Erstellen Sie nun ein Link, der auf die zweite Übung des letzten Kapitels verweist (die Seite, die eine der Themen genauer behandelte).
2. Öffnen Sie nun die Seite aus der 2. Übung des gestrigen Kapitels, und erstellen Sie ein Link, der zurück zur ersten Seite verweist. Suchen Sie ebenfalls nach Seiten im World Wide Web, die dasselbe Thema haben und erstellen Sie ebenfalls Links zu diesen Seiten. Viel Glück!



Mehr HTML

**Woche
1**

Am Tag 4 und 5 haben Sie die Grundlagen von HTML einschließlich einiger grundlegender Seitenelemente und Verknüpfungen kennen gelernt. Mit diesem Hintergrund sind Sie nun bereit, mehr darüber zu erfahren, wozu HTML im Hinblick auf Textformatierung und -Layout in der Lage ist. Dieses Kapitel beschreibt die meisten der noch verbleibenden HTML-Kennungen, die Sie kennen müssen, um Seiten aufbauen zu können, also sowohl die Standard-HTML-2.0- bis hin zu 4.0-Kennungen als auch die HTML-Erweiterungen von verschiedenen Browsern. Heute lernen Sie

- ▶ die Gestaltung einzelner Zeichen (fett, kursiv usw.),
- ▶ das Einfügen spezieller Zeichen (Zeichen mit Akzenten, Copyright-Zeichen, Registriermarken usw.),
- ▶ das Erzeugen vorformatierten Textes (Text mit darin enthaltenen Leerzeichen und Tabulatoren),
- ▶ die Ausrichtung von Text (linksbündig, rechtsbündig, Blocksatz und zentriert),
- ▶ den Wechsel von Schriftart und -größe,
- ▶ das Erzeugen verschiedener anderer HTML-Textelemente wie Zeilenumbrüche, Trennlinien, Adressen und Zitate.

Darüber hinaus werden Sie die Unterschiede zwischen Standard-HTML und HTML-Erweiterungen kennen lernen und auch, welche Kennungen Sie am besten wann in Ihren Seiten einsetzen. Am Ende dieses Kapitels werden Sie eine komplette Webseite schreiben, die viele der in diesem Kapitel vorgestellten Tags enthält und auch auf die vorherigen vier Kapitel zurückgreift.

Dieses Kapitel ist ziemlich lang und behandelt verschiedene Tags und Optionen, und es mag ein wenig überwältigend wirken. Doch kümmern Sie sich nicht darum, alles gleich behalten zu wollen; verschaffen Sie sich einfach einen Eindruck davon, welche Arten der Formatierung Sie mit HTML schaffen können, die spezifischen Tags dafür können Sie sich dann später zusammensuchen. Im nächsten Kapitel werden die Dinge wieder ein wenig ruhiger, und wir werden uns die Werkzeuge und Programme anschauen, mit deren Hilfe Sie HTML schreiben können, sodass Sie nicht alles im Kopf behalten müssen, während Sie noch dabei sind, zu lernen, wie man Seiten zusammenstellt.

Zeichen-Gestaltung

Wenn Sie HTML-Kennungen für Absätze, Überschriften und Listen verwenden, so beeinflussen diese Kennungen den Text als Ganzes, sei es durch Wechseln des Schrifttyps, Veränderung der Abstände über und unter der Zeile oder durch Hinzufügen zusätzlicher Zeichen (im Falle der Listen). So genannte *Character Styles* (Zeichen-Stile)

sind Kennungen, die Wörter oder Zeichen innerhalb anderer HTML-Einheiten beeinflussen und die Erscheinungsweise des Textes so verändern, dass er sich irgendwie vom restlichen Text abhebt – beispielsweise indem er **fett** oder unterstrichen dargestellt wird.

Um die Erscheinungsweise einer Folge von Zeichen innerhalb eines Textes zu verändern, können Sie eine von zwei Kennungsarten benutzen: entweder *logische* Stil-Tags oder *physische* Stil-Tags.

Logische Styles

Logische Stil-Tags bestimmen, wie der jeweils hervorgehobene Text verwendet, nicht aber wie er dargestellt werden soll. Das ist vergleichbar mit den gewöhnlichen Element-Tags für Absätze oder Überschriften; auch sie geben nicht vor, wie der Text zu formatieren, sondern wie er in dem Dokument zu benutzen ist. Logische Stil-Tags bezeichnen einen Text als Definition oder Code oder als ein hervorgehobenes Wort.



Logische Style-Tags bestimmen die Art, wie der Text verwendet wird (Hervorhebung, Zitat, Definition).

Wenn Sie logische Stil-Tags verwenden, ist es die Sache des Browser, zu entscheiden, auf welche Weise der Text innerhalb der Kennung dargestellt wird, sei es in fetter Schrift, kursiv oder durch eine andere Veränderung der Erscheinungsweise. Sie können nicht garantieren, dass mit Hilfe dieser Tags hervorgehobener Text immer fett erscheinen oder immer kursiv dargestellt wird (und Sie sollten sich daher auch nicht darauf verlassen).



HTML 4.0 wird das Modell der physikalischen und logischen Stile von HTML durch die Unterstützung von Stilvorlagen erweitern. Mit Stilvorlagen werden Autoren in der Lage sein, das Erscheinungsbild (darunter die Schriftfamilie, den Schriftstil und die Schriftgröße) von einzelnen Elementen oder auch ganzen Elementklassen (wie zum Beispiel allen ungeordneten Listen) in einem Dokument genauer zu definieren. Stilvorlagen werden in den Kapiteln 10 und 18 behandelt.

Jede Zeichenstil-Kennung wird sowohl eingeleitet als auch abgeschlossen und beeinflusst den Text zwischen den beiden Kennungen. Standard-HTML stellt acht logische Stil-Kennungen zur Verfügung:

<code></code> :	gibt an, dass die Zeichen in irgendeiner Weise hervorgehoben (<i>emphasized</i>) werden sollen, das heißt, sie werden anders als der restliche Text formatiert. Bei grafischen Browsern wird <code></code> typischerweise kursiv. Zum Beispiel: <code><p>We'd all get along much better if you'd stop being so silly.</code>
<code></code> :	Die Zeichen sollen stärker hervorgehoben werden als bei <code></code> . Als <code></code> ausgezeichnete Text wird anders als <code></code> -Text dargestellt, beispielsweise in Fettschrift. Zum Beispiel: <code><p>You must open the can before drinking</p></code>
<code><code></code> :	Ein Code-Beispiel (Schrifttyp mit fester Zeichenbreite wie etwa »Courier« auf grafischen Bildschirmen): <code><p><code>#include trans.h</code></p></code>
<code><samp></code> :	Beispiel-Text, ähnlich wie <code><code></code> : <code><p>The URL for that page is <samp>http://www.cern.ch/</samp></p></code>
<code><kbd></code> :	Text, der über die Tastatur einzugeben ist: <code><p>Type the following command: <kbd>find . -name "prune" -print</kbd></p></code>
<code><var></code> :	Der Name einer Variablen oder von etwas, das mit einem aktuellen Wert ersetzt werden soll. Oft kursiv oder unterstrichen dargestellt, zum Beispiel: <code><p><code>chown </code><var>your_name the_file</var></p></code>
<code><dfn></code> :	<code><p>Styles that are named on how they are used are called <dfn>character styles</dfn></p></code>
<code><cite></code> :	<code><p>Eggplant has been known to cause extreme nausea in many unsuspecting people<cite> (Lemay, 1994)</cite></p></code>



Alle hier aufgeführten Kennungen sind Teil der offiziellen HTML-2.0-Spezifikationen, mit Ausnahme von `<dfn>`. Letzteres gehört zu HTML 3.2, wird aber im Allgemeinen von den meisten Browsern unterstützt, sodass ich es hier mit eingeschlossen habe.

HTML 4.0 enthält zwei zusätzliche Style-Tags, die für Audio-Player äußerst nützlich sind. Ein Grafik-Browser, wie etwa Netscape und Internet Explorer, wird diese Tags gleich gut aufzeigen. Wenn ein Audio-Browser den, sich innerhalb dieser Tags befindenden Inhalt liest, wird jeder Buchstabe einzeln ausgesprochen. Das Wort »Fox«, zum Beispiel, wird wie folgt ausgesprochen: F-O-X.

Diese Tags verwenden ebenfalls einleitende und abschließende Tags. Die zwei neuen Style-Tags sind:

<code><abbr></code>	Dieses Tag zeigt die Abkürzung eines Wortes wie folgt an: <code><p>Use the standard two-letter state abbreviation (such as <code><abbr>CA</abbr></code> for California)<code></p></code></code>
<code><acronym></code>	Ist dem <code><abbr></code> -Tag ähnlich und designiert ein Wort, das aus Anfangsbuchstaben verschiedener Wörter geformt wird. Beispiel: <code><p>Jonathan learned his great problem-handling skills from <code><acronym>STEPS</acronym></code> (Simply Tackle Each Problem Seriously)<code></p></code></code>

Sie haben sich alles gemerkt? Gut! Es gibt ein Quiz am Ende des Kapitels. Der folgende kurze Code liefert für jedes Stil-Tag ein kurzes Beispiel, und die Abbildung 6.1 illustriert, wie diese Tags vom Internet Explorer dargestellt werden.



```
<p>The anteater is the <em>strangest</em> looking animal, isn't it?</p>
<p>Take a <strong>left turn</strong> at <strong>Dee es Hop Stop
</strong></p>
<p><code>#include "trans.h"</code></p>
<p>The URL for that page is <samp>http://www.cern.ch/</samp></p>
<p>Type the following command: <kbd>find . -name "prune" -print</kbd></p>
<p><code>chown </code><var>your_name the_file</var></p>
<p>Styles that are named after how they are used are called <dfn>logical
styles</dfn></p>
<p>Eggplant has been known to cause nausea in some
people<cite> (Lemay, 1994)</cite></p>
<p>Use the standard two-letter state abbreviation (such as
<abbr>CA</abbr> for California)</p>
<p>Jonathan learned his great problem-handling skills from
<acronym>STEPS</acronym> (Simply Tackle Each Problem Seriously)</p>
```

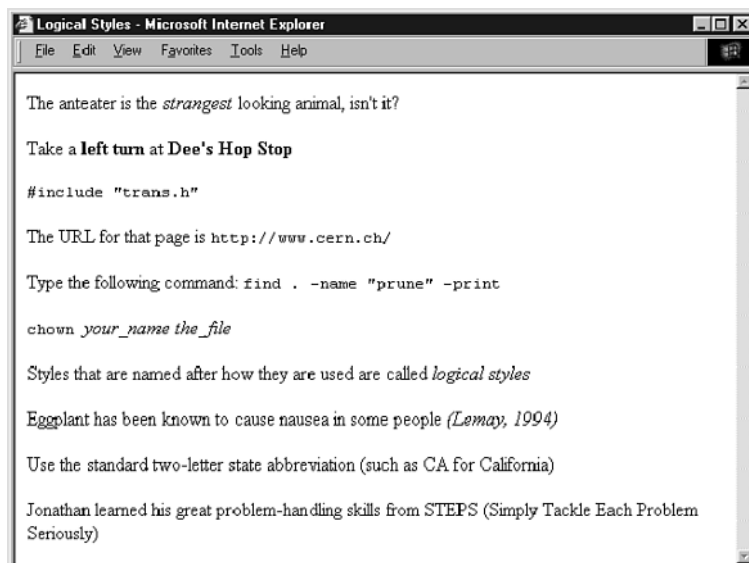


Abbildung 6.1:
Die Darstellung
im Internet Explorer

Physische Stile

Zusätzlich zu den Stil-Tags aus dem vorherigen Abschnitt gibt es noch einige Tags, welche die aktuelle Darstellung des Textes verändern – ihn fett erscheinen lassen oder ihn kursiv oder mit fester Schriftweite (*monospaced*) anzeigen.



Physische Stil-Tags beschreiben genau, wie der Text formatiert werden soll (z.B. fett gedruckt oder unterstrichen).

Wie bei den Kennungen der Zeichenstile hat hier jedes Tag sowohl eine einleitende als auch eine abschließende Kennung. Standard-HTML 2.0 kennt drei physische Stil-Tags:

	für Fettschrift (boldface),
<i>	für kursiv (italic),
<tt>	für eine Schrift mit fester Schriftweite (monospaced).

HTML 3.2 besitzt auch noch weitere Stil-Tags, unter anderem:

<code><u></code>	unterstrichen (Underline)
<code><s></code>	durchgestrichen (Strike through)
<code><sup></code>	Hochgestellt
<code><big></code>	größer als der umgebende Text
<code><sub></code>	Tiefgestellt

<code><u></code>	unterstrichen (Underline)
<code><s></code>	durchgestrichen (Strike through)
<code><sup></code>	Hochgestellt
<code><big></code>	größer als der umgebende Text
<code><sub></code>	Tiefgestellt

Wenn Sie sich dafür entscheiden, die physischen Stil-Tags – speziell die neueren HTML-3.2-Tags – zu benutzen, seien Sie davor gewarnt, dass ein Browser, der mit einem dieser Stile nicht umgehen kann, ihn möglicherweise durch einen anderen ersetzt oder diesen gänzlich ignoriert. Obwohl die meisten der neuesten Browser wie Netscape Communicator und Internet Explorer 4 mit diesen Tags glücklich sind, gibt es genügend Benutzer, die ältere Versionen dieser Browser verwenden, die diese HTML-3.2-Tags in unterschiedlichem Maß unterstützen. Dies sehen wir bei textorientierten Browsern wie zum Beispiel Lynx, in denen einige dieser Tags nicht visuell umgesetzt werden können. In diesen Fällen wird es nötig, die Idee auf Umwegen zu realisieren.

Sie können Zeichen-Tags miteinander kombinieren – beispielsweise sowohl fett als auch kursiv für einige Zeichen benutzen – wie hier:

```
<b><i>Text that is both bold and italic</i></b>
```

Gleichwohl ist das Ergebnis auf dem Bildschirm wie bei allen HTML-Tags vom Browser abhängig. Ihr Text wird eventuell nicht kursiv und fett dargestellt, jedoch erscheint er vielleicht zumindest in einer dieser Darstellungsweisen.

Das folgende Beispiel zeigt einige der physischen Stil-Tags und wie sie in Internet Explorer (Abbildung 6.2) erscheint:



```
<p>In Dante es <i>Inferno</i>, malaboge was the eighth circle of hell,
and held the malicious and fraudulent.</p>
<p>All entries must be received by <b>September 26, 1999</b>.</p>
<p>Type <tt>lpr -Pbirc myfile.txt</tt> to print that file.</p>
<p>Sign your name in the spot marked <u>Sign Here</u>:</p>
<p>People who wear orange shirts and plaid pants <s>have no taste</s>
are fashion-challenged.</p>
<p>RCP floor mats give you <big>big</big> savings over the
competition!</p>
<p>Then, from the corner of the room, he heard a <small>tiny voice
</small>.</p>
<p>In heavy trading today. Consolidated Orange Trucking
rose <sup>1</sup></sup>/<sub>4</sub> points on
points on volume of 1,457,900 shares.</p>
```

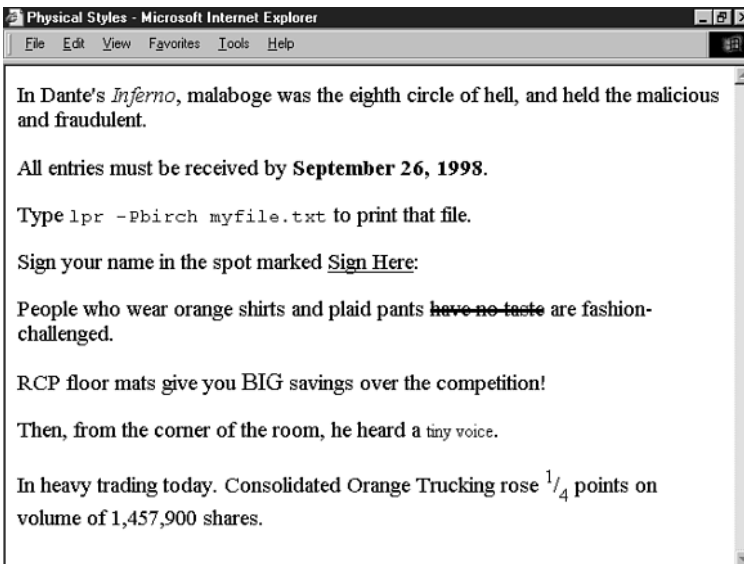


Abbildung 6.2:
Die Darstellung
im Internet
Explorer

Vorformatierter Text

Die meiste Zeit wird der Text in einer HTML-Datei formatiert, ausgehend von den HTML-Kennungen, die verwendet wurden, um den Text auszuzeichnen. Wie ich schon in Kapitel 3 bemerkt habe, werden alle zusätzlichen Leerräume (*white spaces*: Leerzeichen, Tabulatoren, Zeilenumbrüche), die Sie in den Text schreiben, vom Browser ignoriert.

Die einzige Ausnahme von dieser Regel ist die `<pre>`-Kennung für vorformatierten (*preformatted*) Text. Jeder Leerraum (*white space*), den Sie im Text schreiben, der von `<pre>` und `</pre>` eingeschlossen ist, bleibt in der endgültigen Ausgabe enthalten, sodass Sie den Text auf eben diese Weise formatieren können, wie Sie ihn aussehen lassen möchten, und er wird auch in genau dieser Formatierung dargestellt.

Der Kniff dabei ist, dass vorformatierter Text auch mit fester Schriftweite (*monospaced*), wie es auch bei Courier der Fall ist, dargestellt wird. Vorformatierter Text eignet sich ausgezeichnet für Code-Beispiele, in denen Sie die Zeilen passend einrücken und formatieren wollen. Da es Ihnen auch möglich ist, den Text durch eingefügte Leerzeichen auszurichten, können Sie die `<pre>`-Kennung auch für Tabellen verwenden, obwohl der Umstand, dass diese Tabellen mit einer festen Schrift (*monospaced*) dargestellt werden, sie nicht eben ideal erscheinen lassen (Sie werden in Kapitel 11, die Konstruktion richtiger Tabellen erlernen). Hier ist ein Beispiel für eine mit `<pre>` erzeugte Tabelle. (Abbildung 6.3 zeigt, wie sie in Internet Explorer aussieht.)



`<pre>`

	Diameter (miles) from Sun	Distance (millions of miles)	Time to Orbit	Time to Rotate
Mercury	3100	36	88 days	59 days
Venus	7700	67	225 days	244 days
Earth	7920	93	365 days	24 hrs
Mars	4200	141	687 days	24 hrs 24 mins
Jupiter	88640	483	11.9 years	9 hrs 50 mins
Saturn	74500	886	29.5 years	10 hrs 39 mins
Uranus	32000	1782	84 years	23 hrs
Neptune	31000	2793	165 days	15 hrs 48 mins
Pluto	1500	3670	248 years	6 days 7 hrs

`</pre>`



Pre-formatted Paragraphs - Microsoft Internet Explorer

	Diameter (miles)	Distance from Sun (millions of miles)	Time to Orbit	Time to Rotate
Mercury	3100	36	88 days	59 days
Venus	7700	67	225 days	244 days
Earth	7920	93	365 days	24 hrs
Mars	4200	141	687 days	24 hrs 24 mins
Jupiter	86640	483	11.9 years	9 hrs 50 mins
Saturn	74500	886	29.5 years	10 hrs 39 mins
Uranus	32000	1782	84 years	23 hrs
Neptune	31000	2793	165 days	15 hrs 48 mins
Pluto	1500	3670	248 years	6 days 7 hrs

Abbildung 6.3:
Eine Tabelle, die
mit `<pre>` im Inter-
net Explorer darge-
stellt wurde.

Wenn Sie Text für die `<pre>`-Kennung schreiben, können Sie Verknüpfungen und verschiedene Schriftarten verwenden, aber keine Element-Kennungen wie für Überschriften oder Absätze. Umbrechen Sie Ihre Zeilen mit einem »Return«, und versuchen Sie, Ihre Zeilen 60 Zeichen oder kürzer zu halten. (Manche Browser stellen nur eine begrenzte Anzahl von Zeichen pro Zeile dar, und da Browser vorformatierten Text nicht neu formatieren, um ihn in diesen Raum einzupassen, sollten Sie darauf achten, innerhalb dieser Grenzen zu bleiben, damit Ihre Leser die Ansicht nicht hinausverschieben müssen.)

Passen Sie auf, wenn Sie Tabulator-Zeichen innerhalb des vorformatierten Textes benutzen. Die verschiedenen Browser rücken für jedes Tabulatorzeichen eine unterschiedliche Anzahl von Leerzeichen weiter. Ein Browser rückt vielleicht vier Stellen für eine Tabulatormarke vor, während ein anderer acht weiterrückt. Wenn Ihr vorformatierter Text sich an einigen Stellen auf Tabulatormarken verlässt, sollten Sie diese vielleicht besser durch Leerzeichen ersetzen.

Das `<pre>`-Tag eignet sich auch hervorragend dafür, Dateien, die zuvor in einer reinen Textform vorlagen, wie etwa E-Mail oder »Usenet News«, schnell und einfach nach HTML zu konvertieren. Umgeben Sie einfach den gesamten Inhalt der Nachricht mit der `<pre>`-Kennung, so erhalten Sie »fertigen« HTML-Code:

```
<pre>
To: lemay@lne.com
From: jokes@lne.com
Subject: Tales of the Move From Hell, pt. 1
Date: Fri, 26 Aug 1994 14:13:38 +0800
```

I spent the day on the phone today with the entire household services division of northern California, turning off services, turning on services, transferring services and other such fun things you have to do when you move.

It used to be you just called these people and got put on hold for and interminable amount of time, maybe with some nice music, an then you got a customer representative who was surly and hard of hearing, but with some work you could actually get your phone turned off.

```
</pre>
```

Das folgende Beispiel zeigt eine einfache ASCII-Kuh und deren Darstellung in Internet Explorer (Abbildung 6.4):



```
<pre>
      ( )
"Moo!" (oo)
  \ /-----\
  ||         | \
  ||         | *
  ||---W---||
  ||         ||
```

```
</pre>
```

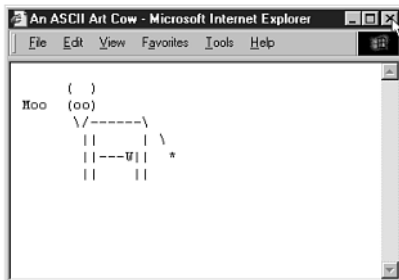


Abbildung 6.4:
Darstellung im Internet Explorer

Horizontale Linien

Die `<hr />`-Kennung, ohne abschließende Kennung und ohne zugehörigen Text, erzeugt eine horizontale Linie auf der Seite. Solche Linien eignen sich ausgezeichnet für die visuelle Abgrenzung von Abschnitten auf der Webseite – insbesondere vor Überschriften oder um den Haupttext von einer Liste abzugrenzen. Abbildung 5.8 illustriert den effektiven Nutzen von solchen Linien.



Das `<hr>`-Tag verfügt über kein schließendes Tag. Um es in XHTML verwenden zu können, und um die Kompatibilität mit HTML-Browsern zu garantieren, fügen Sie eine Leerstelle und einen Schrägstrich (/) dem Ende des Tags hinzu. Achten Sie auch darauf, dass Sie die Tags in XHTML mit Kleinbuchstaben schreiben:

```
<hr />
```

Soll die horizontale Zeile Attribute enthalten, die auf diese Tags hinweisen, erscheint der Schrägstrich (/) weiterhin am Ende des Tag. Sehen Sie hierzu folgendes Beispiel:

```
<hr size="2" />
<hr width="75%" />
<hr align="center" size="4" width="200" />
```

Das folgende Beispiel zeigt eine horizontale Linie und eine Liste.



```
<hr />
<h2>To Do on Friday</h2>
<ul>
<li>Do laundry</li>
<li>Send FedEx with pictures</li>
<li>Have lunch with Mollie</li>
<li>Read Email</li>
<li>Set up Ethernet</li>
</ul>
<hr />
```

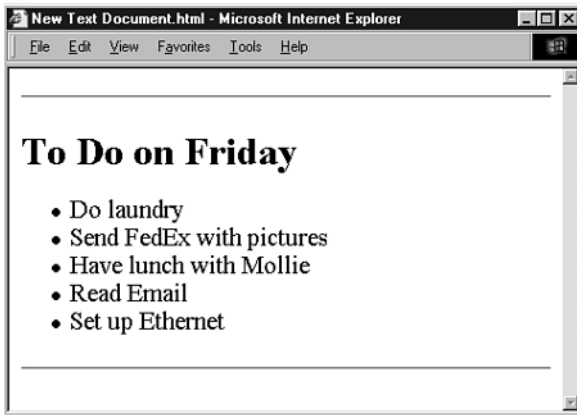


Abbildung 6.5:
Darstellung im Internet Explorer

Attribute der <hr />-Kennung

In HTML 2.0 wird die <hr />-Kennung genauso, wie Sie sie sehen, verwendet, also ohne abschließende Kennung oder Attribute. Netscape führte einige Attribute für das <hr />-Tag ein, die von einigen anderen Browsern unterstützt werden. Sämtliche Attribute wurden zugunsten der Style Sheets in der HTML-Spezifikation verworfen.



Obwohl Präsentations-Attribute, wie size, width und align immer noch in HTML 4.0 unterstützt werden, ist das Arbeiten mit Style Sheets die Methode, die für die Kontrolle des Designs empfohlen wird.

Das size-Attribut zeigt die Dicke der horizontalen Trennlinie in Pixel an. 2 ist hierbei die Voreinstellung und gleichzeitig die geringste Dicke, die eine Trennlinie haben kann. Abbildung 6.6 zeigt einige Beispiele wie die Trennliniendicke mit folgendem Code erzeugt wird.



```
<h2>2 Pixels</h2>
<hr size="2" />
```

```
<h2>4 Pixels</h2>
<hr size="4" />
<h2>8 Pixels</h2>
<hr size="8" />
<h2>16 Pixels</h2>
<hr size="16" />
```

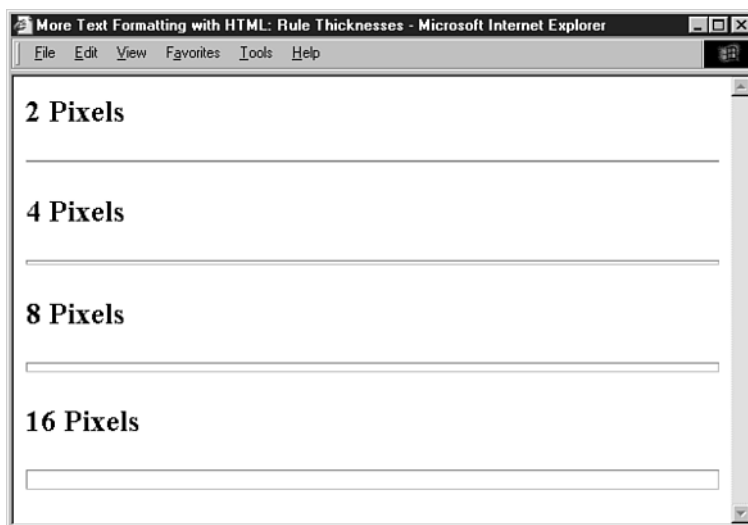


Abbildung 6.6:
Beispiele der
Trennliniendicke

Das `width`-Attribut bestimmt die horizontale Breite der Trennlinie. Sie können entweder die genaue Breite in Pixel angeben oder den Wert als Prozentsatz der Bildschirmbreite festlegen (zum Beispiel 30 oder 50 Prozent), sodass sich die dargestellte absolute Breite ändert, wenn Sie die Fensterbreite verändern. Abbildung 6.7 zeigt einige Beispiele von Trennlinien-Breiten.



```
<h2>100%</h2>
<hr />
<h2>75%</h2>
<hr width="75%" />
<h2>50%</h2>
```



```
<hr width="50%" />  
<h2>25%</h2>  
<hr width="25%" />  
<h2>10%</h2>  
<hr width="10%" />
```

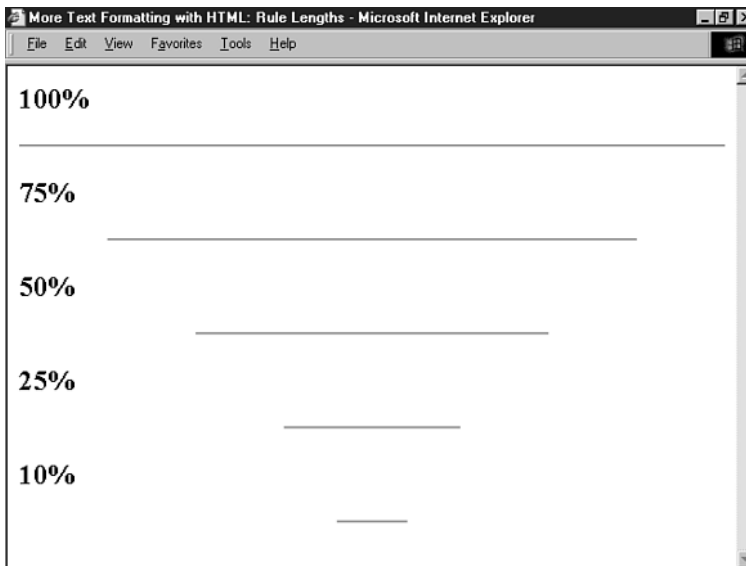


Abbildung 6.7:
Beispiele für ver-
schiedene Trenn-
linienbreiten

Wenn Sie eine Breite festlegen, die kleiner ist als die Breite des Bildschirms, dann können Sie die Ausrichtung der Trennlinie auch durch das `align`-Attribut festlegen, und zwar haben Sie die Wahl zwischen linksbündig (`align="left"`), rechtsbündig (`align="right"`) oder zentriert (`align="center"`). Die Voreinstellung für Trennlinien ist »zentriert«.

Ein beliebter Trick von Web-Designern ist es mit verschiedenen kleineren Trennlinien Muster zu erzeugen, wie in Abbildung 6.8 gezeigt wird.



```
<hr align="center" size="4" width="200" />
<hr align="center" size="4" width="300" />
<hr align="center" size="4" width="400" />
<h1 align="center">NorthWestern Video</h1>
<hr align="center" size="4" width="400" />
<hr align="center" size="4" width="300" />
<hr align="center" size="4" width="200" />
<h2 align="center">Presents</h2>
```

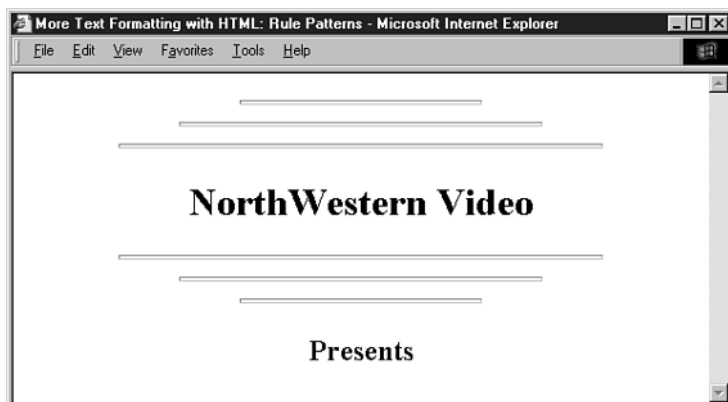


Abbildung 6.8:
Beispiel eines Musters, das durch verschiedene kleinere Trennlinien erzeugt wird.

Schließlich können Sie das `noshade`-Attribut einsetzen, welches den Browser veranlasst, die Trennlinie als einfache schwarze Linie ohne dreidimensionale Schattierung zu zeichnen, wie es in Abbildung 6.9 gezeigt wird.



In HTML 4.0 und früheren Versionen braucht man für `noshade` nicht unbedingt einen Attribut-Wert. Die Methode, die Sie für dieses Attribut verwenden sieht wie folgt aus:

```
<hr align="center" size="4" width="200" noshade />
```

In XHTML 1.0 jedoch, ist ein Wert obligatorisch. Folgendes Beispiel demonstriert, wie das `noshade`-Attribut auf das `<hr />`-Tag, entsprechend der vorgeschlagenen XHTML 1.0-Spezifikation, angewandt werden muss.



```

<hr align="center" size="4" width="200" noshade="noshade" />
<hr align="center" size="4" width="300" noshade="noshade" />
<hr align="center" size="4" width="400" noshade="noshade" />
<h1 align="center">NorthWestern Video</h1>
<hr align="center" size="4" width="400" noshade="noshade" />
<hr align="center" size="4" width="300" noshade="noshade" />
<hr align="center" size="4" width="200" noshade="noshade" />
<h2 align="center">Presents</h2>

```



Abbildung 6.9:
Trennlinien ohne
Schattierung

Zeilenumbrüche

Die `<br />`-Kennung, ohne abschließende Kennung, bricht eine Zeile an der Stelle, an der sie gesetzt wird um. Wenn ein Web-Browser auf eine `<br />`-Kennung stößt, beginnt er mit dem folgenden Text wieder am linken Rand (was immer der linke Rand für das aktuelle Element auch sein mag). Sie können die `<br />`-Kennung innerhalb anderer Tags verwenden, wie etwa in Absätzen oder Listeneinträgen; `<br />` fügt keinen zusätzlichen Platz über oder unter der neuen Zeile ein und ändert auch den Schrifttyp oder -stil des gegenwärtigen Textabschnitts nicht. Alles, was es bewirkt, ist, mit dem Text in einer neuen Zeile weiterzumachen.



Wie das `<hr>`-Tag auch, braucht das `<br>`-Tag in HTML kein abschließendes Tag.

Um dieses Tag in XHTML zu verwenden, und um sicherzugehen, dass es mit älteren Browsern kompatibel ist, fügen Sie eine Leerstelle gefolgt von einem Schrägstrich ein. Hiermit wird das Tag und seine Attribute geschlossen. Siehe hierzu folgendes Beispiel:

```
And then is heard no more: it is a tale <br />
Told by an idiot, full of sound and fury, <br />
Signifying nothing.</p>
```

Das folgende Beispiel zeigt einen einfachen Absatz, bei dem jede Zeile mit einem `<br />` endet. Die Abbildung 6.10 zeigt, wie das im Internet Explorer aussieht.



```
<p>Tomorrow, and tomorrow, and tomorrow<br />
Creeps in this petty pace from day to day<br />
To the last syllable of recorded time;<br />
And all our yesterdays have lighted fools<br />
The way to dusty death. Out, out, brief candle!<br />
Life es but a walking shadow, a poor player,<br />
That struts and frets his hour upon the stage<br />
And then is heard no more. It is a tale <br />
Told by an idiot, full of sound and fury, <br />
Signifying nothing.</p>
```

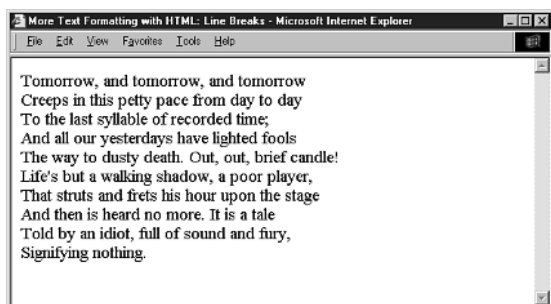


Abbildung 6.10:
Die Darstellung im Internet Explorer



Das `clear`-Attribut ist eine HTML-3.2-Attribut der `<br />`-Kennung. Es wird bei Bildern eingesetzt, um die der Text herumfließt. Sie werden mehr über diese Erweiterung in Kapitel 7 erfahren.

Adressen

Die Adressen-Kennung `<address>` wird auf Webseiten als eine Art Unterschrift verwendet. Adress-Kennungen finden sich normalerweise am Ende jeder Webseite und werden benutzt, um anzuzeigen, wer die Seite geschrieben hat, an wen man sich für weitere Informationen wenden kann, das Datum, einen Urheberrechts-Hinweis oder andere Warnungen sowie alles, was passend erscheint. Adressen werden häufig durch eine horizontale Linie (`<hr />`) eingeleitet, und die `<br />`-Kennung kann benutzt werden, um die Zeilen zu unterteilen.

Ohne eine Adresse oder irgendeine andere Methode, Ihre Webseite zu »unterzeichnen«, wird es nahezu unmöglich, herauszufinden, wer die Seite geschrieben hat und an wen man sich für weitere Informationen wenden kann. Das Signieren jeder Ihrer Webseiten mit der `<address>`-Kennung ist eine ausgezeichnete Möglichkeit, um sicherzustellen, dass Leute, die mit Ihnen in Kontakt kommen wollen, dies auch tun können.

Im folgenden einfachen Beispiel sehen Sie eine Adresse im Internet Explorer (Abbildung 6.11).



```
<hr />
<address>
Laura Lemay lemay@lne.com <br />
A service of Laura Lemay, Incorporated <br />
last revised April 30 1997 <br />
Copyright Laura Lemay 1994 all rights reserved <br />
Void where prohibited. Keep hands and feet inside the vehicle at all times.
</address>
```



Abbildung 6.11:
Die Darstellung
im Internet
Explorer

Zitate

Die `<blockquote>`-Kennung wird für Zitate verwendet. (Im Unterschied zur `<cite>`-Kennung, die kürzere Zitate hervorhebt, wird `<blockquote>` für ausführlichere Zitate eingesetzt, die nicht innerhalb anderer Absätze verschachtelt sein sollten.) Zitate werden im Allgemeinen vom regulären Text durch Einrückung oder durch eine andere Methode abgesetzt. Der Macbeth-Monolog, den ich im Beispiel zu den Zeilenumbrüchen verwendet habe, würde als `<blockquote>` besser wirken als ein einfacher Absatz. Hier ist ein weiteres Beispiel:



```
<blockquote>
```

```
"During the whole of a dull, dark, and soundless day in the autumn of the year,
when the clouds hung oppressively low in the heavens, I had been passing alone, on
horseback, through a singularly dreary tract of country, and at length found
myself, as the shades of evening grew on, within view of the melancholy House of
Usher."--Edgar Allen Poe
```

```
</blockquote>
```

So wie auch in Absätzen, können Sie Zeilen innerhalb von `<blockquote>` durch die Zeilenumbruch-Kennung `<br />` trennen.



```
<blockquote>
Guns aren't lawful, <br />
nooses give.<br />
gas smells awful.<br />
You might as well live.<br />
--Dorothy Parker
</blockquote>
```

Im folgenden Beispiel sehen Sie dies im Internet Explorer (Abbildung 6.12).

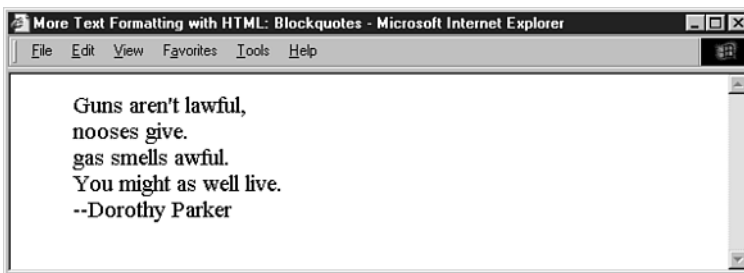


Abbildung 6.12:
Die Darstellung
im Internet
Explorer

Sonderzeichen

Wie Sie bereits in dieser Woche gelernt haben, bestehen HTML-Dateien aus reinem ASCII-Text und sollten keine Formatierungs-Zeichen oder andere ausgefallene Zeichen enthalten. Tatsächlich sind die einzigen Zeichen, die Sie in Ihren HTML-Dateien verwenden sollten, jene, die Sie direkt mit Ihrer Tastatur eingeben können. Falls Sie irgendeine andere Taste außer »Shift« (Großbuchstaben-Umschaltung) drücken müssen oder eine geheimnisvolle Tastenfolge einzugeben haben, um ein bestimmtes Zeichen zu erzeugen, dann können Sie dieses Zeichen nicht in Ihrer HTML-Datei verwenden. Das gilt auch für Zeichen, die Sie vielleicht jeden Tag einsetzen, wie em-Striche oder typographische Anführungszeichen (und wenn Ihr Textverarbeitungsprogramm letztere automatisch erzeugt, sollten Sie sie ausschalten, wenn Sie HTML-Seiten schreiben).

Moment mal, höre ich Sie da sagen. Wenn ich ein Zeichen wie beispielsweise einen dicken Punkt oder einen Akzent auf dem »a« mit einer bestimmten Tastenfolge auf meiner Tastatur eingeben kann und es in eine HTML-Datei einfüge und mein Browser es dann auch noch richtig anzeigt, wo ist da das Problem?

Das Problem besteht darin, dass der interne Zeichensatz Ihres Computers, um das Zeichen zu produzieren (und die es erlaubt, es in Ihrer HTML-Datei und auf Ihrem Bildschirm korrekt anzuzeigen), ziemlich sicher auf anderen Computersystemen nicht funktionieren wird. Irgend jemand anderes im Netz, der Ihre HTML-Datei mit den ausgefallenen Zeichen dann liest, könnte sehr gut irgendein anderes Zeichen oder gar ein Durcheinander sehen. Oder das Zeichen könnte, je nachdem, wie Ihre Datei über das Netz transportiert wird, auch einfach verloren gehen, bevor es den Rechner erreicht, auf dem es dargestellt werden soll.



Im technischen Jargon bedeutet das, dass die Zeichen in HTML-Dateien aus dem (7-Bit-)Standard-ASCII-Zeichensatz stammen müssen und keine Zeichen aus dem »erweiterten« (8-Bit-)ASCII enthalten dürfen, da jede Plattform ihre eigene Definition davon hat, welche Zeichen im oberen ASCII-Bereich enthalten sind. HTML-Browser interpretieren Codes aus dem oberen ASCII-Bereich nach dem ISO-Latin-1-Zeichensatz (ISO 8859-1), einer Obermenge von ASCII.

Was können Sie also tun? HTML bietet eine vernünftige Lösung: Sie definiert spezielle Codes, so genannte »Character Entities« (Zeichen-Entitäten), die Sie innerhalb Ihrer HTML-Dateien einsetzen können, um Zeichen darzustellen, die Sie einsetzen wollen. Von einem Browser interpretiert, werden diese »Character Entities« dann als korrekte Sonderzeichen auf der jeweiligen Plattform und mit der dort verwendeten Schrift dargestellt.

Zeichen-Entitäten für Sonderzeichen

»Character Entities« können eine dieser zwei Formen annehmen: benannte Entitäten und numerische Entitäten.

Benannte Entitäten beginnen mit dem kaufmännischen Und (Ampersand, &) und enden mit einem Semikolon (;). Dazwischen steht der Name des betreffenden Zeichens (oder meistens eher eine Kurzversion des Namens wie `grave` für ein »a« mit einem Akzent darüber oder `reg` für das *registered*-Symbol). Anders als bei den anderen HTML-Kennungen, müssen bei diesen Namen die Groß- und Kleinbuchstaben richtig angegeben werden. Benannte Entitäten sehen beispielsweise so aus:

à
"
Ä
ß

Die numerischen Entitäten beginnen ebenfalls mit dem kaufmännischen Und (&) und enden mit einem Semikolon (;), doch anstelle des Namens enthalten sie das Nummernzeichen (#) und eine Zahl. Die Zahlen korrespondieren mit der Zeichenposition im ISO-Latin-1-Zeichensatz (ISO 8859-1). Jedes Zeichen, das Sie schreiben oder für das Sie eine benannte Entität benutzen können, existiert ebenso als numerische Entität. Numerische Entitäten sehen so aus:

‚
õ

Sie können sowohl numerische als auch benannte Entitäten in Ihren HTML-Dateien benutzen, indem Sie sie an der Stelle einfügen, an der das Zeichen, das dargestellt werden soll, steht. Um also das Wort »Resumé« in Ihrer HTML-Datei zu schreiben, können Sie entweder

r´sum&#acute;
Oder
rßsum233;



Die Integration des ISO-Latin-1-Zeichensatzes in HTML erlaubt es, Akzente auf den meisten Plattformen darzustellen. Dies hat jedoch Beschränkungen. So sind zum Beispiel einige gebräuchliche Zeichen wie Blickfangpunkte (bullets), em-Striche und typographische Anführungszeichen nicht im ISO-Latin-1-Zeichensatz enthalten. Das bedeutet, dass Sie diese Zeichen in Ihren HTML-Dateien überhaupt nicht verwenden können. Außerdem stehen einige der ISO-Latin-1-Zeichen auf einigen Browsern vielleicht überhaupt nicht zur Verfügung, was davon abhängt, ob diese Zeichen überhaupt auf dieser Plattform und in der jeweiligen Schriftart existieren. HTML 4.0 bringt das ganze Thema einen guten Stück voran, indem es vorschlägt, dass Unicode als Zeichensatz für HTML-Dokumente verfügbar ist. Unicode ist ein Quasi-Standard-Zeichen-Kodierungssystem, das die Möglichkeit bietet, nahezu alle Schriftzeichen dieser Erde – darunter auch chinesische und japanische – aufzunehmen und gleichzeitig abwärtskompatibel mit dem gewohnten ASCII-Code ist. Das heißt, Dokumente können leicht in einer beliebigen Sprache erstellt werden und dennoch ohne Schwierigkeiten mehrere Sprachen enthalten. Browser beginnen bereits Unicode zu unterstützen. Der Netscape Communicator zum Beispiel unterstützt Unicode und kann, solange ihm die nötigen Schriften zur Verfügung stehen, Dokumente in vielen der Schriftarten darstellen, die von Unicode angeboten werden. Dies stellt einen wichtigen Schritt dar, da Unicode als der neue De-facto-

Standard für die Kodierung von Schriftzeichen entwickelt wurde. Java zum Beispiel verwendet Unicode als Standard für die Zeichenkodierung und Windows NT unterstützt die Zeichenkodierung nach Unicode ebenfalls.

Zeichen-Entitäten für reservierte Zeichen

Zum größten Teil gibt es die Zeichen-Entitäten, damit Sie Zeichen verwenden können, die im normalen ASCII-Zeichensatzes nicht enthalten sind. Es gibt jedoch einige Ausnahmen für einige Zeichen, die in HTML selbst eine besondere Bedeutung haben. Auch für diese Zeichen müssen Sie Zeichen-Entitäten verwenden.

Lassen Sie uns zum Beispiel annehmen, Sie wollten eine Zeile Code in eine HTML-Datei schreiben, die etwa so aussieht:

```
<p> <code> if x < 0 do print i </code> </p>
```

Das sieht nicht ungewöhnlich aus, oder? Unglücklicherweise kann HTML die Zeile nicht so anzeigen, wie sie geschrieben ist. Weshalb? Das Problem besteht in dem < (kleiner)-Zeichen. Für einen HTML-Browser bedeutet das Kleiner-Zeichen »hier ist der Anfang einer Kennung«. Da in diesem Zusammenhang das Kleiner-Zeichen eben kein Kennungs-Beginn ist, wird der Browser durcheinander kommen. Sie bekommen das gleiche Problem mit dem Größer-Zeichen (>), weil dies in HTML das Ende einer Kennung bezeichnet und mit dem kaufmännischen Und (&) das Ersatzzeichen (die Zeichen-Entitäten) einleitet. Korrekt in HTML geschrieben, würde jene Codezeile also so aussehen:

```
<p> <code> if x &lt; 0 do print i </code> </p>
```

HTML bietet benannte Ersatzzeichen (Escape-Codes) für jedes dieser Sonderzeichen sowie für die Gänsefüßchen an, wie aus Tabelle 6.1 ersichtlich.

Ersatzzeichen	Ergebnis
<	<
>	>
&	&
"	“

Tabelle 6.1: Ersatzzeichen für in Kennungen verwendete Zeichen

Das Gänsefüßchen-Ersatzzeichen ist etwas mysteriös. Rein technisch betrachtet, sollten Sie die Ersatzzeichen benutzen, anstatt die Gänsefüßchen direkt einzugeben, wenn Sie korrekten HTML-Code mit Gänsefüßchen darin erzeugen wollen. Gleichwohl habe ich bei keinem Browser irgendwelche Probleme mit der Darstellung von Gänsefüßchen feststellen können, auch wenn sie direkt in den HTML-Code geschrieben wa-

ren, noch habe ich viele HTML-Dateien gesehen, die sie benutzen. Größtenteils dürfte es sicher genug sein, die guten alten Gänsefüßchen in Ihren HTML-Dateien zu verwenden, anstatt die Ersatzzeichen zu nehmen.

Textausrichtung

Die Textausrichtung legt fest, mit welchem Rand ein Absatz, zum Beispiel eine Überschrift, bündig abschließt. Die Standardausrichtung ist linksbündig. Ebenfalls möglich ist es, einen Absatz rechtsbündig auszurichten. Etwas aus der Reihe fällt die zentrierte Ausrichtung. Hierbei wird der Text des Absatzes um die vertikale Mittelachse des Absatzes ausgerichtet. Der Text ist also zu keinem Rand bündig. In HTML 2.0 war es noch nicht möglich die Textausrichtung festzulegen. Im Normalfall war der Text linksbündig ausgerichtet.

HTML 3.2 liefert Erweiterungen für die Ausrichtung von Text und anderen Elementen; und diese Erweiterungen sind von verschiedenen Browsern zu unterschiedlichen Graden akzeptiert worden. HTML 4.0 unterstützt zwar immer noch die Attribute für Text- und Element-Ausrichtung, dennoch wird empfohlen mit Style Sheets zu arbeiten.

Die Ausrichtung individueller Elemente

Um eine individuelle Überschrift oder einen Absatz auszurichten, wenden Sie das `align`-Attribut an diesem HTML-Element an. Bei `align` handelt es sich um ein HTML-3.2-Attribut; es kann einen dieser drei Werte annehmen: `left`, `right` oder `center`. Sehen Sie sich die folgenden Beispiele an:

```
<h1 align=center>Northridge Paints, Inc.</h1>
<p align=center >We don't just paint the town red.</p>

<h1 align=left>Serendipity Products</h1>
<h2 align=right><a href="who.html">Who We Are</a></h2>
<h2 align=right><a href="products.html">What We Do</a></h2>
<h2 align=right><a href="contacts.html">How To Reach Us</a></h2>
```

Das folgende Eingabe-/Ausgabe-Beispiel zeigt einfache Ausrichtungen einiger Überschriften. Die Abbildung 6.13 zeigt das Ergebnis im Internet Explorer.



```
<h1 align=left>Serendipity Products</h1>
<h2 align=right><a href="who.html">Who We Are</a></h2>
<h2 align=right><a href="products.html">What We Do</a></h2>
<h2 align=right><a href="contacts.html">How To Reach Us</a></h2>
```

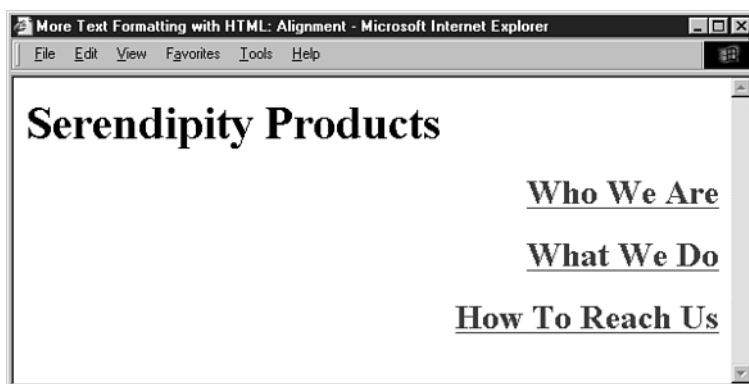


Abbildung 6.13:
Die Ausgabe im
Internet Explorer

Die Ausrichtung ganzer Elemente-Blöcke

Eine etwas flexiblere Methode der Ausrichtung von Textelementen ist die Verwendung von `<div>`-Kennungen.

`<div>` enthält verschiedene Attribute. Unter diesen Attributen befindet sich auch das `align`-Attribut (in HTML 4.0 verworfen), das Elemente nach links oder rechts ausrichtet, sie zentriert, kurz, sie genauso ausrichtet wie Überschriften und Paragraphen auch.

- ▶ `<div>` braucht nur ein einziges Mal eingesetzt zu werden, `align` dagegen muss wiederholt werden;
- ▶ `<div>` kann zur Ausrichtung beliebiger Elemente eingesetzt werden (Überschriften, Absätze, Zitate, Abbildungen, Tabellen usw.); das `align`-Attribut ist nur für eine begrenzte Anzahl von Kennungen einsetzbar.

Um einen Block von HTML-Code auszurichten, umgeben Sie den Code mit den eröffnenden und abschließenden `<div>`-Kennungen und fügen das `align`-Attribut in die eröffnende Kennung ein. So wie auch bei anderen Kennungen kann `align` die Werte `left`, `right` oder `center` haben.

```
<h1 align=left>Serendipity Products</h1>
<div align=right>
<h2><a href="who.html">Who We Are</a></h2>
<h2><a href="products.html">What We Do</a></h2>
<h2><a href="contacts.html">How To Reach Us</a></h2>
</div>
```

Der gesamte HTML-Code zwischen den `<div>`-Kennungen wird entsprechend den Werten des `ALIGN`-Attributs ausgerichtet werden. Wenn individuelle `align`-Attribute innerhalb von Überschriften oder Absätzen im Rahmen von `<div>` auftreten, werden diese Werte die globalen `<div>`-Einstellungen außer Kraft setzen.

Beachten Sie, dass `<div>` selbst kein Absatztyp ist. Sie müssen innerhalb der eröffnenden und abschließenden `<div>`-Kennungen immer noch reguläre Element-Kennungen (`<p>`, `<h1>`, `<ul>`, `<blockquote>` usw.) einsetzen.

Zusätzlich zu `<div>` gibt es auch noch `<center>`, eine Netscape-Erweiterung, die kein Bestandteil von HTML 2.0 oder 3.2 ist (und wahrscheinlich auch nicht von zukünftigen HTML-Versionen sein wird). Die `<center>`-Kennung verhält sich genauso wie `<div align=center>`, indem sie den gesamten HTML-Inhalt innerhalb der eröffnenden und abschließenden Kennungen zentriert. Sie setzen die `<center>`-Kennung vor den Text, den Sie zentrieren wollen, und dann die `</center>`-Kennung als Abschluss, so wie hier:

```
<center>
<h1>Northridge Paints, Inc.</h2>
<p>We don't just paint the town red.</p>
</center>
```

Um der Konsistenz Willen sind Sie wahrscheinlich besser beraten, wenn Sie `<div>` und `align` verwenden, um eine zentrierte Ausrichtung zu erreichen.

Schriftarten und -grade

Die `<font>`-Kennung, die ein Teil von HTML 3.2 ist, aber in HTML 4.0 verworfen ist, wird eingesetzt, um die Eigenheiten bestimmter Zeichen festzulegen, die von den Schriftfestlegungen nicht betroffen wurden. Ursprünglich wurde `<font>` eingesetzt, um den Schriftgrad der Zeichen festzulegen, die es umgibt, ist aber seither daraufhin erweitert worden, die Schriftart selbst und deren Farbe festzulegen.

In diesem Abschnitt werden wir Schriftarten und Schriftgrade besprechen. Am Tag 7, wenn wir über Farbe im Allgemeinen reden, werden Sie etwas über die Veränderung der Schriftfarbe lernen.

Wechsel des Schriftgrads

Die am weitesten verbreitete Anwendung der `<font>`-Kennung ist der Wechsel der Schriftgröße bei einem Zeichen, Wort, Ausdruck oder Textabschnitt. Die `<font>...</font>`-Kennungen umschließen den Text, während das `size`-Attribut den Grad angibt, in dem die Schrift verändert werden soll. Der Wertebereich des `size`-Attributs erstreckt sich von 1 bis 7, wobei 3 die Voreinstellung ist. Betrachten Sie das folgende Beispiel:

```
<p>Langweilen Sie sich mit Ihrer alten Schrift?  
<font size=5>Aendern Sie sie.</font></p>
```

Abbildung 6.14 zeigt die typischen Schriftgrade für jeden Wert von `SIZE`.

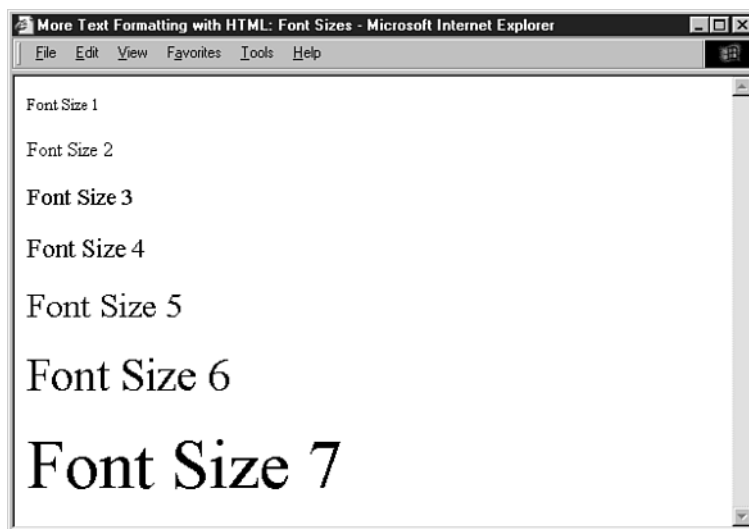


Abbildung 6.14:
Schriftgrade im
Internet Explorer

Sie können die Schriftgröße innerhalb der `<font>`-Kennung auch als relativen Wert angeben, indem Sie die `+`- oder `-`-Zeichen als Wert für `SIZE` einsetzen. Da der Standardwert 3 ist, können Sie die relativen Schriftgrade von `-3` bis `+4` ändern, so wie hier:

```
<p>Aendern wir den <font size=+2>Schriftgrad</font> wieder.</p>
```

Hier wird das Wort »Font« (innerhalb der `<font>`-Kennungen) um zwei Stufen größer als die ursprüngliche Schrift erscheinen, wenn Sie das Beispiel mit einem Browser betrachten, der dieses Feature unterstützt.

Relative Schriftgrade basieren auf einem Wert, den Sie mit der `<basefont>`-Kennung definieren können, welche eine Netscape-Erweiterung ist, die nun zu der Spezifikation von HTML 3.2 gehört. Die `<basefont>`-Kennung hat außerdem das notwendige Attribut `size`. `SIZE` kann Werte von 1 bis 7 annehmen. Alle relativen Schriftgradänderungen, die im Dokument nach der `<basefont>`-Kennung auftreten, werden sich auf deren Wert beziehen.

Versuchen Sie es zu vermeiden, die `<font>`-Kennung einzusetzen, um den Effekt eines größeren Schriftgrads zu simulieren, den inhaltsbasierte HTML-Kennungen wie Überschriften (`<h1>`, `<h2>` usw.) erzeugen, und setzen Sie sie auch nicht dazu ein, ein bestimmtes Wort oder einen Satz hervorzuheben. Wenn Ihr Dokument in einem Browser angesehen wird, der dieses Merkmal nicht unterstützt, verlieren Sie die Schriftgrade, und Ihr Text wird so wie alle anderen Absätze dargestellt werden. Wenn Sie sich jedoch an die inhaltsbasierten Kennungen halten, so wird Ihre Überschrift immer eine Überschrift bleiben. Begrenzen Sie deshalb Ihren Einsatz der `<font>`-Kennung auf einige wenige Spezialeffekte.

Änderung der Schriftart

Netscape führte die `<font>`-Kennung mit der Version 1.0 des Browser in HTML ein. Der Internet Explorer von Microsoft spielte das gleiche Spiel, als er die `<font>`-Kennung dahingehend erweiterte, das `face`-Attribut einzuschließen. Das Tag ist inzwischen Teil von HTML 3.2. Mit HTML 4.0 jedoch wird bevorzugt mit Style Sheets gearbeitet, um die Schriftgröße, die Sie verwenden wollen, zu bestimmen.

`face` nimmt als Wert den Namen der Schriftarten an, die von Anführungszeichen umgeben und durch Kommata voneinander getrennt werden. Wenn ein Browser, der `face` unterstützt, eine Seite mit `face` darin interpretiert, wird er das System nach der jeweiligen Schriftart durchsuchen. Wenn er die erste nicht finden kann, wird er die zweite suchen, dann die dritte usw., bis er eine Schriftart findet, die tatsächlich auf dem System installiert ist. Wenn er keine der aufgeführten Schriftarten finden kann, wird er stattdessen die Voreinstellungs-Schriftart benutzen. So würde beispielsweise der folgende Text in Futura dargestellt werden. Wenn Futura nicht zur Verfügung stünde, würde der Browser Helvetica versuchen und schließlich auf die Voreinstellung zurückfallen, wenn helvetica nicht verfügbar wäre.

```
<p><fontface="futura,helvetica">Serifenlose Schriftarten kommen ohne die kleinen  
Extra-Schnörkel aus. </font></p>
```

Bedenken Sie, wenn Sie das `face`-Attribut einsetzen, dass dies nur sehr wenige Browser momentan unterstützen, sodass es für den größten Teil Ihres Publikums nicht zur Verfügung steht. Außerdem haben viele Schriftarten auf unterschiedlichen Systemen unterschiedliche Namen; so heißt die gute alte Times auf einigen Systemen Times, auf anderen Times-Roman oder auch Times New Roman. Wegen der unterschiedlichen Benennungen der Schriftarten und dem Mangel an weitgehender Unterstützung für das `face`-Attribut sollte eine Festlegung der Schriftart nur für spezielle Präsentationen und nicht auf Ihren alltäglichen Seiten eingesetzt werden.

Das gefürchtete `<blink>`

Sie werden die `<blink>`-Kennung nicht in der offiziellen Dokumentation von Netscape finden. Die Möglichkeit, Text zum Blinken zu bringen, wurde in Netscape als versteckte, undokumentierte Fähigkeit, also als Osterie, eingeführt. Dennoch scheint ein ziemlicher Prozentsatz von Webseiten diese Möglichkeit zu nutzen. Die `<blink>...</blink>`-Kennungen bringen den dazwischen liegenden Text dazu, sich ein- und auszuschalten. Abhängig von der Netscape-Version, die Sie verwenden, kann dies bedeuten, dass der Text selbst entweder in regelmäßigen Intervallen verschwindet und wieder erscheint oder dass ein hässlicher grauer oder weißer Block hinter dem Text auftaucht und verschwindet. Blink wird gemeinhin dafür eingesetzt, die Aufmerksamkeit auf einen Teil der Seite zu lenken.

Das Problem mit `<blink>` ist, dass es zu viel Nachdruck verleiht. Da es sich wiederholt, fährt das Blinken dabei fort, Aufmerksamkeit auf eine Stelle zu richten, und kann in einigen Fällen dazu führen, dass es fast unmöglich wird, sich auf den Inhalt der restlichen Seite zu konzentrieren. Von der Verwendung von `<blink>` raten die meisten Web-Designer (einschließlich meiner Wenigkeit) vehement ab, da viele Leute es als sehr lästig, hässlich und ärgerlich empfinden. `<blink>` ist mit dem Kratzen von Fingernägeln auf einer Schiefertafel gleichzusetzen.

Wenn Sie `<blink>` schon einsetzen müssen, dann wenden Sie es sehr sparsam an (nicht mehr als einige wenige Wörter pro Seite). Seien Sie sich auch darüber im Klaren, dass das Blinken in einigen Versionen von Netscape ein- und ausgeschaltet werden kann. Wenn Sie einem Wort oder einem Satz Nachdruck verleihen wollen, sollten Sie einen konventionelleren Weg zusätzlich zu oder anstelle von `<blink>` einschlagen, da Sie nicht garantieren können, dass es erscheinen wird, auch wenn Ihr Leser Netscape benutzt.

<nobr> und <wbr>

Die <nobr>...</nobr>-Kennungen (no break = kein Umbruch) sind das Gegenteil der
-Kennung. Der Text innerhalb der <nobr>-Kennungen bleibt immer auf einer Zeile, auch wenn er sich ohne die Kennung über zwei oder mehr Zeilen erstreckt hätte.

Die <wbr>-Kennung (word break) gibt den angemessenen Zeilenumbruch innerhalb einer Zeile an (typischerweise innerhalb einer <nobr>...</nobr>-Sequenz). Im Unterschied zu
, welches einen Umbruch erzwingt, wird <wbr> nur angewendet, wenn es erforderlich ist. Wenn die Zeile problemlos auf den Bildschirm passt, wird <wbr> ignoriert. In XHTML müssen Sie das abschließende Tag <wbr /> verwenden.

Weder <nobr> noch <wbr> sind Teil von HTML 3.2 bzw. HTML 4.0. Sie sind Erweiterungen, die von Netscape eingeführt wurden und sowohl vom Netscape Communicator als auch vom Internet Explorer 4 unterstützt werden.

Übung 6.1: Erstellen Sie eine richtige HTML-Seite

Dies ist Ihre Chance, das anzuwenden, was Sie gelernt haben, und eine Webseite aus dem wirklichen Leben zu erzeugen. Keine zerstückelten oder übermäßig albernem Beispiele mehr. Die Webseite, die Sie in diesem Abschnitt erzeugen, ist echt, geeignet für den Gebrauch in der wirklichen Welt (oder zumindest der wirklichen Welt des Web).

Ihre Aufgabe in diesem Beispiel: Gestalten und erzeugen Sie die Homepage für einen Bücherladen namens The Bookworm (Der Bücherwurm), der sich auf alte und seltene Bücher spezialisiert hat.

Planen der Seite

In Kapitel 2 habe ich darauf hingewiesen, dass eine Planung Ihrer Web-Site vor dem Schreiben es zumeist leichter macht, die Sache aufzubauen und zu pflegen. Also überlegen Sie sich zuerst den Inhalt, den Sie auf die Seite setzen wollen. Hier sind ein paar Ideen für die Themen auf der Seite:

- ▶ Die Anschrift und Telefonnummer des Buchladens
- ▶ Eine kurze Beschreibung dessen, was der Buchladen ist und was ihn so einzigartig macht
- ▶ Aktuelle Titel und Autoren
- ▶ Veranstaltungsankündigungen

So, nun überlegen Sie sich ein paar Ideen für die Inhalte, mit denen Sie diese Seite verknüpfen wollen. Jeder Titel aus der Liste der aktuellen Titel dürfte ein logischer Kandidat dafür sein; Sie können zudem weitere Informationen über das Buch selbst und seinen Autor und Verleger anbieten, seinen Preis, gegebenenfalls seine Verfügbarkeit.

Die Veranstaltungsankündigungen könnten gleichfalls eine Reihe von Verknüpfungen ergeben, je nachdem, wie ausführlich Sie jede Veranstaltung vorstellen wollen. Wenn Sie nur ein oder zwei Sätze zu jeder verlieren wollen, dürfte es sinnvoller sein, sie direkt auf der Seite vorzustellen, anstatt jede Veranstaltung mit einer eigenen Seite zu verknüpfen. Warum sollten Ihre Leser warten, bis die neue Seite geladen wird, bloß um eine Handvoll Zeilen Text zu lesen?

Andere interessante Verbindungen könnten im Text selbst auftauchen, doch fürs erste ist unsere grundlegende Planung ausreichend, um damit anzufangen.

Beginnen Sie mit dem Gerüst

Zunächst erzeugen Sie das Gerüst, das alle HTML-Dateien haben müssen: Angaben zur Strukturierung des Dokuments, einen Titel und eine erste Überschrift. Achten Sie darauf, dass der Titel aussagekräftig, aber kurz ist; Sie können sich einen längeren Titel für das `<h1>`-Element auf der Seite aufheben. Die vier `<h2>`-Unterpunkte helfen Ihnen die vier Hauptsektionen Ihrer Web-Site zu definieren.

```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>
<head>
<title>The Bookworm Bookshop</title>
</head>
<body>
<h1>The Bookworm: A Better Book Store</h1>
<h2>Contents</h2>
<h2>About the Bookworm Bookshop</h2>
<h2>Recent Titles (as of 11-Nov-99)</h2>
<h2>Upcoming Events</h2>
</body>
</html>
```

Jede der Überschriften Ihrer Seite markiert den Anfang eines speziellen Abschnitts dieser Seite. Erstellen Sie einen Anchor für jede der Themen-Überschriften, damit Sie leicht von Sektion zu Sektion springen können.

Die Anchor-Namen sind einfach: `top` (für die Hauptüberschrift), `contents` (für die Inhaltstabelle) und `about`, `recent` und `upcoming` für die drei Unterpunkte der Seite. Der revidierte Code mit den Anchor sieht wie folgt aus:

```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>
<head>
<title>The Bookworm Bookshop</title>
</head>
<body>
<a name="top"><h1>The Bookworm: A Better Book Store</h1></a>
<a name="contents"><h2>Contents</h2></a>
<a name="about"><h2>About the Bookworm Bookshop</h2></a>
<a name="recent"><h2>Recent Titles (as of 11-Nov-99)</h2></a>
<a name="upcoming"><h2>Upcoming Events</h2></a>
</body>
</html>
```

Fügen Sie Inhalt hinzu

Nun beginnen Sie damit, die Seite mit Inhalt zu füllen. Da es um ein literarisches Bestreben geht, würde sich ein kleines Zitat über alte Bücher am Anfang der Seite gut machen. Da es ein Zitat ist, können Sie die `<blockquote>`-Kennung benutzen, um es als solches erscheinen zu lassen. Die Herkunft der Verse ist eine Quellenangabe, so dass Sie auch die `<cite>`-Kennung verwenden können.

Geben Sie folgenden Code in die Zeile nach der Überschrift 1. Ordnung ein:

```
<blockquote>
"Old books are best--how tale and rhyme<br />
Float with us down the stream of time!"<br />
- Clarence Urmy, <cite>Old Songs are Best</cite>
</blockquote>
```

Fügen Sie gleich nach den Anführungszeichen die Adresse des »Bookstores« hinzu. Es handelt sich hierbei um einen einfachen Paragraphen, dessen Zeilen mit Zeilenumbruch separiert wurden:

```
<p>The Bookworm Bookshop<br />
1345 Applewood Dr<br />
Springfield, CA 94325<br />
(415) 555-0034
</p>
```

Fügen Sie ein Inhaltsverzeichnis hinzu

In der von Ihnen erstellten Seite muss man viel scrollen, um vom Anfang der Seite zu deren Ende zu gelangen. Sie könnten diese Seite verbessern und ein kleines Inhaltsverzeichnis am Seitenanfang hinzufügen, die die Sektionen in einer Merkpunktliste auflistet. Klickt ein Leser eines der Links dieser Inhaltstabelle an, wird er oder sie sofort zu der Sektion gelangen, die gewählt wurde. Da Sie die Anchor ja schon erstellt haben, ist es für Sie leicht zu sehen, wo Sie diese Links hinführen werden.

Die Überschriften der Inhaltstabellen haben Sie bereits. Sie müssen jetzt eine Liste und eine Trennlinie hinzufügen. Danach erstellen Sie die Links, die zu den anderen Sektionen der Seite führen. Der Code sieht wie folgt aus:

```
<a name="contents"><h2>Contents</h2></a>
<ul>
  <li><a href="#about">About the Bookworm Bookshop</a></li>
  <li><a href="#recent">Recent Titles</a></li>
  <li><a href="#upcoming">Upcoming Events</a></li>
</ul>
<hr />
```

Bild 6.15 zeigt den einführenden Teil der »Bookworm Bookshop«-Seite im Internet Explorer

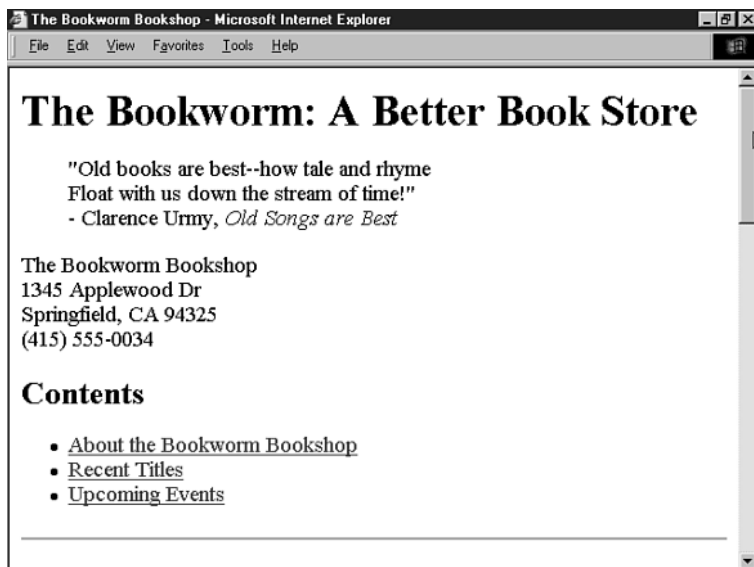


Abbildung 6.15:
Darstellung des
oberen Teils der
»Bookworm Book-
shop«-Seite

Erstellen der »Bookstore«-Beschreibung

Wir kommen nun zum ersten deskriptiven Unterpunkt der Seite. Einen Unterpunkt haben Sie ja bereits erstellt. Diese Sektion liefert Ihnen eine Beschreibung des »Bookstores«. Nach der Überschrift (1. Zeile des folgenden Beispiels), habe ich die Beschreibung so erstellt, dass sie eine Feature-Liste enthält. Die Features stechen somit besser aus dem Text heraus.

```
<a name="about"><h2>About the Bookworm Bookshop</h2></a>
<p>Since 1933, The Bookworm Bookshop has offered
rare and hard-to-find titles for the discerning reader.
The Bookworm offers:</p>
<ul>
<li>Friendly, knowledgeable, and courteous help</li>
<li>Free coffee and juice for our customers</li>
<li>A well-lit reading room so you can "try before you buy"</li>
<li>Four friendly cats: Esmerelda, Catherine, Dulcinea and Beatrice</li>
</ul>
```

Fügen Sie eine Notiz über die Öffnungszeiten hinzu und markieren Sie die Zahlen:

```
<p>Our hours are <strong>10am to 9pm</strong> weekdays,
<strong>noon to 7</strong> on weekends.</p>
```

Beenden Sie die Sektion mit Links zu den Inhaltstabellen und zum Seitenanfang, gefolgt von einem Horizontallineal.

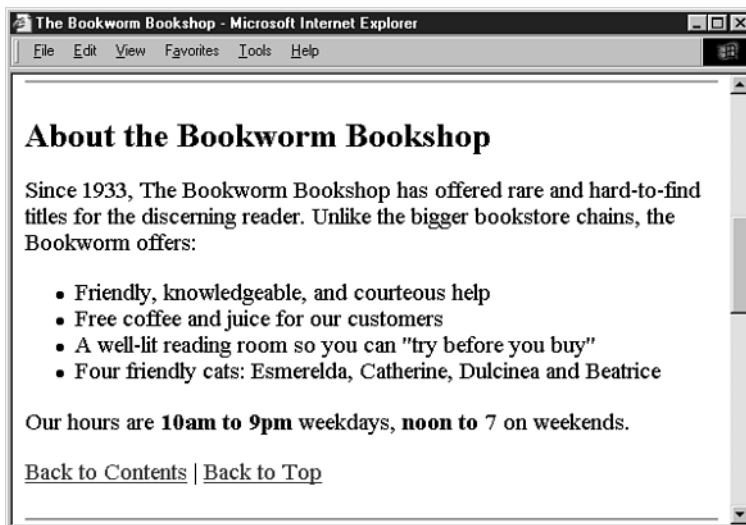


Abbildung 6.16:
Die »Bookstore«-
Beschreibung

```
<p><a href="#contents">Back to Contents</a> | <a href="#top">Back to Top</a></p>
<hr />
```

Erstellen der Recent-Titles-Sektion

Die Recent-Titles-Sektion ist an und für sich ein klassisches Link-Menü. Sie können hier eine Titelliste in eine ungeordnete Liste eingeben, wobei die Titel selbst Zitate sind (wird mit dem `<cite>`-Tag erreicht).

Nach der Recent-Titles-Sektion (in der ersten Zeile des folgenden Beispiels aufgeführt), geben Sie folgenden Code ein:

```
<a name="recent"><h2>Recent Titles (as of 11-Nov-99)</h2></a>
<ul>
<li>Sandra Bellweather, <cite>Belladonna</cite></li>
<li>Jonathan Tin, <cite>20-Minute Meals for One</cite></li>
<li>Maxwell Burgess, <cite>Legion of Thunder</cite></li>
<li>Alison Caine, <cite>Banquo es Ghost</cite></li>
</ul>
<hr />
```

Fügen Sie jetzt die `<a>`-Tags hinzu, um die Links zu erstellen. Wie weit soll das Link erweitert werden? Soll es die gesamte Zeile (Autor und Titel) oder nur den Buchtitel enthalten? Bei dieser Entscheidung sind einzig und allein Ihre Präferenzen ausschlaggebend. Ich persönlich möchte nur zu den wichtigsten Sachen verweisen und sichergehen, dass das Link auch aus dem Text heraussticht. Das ist mir allemal lieber als ein überladener Text. In diesem Beispiel habe ich nur auf die Buchtitel verwiesen und habe Links zur Inhaltstabelle und zum Seitenanfang erstellt:

```
<a name="recent"><h2>Recent Titles (as of 11-Nov-99)</h2></a>
<ul>
<li>Sandra Bellweather, <a href="belladonna.html">
<cite>Belladonna</cite></a></li>
<li>Johnathan Tin, <a href="20minmeals.html">
<cite>20-Minute Meals for One</cite></a></li>
<li>Maxwell Burgess, <a href="legion.html">
<cite>Legion of Thunder</cite></a></li>
<li>Alison Caine, <a href="banquo.html">
<cite>Banquo es Ghost</cite></a></li>
</ul>
<p><a href="#contents">Back to Contents</a> | <a href="#top">Back to Top</a></p>
<hr />
```

Das `<cite>`-Tag befindet sich innerhalb des Link-Tag `<a>`. Ich hätte es auch genauso einfach außerhalb des `<a>`-Tag setzen können; Style-Tags können eben überall eingegeben werden. Doch wie ich schon erwähnt habe, passen Sie auf, dass Sie Tags nicht

überlappen. Ihr Browser würde sich wahrscheinlich nicht mehr auskennen. Mit anderen Worten: unterlassen Sie Folgendes:

```
<a href="banquo.html"><cite>Banquo es Ghost</a></cite>
```

In Bild 6.17 sehen Sie, wie die Recent-Titles-Sektion im Internet Explorer erscheint:



Abbildung 6.17:
Die Recent-Titles-
Sektion

Die Veranstaltungsankündigungen vervollständigen

Fahren wir mit der Sektion der kommenden Ereignisse fort. Es macht nicht viel Sinn Links für ein paar Sätze zu erstellen. Für diese Sektion erstellen wir eine Menü-Liste (dabei wird das `<ul>`-Tag verwendet), die kurze Paragraphen aufzeigt (in einigen Browsern befinden sich Bullets am Anfang eines jeden Paragraphen). Einige Sätze, die sich in der Nähe eines neuen Paragraphen befinden, sind fett gedruckt. *Diese Sätze markieren eine Zusammenfassung der Ereignisse selbst.* Somit wird über jeden Paragraphen ein schneller Überblick verschafft. Sind die Leser jedoch nicht interessiert, können Sie bestimmte Paragraphen auch einfach ignorieren.

Wie in den vorherigen Sektionen auch, beenden Sie diese Sektion mit Links, die auf den Seitenanfang und den Inhalt verweisen, gefolgt von einer horizontalen Trennlinie:

```
<a name="upcoming"><h2>Upcoming Events</h2></a>
<ul>
<li><b>The Wednesday Evening Book Review</b> meets, appropriately, on
Wednesday evenings at 7 pm for coffee and a round-table discussion.
Call the Bookworm for information on joining the group.</li>
<li><b>The Children es Hour</b> happens every Saturday at 1 pm and includes
reading, games, and other activities. Cookies and milk are served.</li>
<li><b>Carole Fenney</b> will be at the Bookworm on Friday, September 17,
to read from her book of poems <cite>Spiders in the Web.</cite></li>
<li><b>The Bookworm will be closed</b> October 1 to remove a family
of bats that has nested in the tower. We like the company, but not
```

```
the mess they leave behind!</li>
</ul>
<p><a href="#contents">Back to Contents</a> | <a href="#top">Back to
Top</a></p>
<hr />
```

Unterschreiben Sie die Seite

Sie sind fertig; unterschreiben Sie also, was Sie erstellt haben, damit Ihre Leser wissen, wer die Arbeit erledigt hat. Ich habe die Signatur hier mit einer Trennlinie vom Text abgesetzt, das letzte Aktualisierungsdatum eingefügt, meinen Namen als »Web-Master« (das ist Web-Jargon und bedeutet: die Person, die für ein Web-System verantwortlich ist) mit einer E-Mail-Anschrift eingesetzt und eine Copyright-Notiz (mit einem Copyright-Symbol, gekennzeichnet durch den numerischen Escape-Code ©):

```
<hr />
<address>
Last Updated: 30-Apr-97<br />
WebMaster: Laura Lemay lemay@bookworm.com<br />
&#169; copyright 1995 the Bookworm<br />
</address>
```

Bild 6.18 zeigt den unteren Teil der Seite, welche die kommende Ereignis-Sektion und die Seitenunterschrift enthält, wie sie im Internet Explorer angezeigt wird.

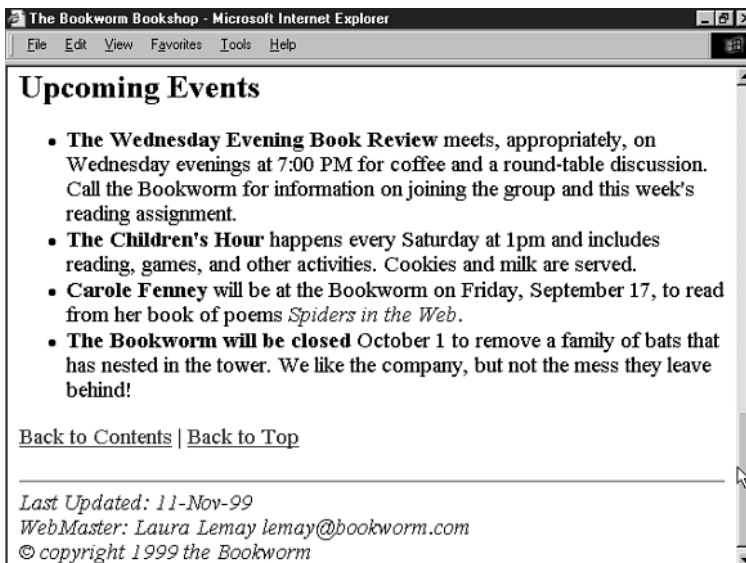


Abbildung 6.18:
Die Veranstaltungs-
ankündi-
gung-Sektion und
die Seitenunter-
schrift

Schauen wir einmal, was wir haben

Hier ist der bisherige HTML-Code (mit den numerischen Zeichenkodierungen der deutschen Umlaute) für die Seite:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">
<html>
<head>
<title>The Bookworm Bookshop</title>
</head>
<body>
<a name="top"><h1>The Bookworm: A Better Book Store</h1></a>
<blockquote>
"Old books are best--how tale and rhyme<br />
Float with us down the stream of time!"<br />
- Clarence Urmay, <cite>Old Songs are Best</cite>
</blockquote>
<p>The Bookworm Bookshop<br />
1345 Applewood Dr<br />
Springfield, CA 94325<br />
(415) 555-0034
</p>
<a name="contents"><h2>Contents</h2></a>
<ul>
<li><a href="#about">About the Bookworm Bookshop</a></li>
<li><a href="#recent">Recent Titles</a></li>
<li><a href="#upcoming">Upcoming Events</a></li>
</ul>
<hr />
<a name="about"><h2>About the Bookworm Bookshop</h2></a>
<p>Since 1933, the Bookworm Bookshop has offered
rare and hard-to-find titles for the discerning reader.
The Bookworm offers:</p>
<ul>
<li>Friendly, knowledgeable, and courteous help</li>
<li>Free coffee and juice for our customers</li>
<li>A well-lit reading room so you can "try before you buy"</li>
<li>Four friendly cats: Esmerelda, Catherine, Dulcinea and Beatrice</li>
</ul>
<p>Our hours are <strong>10am to 9pm</strong> weekdays,
<strong>noon to 7</strong> on weekends.</p>
<p><a href="#contents">Back to Contents</a> | <a href="#top">Back to
Top</a></p>
<hr />
<a name="recent"><h2>Recent Titles (as of 11-Nov-99)</h2></a>
<ul>
```

```

<li>Sandra Bellweather, <a href="belladonna.html">
  <cite>Belladonna</cite></a></li>
<li>Johnathan Tin, <a href="20minmeals.html">
  <cite>20-Minute Meals for One</cite></a></li>
<li>Maxwell Burgess, <a href="legion.html">
  <cite>Legion of Thunder</cite></a></li>
<li>Alison Caine, <a href="banquo.html">
  <cite>Banquo es Ghost</cite></a></li>
</ul>
<p><a href="#contents">Back to Contents</a> | <a href="#top">Back to
Top</a></p>
<hr />
<a name="upcoming"><h2>Upcoming Events</h2></a>
<ul>
<li><b>The Wednesday Evening Book Review</b> meets, appropriately, on
  Wednesday evenings at 7 pm for coffee and a round-table discussion.
  Call the Bookworm for information on joining the group.</li>
<li><b>The Children es Hour</b> happens every Saturday at 1 pm and includes
  reading, games, and other activities. Cookies and milk are served.</li>
<li><b>Carole Fenney</b> will be at the Bookworm on Friday, September 17,
  to read from her book of poems <cite>Spiders in the Web.</cite></li>
<li><b>The Bookworm will be closed</b> October 1 to remove a family
  of bats that has nested in the tower. We like the company, but not
  the mess they leave behind!</li>
</ul>
<p><a href="#contents">Back to Contents</a> | <a href="#top">Back to
Top</a></p>
<hr />
<address>
Last Updated: 11-Nov-99<br />
WebMaster: Laura Lemay lemay@bookworm.com<br />
&#169; copyright 1999 the Bookworm<br />
</address>
</body>
</html>

```

Damit haben wir nun also ein paar Überschriften, etwas Text, einige Themen und einige Verknüpfungen. Das ist die Grundlage für eine gute Webseite. An dieser Stelle, wo der meiste Inhalt schon da ist, überlegen Sie sich, ob Sie noch weitere Verbindungen herstellen wollen oder welche anderen Merkmale Sie der Seite noch hinzufügen möchten.

Beispielsweise wurde im Einleitungsabschnitt auf die vier Katzen hingewiesen, die zum Buchladen gehören. Obwohl Sie es ursprünglich nicht geplant haben, könnten Sie jetzt leicht Webseiten erzeugen, die die Katzen beschreiben (und ein Bild zeigen), und sie dann zurückverbinden mit dieser Seite, eine Verknüpfung (und eine Seite) pro Katze.

Ist es wichtig, die Katzen zu beschreiben? Als Gestalter der Seite liegt es ganz bei Ihnen, das zu entscheiden. Sie könnten diese Seite mit allen möglichen Sachen verbinden, wenn Sie einen interessanten Anlass für eine solche Verknüpfung hätten (und etwas, womit Sie verknüpfen können). Verbinden Sie die Anschrift des Buchladens mit der örtlichen Handelskammer. Verbinden Sie das Zitat mit einer Online-Enzyklopädie von Zitaten. Verbinden Sie den Hinweis auf den kostenlosen Kaffee mit der Kaffee-Homepage.

Am Tag 16 erzähle ich noch mehr interessante Sachen über Verknüpfungen (und darüber, wie man sich beim Verknüpfen nicht ablenken lässt); an diesem Tag erfahren Sie auch noch mehr darüber, was Sie bei guten Webseiten tun oder unterlassen sollten. Der Grund, aus dem ich diesen Punkt hier anspreche, ist: Wenn Sie erst einmal den Inhalt auf Ihrer Seite haben, dann tauchen mannigfaltige Gelegenheiten auf, die Seite zu erweitern und mit anderen zu verbinden; Möglichkeiten, an die Sie gar nicht gedacht haben, als Sie Ihre Grundüberlegungen anstellten. Sobald Sie also mit der Seite einigermaßen fertig sind, ist es gut innezuhalten und sich anzuschauen, was Sie haben, sowohl auf Ihrem Plan als auch das, was Sie bereits geschafft haben.

Für den Zweck unseres Beispiels hören wir hier auf und begnügen uns mit den Verknüpfungen, die wir bereits erstellt haben. Wir sind nahe daran, alles erledigt zu haben, sodass ich dieses Kapitel nicht noch länger machen will, als es ohnehin schon ist!

Testen Sie das Ergebnis

Jetzt, wo der gesamte Code an seinem Platz steht, können Sie sich das Ergebnis mit Ihrem Browser ansehen. Die Abbildung 6.18 zeigt, wie es im Internet Explorer aussieht. So, das ist es, nachdem Sie all Ihre Tippfehler korrigiert haben und die vergessenen abschließenden Kennungen und dergleichen seltsame Fehler, die sich scheinbar immer einschleichen, wenn man zum ersten Mal eine HTML-Datei erzeugt. Und das scheint stets zu passieren, egal wie sorgfältig Sie arbeiten. Wenn Sie einen HTML-Editor oder ein anderes Hilfsprogramm benutzen, wird es ein bisschen leichter, aber Fehler scheint es immer zu geben. Daher brauchen wir eine Vorschau, damit Sie Fehlern auf die Spur kommen, bevor Sie das Dokument tatsächlich für andere Leute verfügbar machen.

Verfeinerungen

Bis jetzt war alles auf der Seite einfaches HTML 2.0, sodass es in allen Browsern lesbar ist und auch ziemlich gleich aussieht. Wenn ich jedoch an diesen Punkt gekommen bin, kann ich zusätzliche Formatierungs-Tags und Attribute hinzufügen, die die Seite für viele Leser nicht ändern werden, sie jedoch in Browsern ein wenig schicker aussehen lassen, sofern diese Erweiterungen unterstützt werden.

Welche Erweiterungen wollen wir also verwenden? Ich habe zwei ausgewählt:

- ▶ Den Titel der Seite, das Zitat und die Adresse des Ladens zentrieren
- ▶ An der Adresse selbst eine geringfügige Änderung des Schriftgrads vornehmen

Um den obersten Teil der Seite zu zentrieren, werden wir die Überschrift, das Zitat und die Adresse des Buchladens mit der `<div>`-Kennung folgendermaßen umschließen:

```
<div align=center>
<h1>The Bookworm: A Better Book Store</h1>
<blockquote>
"Old books are best--how tale and rhyme<br />
Float with us down the stream of time!"<br />
- Clarence Umy, <cite>Old Songs are Best</cite>
</blockquote>
<p>The Bookworm Bookshop<br />
1345 Applewood Dr<br />
Springfield, CA 94325<br />
(415) 555-0034
</p>
</div>
```

Um den Schriftgrad der Adresse zu verändern, umschließen Sie die Zeilen, die die Adresse enthalten, mit der `<font>`-Kennung:

```
<p><font size=+1>The Bookworm Bookshop<br />
1345 Applewood Dr<br />
Springfield, CA 94325<br />
(415) 555-0034
</font></p>
```

Abbildung 6.19 zeigt das Endergebnis mit Erweiterungen im Internet Explorer. Beachten Sie, dass keine dieser Änderungen die Lesbarkeit der Seite in den Browsern beeinträchtigt, die die `<div>`- oder `<font>`-Kennung nicht unterstützen; die Seite sieht auch dort noch gut aus, nur ein wenig anders.

Wann sollten Sie Textformatierungs-Attribute einsetzen? Die Faustregel, der ich am liebsten folge, ist, Erweiterungen nur dann einzusetzen, wenn ihre Verwendung nicht mit anderen Browsern in Konflikt gerät. HTML 4.0 empfiehlt Webseiten-Autoren mit den Style Sheets anstelle der Textformatierungs-Tags, beispielsweise `<font>`, und Attributen, wie etwa `align`, zu arbeiten. Allerdings werden Style Sheets nur von den neuesten Browsern unterstützt.

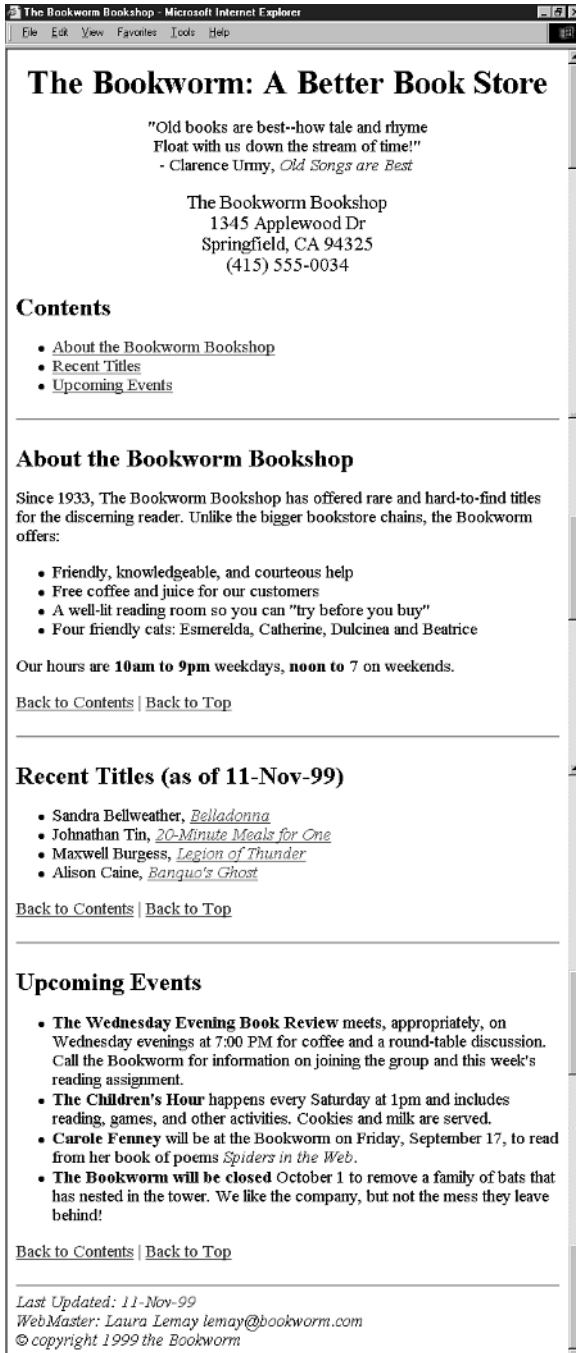


Abbildung 6.19:
Die überarbeitete »Bookworm«-
Homepage mit zusätzlichen
Attributen

Zusammenfassung

Tags, Tags und noch mehr Tags! Nach diesem Kapitel kennen Sie die meisten übriggebliebenen Tags der HTML-Sprache für die Präsentation des Textes, eine gute Anzahl der neuen Tags für zusätzliche Textformatierung und -präsentation, und Sie haben eine echte Webseite zusammengestellt. Sie könnten hier jetzt aufhören und recht ansehnliche Webseiten erzeugen. Aber es kommen noch mehr tolle Sachen, also legen Sie das Buch jetzt nicht weg.

Tabelle 6.2 zeigt eine kurze Zusammenfassung aller Kennungen, die Sie in diesem Kapitel kennen gelernt haben und die in der HTML 4.0-Spezifikation enthalten sind.

Kennung	Attribut	Verwendungszweck
<code><address>...</address></code>		„Signatur“ von Webseiten erscheint für gewöhnlich im unteren Teil eines jeden Dokuments und enthält Kontakt- oder Copyright-Informationen
<code>...</code>		Fetter Text
<code><big>...</big></code>		(HTML-3.2) Text mit größerer Schriftgröße
<code><blink>...</blink></code>		(Netscape-Erweiterung) verursacht einen blinkenden Effekt im eingeschlossenen Text
<code><blockquote>...</blockquote></code>		Ein Zitat, länger als nur ein paar Wörter
<code><cite>...</cite></code>		Eine Quellenangabe
<code><code>...</code></code>		Ein Code-Beispiel
<code><dfn>...</dfn></code>		Eine Definition oder ein zu definierender Ausdruck
<code>...</code>		Hervorgehobener Text
<code><i>...</i></code>		Kursiver Text
<code><kbd>...</kbd></code>		Text, der vom Anwender einzugeben ist
<code><pre>...</pre></code>		Vorformatierter Text; alle Leerstellen, Tabulatorzeichen und Zeilenumbrüche werden beibehalten. Der Text wird außerdem in fester (monospaced) Schrift dargestellt
<code><s>...</s></code>		(HTML-3.2) durchgestrichener Text
<code><samp>...</samp></code>		Beispieltext
<code><small>...</small></code>		(HTML-3.2) Text mit kleinerer Schriftgröße
<code>...</code>		Stärker hervorgehobener Text

Tabelle 6.2: HTML-Tags aus Kapitel 65

Kennung	Attribut	Verwendungszweck
<code><sub>...</sub></code>		(HTML-3.2) tiefgestellter Text
<code><sup>...</sup></code>		(HTML-3.2) hochgestellter Text
<code><tt>...</tt></code>		Text in Schreibmaschinenschrift (eine Schrift mit fester Weite wie beispielsweise Courier)
<code><u>...</u></code>		Unterstrichener Text
<code><var>...</var></code>		Ein Variablen-Name
<code><hr /></code>		Eine horizontale Trennlinie an der gegebenen Stelle im Text
<code><hr /></code>	size	(HTML-3.2) die Dicke der Trennlinie, in Pixel
<code><hr /></code>	width	(HTML-3.2) die Breite der Trennlinie, entweder in genauer Pixel-Anzahl oder als Prozentsatz der Seitenbreite (z.B. 50 Prozent)
<code><hr /></code>	align	(HTML-3.2) die Ausrichtung der Trennlinie auf der Seite. Mögliche Werte sind left, right und center
<code><hr /></code>	noshade	(HTML-3.2) Darstellung der Trennlinie ohne die dreidimensionale Schattierung
<code>
</code>		Ein Zeilenumbruch; beginnt das nächste Zeichen auf der folgenden Zeile (ohne jedoch einen neuen Absatz oder Listeneintrag anzufangen)
<code><nobr>...</nobr></code>		(Erweiterung) kein Zeilenumbruch für Text innerhalb dieser Kennung
<code><wbr></code>		(Erweiterung) Zeilenumbruch an dieser Stelle, wenn Bildschirmteilung dies erfordert
<code><p>, <h1 - 6></code>	align=left	(HTML-3.2-Erweiterung) richtet den Text innerhalb des Absatzes oder der Überschrift linksbündig aus
<code><p>, <h1 - 6></code>	align=right	(HTML-3.2-Erweiterung) richtet den Text innerhalb des Absatzes oder der Überschrift rechtsbündig aus
<code><p>, <h1 - 6></code>	align=center	(HTML-3.2-Erweiterung) zentriert den Text innerhalb des Absatzes oder der Überschrift
<code><div>...</div></code>	align=left	(HTML-3.2-Erweiterung) richtet den Inhalt zwischen den öffnenden und abschließenden Kennungen linksbündig aus
<code><div>...</div></code>	align=right	(HTML-3.2-Erweiterung) richtet den Inhalt zwischen den öffnenden und abschließenden Kennungen rechtsbündig aus

Tabelle 6.2: HTML-Tags aus Kapitel 65 (Forts.)

Kennung	Attribut	Verwendungszweck
<code><div>...</div></code>	<code>align=</code> <code>center</code>	(HTML-3.2-Erweiterung) zentriert den Inhalt zwischen den eröffnenden und abschließenden Kennungen
<code><center>...</center></code>		(HTML-3.2-Erweiterung) zentriert den Inhalt zwischen den eröffnenden und abschließenden Kennungen
<code>...</code>	<code>size</code>	(HTML-3.2-Erweiterung) gibt die Größe der Schrift an, zu der gewechselt werden soll, entweder von 1 bis 7 (Voreinstellung ist 3) oder als relativen Wert +N oder -N. Relative Schriftgrößen basieren auf dem Wert von <code><basefont></code>
	<code>face</code>	(Internet-Explorer-Erweiterung) der Name der Schriftart, zu der gewechselt werden soll, als Liste von Schriftarten, aus denen ausgewählt werden kann
<code><basefont></code>	<code>size</code>	(Netscape-Erweiterung) die Voreinstellungsgröße der Schrift, von der aus relative Größenveränderungen vorgenommen werden

Tabelle 6.2: HTML-Tags aus Kapitel 65 (Forts.)

Workshop

Ein weiteres (zugegeben langes) Kapitel liegt hinter uns! Sie haben heute viele Grundkenntnisse vermittelt bekommen und haben bestimmt schon den ganzen Kopf voll davon. Ich werde also versuchen, die heutigen Fragen so einfach wie möglich zu halten.

Fragen und Antworten

F Wenn es also in HTML Zeilenumbrüche gibt (`<br />`), wie ist es dann mit Seitenumbrüchen?

A Es gibt in HTML kein Tag für Seitenumbrüche. Denken Sie daran, was der Ausdruck Seite in einem Web-Dokument bedeutet. Wenn jedes Dokument im Web eine einzelne Seite ist, dann ist der einzige Weg, einen Seitenumbruch zu erzeugen, ein HTML-Dokument in verschiedene Dateien aufzuteilen und sie miteinander zu verbinden.

Selbst innerhalb eines einzigen Dokuments haben Browser kein Konzept einer Seite; jedes Dokument wird einfach fortlaufend angezeigt. Setzt man einen Bildschirminhalt mit einer Seite gleich, so kann man in HTML nicht von einem Seitenumbruch ausgehen, weil die Größe des Bildschirmfensters ja bei jedem Browser anders ist, was nicht nur vom Browser selbst abhängt, sondern

auch von dem Monitor, auf dem er läuft, der eingestellten Zeilenzahl, der gegenwärtigen Schriftart und anderen Faktoren,, die man mit HTML nicht kontrollieren kann.

Wenn Sie Ihre Webseite gestalten, sollten Sie sich nicht zu sehr am Konzept einer Seite aufhängen, wie es sie bei Papierdokumenten gibt. Erinnern Sie sich, dass die Stärke von HTML in der Flexibilität bei verschiedenen Arten von Systemen und Formaten liegt. Stellen Sie sich kleine Informationshäppchen vor und wie sie für eine komplette Web-Präsentation miteinander verbunden werden.

F Wie kann ich em-Striche oder typographische Anführungszeichen in meine HTML-Dateien aufnehmen?

A Das können Sie nicht. Weder em-Striche noch typographische Anführungszeichen sind als Teil des ISO-Latin-1-Zeichensatzes definiert, weshalb diese Zeichen für HTML im Moment nicht verfügbar sind. HTML 4.0 verspricht dieses Problem über die Unterstützung von Unicode, das Zugriff auf einen wesentlich umfangreicheren Zeichensatz bietet, zu beheben.

F »<blink> ist das HTML-Equivalent von Fingernägeln auf einer Schiefertafel?« Ist das nicht ein wenig übertrieben?

A Ich konnte der Versuchung nicht widerstehen.

Viele Leute verabscheuen `<blink>` und werden Ihnen das auch sofort mit einer Leidenschaft mitteilen, die normalerweise für Politik und Religion reserviert ist. Es gibt sogar Leute, die Ihre Seiten nur deshalb ignorieren werden, weil Sie `<blink>` eingesetzt haben. Warum wollen Sie Ihr Publikum verscheuchen und vom Inhalt ablenken, nur um einen kitschigen Effekt zu verwirklichen?

Quiz

1. Welches ist der Unterschied zwischen logischen und physischen Styles?
2. Wofür kann das `<pre>`-Tag verwendet werden?
3. Wofür wird das `<address>`-Tag am meisten benutzt?
4. Mit älteren HTML-Versionen konnten Sie den Text einer Webseite ausrichten und zentrieren. Welche Methode wird von HTML 4.0 für diese Aufgaben empfohlen?
5. Nennen Sie acht logische Style-Tags und sehen Sie dabei nicht in Tabelle 6.2 nach. Erklären Sie, warum Sie logische Tags anstelle der physischen Tags verwenden sollen.

Antworten

1. Logische Styles zeigen an, wie der markierte Text verwendet wird (Zitat, Definition, Code, usw). Physische Styles zeigen an, wie der markierte Text angezeigt wird (fett gedruckt, kursiv, monospaced, usw).
2. Vorformatierter Text (preformatted) kann für text-basierende Tabellen, Code-Beispiele, ASCII art und jede andere Webseite verwendet werden, die extra Platz für das Ausrichten der Zeichen benötigt.
3. Das <address>-Tag wird meistens für Unterschriften-ähnliche Entitäten auf einer Webseite benutzt. Diese Entitäten enthalten Webseiten Autor, Kontakt-Informationen, Daten, Copyright-Notizen oder Warnungen und erscheint für gewöhnlich am Ende einer Seite.
4. Ausrichtung und Zentrierung eines Textes kann man mit Style Sheets machen. Dieser Ansatz wird in HTML 4.0 empfohlen.
5. Die acht logischen Styles sind: (für markierten Text), (für fett gedruckten Text), <code> (für den Programmiercode), <samp> (ähnlich wie <code>), <kbd> (um Tastatureingaben der Anwender anzuzeigen), <var> (Namen für Variablen), <dfn> (für Definitionen) und <cite> (für kurze Notizen oder Zitate). Logische Tags hängen vom Browser ab.

Übungen

1. Was sollte sich auf Ihren Seiten befinden, damit Besucher sich intensiver mit diesen beschäftigen? Vergessen Sie nicht mit Links auf andere Seiten Ihrer Site zu verweisen.
2. Mit dieser Übung soll Ihre kreative Ader unterstützt werden. Stellen Sie sich vor, Sie hätten ein Produkt erfunden, das Sockenpaare in Zukunft zusammenbleiben und nicht mehr in der Waschmaschine verloren gehen. Erstellen Sie eine Seite, die sämtliche Vorteile dieses Produkts aufzählt und erklärt, warum dieses in keinem Haushalt fehlen darf. Verwenden Sie verschiedene Styles zum Betonen oder Markieren der wichtigsten Punkte dieser Seite.



Die Verwendung von Abbildungen, Farben und Hintergründen

**Woche
1**

Wenn Sie Schwierigkeiten dabei haben, alle die HTML-Kennungen zu behalten, die ich Ihnen in den letzten paar Tagen an den Kopf geworfen habe, können Sie jetzt aufatmen: Heute wird es einfacher. Tatsächlich werden Sie heute nur sehr wenige neue HTML-Tags kennen lernen. Der Fokus liegt heute darauf, Abbildungen und Farben in Ihre Webseiten einzufügen. In diesem Kapitel werden Sie etwas über die HTML-Kodierungen für das Hinzufügen von Abbildungen, Farben und Hintergründen lernen, und zwar:

- ▶ Die Abbildungsarten, die Sie in Webseiten benutzen können
- ▶ Wie Sie Abbildungen auf Ihrer Webseite einsetzen, entweder allein oder zusammen mit Text
- ▶ Wie Abbildungen als anwählbare Verbindungen benutzt werden
- ▶ Wie sie externe Bilder als Ersatz für oder zusätzlich zu internen Abbildungen einsetzen können
- ▶ Wie für Browser, die keine Abbildungen darstellen können, Alternativen dazu angeboten werden
- ▶ Bildabmessungen und Bildskalierung sowie Bildvorschau
- ▶ Wechsel der Schrift und der Hintergrundfarben in Ihren Webseiten
- ▶ Die Verwendung von »gekachelten« Seitenhintergründen
- ▶ Wie und wann Sie Abbildungen in Ihren Webseiten verwenden sollen

Nach diesem Kapitel werden Sie alles, was es über das Einfügen von Abbildungen in Webseiten zu wissen gibt, kennen.

Abbildungen im Web

Interne Abbildungen sind Abbildungen, die direkt auf Ihrer Webseite zwischen dem Text und den Verknüpfungen erscheinen und automatisch geladen werden, wenn Sie die Seite selbst laden – vorausgesetzt natürlich, dass Sie einen grafischen Browser haben und das automatische Laden interner Abbildungen eingeschaltet ist. *Externe* Abbildungen sind Abbildungen, die nicht direkt abgebildet werden, wenn Sie eine Seite laden. Sie werden nur auf Anforderung Ihrer Leser geladen, normalerweise stehen sie am anderen Ende einer Verknüpfung. Externe Abbildungen benötigen keinen grafikfähigen Browser, um betrachtet zu werden – Sie können eine Abbildung problemlos mit einem rein textbasierten Browser herunterladen und dann einen grafischen Editor oder Betrachter verwenden, um sich das Bild später anzuschauen. Sie werden noch später in diesem Kapitel lernen, wie Sie sowohl interne als auch externe Abbildungen einsetzen können.



Interne Abbildungen (Inline Images) erscheinen auf einer Webseite zusammen mit Text und Verknüpfungen und werden automatisch geladen, wenn die Seite selbst aufgerufen wird.



Externe Abbildungen (External Images) werden getrennt von der Webseite gespeichert und nur auf Anforderung geladen, zum Beispiel wenn eine Verknüpfung angeklickt wird.

Egal, ob Sie nun interne oder externe Abbildungen einsetzen, diese Bilder müssen in einem spezifischen Format gespeichert sein. Bei internen Abbildungen muss es eines dieser beiden Formate sein: GIF oder JPEG. GIF ist eigentlich der beliebtere Standard, und es gibt mehr Browser, die interne GIF-Dateien darstellen können, als es Browser gibt, die dies mit JPEG-Dateien tun. Die Unterstützung für JPEG verbreitet sich immer weiter, ist aber noch nicht so weitgehend wie für GIF, und deshalb ist es die beste Methode, sicherzugehen, dass Ihre Abbildungen von dem größtmöglichen Publikum betrachtet werden können, die sich auf GIF zu beschränken. Sie werden noch mehr über den Unterschied zwischen GIF und JPEG und wie man Abbildungen in diesen Formaten erstellt in Kapitel 8 lernen. Sie werden außerdem später in diesem Kapitel noch mehr über externe Abbildungen und die Formate, die Sie dafür einsetzen können, erfahren.

Lassen Sie uns für dieses Kapitel einfach mal annehmen, dass Sie bereits eine Abbildung haben, die Sie auf Ihre Webseite packen möchten. Wie bekommen Sie sie ins GIF- oder JPEG-Format, sodass die Abbildung auf Ihrer Seite dargestellt werden kann? Die meisten Grafikbearbeitungsprogramme wie Adobe Photoshop (<http://www.adobe.com>), Paint Shop Pro (<http://www.jasc.com/>), Corel Draw (<http://www.corel.com/>) bieten Möglichkeiten an, Ihr Bild in verschiedene Abbildungsformate umzuwandeln. Sie müssen da vielleicht unter der Option »SPEICHERN ALS« oder »EXPORTIEREN« nachsehen, um eine Möglichkeit zu finden. Freeware und Shareware-Programme, die *zwischen* Bildformaten nichts konvertieren sind auch für Plattformen erhältlich. Viel Shareware – und Demo-Versionen von Bildbearbeitungs-Software gibt es unter <http://www.download.com> (Suchen Sie *image editors* indem Sie die Software Plattformoptionen wählen).



Sie werden mehr über Bildbearbeitungsprogramme am Tag 8 lernen.

Wenn Sie Dateien im GIF-Format speichern wollen, suchen Sie nach einer Option, die CompuServe GIF, GIF87, GIF89 oder einfach nur GIF genannt wird. Jede von diesen wird ausreichen. Wenn Sie Ihre Dateien als JPEG speichern, wird die einzige Möglichkeit normalerweise einfach JPEG sein.

Erinnern Sie sich, dass Ihre HTML-Dateien immer eine `.html` oder `.htm`-Endung haben mussten, um richtig zu funktionieren? Auch Bilddateien haben Suffixe. Für GIF-Dateien gilt das Suffix `.gif`. Für JPEG-Dateien ist das Suffix entweder `.jpg` oder `.jpeg`; beide werden gleichermaßen gut funktionieren.



Einige Grafik-Editoren versuchen immer, alle Dateien mit großgeschriebenen Suffixen (`.GIF`, `.JPEG`) zu speichern. Obwohl dies korrekte Suffixe sind, sind Abbildungsamen wie auch HTML-Datei-Namen von der Groß- und Kleinschreibung abhängig, weshalb GIF nicht dasselbe Suffix ist wie `gif`. Die Schreibweise des Suffix ist nicht wichtig, wenn Sie es auf Ihrem lokalen System testen, wird aber entscheidend, wenn Sie Ihre Dateien auf den Server bewegen; deshalb sollten Sie, wann immer möglich, Kleinbuchstaben benutzen.

Interne Abbildungen in HTML: die `<img />`-Kennung

Nachdem Sie ein Bild im GIF- oder JPEG-Format fertig vorbereitet haben, können Sie es nun in Ihre Webseite einfügen. Interne Abbildungen werden in HTML durch die `<img />`-Kennung ausgezeichnet. Die `<img />`-Kennung hat wie die `<hr />`- und `<br />`-Kennungen keine abschließende HTML-Kennung. Letzter Satz im Absatz: Bei XHTML wird ein zusätzlicher Abstand gefolgt von einem Schrägstrich am Ende des Tag und seinen Attributen hinzugefügt.

Das `<img />`-Tag hat zahlreiche Attribute, die verschiedene Präsentations- und Verwendungsmethoden mit Inline Bildern ermöglichen. Viele dieser Attribute gehören zu HTML 3.2 oder HTML 4.0 und sind in einigen älteren Browsern nicht verfügbar. Auch andere Attribute gelten als verworfen, da sie von Style Sheets in den HTML 4.0- und den XHTML-Spezifikationen ersetzt wurden.

Das wichtigste Attribut der `<img />`-Kennung ist `SRC` (von Source = Quelle). Das `src`-Attribut bezeichnet in Gänsefüßchen den Datei-Namen der Abbildung, die Sie einfügen wollen. Der Pfad-Name der Datei folgt den gleichen Regeln wie die `href`-Attribute in den Verknüpfungen mit anderen Dokumenten. Sie können also für eine GIF-Datei namens `image.gif` im aktuellen Verzeichnis die folgende Kennung benutzen:

```

```

Für eine Abbildung in einem Verzeichnis über dem aktuellen benutzen Sie folgende Kennung:

```

```

Für eine Abbildung aus dem Verzeichnis oberhalb des aktuellen, schreiben Sie weiter nach den gleichen Regeln, wie für Dokument-Namen im `href`-Teil der `<a>`-Kennung.

Eine Alternative zu Abbildungen hinzufügen

Bilder können eine einfache Text-Webseite in eine lebendige, bunte Seite umwandeln. Aber was geschieht, wenn ein Besucher Ihre Webseite mit einem Browser betrachtet, der nur Text anzeigt? Was ist, wenn, der Ihr Besucher den Vorgang, der das Herunterladen von Bildern ermöglicht, ausgeschaltet hat und alle Ihre so mühsam erstellten Grafiken nur als einfache Icons erscheinen. Plötzlich sieht diese lebendig bunte Seite gar nicht mehr lebendig und bunt aus. Und wenn Sie beim Erstellen Ihrer Webseite diese Möglichkeit völlig außer Acht gelassen haben, wird Ihr Publikum Ihre Webseite nicht lesen oder verwenden können.

Zu diesen beiden Problemen gibt es eine ganz einfache Lösung:

Wenn Sie das `alt`-Attribut des `<img />`-Tag verwenden, können Sie für die Bilder, die der Browser als solche nicht erkennt, einen sinnvollen Inhalt eingeben.

In textorientierten Browsern, wie etwa Lynx, werden Grafiken, die das `<img />`-Tag in der Originaldatei verwenden, mit eckigen Klammern angezeigt. Zum Beispiel: `[IMAGE]`:

Das `alt`-Attribut im `<img />`-Tag liefert Ihren Besuchern, die Browser verwenden, die nur Text lesen können, oder die Grafik in ihrem Browser ausgeschaltet haben, eine sinnvollere Textalternative zum faden `[IMAGE]`. Das `alt`-Attribut enthält einen Anhang mit dem Text, der die Grafik ersetzen soll.

```

```

Die meisten Browser werden diesen Textanhang im `alt`-Attribut als literarischen Anhang interpretieren. Wenn Sie beliebige HTML-Tags in diesen Anhang eingeben, werden diese als HTML-Code interpretiert und angezeigt. Sie können deshalb als Ersatz für ein Bild keine ganzen HTML-Code-Blöcke verwenden – nur einige Wörter oder Sätze.

Ich behandle Alternativen zu Bildern mit gutem Grund schon zu Anfang dieses Kapitels. Alternativen zu Bildern sind Optionen in den älteren HTML-Versionen, aber sie sind obligatorisch in HTML 4.0 *Strict* und HTML 1.0-Spezifikationen.

Übung 7.1: Bilder hinzufügen

Lassen Sie uns ein einfaches Beispiel versuchen. Hier ist die Webseite für ein Spukhaus, in dem es jedes Jahr zu Halloween spukt. Mit Hilfe all der ausgezeichneten Hinweise, die ich Ihnen in den letzten sechs Kapiteln gegeben habe, sollten Sie in der Lage sein, ohne weiteres so eine Seite wie die folgende zu erstellen. Hier folgt der HTML-Code für diese Datei und Abbildung 7.1 zeigt, wie das aussieht:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">
<html>
<head>
<title>Welcome to the Halloween House of Terror</title>
</head>
<body>
<h1>Welcome to The Halloween House of Terror!!</h1>
<hr />
<p>Voted the most frightening haunted house three years in a row,
the <strong>Halloween House of Terror</strong> provides the
ultimate in Halloween thrills. Over <strong>20 rooms of thrills
and excitement</strong> to make your blood run cold and your hair
stand on end!</p>
<p>The Halloween House of Terror is open from <em>October 20 to
November 1st</em>, with a gala celebration on Halloween night.
Our hours are:</p>
<ul>
<li>Mon-Fri 5PM-midnight</li>
<li>Sat & Sun 5PM-3AM</li>
<li><strong>Halloween Night (31-Oct)</strong>: 3PM-??</li>
</ul>
<p>The Halloween House of Terror is located at:<br />
The Old Waterfall Shopping Center<br />
1020 Mirabella Ave<br />
Springfield, CA 94532</p>
</body>
</html>
```

So weit, so gut. Jetzt können Sie Ihrer Webseite ein Bild hinzufügen.

Stellen Sie sich vor, Sie hätten ganz zufällig ein Spukhaus auf Ihrem Festplattenlaufwerk; würde sich das nicht gut am Anfang Ihrer Webseite machen. Das Bild, das man house.jpg nennt, ist im gleichen Verzeichnis zu finden wie die halloween.html page. Es liegt sozusagen schon für Ihre Webseite bereit.

Stellen Sie sich nun vor, dass Sie dieses Bild Ihrer Seite hinzufügen möchten, und zwar so, dass es eine Zeile für sich beansprucht und die Überschrift unter dem Bild erscheint: Dafür fügen Sie ein `<img />`-Tag der Datei in einem eigenen Absatz, kurz vor der Überschrift ein. (Bilder wie Links definieren nicht ihre eigenen Textelemente. Deshalb muss das `<img />`-Tag innerhalb eines Absatz- oder Kopfzeilenelements erscheinen).

```
<p></p>
<h1>Welcome to The Halloween House of Terror!!</h1>
```


Wenn Sie jetzt die `halloween.html`-Seite öffnen, sollte Ihr Browser das Geisterhaus-Bild auf der Seite, wie in Abbildung 7.2, anzeigen.



Abbildung 7.1:
Die Halloween-
Haus-Homepage



Abbildung 7.2:
Die Halloween-
Haus-Homepage
mit einer Abbil-
dung des verwun-
schenen Hauses

Falls es nicht klappt (wenn Ihr Browser ein seltsames Bildsymbol anstelle des verwunschenen Hauses zeigt), überprüfen Sie, ob Sie den Datei-Namen in der HTML-Datei korrekt angegeben haben. Bilddatei-Namen unterscheiden zwischen Groß- und Kleinbuchstaben, deshalb müssen alle Groß- und Kleinbuchstaben übereinstimmen (mit dem tatsächlichen Namen der Datei, die geladen werden soll). Wenn es immer noch nicht funktioniert, prüfen Sie genau, ob es sich bei der Abbildung wirklich um eine GIF- oder JPEG-Datei handelt und ob sie mit dem richtigen Suffix ausgestattet ist.

Und schließlich stellen Sie sicher, dass Sie bei Ihrem Browser das Laden von Abbildungen auch eingeschaltet haben. (Diese Option nennt man in Netscape Grafiken automatisch laden und im Internet Explorer Bilder anzeigen).

Wenn ein Bild gut ist, dann wären zwei Bilder doch noch viel besser, finden Sie nicht? Versuchen Sie ein zweites `<img />`-Tag in der Nähe des ersten hinzuzufügen und sehen Sie, was passiert:

```
<p>
</p>
<h1>Welcome to The Halloween House of Terror!!</h1>
```

Die Abbildung 7.3 zeigt das Ergebnis im Internet Explorer.



Abbildung 7.3:
Mehrere Abbil-
dungen des
verwunschenen
Hauses

Und das ist es auch schon! Unabhängig davon, wie klein oder groß das Bild ist, Sie fügen es auf die gleiche Weise in Ihre Webseite ein.

Abbildungen und Text

Die vorige Übung zeigte, wie eine interne Abbildung in einem eigenen Absatz mit Text darunter in eine Seite eingefügt wird. Sie können ebenso Abbildungen neben (oder in) eine Textzeile einfügen. (Genaugenommen kommt der Ausdruck »inline image« genau daher.)

Um eine Abbildung in eine Textzeile zu integrieren, fügen Sie sie einfach innerhalb des Textes ein, zwischen die Element-Tags (<h1>, <p>, <address> und so weiter). Abbildung 7.4 zeigt den Unterschied, wenn die Abbildung in eine Textzeile eingefügt wird. (Ich habe den Titel etwas gekürzt.)

```
<h2>The Halloween House of Terror!!</h2>
```

Die Abbildung sollte nicht allzu groß sein, und sie muss auch nicht am Anfang des Textes stehen. Sie können eine Abbildung irgendwo in einem Textblock einsetzen:



Abbildung 7.4:
Die Halloween-
Haus-Homepage
mit Abbildung
inner der Über-
schrift



```
<blockquote>
Love, from whom the world  begun,<br />
Hath the secret of the sun.  <br />
Love can tell, and love alone,
Whence the million stars  were strewn <br />
Why each atom  knows its own. <br />
--Robert Bridges
</blockquote>
```

Abbildung 7.5 zeigt, wie dies aussieht.

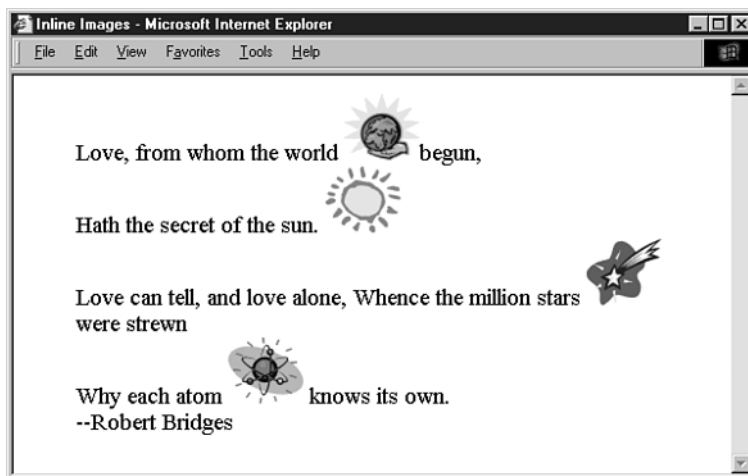


Abbildung 7.5:
Abbildungen passen überall in den Text

Ausrichtung von Text und Abbildungen

Beachten Sie, dass bei diesen Beispielen die Abbildung so dargestellt wurde, dass der untere Rand des Bildes und der untere Rand des Textes nebeneinander lagen. Die `<img />`-Kennung unterstützt auch das `align`-Attribut, das es Ihnen ermöglicht, die Abbildung oben oder unten mit dem umgebenden Text oder anderen Bildern auf der Zeile auszurichten.



Das `align`-Attribut für das `<img />`-Tag wurde in HTML 4.0 zu Gunsten der Style-Sheet-Attribute verworfen. Sie werden am Tag 10 mehr über Style Sheets lernen.

Standard-HTML 2.0 definiert drei grundlegende Werte für `align`:

- ▶ `align="top"` Richtet den oberen Rand der Abbildung mit dem obersten Teil der Zeile aus (was entweder die obere Kante des Textes oder die einer anderen Abbildung sein kann).
- ▶ `align="middle"` Richtet das Zentrum der Abbildung mit der Mitte der Zeile aus (meist die Grundlinie der Textzeile, nicht die tatsächliche Mitte der Zeile).
- ▶ `align="bottom"` Richtet den unteren Teil der Abbildung mit dem unteren Teil der Textzeile aus.

HTML 3.2 bietet zwei weitere Werte: `left` und `right`. Diese Werte werden im nächsten Abschnitt, »Text neben Bildern darstellen«, behandelt.

Abbildung 7.6 zeigt das Gedicht von Robert Bridges von vorhin. Das Bild der Erdkugel wurde außerhalb einer Textzeile ausgerichtet, das Sonnenbild wurde zum Anfang des Textes, das Sternbild in die Mitte, und das Atombild wurden ans Ende des Textes jeweils in eine Zeile gesetzt.



```
<blockquote>
Love, from whom the world
begun,<br />
Hath the secret of the sun.
<br />
Love can tell, and love alone, Whence the million stars

were strewn<br />
Why each atom

knows its own.<br />
</blockquote>
```

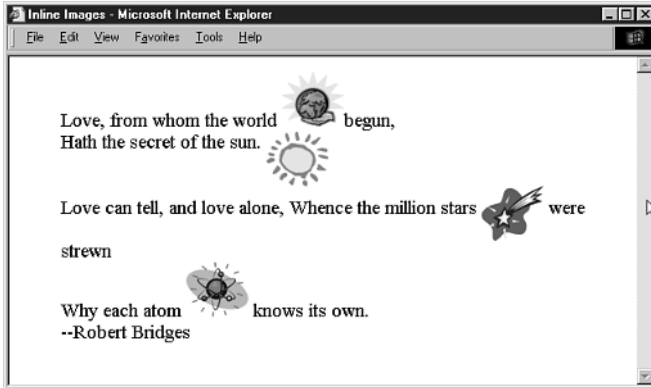


Abbildung 7.6:
»Auf der Seite befinden sich oben, in der Mitte und unten ausgerichtete Bilder in der Textzeile sowie ein Bild, das außerhalb des Textes ausgerichtet wurde.

Zusätzlich zu den vorhergehenden Werten bieten verschiedene nicht-Standard-Werte für align bessere Kontrolle und Präzision bei der Ausrichtung des Bildes in eine Textzeile. Die folgenden Werte werden alle vom Netscape Navigator (und manchmal auch vom Internet Explorer) unterstützt. Sie sind aber nicht Teil von HTML 3.2 oder HTML 4.0 Die folgenden vier Attribute werden in den angezeigten Spezifikationen von XHTML 1.0 nicht unterstützt. Wenn Sie diese Attribute bei XHTML 1.0 benutzen, werden Ihre Webseiten nicht besonders attraktiv aussehen:

align="texttop"	Richtet den oberen Rand des Bildes bündig am größten Textelement in der Zeile aus (wohingegen align="top" das Bild am größten Element in der Zeile ausrichtet).
align="absmiddle"	Richtet die Mitte des Bildes an der Mitte des größten Elements der Zeile aus. (align="middle" richtet normalerweise die Mitte des Bildes mit der Grundlinie des Textes aus, nicht mit der Mitte)
align="baseline"	Richtet den unteren Rand des Bildes an der Grundlinie des Textes aus. align="baseline" ist dasselbe wie align="bottom", aber align="baseline" ist als Bezeichnung beschreibender.
align="absbottom"	Richtet das untere Ende des Bildes am tiefsten gelegenen Element einer Zeile aus (was unter der Grundlinie des Textes liegen kann).

Die folgenden Code-Beispiele zeigen diese Ausrichtungsoptionen in einem praktischen Beispiel. In Abbildung 7.7 sehen Sie Beispiele von allen Optionen, wie sie im Netscape Navigator erscheinen. In allen Fällen ist die Zeile der linken Seite und der Text zusammen ausgerichtet, und die Pfeilposition variiert.



```

<h2>Middle of Text and Line aligned, arrow varies:</h2>

Align: Top

Align: Text Top

<h2>Top of Text and Line aligned, arrow varies:</h2>

Align: Absolute Middle

Align: Middle

<h2>Top of Text and Line aligned, arrow varies:</h2>

Align: Baseline / Bottom

Align: Absolute Bottom


```

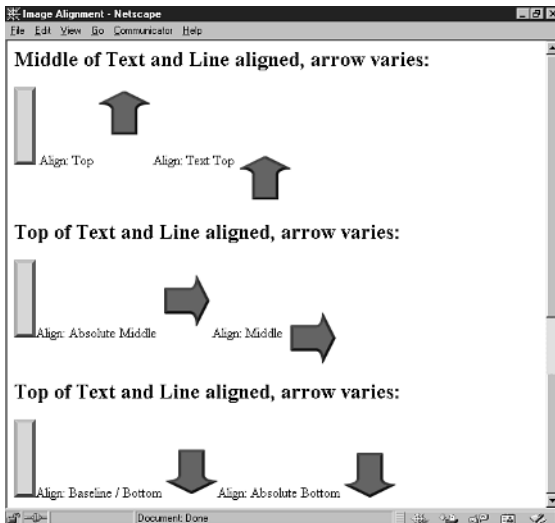


Abbildung 7.7:
Ausrichtungsoptionen im Netscape
Navigator

Text neben Bildern darstellen

Eine Abbildung innerhalb einer Textzeile einzufügen ist kein Problem, solange Sie nur eine einzige Zeile Text haben. Ein Aspekt von internen Bildern, den ich bis jetzt verschwiegen habe, ist, dass dies alles in HTML 2.0 nur mit einer einzelnen Textzeile funktioniert. Wenn Sie mehrere Textzeilen haben und eine Abbildung in deren Mitte einfügen wollen, wird der Text um das Bild (mit Ausnahme einer einzelnen Zeile) ober- und unterhalb der Abbildung erscheinen – siehe Abbildung 7.8.

Was ist nun aber, wenn Sie mehrere Textzeilen um ein Bild wickeln wollen, sodass der Text das Bild von allen Seiten umschließt? Mit HTML 2.0 können Sie das nicht tun. Sie sind auf eine einzelne Zeile Text auf einer der Seiten des Bildes beschränkt, was ihre Gestaltungsmöglichkeiten einschränkt.

Um diese Beschränkung von HTML 2.0 zu umgehen, definierte Netscape zwei neue Werte für das `align`-Attribut der `<img />`-Kennung – `left` und `right`. Diese beiden Werte wurden in HTML 3.2 aufgenommen und werden inzwischen von vielen anderen Browsern unterstützt.

`align="left"` und `align="right"`

`align="left"` richtet ein Bild zum linken Seitenrand hin aus und `align="right"` zum rechten. Aber die Anwendung dieser Attribute verursacht ebenfalls, dass der der Abbildung folgende Text im Raum auf der linken oder rechten Seite der Abbildung dargestellt wird, je nach der gewünschten Ausrichtung. Abbildung 7.8 zeigt eine Abbildung mit etwas Text, der daneben ausgerichtet ist.



```

<h1>Mystery Tulip Murderer Strikes</h1>
<p>Someone, or something, is killing the tulips of New South
Haverford, Virginia. Residents of this small town are shocked and
dismayed by the senseless vandalism that has struck their tiny
town.</p>
<p>New South Haverford is known for its extravagant displays of
tulips in the springtime, and a good portion of its tourist trade
relies on the people who come from as far as New Hampshire to see
what has been estimated as up to two hundred thousand tulips that
bloom in April and May.</p>
<p>Or at least the tourists had been flocking to New South
```


Haverford until last week, when over the course of three days the flower of each and every tulip in the town was neatly clipped off while the town slept.</p>

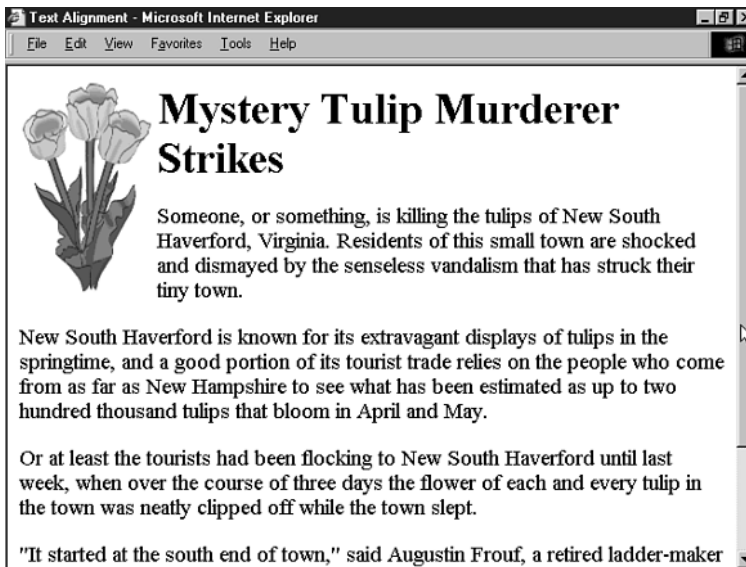


Abbildung 7.8:
Text und Abbil-
dung sind ausge-
richtet

Sie können beliebigen HTML-Text (Absätze, Überschriften, Listen, andere Abbildungen) nach einem ausgerichteten Bild einfügen, wobei der Text in den Raum zwischen dem Bild und dem gegenüber liegenden Seitenrand hineingepackt wird (oder Sie können sogar Abbildungen an beiden Seitenrändern haben und den Text in die Mitte setzen). Der Browser füllt den Raum mit Text, bis er das untere Ende der Abbildung erreicht, und fährt dann fort, den Text unterhalb des Bildes einzufügen.

Anhalten des Textflusses

Was können Sie tun, wenn Sie den Raum neben der Abbildung nicht weiterhin auffüllen, sondern eine neue Zeile unterhalb der Abbildung beginnen wollen? Ein normaler Zeilenumbruch wird da nicht genügen – er wird die Zeile nur um einen Zeilenabstand weiterschieben. Auch eine neue Absatzmarke wird den Text weiterhin um das Bild herumfließen lassen. Um das Herumfließen des Textes um die Abbildung endgültig zu

unterbrechen, setzen Sie eine Zeilenumbruch-Kennung (`<br />`) mit dem nennen Attribut `clear` (löschen) ein. Mit dem `clear`-Attribut können Sie die Zeile so umbrechen, dass die nächste Textzeile erst nach dem Ende der Abbildung beginnt (und zwar geht sie dann zum Seitenrand unterhalb der Abbildung).

Das `clear`-Attribut kann einen dieser drei Werte annehmen:

- ▶ **Left**
Umbruch zum freien linken Rand bei allen linksbündig ausgerichteten Abbildungen
- ▶ **Right**
Umbruch zum freien rechten Rand bei allen rechtsbündig ausgerichteten Abbildungen
- ▶ **All**
Umbruch zu einer Zeile, die auf beiden Seiten frei ist



Das `clear`-Attribut für das `<br />`-Tag gilt in HTML 4.0 als verworfen und wurde von Style-Sheet-Attributen ersetzt.

Das folgende Code-Beispiel zeigt die Abbildung einer Tulpe, um die etwas Text gelegt wird. Ein Zeilenumbruch mit `clear="left"` bricht den Textfluss um die Grafik ab und führt den Text nach der Abbildung fort. Abbildung 7.9 zeigt das Ergebnis im Internet Explorer.



```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
</head>
<body>

<h1>Mystery Tulip Murderer Strikes</h1>
<br clear="left" />
<p>Someone, or something, is killing the tulips of New South
Haverford, Virginia. Residents of this small town are shocked and
dismayed by the senseless vandalism that has struck their tiny
town.</p>
<p>New South Haverford ist known for its extravagant displays of
tulips in the springtime, and a good portion of its tourist trade
relies on the people who come from as far as New Hampshire to see
what has been estimated as up to two hundred thousand tulips that
```

bloom in April and May.</p>

<p>Or at least the tourists had been flocking to New South Haverford until last week, when over the course of three days the flower of each and every tulip in the town was neatly clipped off while the town slept.</p>

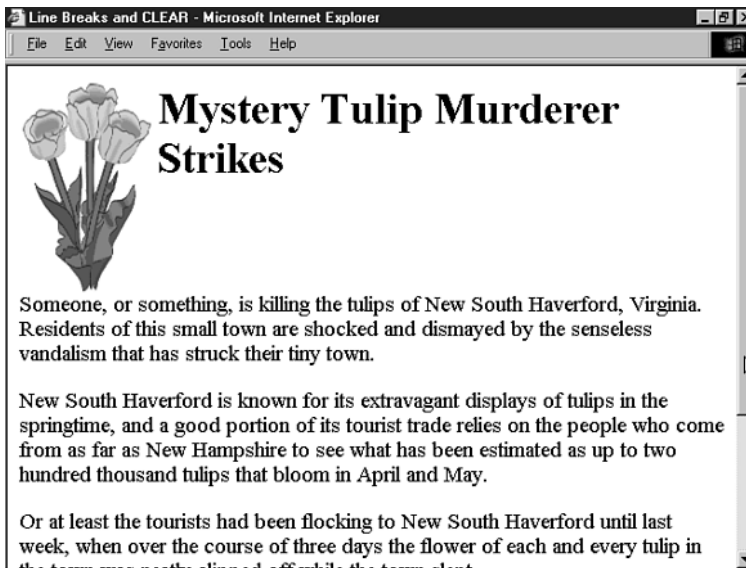


Abbildung 7.9:
Zeilenumbruch
zum freien Sei-
tenrand unter-
halb der Grafik

Den Abstand um die Abbildungen einstellen

Mit der Fähigkeit, Text um ein Bild herumfließen zu lassen, möchten Sie vielleicht auch den Abstand um das Bild einstellen. Die `vspace`- und `hspace`-Attribute ermöglichen Ihnen genau das (beide sind HTML-3.2-Merkmale). Beide können als Pixel-Wert angegeben werden; `vspace` (von vertical space, vertikaler Abstand) kontrolliert den Abstand über und unter der Abbildung, und `hspace` (von horizontal space, horizontaler Abstand) kontrolliert den Abstand nach links und rechts.



Die `vspace` und `hspace` Attribute für das `<img />`-Tag sind in HTML 4.0 zugunsten der Style Sheets verworfen.

Der folgende HTML-Code, der in Abbildung 7.10 angezeigt wird, illustriert zwei Beispiel: das oberste Beispiel zeigt die standardhorizontalen und -vertikalen Abstände um das Bild herum, während das untere Beispiel das Ergebnis zeigt, das mit Hilfe von `hspace`- und `vspace`-Attributen erzielt wurde. In beiden Bildern wurde das `align="left"`-Attribut verwendet, sodass automatisch ein Zeilenumbruch auf der linken Seite des Textes neben dem Bild erfolgt. Im unteren Beispiel jedoch wird der Text durch den zusätzlichen Abstand oberhalb des Bildes angezeigt (der zusätzliche Abstand wurde mit Hilfe des `vspace`-Attributs erzielt).



```

<p>This is an eggplant. We intend to stay a good ways away from
it, because we really don't like eggplant very much.</p>
<br clear="left" />
<hr />

<p>This is an eggplant. We intend to stay a good ways away from
it, because we really don't like eggplant very much.</p>
```

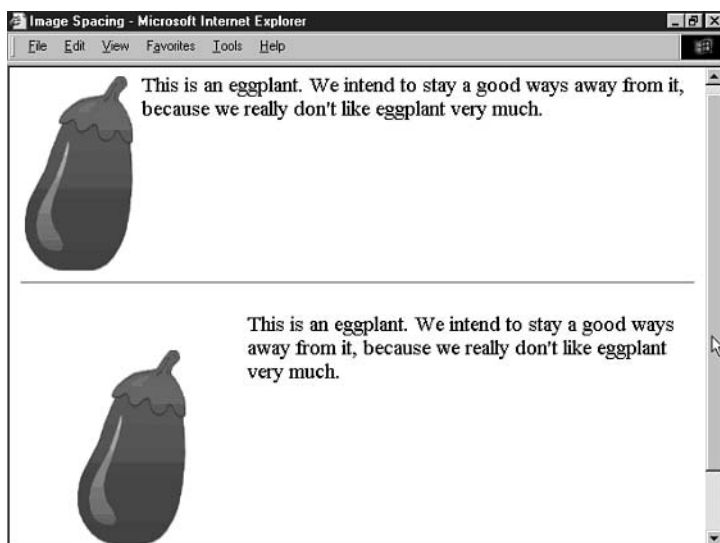


Abbildung 7.10:
Oberes Beispiel mit
Bildausrichtung;
unteres Beispiel
ohne Bildausrich-
tung

Abbildungen und Verknüpfungen

Kann eine Abbildung als Verknüpfung dienen? Aber natürlich! Wenn Sie eine `<img />`-Kennung zwischen den einleitenden und abschließenden Teil eines Verknüpfungstags (`<a>`) schreiben, so dient die Abbildung für die Verknüpfung selbst als anwählbare Stelle auf der Seite:

```
<a href="index.html">Up to Index</a>
```

Wenn Sie sowohl eine Abbildung als auch Text in den Anker setzen, so können sowohl die Abbildung als auch der Text angewählt werden, um die Verknüpfung zu aktivieren:

```
<a href="index.html">Up to Index</a>
```

Abbildungen, die zugleich auch Auswahlpunkte für eine Verknüpfung sind, werden in HTML 2.0 mit einem Rahmen darum dargestellt, um sie von normalen, nichtanwählbaren Abbildungen zu unterscheiden

In Abbildung 7.11 sehen Sie ein entsprechendes Beispiel. Das Schmetterlingsbild ist ein Bild, das sich nicht anklicken lässt, also gibt es auch keinen Rand um dieses Bild.

Sie können die Dicke der Bildumrandung mit dem `border`-Attribut der `<img />`-Kennung bestimmen. Das `border`-Attribut, das eine Netscape-Erweiterung ist, wurde in HTML 3.2 aufgenommen, aber es wurde bei HTML 4.0 zu Gunsten von Style Sheets verworfen. Dieses Attribut wird mit einer Zahl angegeben, die den Durchmesser der Bildumrandung in Pixeln festlegt. `border=0` versteckt die Umrandung völlig.

Seien Sie jedoch vorsichtig, wenn Sie `border` auf 0 (Null) setzen, wenn eine Abbildung als Verknüpfung dient. Die Umrandung gibt einen visuellen Aufschluss darüber, dass die Abbildung auch als Verknüpfung dient. Wenn Sie diese Umrandung entfernen, erschweren Sie es dem Leser, herauszufinden, welches denn nun normale Abbildungen sind und welche als Ausgangspunkt für Verknüpfungen dienen, ohne dass er die Maus bewegt, um es herauszubekommen. Vergewissern Sie sich, wenn Sie schon unbedingt die Umrandung entfernen wollen, dass Ihr Design einen Hinweis darauf gibt, dass es sich um ein anklickbares und nicht nur um ein »normales« Bild handelt. Sie können Ihre Bilder zum Beispiel so gestalten, dass sie tatsächlich wie Buttons aussehen (siehe Abbildung 7.12).

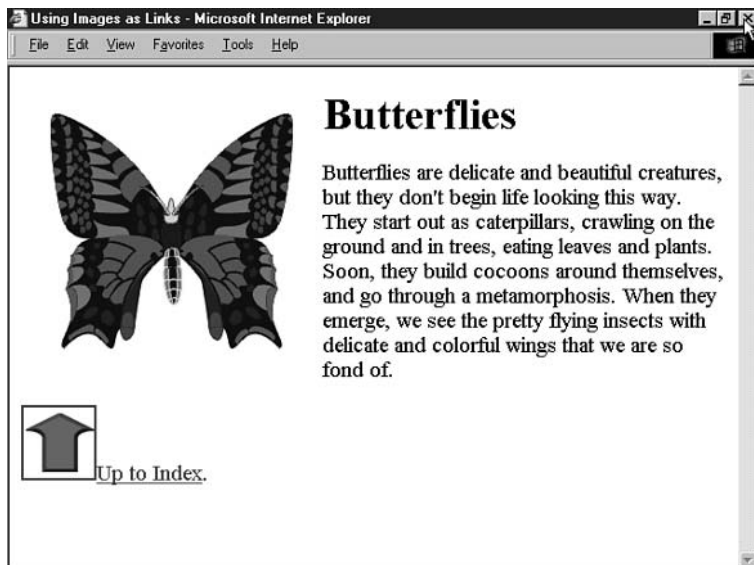


Abbildung 7.11:
Eine Abbildung,
die als Verknüpfung
dient

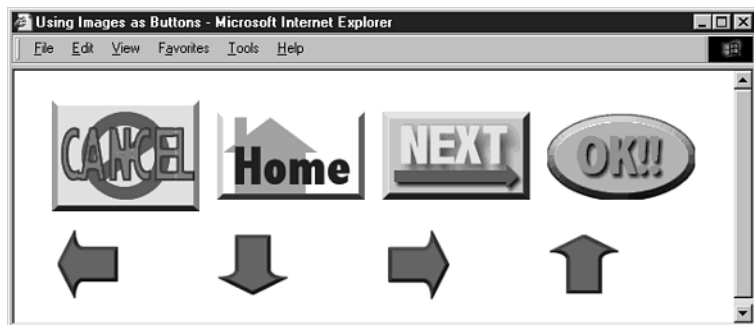


Abbildung 7.12:
Abbildungen, die
wie Buttons aus-
sehen

Übung 7.2: Navigationssymbole

Lassen Sie uns ein einfaches Beispiel für die Verwendung einer Abbildung als Verknüpfungspunkt konstruieren. Wenn Sie mehrere zusammengehörige Webseiten haben, zwischen denen man sich in gleichmäßiger Weise bewegen kann (beispielsweise vorwärts, rückwärts, zurück, hoch, an den Anfang und so weiter), ist es sinnvoll, ein Menü mit solchen Bewegungsoptionen am Anfang oder am Ende jeder Seite anzubieten, damit Ihre Leser genau wissen, wie sie ihren Weg durch Ihre Dokumente finden können.

Dieses Beispiel zeigt, wie Sie Symbole (*Icons*) verwenden, die zum Navigieren durch eine Gruppe linear zusammengehöriger Seiten benutzt werden können. Sie haben drei Bildchen im GIF-Format: eines für vorwärts, eines für zurück, und ein drittes erlaubt Ihren Lesern, in den Hauptindex der gesamten Dokumentenstruktur zurückzuspringen.

Zunächst schreiben wir die HTML-Struktur, um diese Bildchen (*Icons*) verwenden zu können. Hier ist die Seite selbst nicht sonderlich wichtig, sodass wir nur den notwendigen Rahmen erzeugen. Abbildung 7.13 zeigt, wie das Dokument anfangs aussieht.



```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>
<head>
<title>Motorcycle Maintenance: Removing Spark Plugs</title>
<h1>Removing Spark Plugs</h1>
<p>(include some info about spark plugs here)</p>
<hr />
</body>
</html>
```

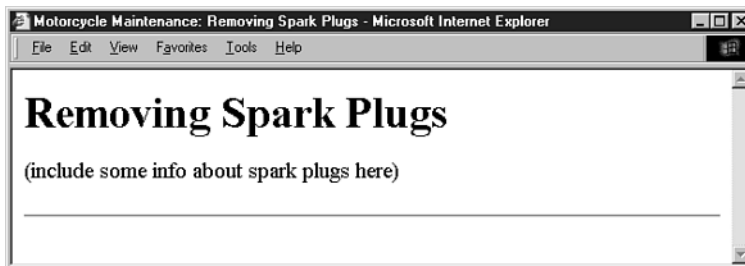


Abbildung 7.13:
Das Grunddokument ohne Icons

Nun ergänzen Sie am Ende des Dokuments mit Hilfe der `<img />`-Kennung die drei Abbildungen (Abbildung 7.14 zeigt das Ergebnis):



```



```

Ergänzen Sie nun die Anker um die Abbildungen, damit diese aktivierbar werden. Abbildung 7.15 zeigt das Ergebnis.

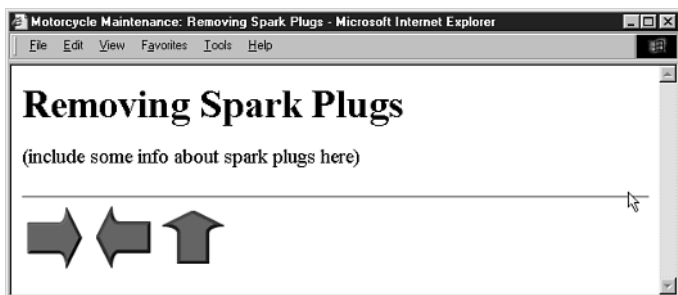


Abbildung 7.14:
Das Grunddokument mit
Icons



```
<a href="replacing.html"></a>
<a href="ready.html"></a>
<a href="index.html"></a>
```



Abbildung 7.15:
Das Grunddokument mit
Icon-Verknüpfungen

Wenn Sie nun eines der Icons anwählen, so springt der Browser zu der im Verbindungs-Tag angegebenen Seite, gerade so, als hätten Sie normalen Verknüpfungstext dafür benutzt.

Wo wir gerade bei Text sind: Reichen die Icons, so wie sie sind, aus? Wie wäre es, etwas Text hinzuzufügen, um zu beschreiben, was einen auf der anderen Seite der Verknüpfung erwartet? Sie können den Text sowohl innerhalb als auch außerhalb der Anker platzieren, je nachdem, ob er auch zum Anwählen der Verbindung benutzt werden soll oder nicht. In unserem Beispiel schreiben wir ihn außerhalb der Anker, sodass nur die Icons angewählt werden können. Zudem richten wir den unteren Teil des Textes und der Abbildungen mit Hilfe des `align`-Attributs der `<img />`-Kennung aus. Schließlich arrangieren wir alles so, dass jedes Element in einer eigenen Zeile steht, da der zusätzliche Text sonst die Icons irgendwie auf mehrere Zeilen verteilen würde. Abbildung 7.16 zeigt das fertige Menü.



```
<hr />
<p><a href="replacing.html"></a>
On to "Gapping the New Plugs"<br />
<a href="ready.html"></a>
Back to "When You Should Replace your Spark Plugs"<br />
<a href="index.html"></a>
Up To Index
</p>
```



Abbildung 7.16:
Das Grunddokument mit Icon-Verknüpfungen und Text

Externe Abbildungen einsetzen

Im Unterschied zu internen Abbildungen erscheinen externe Abbildungen nicht direkt auf Ihrer Webseite, sondern werden separat gespeichert und mit der Seite so verknüpft, wie auch andere Seiten verknüpft werden.

Der Grund, warum es sich lohnt, externe Bilder in diesem Kapitel zu erwähnen, ist der, dass externe Abbildungen oft eine komplementäre Rolle zu internen Abbildungen spielen können. Zum Beispiel:

- ▶ Die meisten Web-Browser unterstützen interne GIF-Bilder, und viele unterstützen ebenfalls interne JPEG-Bilder. Gleichzeitig unterstützen die meisten Browser jedoch eine viel größere Auswahl von Bildformaten durch die Verwendung externer Bilddateien und Hilfsprogramme. Sie können also durch die Verwendung externer Bilder viele andere Bildformate neben GIF und JPEG verwenden, zum Beispiel BMP (Windows bitmaps) oder PICT (Macintosh bitmaps).
- ▶ Rein textbasierte Browser können keine internen Bilder auf Webseiten abbilden, Sie können jedoch mit ihnen externe Bilder herunterladen und sie dann mit einem Bildbearbeitungs- oder Betrachtungsprogramm anschauen.
- ▶ Sie können ein kleines, internes Bild auf Ihrer Webseite, das schnell heruntergeladen werden kann, mit einem größeren, detaillierten externen Bild kombinieren, sodass die Leser, die mehr sehen möchten, das größere Bild selbst herunterladen können.
- ▶ Um externe Bilder zu verwenden, erstellen Sie das Bild genauso, wie Sie es bei einem internen tätigen, und speichern es dann mit dem entsprechenden Dateinamen. So wie bei anderen Dateien im Web, ist auch hier das Datei-Suffix wichtig. Abhängig vom Bildformat, benutzen Sie eine der in Tabelle 7.1 aufgezählten Suffixe.

Format	Endung
GIF	.gif
JPEG	.jpg, .jpeg
XBM	.xbm
TIFF	.tiff, .tif
BMP	.bmp
PICT	.pict

Tabelle 7.1: Bild-Formate und Datei-Endungen

Haben Sie schließlich ein externes Bild, so müssen Sie nur noch mit einer Verknüpfung zu ihm hinführen, genauso wie Sie auch zu einer anderen HTML-Seite verknüpfen würden:

```
<p> Letztes Jahr zuechtete ich einige wirklich riesige <a href="grosse tomaten.jpeg">
Tomaten </a> in meinem Garten. </p>
```

Im nächsten Beispiel werden Sie interne und externe Abbildungen zusammen einsetzen.

Übung 7.3: Verbindungen mit externen GIF- und JPEG-Dateien

Eine auf Webseiten übliche Praxis ist das Anbieten sehr kleiner GIF-Bilder (ein »Dau-mennagel« oder *thumbnail*) auf der Seite selbst; dieses Bildchen wird dann mit seinem größeren Gegenstück verknüpft. Dies bietet zwei wesentliche Vorzüge gegenüber dem internen Einbinden der gesamten Abbildung:

- ▶ Es hält die Webseite klein, sodass sie schnell geladen werden kann.
- ▶ Es vermittelt Ihren Lesern einen Eindruck von dem Bild, bevor sie die große Version laden.

In diesem Beispiel schaffen Sie eine Verbindung zwischen einem kleinen Bild und einer externen größeren Version des gleichen Bildes. Die große Abbildung ist eine Fotografie eines Schlosses an einem Flusssufer gelegen, genannt `pcastle.jpg` (dargestellt in Abbildung 7.17).



Abbildung 7.17:
Das Schloss-Bild

Zunächst erzeugen Sie mit Ihrem Bildverarbeitungsprogramm eine »Thumbnail«-Version der Schloss-Fotografie. Der »Thumbnail« kann eine verkleinerte Version der Originaldatei sein, ein Ausschnitt aus dem Bild (sagen wir, einer der Pinguine aus der Gruppe) oder irgendetwas anderes, mit dem Sie auf das größere Bild hinweisen wol-

len. Paint Shop Pro ist ein gutes Shareware-Programm für Windows. Während Adobe Photoshop auf Macintosh oder Windows PCs eher ein Standardprogramm für professionelle Designer ist. Das Minibild kann eine kleine Version der Originaldatei, ein Ausschnitt der Datei (zum Beispiel nur das Schloss anstelle des ganzen Bildes) oder etwas anderes sein, mit dem Sie das ganze Bild ankündigen möchte.

```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd"><html>
<head>
<title>Castle at Sunrise</title>
</head>
<body>
<h1>Castle at Sunrise</h1>

</body>
</html>
```

Nun können Sie mit dem Verknüpfungs-Tag das kleine Symbol mit dem größeren Bild verbinden, indem Sie die `<img />`-Kennung mit dem `<a>`-Tag umgeben:

```
<a href="castle.jpg"></a>
```

Das Resultat ist in Abbildung 7.18 dargestellt. Wenn Sie jetzt die kleine Schloss-Abbildung anklicken, wird das große Bild geladen und entweder vom Browser selbst oder von dem Hilfsprogramm angezeigt, das im Browser für JPEG-Dateien angegeben ist.



Abbildung 7.18:
Die Schloss-
Homepage mit
Verknüpfung

Eine Alternative zur direkten Verknüpfung der kleinen Abbildung mit dem größeren Bild besteht darin, die externen Abbildungen in verschiedenen Formaten anzubieten und dann verschiedene textbasierte Verknüpfungen zu den unterschiedlichen externen Versionen herzustellen (Sie können dies für Leser tun, die die Software nur für ein bestimmtes Format, nicht aber für ein anderes besitzen). An dieser Stelle unseres Beispiels füge ich eine Verbindung mit einer GIF-Version der gleichen Schloss-Datei hinzu.

Um eine GIF-Version der Pinguin-Fotografie zu erzeugen, benötigen Sie Ihr Bildverarbeitungsprogramm oder Ihren Bild-Konverter nochmals, um damit die ursprüngliche Fotografie zu konvertieren. Hier habe ich es `castle.gif` genannt.

Um nun sowohl GIF- als auch JPEG-Versionen der Pinguin-Fotografie anzubieten, verwandeln wir die Verknüpfung der Abbildung in ein einfaches Verbindungs-Menü mit den GIF- und JPEG-Dateien, und geben zusätzliche Angaben zur Dateigröße ein (das Ergebnis wird in Abbildung 7.19 gezeigt):



```
<p></p>
<ul>
<li>Castle at Sunrise (<a href="castle.jpg">25K JPEG
file</a></li>
<li>Castle at Sunrise (<a href="castle.gif">49K GIF
file</a></li>
</ul>
```

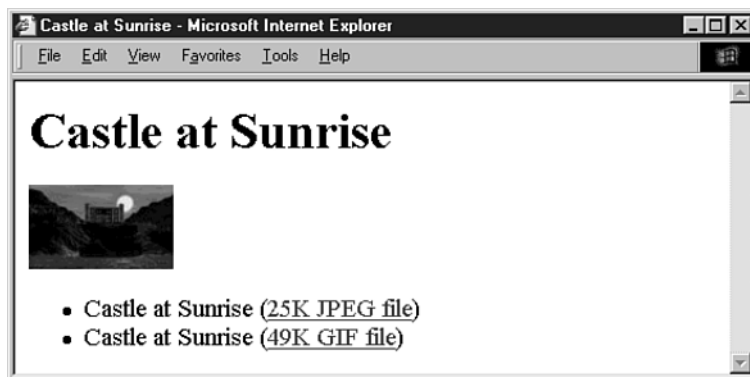


Abbildung 7.19:
Das Schloss mit
Verknüpfungs-
menü



Abbildungen sind nicht die einzigen Dateitypen, die Sie außerhalb Ihrer Webseite speichern können. Sounddateien, Video, Zip-Archive – so gut wie alles kann als externe Datei verknüpft werden. Sie werden mehr darüber in Kapitel 13 erfahren.

Andere nette Tricks mit Bildern

Jetzt, wo Sie etwas über interne und externe Bilder wissen sowie über Bilder, die als Verknüpfungen dienen und etwas darüber gelernt haben, wie man Text um Bilder herumfließen lässt, wissen Sie darüber Bescheid, was die meisten Leute mit Bildern auf Webseiten anfangen – und Ihnen ist alles darüber bekannt, was HTML 2.0 mit Bildern anfangen kann. Es gibt aber einige neuere Tricks, mit denen man spielen kann, und davon handelt dieser Abschnitt.

Alle Attribute in diesem Abschnitt waren ursprünglich Netscape-Erweiterungen, die jetzt in HTML 3.2 aufgenommen wurden, aber die meisten wurden von HTML 4.0 verworfen.

Größe und Skalierung von Bildern

Zwei Attribute des `<img />`-Tags, `height` und `width`, legen Höhe und Breite des Bildes in Pixeln fest. Beide Attribute sind Teil der HTML-Spezifikation, aber sie wurden von HTML 4.0 zu Gunsten der Style Sheets verworfen.

Wenn Sie für diese Werte die tatsächliche Höhe und Breite des Bildes angeben (die Sie mit Hilfe der meisten Bildverarbeitungsprogramme ermitteln können), benötigen einige Browser für die Formatierung und die Anzeige von Text und Bildern auf Ihrer Seite sehr viel weniger Zeit, als wenn Sie die Werte nicht angeben.

Warum? Wenn ein Browser den HTML-Code in Ihrer Datei übersetzt, muss er alle Bilder laden und ihre Breite und Höhe ermitteln, bevor er den Text entsprechend formatieren kann. Dies bedeutet meistens, dass er einen Teil des Textes lädt und formatiert, dann auf das Laden des Bildes wartet, dann um das Bild herum formatiert, wenn er die Ausmaße herausgefunden hat, und dann mit dem Rest der Seite fortfährt.



Wenn Sie Ihre Seite mit Bildern darin in Netscape 2.0 oder 3.0 testen, wählen Sie doch mal DOKUMENTEN-INFO aus dem Menü ANSICHT. Sie werden ein Fenster bekommen, das alle Bilder auf Ihrer Seite anzeigt. Wenn Sie dann jedes Bild einzeln auswählen, werden Sie Informationen darüber erhalten – einschließlich seiner Größe, die Sie dann in Ihre HTML-Datei einfügen können.

Wenn sich die Werte für `width` und `height` von der tatsächlichen Breite und Höhe des Bildes unterscheiden, skaliert der Browser es so, dass es in den vorgegebenen Leerraum passt. Da kleinere Bilder weniger Speicherplatz belegen als größere und damit in kürzerer Zeit über das Netzwerk übertragen werden können, ist das eine ganz elegante Möglichkeit, große Bilder auf Ihren Seiten loszuwerden. Sie erzeugen einfach eine kleinere Ausgabe des Bildes und skalieren es dann auf Ihrer Webseite entsprechend.

Beachten Sie jedoch, dass auch die Pixel skaliert werden, sodass die schließlich wieder vergrößerte Version eine gröbere Körnung aufweist. Experimentieren Sie mit den verschiedenen Größen und den Skalierungsfaktoren, um den richtigen Effekt zu erzielen.



Führen Sie keine umgekehrte Skalierung durch – indem Sie ein größeres Bild erzeugen und es dann mit `width` und `height` herunterskalieren. Kleinere Dateigrößen sind besser, da sie weniger Zeit zum Laden brauchen. Wenn Sie eh nur ein kleines Bild darstellen wollen, machen Sie es von Anfang an kleiner.

Mehr über Bildumrandungen

Sie haben das `border`-Attribut der `<img />`-Kennung als Teil des Abschnitts über Verknüpfungen kennen gelernt, wo die Festlegung von `border` auf einen Zahlenwert oder auf Null die Dicke der Bildumrandung festlegte (oder diese völlig versteckte).

Normalerweise haben einfache Bilder keine Umrandungen; nur solche mit Verknüpfungen. Trotzdem können Sie das `border`-Attribut mit einfachen Bildern verwenden, um eine Umrandung um das Bild zu ziehen, so wie hier:

```
<p>  
This is an eggplant. We intend to stay a good ways away from it,  
because we really don't like eggplant very much.</p>
```

Abbildung 7.20 zeigt ein Beispiel eines Bildes mit einer Umrandung.

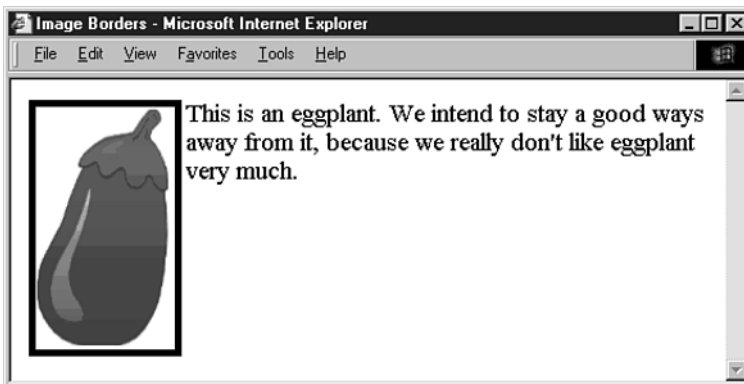


Abbildung 7.20:
Eine Bildumrandung

Bildvorschau

Eine optionale Netscape-Erweiterung für Abbildungen ist die Anwendung des `lowsrc`-Attributs (von `low source`, Quelle mit geringerer Auflösung) zu `<img />`, welches eine Art von Vorschau auf das tatsächliche Bild auf der Seite ermöglicht. `lowsrc` wird so wie `src`, mit einem Pfad-Namen zu einer anderen Bilddatei eingesetzt:

```

```

Wenn ein Browser, der `lowsrc` unterstützt, diese Kennung vorfindet, lädt er das `lowsrc`-Bild zuerst im ersten Durchgang für das allgemeine Seitenlayout. Dann, nachdem das Layout und die `lowsrc`-Bilder geladen und dargestellt sind, wird das andere Bild, das durch `src` festgelegt ist, geladen und eingeblendet, um das `lowsrc`-Bild zu ersetzen.

Warum sollten Sie das tun? Das Bild in `lowsrc` ist gewöhnlich eine kleine Vorschau mit geringerer Auflösung als das eigentliche Bild, sodass es sehr schnell geladen werden und dem Leser eine Vorstellung vom allgemeinen Erscheinungsbild der Seite geben kann. (Vergewissern Sie sich, dass das `lowsrc`-Bild tatsächlich kleiner ist, da es sonst zwecklos wäre, es einzufügen.) Dann, nachdem das Layout fertig ist, kann der Leser nach unten rollen und den Text lesen, während die »besseren« Bilder unauffällig im Hintergrund geladen werden.

Die Verwendung von `lowsrc` ist optional; es wird von älteren Browsern einfach ignoriert.

Verwendung von Farben

Eine Möglichkeit, Farbe in Ihre Webseiten zu bringen, besteht darin, Bilder einzufügen; Bilder können Farbspritzer zwischen dem Schwarz, Grau und Weiß erzeugen. Einige Netscape-Erweiterungen ermöglichen es Ihnen jedoch, die Farben der Seite selbst zu bestimmen, einschließlich der Hintergrundfarbe der Seite, sowie der Farben von Text und Verknüpfungen. Außerdem können Sie auch noch »Farbtupfen« für einzelne Zeichen auf der Seite festlegen. In diesem Abschnitt werden Sie alle diese Dinge erlernen.

In diesem Teil des Buchs lernen Sie, wie Sie mit all diesen Veränderungen in HTML 3.2 zurechtkommen können. Farbattribute werden ebenfalls, wie es bei den meisten Präsentationsattributen, die wir bis jetzt kennen gelernt haben, der Fall ist, von HTML 4.0 zu Gunsten der Style Sheets verworfen. Über Style Sheets werden Sie mehr am Tag 10 lernen.

Farbbezeichnungen

Bevor Sie die Farbe von irgendeinem Teil Ihrer HTML-Seite ändern können, müssen Sie wissen, welche Farbe das sein soll. Es gibt in HTML zwei Arten, die Farben mit den Farberweiterungen festzulegen:

- ▶ Sie können eine hexadezimale Zahl einsetzen, die diese Farbe repräsentiert.
- ▶ Sie können einen Farb-Namen aus einer im voraus definierten Liste verwenden.

Die flexibelste und am weitesten unterstützte Methode zur Farbfestlegung besteht darin, den numerischen Wert der Farbe herauszufinden, die Sie benutzen wollen. Die meisten Bildverarbeitungsprogramme haben einen so genannten Farbauswähler – eine Möglichkeit, mit der Sie eine einzelne Farbe aus einer ganzen Reihe verfügbarer Farben auswählen können. Die meisten Farbauswähler werden Ihnen im Gegenzug den Wert dieser Farbe in RGB-Form als drei Zahlen angeben (jeweils eine für Rot, Grün oder Blau – wofür der Name RGB ja auch steht). Jede Zahl reicht normalerweise von 0 bis 255, wobei 000 Schwarz und 255 Weiß ist.

Nachdem Sie Ihre Farbe(n) in der Form von drei Zahlenwerten von 0 bis 255 haben, müssen Sie diese noch in hexadezimale Werte umwandeln. Sie können einen beliebigen wissenschaftlichen Taschenrechner benutzen, der zwischen ASCII und Hex umwandelt, um diese Zahlen zu bekommen. Es gibt auch einen Haufen Freeware- und Shareware-Farbauswähler, die hier weiterhelfen, einschließlich HTML Color Reference und ColorFinder für Windows und ColorMeister und ColorSelect für Macintosh. Sie können auch `rgb.html` benutzen, ein Formular, das die Konvertierung für Sie vornimmt, und dessen Anwendung Sie später in diesem Buch noch erlernen werden. Inzwischen können Sie das `rgb.html`-Formular unter <http://www.lne.com/rgb.html> ausprobieren, wo Sie die Hexadezimalzahl für eine beliebige Kombination dreier Zahlen erfahren können. Beispielsweise wird aus den RGB-Zahlenwerten 000 die Hexadezimal-Kombination 00 00 00, und aus RGB 255 255 255 wird FF FF FF.

Die endgültige hexadezimale Zahl, die Sie benötigen, setzt sich aus allen drei Zahlen mit dem Nummernzeichen (#) am Anfang zusammen:

```
#000000  
#DE04E4  
#FFFFFF00
```

Netscape und der Internet Explorer bieten einen weitaus einfacheren Weg Farben festzulegen. Anstelle von geheimnisvollen Zahlenspielen wählen Sie einfach einen Farb-Namen aus: Black, White, Green, Maroon, Olive, Navy, Purple, Gray, Red, Yellow, Blue, Teal, Lime, Aqua, Fuchsia oder Silver (diese Farben stammen von der Windows-Farbpalette, die nur 16 Farben zulässt.)

Obwohl Farb-Namen einfacher zu behalten und herauszufinden sind als die Nummern, bieten sie weniger Flexibilität in den Farben, die Ihnen zur Verfügung stehen. Außerdem werden die Farb-Namen in den Browsern nicht so weitgehend unterstützt wie die Farbnummern (obwohl sowohl Netscape als auch der Internet Explorer die Farb-Namen unterstützen). Behalten Sie das im Hinterkopf, wenn Sie Farb-Namen einsetzen, da Sie diese Farben in anderen Browsern verlieren können.

Nachdem Sie also einen Farb-Namen oder eine Farbnummer in der Hand haben, können Sie losgehen und diese Farbe auf verschiedene Teile Ihrer HTML-Seite anwenden.

Ändern der Hintergrundfarbe

Um die Hintergrundfarbe der Seite zu ändern, entscheiden Sie sich für eine Farbe, und fügen Sie sie dann der `<body>`-Kennung mittels eines Attributs namens `bgcolor` (backgroundcolor = Hintergrundfarbe) hinzu. Die `<body>`-Kennung ist, falls Sie das vergessen haben, die Kennung, die den gesamten Inhalt Ihrer HTML-Seite umschließt. `<head>` enthält den Titel und `<body>` fast alles andere. `bgcolor` ist eine HTML-Erweiterung, die von Netscape mit der Version 1.1 des Browser eingeführt wurde und Eingang in HTML 3.2 gefunden hat.

Um für Hintergründe Farbnummern einzusetzen, setzen Sie als Wert des `bgcolor`-Attributs die hexadezimale Zahl, die Sie in der vorigen Sektion herausgefunden haben, innerhalb von `<body>` ein. Das sieht dann so aus:

```
<body bgcolor="#ffffff">
<body bgcolor="#934ce8">
```

Um Farb-Namen zu verwenden, benutzen Sie einfach den Namen der Farbe als den Wert von `bgcolor`:

```
<body bgcolor="#white">
<body bgcolor="#green">
```



Einige Browser erlauben es Ihnen, Farb-Namen ohne das einleitende Nummernzeichen (#) anzugeben. Obwohl dies bequemer erscheinen mag, ist die Einfügung eines einzelnen Extra-Zeichens doch wohl keine so große Mühe, wenn man bedenkt, dass dies mit vielen anderen Browsern nicht vereinbar ist.

Ändern der Textfarben

Wenn Sie schon die Hintergrundfarben ändern, dann ist es nur logisch, dass Sie auch die Farbe des eigentlichen Textes verändern möchten. Eine größere Anzahl von HTML-Attributen ermöglichen Ihnen das Ändern der Textfarbe Ihrer Seite.

Um die Text- und Verknüpfungsfarben zu ändern, verwenden Sie die Farb-Namen oder -nummern genauso wie für das Ändern der Hintergrundfarben. Wenn Sie diese Farbe bestimmt haben, können Sie dann eines der folgenden Attribute zur `<body>`-Kennung hinzufügen, wobei Sie den Farb-Namen oder die Farbnummer als deren Wert einsetzen:

- ▶ `text`
Bestimmt die Farbe des gesamten Textes im `<body>`-Teil der Seite, einschließlich der Überschriften, des normalen Textes, des Textes innerhalb von Tabellen etc. mit Ausnahme des Verknüpfungstextes
- ▶ `link`
Bestimmt die Farbe der normalen, noch nicht angeklickten Verknüpfungen auf der Seite (die oft per Voreinstellung blau sind).
- ▶ `vlink`
Bestimmt die Farbe der Verknüpfungen, die bereits verfolgt wurden (die oft per Voreinstellung violett oder rot sind).
- ▶ `alink`
Bestimmt die Farbe einer Verknüpfung, die in dem Moment durch einen Mausklick aktiviert wurde, solange der Maus-Button noch nicht losgelassen wurde («aktiver Link»). Diese sind in der Browser-Voreinstellung oft rot.

Erinnern Sie sich an das Geisterhaus-Bild am Anfang dieses Kapitels, das in eine Webseite eingefügt wurde? Diese Seite wäre entschieden gruseliger, wenn der Bildhintergrund schwarz wäre, und ein orangefarbener Text würde viel besser zu Urlaub passen. Zur Erstellung einer Seite mit orangefarbenen Hintergrund, schwarzem Text und tiefroten *unfollowed* Links, sollten Sie folgendes `<body>`-Tag verwenden:

```
<body bgcolor="#ff9933" text="#000000" link="#800000">
```

Die Verwendung der folgenden Farb-Namen und *unfollowed* Verknüpfungen würden den gleichen Effekt erzielen:

```
<body bgcolor="orange" text="black" link="#800000">
```

Jedes dieser Verknüpfungen würde eine Seite erzeugen, die ungefähr so aussieht, wie die in Abbildung 7.21 zeigte.

Farbtupfer

Wenn Sie die Textfarben innerhalb einer Seite mit den Attributen der `<body>`-Kennung verändern, dann verändert das den gesamten Text auf der Seite. Mit Farbtupfern können Sie die Farbe einzelner Zeichen auf der Seite bestimmen, die Sie zusätzlich zu oder anstelle der übergreifenden Textfarbe einsetzen können.

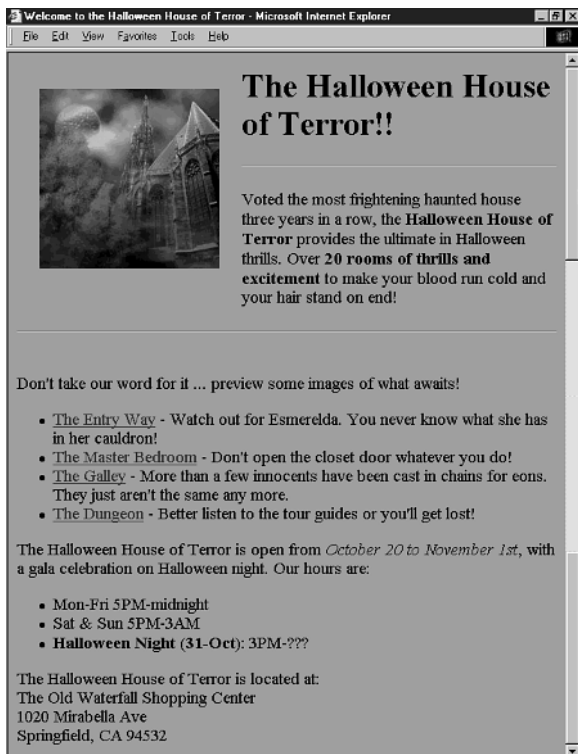


Abbildung 7.21:
Hintergrund- und Textfarben

Gestern haben Sie etwas über die HTML-3.2-Erweiterung `<font>` gelernt, welche die Schriftart und -größe verändert. Es gibt inzwischen aber ein drittes und noch neueres Attribut zu `<font>`, nämlich `color`, das es Ihnen erlaubt, die Farbe von individuellen Wörtern oder Sätzen zu bestimmen. Der Wert von `color` ist entweder ein Farb-Name oder eine Nummer:

```
<p>When we go out tonight, we're going to paint the town  
<font color="#ff0000">RED</font>.
```

Sie können Schrift-Farbtupfer selbstverständlich gleichzeitig mit verschiedenen Schriftarten und -größen einsetzen.

Bilder als Hintergründe

Ein letztes Thema in diesem Kapitel ist die Möglichkeit, ein Bild und nicht nur einen einfarbigen Hintergrund als Hintergrund für Ihre Seiten einzusetzen. Wenn Sie ein Bild als Hintergrund einsetzen, ist dieses Bild »gekachelt« – das heißt, das Bild wird reihenweise wiederholt, um das Browser-Fenster aufzufüllen.

Um einen gekachelten Hintergrund zu erzeugen, brauchen Sie zuerst einmal ein Bild, das als Kachel dienen kann. Wenn Sie ein Bild für diese »Kacheln« erzeugen, sollten Sie sicherstellen, dass das Muster zwischen den einzelnen Bildern glatt ineinander übergeht, wenn sie nebeneinander angeordnet werden. Dazu müssen Sie normalerweise das Bild in Ihrem Bildverarbeitungsprogramm sorgfältig editieren, damit die Kanten nicht sichtbar werden. Ziel dabei ist, dass sich die Kanten so sauber treffen, dass keine »Naht« zwischen den Kacheln entsteht, nachdem sie aneinandergelegt wurden (ein Beispiel für eine weniger saubere Lösung sehen Sie in Abbildung 7.22). Sie können aber auch ein kommerzielles Clipart-Paket verwenden, das Muster bereitstellt, die oft genau für diesen Einsatzzweck ausgelegt wurden.

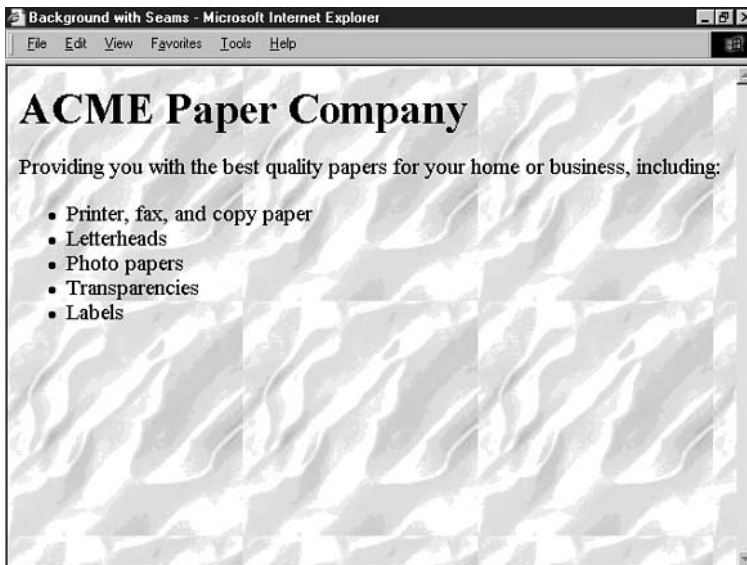


Abbildung 7.22:
Kachelartig ange-
ordnete Bilder
mit »Naht«

Wenn Sie ein Bild haben, das Sie sauber aneinanderreihen können, brauchen Sie nur noch das `background`-Attribut, das Bestandteil des `<body>`-Tag ist. Der Wert von `background` ist ein Datei-Name oder eine URL, die auf Ihre Bilddatei zeigt, wie im folgenden Beispiel:

```
<body background="tiles.gif">
<body background="backgrounds/rosemarble.gif">
```

Abbildung 7.23 zeigt das Ergebnis eines Hintergrunds mit einfachen Bildern.

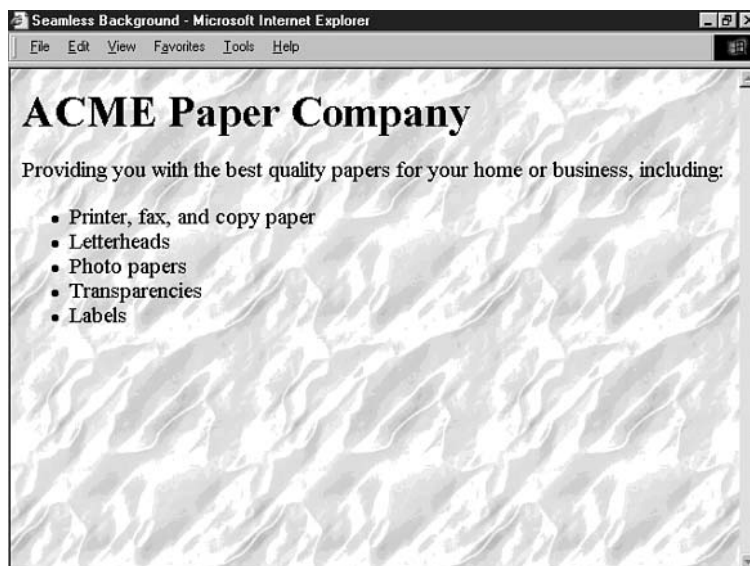


Abbildung 7.23:
Ein Hintergrund
mit kachelartig
angeordneten Bil-
dern im Internet
Explorer

Der Internet Explorer bietet einen besonderen Dreh für gekacheltes Hintergrunddesign: ein fixiertes Kachelmuster, welches Wasserzeichen genannt wird. Die Idee, die dahinter steckt, ist, dass, wenn Sie die Seite herunterrollen, nicht mehr alles (einschließlich des Hintergrundbildes) an Ihnen vorbeirollt, sondern nur der Vordergrund (Text und Bilder). Die Hintergrundkacheln bleiben an ihrem Platz verwurzelt. Um diesen Effekt zu erzeugen, verwenden Sie das `bgproperties="fixed"`-Attribut zur `<body>`-Kennung:

```
<body background="backgrounds/rosemarble.gif" bgproperties="fixed">
```

Tipps für eine bessere Nutzung von Abbildungen

Die Verwendung von Abbildungen auf Webseiten ist heute eine der meistdiskutierten Fragen zwischen Nutzern und Anbietern von Webseiten. Für jeden, der eine Webseite mit mehr, größeren und glänzenden Abbildungen gestalten will, um die Vorzüge der grafischen Fähigkeiten des Web auszunutzen, gibt es jemanden mit einer langsamen Netzwerkverbindung, der um weniger Abbildungen bittet, damit ihr oder sein Browser nicht stundenlang braucht, um eine Seite zu laden.

Als Gestalter von Webseiten sollten Sie beide Sichtweisen berücksichtigen. Schaffen Sie einen Mittelweg zwischen dem Spaß, eine optisch ansprechende, farbenfrohe Webseite zu erzeugen, und der Notwendigkeit, dass jeder, der es möchte, Ihre Informationen auch erhalten kann – und das sind auch jene Leute, die Ihre Abbildungen (weshalb auch immer) nicht brauchen können.

Dieser Abschnitt bietet Ihnen ein paar Hinweise und Kompromisse, die Sie bei der Gestaltung Ihrer Webseiten machen können, sodass Sie jeden zufrieden stellen können (oder auch unglücklich machen, je nachdem, wie Sie es betrachten).

Brauchen Sie die Abbildung wirklich?

Bei jeder Abbildung, die Sie in Ihre Seite einfügen, sollten Sie sich fragen, weshalb Sie das tun. Was bringt die Abbildung für die Gestaltung der Seite? Bietet sie Informationen, die stattdessen auch im Text präsentiert werden könnten? Ist sie nur da, weil Sie sie mögen?

Bemühen Sie sich, Ihre Webseiten nicht mit hübschen, aber ansonsten unnötigen Abbildungen zu überhäufen. Eine einfache Webseite mit nur wenigen Symbol-Abbildungen ist häufig effektiver als eine, aus der Ihnen ausgefallene 3-D-Buttons entspringen oder große Fotografien, schattierte Knöpfe oder psychedelische Trennlinien.

Verwenden Sie kleine Bilder

Ein kleines Bild benötigt weniger Zeit bei der Übertragung durchs Netz. Daher bewirkt die Verwendung kleinerer Abbildungen, dass Ihre Seite schneller geladen wird und Leute, die sie über langsame Netzwerk-Verbindungen lesen wollen, weniger frustriert sind. Was könnte leichter sein?

Mit »kleinerer Abbildung« kann gemeint sein: geringere Größe auf dem Bildschirm. Doch Sie können kleinere Abbildungen auch erzeugen, indem Sie die Anzahl der Farben verringern, die von dem Bild verwendet werden. Das Ziel ist also, die Dateigröße der Abbildung zu verringern, damit sie schneller übertragen wird, doch ein 10 mal 10 cm großes Schwarzweißbild (zwei Farben) kann eine geringere Dateigröße haben als eine zwei Quadratzentimeter große Fotografie mit 8-Bit-Farbtiefe. Mit den meisten Bildverarbeitungsprogrammen können Sie die Anzahl der verwendeten Farben reduzieren und das Ergebnis so bearbeiten, dass es auch mit weniger Farben noch gut aussieht.

Eine gute Regel, an die man sich halten kann, ist, zu versuchen, die Abbildungen kleiner als 20 Kbyte zu halten. Das mag wenig scheinen, doch eine einzige 20-Kbyte-Datei braucht etwa 10 Sekunden, um über eine 28.8 Kbps-Verbindung geladen zu werden. Multiplizieren Sie diese Zeit mit der Anzahl der Abbildungen in Ihrer Websei-

te, und Sie kommen womöglich auf eine spürbare Zeitspanne für das Laden der Seite (auch wenn Sie einen Browser verwenden, der mehrere Bilder gleichzeitig laden kann, bleibt die Übertragungsgeschwindigkeit doch gleich schnell). Wird sich jemand für Ihre Webseite interessieren, wenn man essen und spazieren gehen kann, während sie geladen wird?

Abbildungen so oft wie möglich wiederverwenden

Zusätzlich dazu, dass Sie einzelne Bilder klein halten sollten, versuchen Sie, dasselbe Bild so oft wie möglich auf einzelnen Seiten und über mehrere Seiten hinweg zu wiederholen: Wenn Sie zum Beispiel Abbildungen von Hinweispunkten haben, benutzen Sie jedesmal dasselbe Bild, anstatt ein neues einzusetzen. Die Wiederverwendung von Bildern hat zwei bedeutsame Vorteile gegenüber der Verwendung unterschiedlicher Bilder:

- ▶ Die Wiederverwendung von Bildern bietet eine Einheitlichkeit im Design über Ihre Seiten hinweg und hilft dabei, ein bestimmtes »Image« für Ihre Site zu schaffen.
- ▶ Was noch wichtiger ist: Bilder wieder zu verwenden bedeutet, dass Ihr Browser sie nur ein einziges Mal herunterladen braucht. Sobald sich das Bild einmal im Arbeitsspeicher befindet, kann es wiederholt dargestellt werden, ohne dass der Browser wieder neue Verbindungen zum Server herstellen muss.

Um ein Bild wieder zu verwenden, brauchen Sie nichts Besonderes zu tun; vergewissern Sie sich bloß, dass Sie sich auf das Bild jedesmal mit der gleichen URL beziehen. Der Browser kümmert sich dann um den Rest.

Alternativen zu Abbildungen anbieten

Wenn Sie bislang bei Ihren Abbildungen das `alt`-Attribut noch nicht benutzt haben, sollten Sie es nun tun. Das `alt`-Attribut ist außerordentlich nützlich, um Ihre Webseiten auch für textbasierte Browser lesbar zu machen. Was ist jedoch mit Leuten, welche die Abbildungen in ihrem Browser ausschalten, weil sie eine langsame Internet-Verbindung haben? Die meisten Browser verwenden in diesem Fall das `alt`-Attribut nicht (obgleich sie es sollten). Und bisweilen ist `alt` nicht ausreichend; da Sie lediglich Text im `alt`-Attribut verwenden können, kann als Ersatz für die Abbildung kein HTML-Code benutzt werden.

Um all diese Probleme zu umgehen und Ihre fescbe, grafische Webseite zu bewahren, ziehen Sie die Erzeugung einer zusätzlichen reinen Textversion Ihrer Webseiten in Betracht, und fügen Sie Verbindungen zu diesen in die grafischen Versionen der Seiten ein, etwa so:

<p> Eine reine Text-Version dieser Seite ist ebenfalls verfügbar. </p>

Die Verknüpfung mit der Textversion der Seite benötigt nur einen kleinen Absatz auf der »eigentlichen« Webseite, macht jedoch die Informationen, die Sie anbieten, für Ihre Leser viel besser erreichbar. Dies ist eine Gefälligkeit, für die Ihnen die Leser mit langsamen Verbindungen dankbar sein werden, während Sie zugleich auch Ihre »eigentliche« Webseite mit so viel Grafik, wie Sie mögen, für diejenigen bereithalten können, die schnelle Netzwerk-Verbindungen haben.

Zusammenfassung

Eine der wichtigsten Eigenschaften, die das World Wide Web von anderen Formen der Internet-Informationen abhebt, ist, dass Dokumente im Web farbige Grafiken enthalten können. Man könnte behaupten, dass es die Existenz der Grafik war, die das Web so schnell um sich greifen und es in einem so kurzen Zeitraum so beliebt werden ließ.

Um Abbildungen in Ihre Webseite aufzunehmen, müssen jene Abbildungen im GIF- oder JPEG-Format vorliegen (GIF ist weitgehender unterstützt) und klein genug sein, um auch über langsame Internet-Verbindungen schnell übertragen werden zu können. Die HTML-Kennung `<img />` ermöglicht das Einbinden einer Grafik in Ihre Webseite, entweder gemeinsam mit Text in einer Zeile oder in einer eigenen Zeile. Die `<img />`-Kennung hat drei grundsätzliche Attribute, die in Standard-HTML unterstützt werden:

- ▶ **src:**
Speicherort und Datei-Name der einzubindenden Abbildung.
- ▶ **align:**
Wie der umgebende Text vertikal zur Abbildung auszurichten ist. `align` kann einen von drei Werten annehmen: `top`, `middle` oder `bottom`.
- ▶ **alt:**
Ein alternativer Text, um die Abbildung in textbasierten Browsern zu ersetzen. Sie können Abbildungen auch innerhalb eines Verknüpfungs-Tag (`<a>`) verwenden, wobei die Abbildung genauso wie der Text zum Anwählen der Verknüpfung benutzt werden kann.
Zusätzlich zu den Standard-Attributen gibt es verschiedene neue Attribute der `<img />`-Kennung, die größere Kontrolle über Abbildungen und das Layout von Webseiten ermöglichen. Diese neuen Attribute, von denen die meisten inzwischen in HTML 3.2 aufgenommen wurden, sind unter anderem die folgenden:
 - ▶ `align="left"`, `align="right"`
Setzen die Abbildung an den entsprechenden Seitenrand, wobei sie den fol-

genden Text in den Raum neben der Abbildung fließen lassen. Außerdem gibt es eine HTML-Erweiterung zu `<br />` namens `clear`, die das Weiterfließen des Textes neben der Abbildung unterbricht. `clear` kann drei Werte einnehmen: `left`, `right` und `all`.

- ▶ `align="texttop", align="absmiddle", align="baseline", align="absbottom"`
(Netscape) Ermöglichen größere Kontrolle über die Ausrichtung von internen Bildern und über den umgebenden Text.
- ▶ `vspace, hspace`
Legen den Abstand zwischen einem Bild und dem umgebenden Text fest.
`border`
- ▶ Bestimmt die Dicke der Umrandung einer Abbildung (die eine Verknüpfung sein kann, aber nicht sein muss). `border=0` versteckt die Umrandung völlig.
- ▶ `lowsrc`
(Nur Netscape) Bestimmt ein alternatives Bild von geringerer Auflösung, das vor dem durch `src` bestimmten Bild als Vorschau geladen wird.

Indem Sie Attribute der `<body>`-Kennung verwenden, können Sie eine Seite mit farbigem Hintergrund oder farbigem Text versehen, oder einzelne Buchstaben mit dem `color`-Attribut der `<font>`-Kennung einfärben. Außerdem können Sie gemusterte oder gekachelte Hintergrundbilder einsetzen, die hinter dem Text und den Bildern erscheinen. Hierzu legen Sie die zu verwendende Bilddatei im `background`-Attribut des `<body>`-Tag fest.

Workshop

Da Sie nun wissen, wie man Bilder und Farben in Webseiten einfügt, können Sie richtig kreativ bei der Erstellung Ihrer Webseiten sein. Dieser Workshop wird Ihnen helfen, sich an die wichtigsten Punkte zur Verwendung von Bildern und Farben auf Ihren Seiten zu erinnern, sodass Ihre Webseiten mit HTML 3.2- und HTML 4.0-Browsern kompatibel sind. Wenn Sie Ihre Webseite unbedingt mit den HTML 4.0-Spezifikationen erstellen möchten, müssen Sie auf viele der Präsentationsoptionen, die Sie heute kennen gelernt haben, verzichten und stattdessen mit Style Sheets arbeiten.

Fragen und Antworten

F Was ist der Unterschied zwischen einem GIF und einem JPEG-Bild? Gibt es eine bestimmte Regel, die besagt, wann man lieber das eine Bild dem anderen vorziehen soll und umgekehrt?

A Beachten Sie folgende Regel: verwenden Sie GIF-Dateien, wenn Ihr Bild 256 oder weniger Farben enthält. Einige Beispiele dafür sind Cartoons, Cliparts, Schwarzweißfotos oder Bilder mit farbigen Flächen. Benutzen Sie GIF-Dateien auch dann, wenn Sie möchten, dass Ihre Bilder transparente Flächen enthalten oder wenn Sie eine Animation erstellen möchten, für die kein spezielles Plugin oder Browserhilfsprogramm erforderlich ist. Denken Sie daran, dass Sie Ihre Bildbearbeitungs-Software verwenden, um die Anzahl der Farben in der Bildpalette zu reduzieren, denn so wird auch die Dateigröße reduziert. JPEG-Bilder eignen sich am besten für Grafiken, die Photoqualität haben oder 3D sein sollen, da sie farbechte Bilder sehr gut anzeigen können. Die meisten Bildbearbeitungsprogramme ermöglichen Ihnen die genaue Bestimmung des Komprimierungswerts eines JPEG-Bildes. Je komprimierter das Bild ist, desto kleiner wird die Dateigröße. Wenn Sie mit der Kompression übertreiben, zerstört dies jedoch die Qualität und Erscheinung des Bildes. Sie müssen das Gleichgewicht zwischen Qualität und Dateigröße finden, und das kann von Bild zu Bild unterschiedlich sein.

F Wie kann ich Verkleinerungen meiner Abbildungen erzeugen, sodass ich sie mit größeren externen Abbildungen verknüpfen kann?

A Das müssen Sie mit irgendeinem Bildverarbeitungsprogramm machen; das Web wird es nicht für Sie erledigen. Laden Sie einfach das Bild, und verkleinern Sie es auf die richtige Größe.

F Was sind das für Bilder, von denen nur ein kleiner Teil auf dem Seitenhintergrund angezeigt wird? Sie sehen aus, als würden sie auf der Seite irgendwie schweben. Wie kann ich solche Bilder erstellen?

A Es handelt sich hierbei um eine weitere Aufgabe, die Sie mit einem Bildbearbeitungsprogramm meistern können. Diese Bilder sind als Transparent GIFs bekannt und Sie können diesen Effekt nur mit einem GIF-Bild erzeugen. Ich werde Ihnen im nächsten Kapitel zeigen, wie man ein Transparent GIF erstellt.

F Kann ich HTML-Kennungen in die Zeichenfolge für das alt-Attribut schreiben?

A Das wäre ganz nett, nicht wahr? Unglücklicherweise geht das aber nicht. Alles, was man einfügen kann, ist ganz gewöhnlicher Text. Halten Sie ihn möglichst einfach, und alles sollte in Ordnung sein.

- F Sie haben eine Technik besprochen, bei der man `lowsrc`-Abbildungen auf einer Seite einfügt, die vor den regulären Abbildungen heruntergeladen werden. Ich habe beim Laden einer Webseite bei einer Abbildung einen besonderen Effekt bemerkt. Das Bild wurde zunächst nur verschwommen dargestellt. Mit der Zeit wurde die Darstellung immer klarer. Ist das der `lowsrc`-Effekt?**
- A** *Nein, dieser Effekt wird `interlaced GIF` genannt. Dabei handelt es sich nur um eine Abbildung, die beim Laden anders als normale GIFs dargestellt wird. Sie werden darüber mehr im nächsten Kapitel erfahren.
`lowsrc`-Abbildungen laden ganz normal (ohne spezielle visuelle Effekte).*
- F Ich habe ein paar Webseiten gesehen, bei denen man verschiedene Stellen einer Abbildung anklicken kann und dann unterschiedliche Ergebnisse bekommt, zum Beispiel eine Landkarte der Vereinigten Staaten von Nordamerika, in der mit jedem Staat eine andere Seite verknüpft war. Wie kann man das mit HTML machen?**
- A** *Das wird `Abbildungsplan (image map)` genannt und ist schon eine fortgeschrittene Form der Entwicklung von Webseiten. Zu diesem Thema erfahren Sie mehr in Kapitel 9.*

Quiz

1. Beschreiben Sie die zwei verschiedenen Bildklassen, die in Webseiten verwendet werden.
2. Was ist das bekannteste Attribut des `<img />`-Tag? Was tut es?
3. Wenn sie ein eher komisches Icon anstelle eines Bildes auf Ihrer Seite in einem Browser sehen, lädt das Bild nicht. Warum kann das passieren?
4. Geben Sie die Regel an: wenn jemand eine Webseite im Web-Browser betrachtet, woran erkennt er den Unterschied zwischen einem anklickbaren Bild (das als Link verwendet wird) und einem nicht anklickbaren Bild?
5. Warum ist das Verwenden eines `alt`-Attributs so wichtig, wenn ein Text alternativ zu einem Bild angeboten werden soll? Wann muss man einen solchen Alternativtext erstellen?

Antworten:

1. Inline-Bilder erscheinen direkt auf der Webseite, zwischen dem Text und den Links. Externe Bilder werden bei Wunsch des Besuchers, normalerweise nach dem Anklicken des Links, heruntergeladen.
2. Das wichtigste Attribut des `<img />`-Tag ist das `src`-Attribut. Dies gibt den Dateinamen oder die URL des Bildes an, das Sie in Ihre Seite einfügen möchten.
3. Es gibt mehrere Gründe, warum ein Bild nicht lädt. Die URL könnte falsch sein; der Dateiname könnte falsch sein (Genauigkeit ist hier sehr wichtig!); die Dateierweiterung oder der Dateityp könnten nicht stimmen; es könnte auch sein, dass das Bilderladen in Ihrem Browser ausgeschaltet ist.
4. Bei Standardeinstellungen sind anklickbare Bilder (die als Links dienen) umrahmt, nicht-anklickbare Bilder besitzen keinen Rahmen.
5. Es ist eine gute Idee, Alternativtexte zu Bildern zu verwenden, da manche Ihrer Besucher wahrscheinlich nur Browser benutzen, die nur Text anzeigen, oder die Grafik in ihren Browsern ausgeschaltet ist. Alternativtexte sollten vor allem bei Bildern verwendet werden, die als Links dienen.

Übungen

1. Erstellen oder finden Sie Bilder, die Sie als Navigationsicons oder Schaltflächen auf einer oder mehrerer Seiten Ihrer Web-Site verwenden können. Denken Sie daran, dass es immer besser ist, viele Bilder anstelle eines Bildes zu benutzen. Erstellen Sie eine einfache Navigationsleiste, die am Seitenkopf oder am Seitenende benutzt werden kann.
2. Erstellen oder finden Sie Bilder, die Sie zur Verbesserung Ihres Webseiten-Layouts verwenden können. Bilder wie kleine Spruchbänder (für Seitentitel), Kugeln, horizontale Linien und Hintergrundbilder machen sich immer gut. Nachdem Sie sich ausgesucht haben, was Ihnen gefällt, schreiben Sie Hintergrund-Text und suchen Linkfarben, die zu Ihrer Web-Site passen.



Animierte Grafiken erstellen

**Woche
2**

Wahrscheinlich haben Sie erst einmal genug vom Kodieren.

Dieses Kapitel verschont Sie einstweilen vom Tippen und *befasst sich mit bewegten Bildern*. Animierte Grafiken beleben Ihre Web-Site, heben gewisse Sachen hervor und sind gar nicht so schwierig zu erstellen, wie Sie vielleicht denken. Mit der richtigen Software ausgerüstet, können Sie eine eigene und originelle Web-Site kreieren. Und genau das werden Sie in diesem Kapitel lernen. Sind Sie bereit?

- ▶ Um herauszufinden, warum das animierte GIF den Eindruck erweckt als würde es sich bewegen, untersuchen Sie die einzelnen Frames (Einzelbild einer Animation).
- ▶ Wann und wo werden Animationen (Folge von bewegten Bildern) angewandt?
- ▶ Wie erzeugt man transparente GIF-Dateien und wie definiert man die Farbe als »transparent«?
- ▶ Wissenswertes über die Bildbearbeitung und die Werkzeuge zur Erstellung von GIF-Dateien. Letztere werden Ihnen helfen, Ihre eigene animierte GIF-Datei zu erstellen.
- ▶ Überblick über die nützlichsten Software-Eigenschaften in einer Bildbearbeitungs-Software.
- ▶ Wie kompiliert und reduziert man die Größe einer animierten GIF-Datei?
- ▶ Wie erstelle ich Frames für meine Animation?

Was versteht man unter einer Animation?

Stellen Sie sich vor, Sie hätten einen Stoß Bilder, eines über dem anderen geschichtet. Jedes Bild sieht dabei etwas anders als das vorherige aus und alle Bilder müssen in einer bestimmten Reihenfolge aufeinander folgen. Wenn Sie diese Bilder nun durchblättern, entsteht ein Daumenkinoeffekt, wobei die Geschwindigkeit der Bewegung davon abhängt, wie schnell Sie die Seiten durchblättern.

Im Grunde genommen handelt es sich bei den Animations-Dateien (sei es nun ein animiertes GIF, eine Windows AVI-Datei, Quick Time Movie- oder eine MPEG-Datei) um nichts anderes als um ein Daumenkino. Die Bilder, die gewöhnlich die gleiche Größe haben, sind dabei mittels Software so in Reihe geschaltet, dass sie eine Art Script erzeugen. Dieses Script befindet sich in der Animations-Datei und legt Parameter, wie zum Beispiel Ablaufgeschwindigkeit der Animation oder die Art und Weise, wie ein Bild über dem anderen angeordnet sein soll, fest. Sie müssen also nicht jedes einzelne Bild herunterladen, um diesen Trickfilmeffekt zu erzeugen, denn Sie haben ja eine einzige Datei, die aus zahlreichen Frames besteht. Wenn diese in einer bestimmten Reihe geschaltet werden, wird der Trickfilmeffekt erzielt.

Vielleicht kennen Sie ja das tanzende Baby, das das Internet im Sturm erobert hat. Dieses Baby ist doch tatsächlich eine Berühmtheit geworden, denn der Produzent der beliebten TV-Show Ally McBeal hat es beim Surfen im Internet entdeckt und in seiner Show auftreten lassen. Baby Cha wurde durch vereinte Bemühungen von Viewpoint Datalabs (Erfinder der Baby-Figur) und Kinetix – einer Tochterfirma von Autodesk und Hersteller von 3D Studio Max und der Plugin-Software Charakter Studio – zum Leben erweckt. Sollten Sie dieses Baby nicht kennen, so müssen Sie auf alle Fälle auf die Kinetix Webseite gehen (<http://www.ktx.com/cs/html/babycha.html>) und dieses wunderbare Animationsbeispiel selbst sehen.

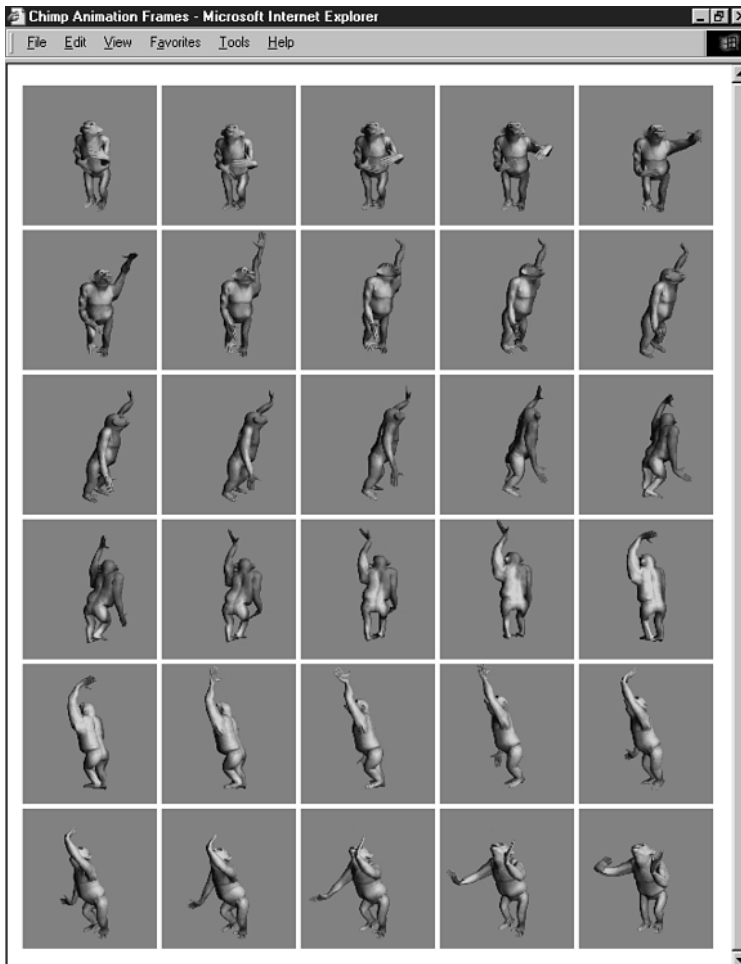


Abbildung 8.1:
30 Animations-
Frames des tan-
zenden Schimpan-
sen

Alle Animationen liegen im Windows AVI Format vor (stellen Sie also sicher, dass Ihr Browser sie auch unterstützt). Einige der Dateien benötigen etwas Zeit beim Laden; das Warten lohnt sich aber auf alle Fälle.

Bild 8.1. zeigt ein Pendant zu »Baby Cha«. Der tanzende Schimpanse wurde ebenfalls mit 3D Studio Max und Character Studio kreiert, also derselben Software, die für die Baby Cha-Animation benutzt wurde. Bild 8.1 zeigt 30 Frames der Animation, wobei nur jeder 5. Frame der gesamten Sequenz wiedergegeben wurde, um sie in einem einzigen Fenster anzeigen zu können. Sämtliche Bilder sind in verschiedene Sequenzen unterteilt. Die 1. Sequenz beginnt mit dem ersten Bild links oben und endet rechts oben mit dem letzten Bild. Jeder Frame dieser Animation ist dabei anders und wenn die Frames in einer Sequenz abgespielt werden, erscheint ein sich drehender und springender Affe, der dabei seinen linken Arm hebt und senkt – kurz, eine Animation.

Beim Erstellen von Animationen müssen die Frames nicht immer so nahtlos ineinander übergehen wie sie es in diesem Beispiel tun. Sie können auch eine automatisch ablaufende Dia-Show mehrerer verschiedener Bilder programmieren (zum Beispiel Fotos und Banner).

Wann und wo Animation anwenden?

Baby Cha und der tanzende Schimpanse sind zwar lieb und nett, aber von geringem Nutzen. Wann sollte man also bewegte Elemente in eine Webseite einbinden?

Bewegte Elemente sollen auf etwas aufmerksam machen. Zuviel davon lenken aber von der Seite, bzw. vom Text ab und der Leser wird nicht mehr wissen, wohin er zuerst sehen soll. Deshalb sollten Sie Animationen sparsam und wirksam einsetzen. Auf keinen Fall sollte man mehrere Animationen auf eine Seite quetschen, das Laden der Seite würde nämlich eine Ewigkeit dauern.

Lesen Sie hierzu einiges über die übliche Verwendung von Animationen:

- ▶ **Banner**
Manche mögen keine Spruchbänder und vermeiden deren Anwendung. Spruchbänder werden aber meistens angewandt wenn Ihre Webseite gesponsert wird oder wenn Sie auf Ihre Produkte und Fähigkeiten aufmerksam machen wollen.
- ▶ **Bullets (Blickfangpunkte)**
Bullets werden auf WWW-Seiten gerne verwendet, um Verweise grafisch und attraktiv darzustellen. Bei wenigen Verweisen sind kleine grafische Bullets durchaus akzeptabel, nicht aber bei Dutzenden von Verweisen auf einer Seite.

► Horizontale Trennlinien

Auf einer Webseite sind oft dünne Linien zu sehen, die der übersichtlichen Strukturierung von Information dienen. Wer auf seinen WWW-Seiten jedoch etwas Besonderes bieten will, verwendet grafische Linien: einen mit Liebespfeilen schießenden Amor, zwei Tennisspieler, die sich den Ball zuspielen, ein Haifisch, der von links nach rechts schwimmt, kurzum, eine Linie, die dem Thema der Webseite entspricht. Breite und Höhe der Animation sollten aber nicht zu groß sein, denn die Datei wäre sonst zu umfangreich.

► Animierte Logos

Das animierte Logo Ihrer Homepage macht auf Ihre Firma oder Ihr Produkt aufmerksam. Ist Ihr Logo lang, so können Sie es teilen und in jedem Teil bewegte Elemente einfügen. Wenn Sie danach eine rahmenlose Tabelle verwenden, um diese Teile wieder zusammenzufügen, erscheinen sie als eine Grafik auf Ihrer Seite.

► Symbole

Wie schon erwähnt, sind Animationen von großem Nutzen, wenn es darum geht, die Aufmerksamkeit auf wichtige Informationen Ihrer Seite zu ziehen. Ein Briefumschlag oder Briefkasten neben Ihrer E-Mail-Adresse, ein blinkendes »Neu!« neben lesenswerten Neuigkeiten, ein läutendes Telefon neben Ihrer Telefonnummer, usw...

Erstellen transparenter GIF-Dateien

Sicherlich haben Sie schon Grafiken auf Webseiten gesehen, die den Anschein hatten als würden sie auf der Seite schweben. Im Gegensatz zu einem Vier- oder Dreieck, hat ein transparentes GIF keine festgelegte Form und lässt gleichzeitig den Hintergrund erkennen.

Je nachdem welche Software sie verwenden, haben Sie die Möglichkeit eine in der Grafik vorkommende Farbe als transparent zu definieren. Einige GIF Animations-Compiler ermöglichen es Ihnen, mit Hilfe dieses Features transparente Hintergründe bei Grafiken zu definieren, andere wiederum nicht. GIF- Animationen lassen sich auf zweierlei Art und Weise erstellen:

- Falls Sie mit Ihrem GIF-Animations-Programm die Farbe »transparent« definieren können (wobei die Farbe erst in einem Frame, dann im nächsten usw. definiert wird) können Sie diese Eigenschaft (transparent) hinzufügen, bevor Sie die Animation speichern.

- Sollten Sie in Ihrer GIF-Animation die Farbe transparent nicht definieren können, müssen Sie diese Farbeigenschaft für jedes Einzelbild in Ihrer Bildbearbeitungs-Software definieren. Speichern Sie jeden einzelnen Frame als Transparent-GIF und fügen Sie sie in Ihrem GIF-Animations-Programm zusammen. Das Erstellen Ihrer Animation dauert zwar etwas länger, aber es funktioniert.

Das transparente GIF

Normalerweise benutzen Sie Bilder, die sich auf einer soliden Hintergrundfarbe befinden, damit die Stellen, die sie sichtbar auf Ihrer Webseite haben wollen, erkennbar bleiben. Definieren Sie hierfür die Hintergrundfarbe Ihres Bildes als transparent. Sollten Sie die Animation einer Webseite mit Hintergrundbild gestalten, definieren oder wählen Sie eine GIF-Datei, deren Hintergrundfarbe der des Bildes ähnelt. Beispiel: Bei einem schwarzen Sternenhimmel als Hintergrund, sollten Sie einen schwarzen Hintergrund für Ihre transparente GIF verwenden. Ist der Hintergrund Ihrer Webseite in Pastelltönen gehalten, definieren oder wählen Sie GIF-Bilder auf weißem oder hellem Hintergrund.

Nun fragen Sie sich sicherlich, warum das so wichtig ist. Viele Bildbearbeitungsprogramme verwenden Anti-Aliasing (Algorithmus für Pixelgrafiken), um sichtbare Treppcheneffekte an harten Kanten oder Farbübergängen in Grafiken auszugleichen. Wenn Sie eine schwarze Linie auf einem dunklen Hintergrund haben, so wirkt diese hart und man kann so genannte Pixeltreppchen erkennen. Es ist also besser, einen Hintergrund mit ähnlichen Farbtönen wie den der Seite, auf der Sie Ihre Grafik abbilden werden, zu verwenden.



Anti-Aliasing ist ein Algorithmus für Pixelgrafiken, um sichtbare Treppcheneffekte an harten Kanten oder Farbübergängen in Grafiken auszugleichen. Dies wird erzielt, indem man Pixel, die zwischen den beiden ursprünglichen Farben stehen, mit der Grafik vermischt. Beispiel: Eine schwarze Linie auf weißem Hintergrund erscheint mit dem Anti-Aliasing-Programm weicher, weil graue Schattierungen um die schwarze Linie herum eingefügt wurden.

Bild 8.2. zeigt ein Beispiel des Anti-Aliasing-Programms. Die zu sehende Zeichnung wurde auf weißem Hintergrund erstellt. Während auf dem linken Bild mit schwarzem Hintergrund Pixel zu erkennen sind, die um die Zeichnung geistern, erscheint die gleiche transparente GIF-Datei tadellos auf der rechten Seite mit weißem Hintergrund. Dieses extreme Beispiel illustriert den Vorteil von Anti-Aliasing bei transparenten GIF-Dateien.

Sollten Sie eigene Bilder für transparente GIFs erstellen, stellen Sie sicher, dass Sie die Hintergrundfarbe auf keine der Stellen anwenden, die Sie nicht transparent haben wollen. Der Grund hierfür ist, dass jede Stelle, die Sie als transparent definieren werden, auch transparent sein wird. Sollte irgend eine Stelle Ihres Bildes also transparent sein, so wird Ihr Kunstwerk leider Löcher haben.

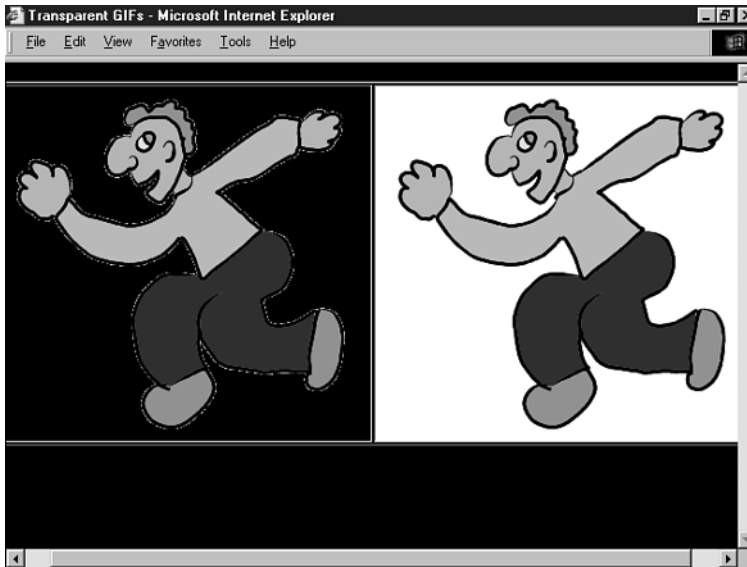


Abbildung 8.2:
Anti-Aliasing-
Effekte auf dem
transparenten
GIF.

Was ist zu tun, wenn Sie innerhalb Ihres Bildes die Farbe transparent dennoch verwenden wollen?

Angenommen, Sie möchten nun eine Zeichentrickfigur auf weißem Hintergrund haben. Die Augäpfel dieser Figur sollen aber weiß sein. Ganz einfach: Benutzen Sie reines weiß als Hintergrundfarbe (Red 255, Green 255, Blue 255) und *fast* weiß für die Augen (zum Beispiel R 255, G 255, B 250). Die Farben sehen zwar fast gleich aus, sind aber im Bildbearbeitungsprogramm verschieden.

Programme, die Ihnen beim Erstellen animierter GIFs helfen

Nehmen wir an, Sie haben eine Reihe Bilder erstellt, die Sie gerne in Ihren Animations-Frames haben möchten. Der nächste Schritt wäre, die einzelnen Bilder in einer einzigen animierten Datei zusammenzustellen. Eine große Anzahl von Werkzeugen steht Ihnen beim Erstellen animierter GIF-Grafiken zur Verfügung. Animierte GIFs

sind inzwischen schon so beliebt, dass viele Bildbearbeitungsprogramme mittlerweile, wie etwa Paintshop Pro ab Version 5 oder PhotoImpact mit Web-Extensions ab Version 3, auch Zusatzprogramme zum Erstellen animierter GIF-Grafiken enthalten. Diese Applikationen werden später noch genauer beschrieben.

Sie können sogar mit Präsentations-Programmen (zum Beispiel PowerPoint) Präsentationen erstellen und diese dann als animierte GIF-Dateien abspeichern.

Zwar kann man mit den meisten Bildbearbeitungsprogrammen solche GIF-Dateien erstellen, aber trotzdem kann man nicht mit allen Programmen animierte GIF-Dateien kreieren. Dann gibt es auch noch Programme, die das Erstellen von animierten GIF-Dateien unterstützen, allerdings die Bilder nicht selbst erstellen. Aber keine Angst: Falls Sie mit einem Grafikprogramm arbeiten sollten, das animierte GIFs nicht speichert, gibt es Lösungen. Auf den nächsten Seiten stellen wir Ihnen einige Zusatzprogramme vor, mit deren Hilfe Sie bereits vorhandene Grafiken zu animierten Grafiken zusammenstellen können.

GIF Construction Set (für Windows)

GIF Construction Set von Alchemy Mindworks ist eines der beliebtesten Shareware-GIF-Animationsprogramme für Windows (erhältlich für Windows 3.1 und Windows 95/NT). Mit dieser relativ preisgünstigen Shareware ist das Erstellen von animierten GIFs ein Kinderspiel: GIF-Dateien werden mit der Drag-And-Drop-Methode zusammengestellt. Clevere Features helfen Ihnen beim Organisieren der Farbpalette, wenn Sie Transparenz verwenden wollen, fügen Übergangseffekte hinzu, helfen Ihnen bei Verzögerungszeiten, konvertieren AVI Video Clips in animierte GIF-Grafiken, und vieles mehr.

GifBuilder (für Macintosh)

Für Macintosh erfreut sich dieses Programm von Yves Piquet größter Beliebtheit: GifBuilder. Hier handelt es sich um Scripting-fähiges Zusatzprogramm, mit dem Sie Grafiken in mehrere verschiedene Formate konvertieren können. Entweder ändern Sie schon existierende animierte GIFs oder aber sie fügen eine Sammlung von GIF-, TIFF-, PICT- oder Photoshop- (PDS) Dateien hinzu. Mit GifBuilder können Sie auch andere Animations-Formate erstellen, wie etwa QuickTime-Filme, PICS-Dateien, Adobe Premiere FilmStrip 1.0-Datei und die Ebenen eines RGB oder graustufende Adobe Photoshop 3.0-Datei. Die aktuelle Version dieses Programms ist 0.5. Sie verfügt über Icons in den Frame-Fenstern, für Funktionen wie Filter, Übergänge, Beschneiden einer Animation (cropping) und mehr.

Die Homepage hierzu finden Sie unter <http://iawww.epfl.ch/Staff/Yves.Piquet/clip2gif-home/GifBuilder.html>.

Übung 8.1: Erstellen einer Animation

Mit dieser Übung werden Sie 10 Bilder eines springenden Balles zusammenstellen. Dateinamen werden durchnummeriert, beginnend mit `ball01.tga` und endend mit `ball10.tga`.

Etwas später in diesem Kapitel wird Ihnen dann gezeigt, wie diese Bilder erstellt werden.

In dieser Übung wird Ulead PhotoImpact GIF Animator zum Zusammenstellen der GIF Animation verwendet. Sollten Sie einen anderen Windows Grafik Editor verwenden, der keinen GIF-Animator hat, können Sie eine Testversion von Ulead PhotoImpact GIF Animator von der Uleads Web Utilities Site unter http://www.webutilities.com/ga/ga_main.htm herunterladen.

Befehle und Terminologie unterscheiden sich vielleicht von denen Ihrer Software, doch das Konzept an sich ist sehr ähnlich. Jeder GIF Animations-Compiler geht im Grunde genommen auf die gleiche Art und Weise vor: Welches Bild soll eingefügt werden, in welcher Reihenfolge sollen sie eingefügt werden, Ablaufzeiten der Frames, usw...

Gehen Sie folgendermaßen beim Zusammenstellen einer GIF Animation mit Ulead GIF Animator vor:

1. Laden Sie die Testversion des Ulead GIF Animators herunter und wählen Sie **START | PROGRAMME | ULEAD GIF ANIMATOR 3.0A | GIF ANIMATOR**. Wenn der Startup-Assistent angezeigt wird, rufen Sie den Befehl **FILE | ANIMATION WIZARD** auf.
2. Im ersten Dialog mit dem Assistenten können Sie die Dateien wählen, die Sie in Ihrer Animation haben möchten. Klicken Sie dabei auf den **ADD IMAGE**-Button an und lokalisieren Sie Laufwerk und Datei für die Speicherung Ihrer Bild-Dateien. Rufen Sie alle Frames auf. Klicken Sie dabei den letzten Dateinamen an (`ball10.tga`), dann mit Shift-Klick den ersten Dateinamen. Wenn Sie **Open** anklicken, werden sämtliche Dateinamen auf einer Liste platziert. (Klicken Sie den ersten Dateinamen wie gewohnt an und den zweiten Dateinamen mit Shift-Klick, befindet sich der erste Dateiname dort, wo die letzte sein sollte und vice-versa. Gerade deshalb ist es so wichtig, Ihre Dateien in numerischer Reihenfolge abzuspeichern. Kleiner Tipp: Mit Drag-and-Drop können Sie die Dateinamen in der richtigen Reihenfolge anordnen.)
3. Klicken Sie auf **NEXT** für den nächsten Dialog mit dem Assistenten. Sie werden gefragt, ob die Animation text- oder photo-orientiert sein soll. Soll Ihre Animation photo-orientiert sein, *müssen die Bilder gedithert werden*. Mit »dither« werden

Farben simuliert, die nicht in der Farbpalette sind. Wenn sich Ihr Bild aus mehr als 256 Farben zusammenstellt, kann sich das als sehr nützlich erweisen. Wählen Sie also Dither und klicken Sie auf NEXT, um fortzufahren.

4. Im FRAME DURATION-Dialog legen Sie die Zeitverzögerung fest, also wie lange jedes Bild erscheinen soll bevor es durch das nächste ersetzt wird. Die Zeitverzögerung wird in 1/100 Sekunden bemessen oder mit der Anzahl der Frames die pro Sekunde erscheinen sollen, festgelegt. Legt man sich auf einen der beiden Werte fest, so wird der andere automatisch in der Relation angepasst. Eine kleine Vorschau der gewählten Ablaufzeit wird Ihnen zeigen, wie schnell Ihre Bilder abgespielt werden. Klicken Sie Next an.
5. Das war es. Der Animations-Assistent sagt Ihnen noch, wie er die Animation zusammenstellt. Sie müssen jetzt nur noch den Finish-Button anklicken. Nachdem der Animations-Assistent Ihre Datei erstellt hat, wird diese in einem Fenster angezeigt (Bild 8.3).

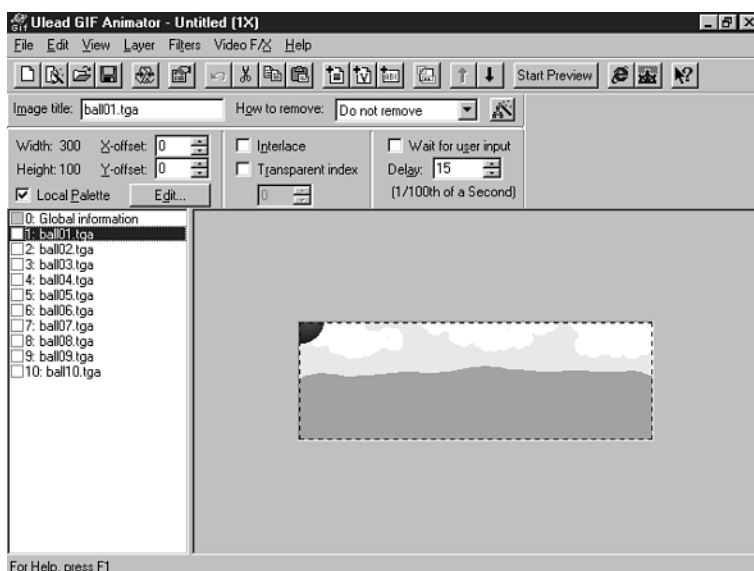


Abbildung 8.3:
Alle Animations-
Dateien in einer
einzigen Datei
zusammengestellt

6. Eine Vorschau Ihrer Animation erhalten Sie, wenn Sie den START PREVIEW-Button in der obersten Anzeigeleiste anklicken. Mit END beenden Sie die Vorschau.

Das waren jetzt nur einige Grundkenntnisse für das Erstellen und Kompilieren animierter GIF-Dateien. Jeder GIF-Animation-Compiler hat eigene zusätzliche Features. Es lohnt sich, sie näher zu untersuchen. Im Ulead GIF Animator können Sie Ihrer Animation Farbübergänge, Spezialeffekte und noch vieles mehr hinzufügen. Jedes Programm hat ganz spezifische Effekte. Probieren Sie sie doch einmal aus ...

Bevor Sie Ihre Ulead GIF Animator-Datei in einer GIF-Datei abspeichern, müssen Sie noch etwas über die Optimierung von GIF-Grafiken erfahren. Bevor der GIF Animator nämlich Ihre Datei abspeichert, verlangt er von Ihnen noch etliche Eingaben.

Durch die Optimierung verringern Sie die Größe Ihrer Animations-Datei. Dadurch wird die Ladezeit der Datei verkürzt. Das Konzept, das hinter dem Assistenten steht, ist nicht nur sehr wichtig, sondern auch sehr hilfreich. Allerdings verfügen nicht alle GIF-Animatoren über denselben Optimierungs-Assistenten. Folgende aufgeführte Konzepte werden Ihnen zeigen, wie Sie eine Animations-Datei gestalten können. Übung 8.2 wird also etwas schwieriger werden.

Verringerung der Animationsgröße

Können Sie sich noch an den tanzenden Schimpansen erinnern? Bevor die einzelnen Bilder gezeigt werden konnten, musste man zuerst die Dateigröße bestimmen, denn schließlich sollte die Ladezeit der Seite nicht zu lang sein.

Die erste Überlegung war also:

Wie groß soll die Datei sein? Die Ladezeit steht in Relation zum Dateiumfang, also je kleiner der Dateiumfang, umso kürzer die Ladezeit. Verschiedene Größen wurden ausprobiert und schließlich wurde eine 115 x 120 Pixel Animation gewählt. (Bei noch kleineren Abmessungen konnte man den Affen überhaupt nicht mehr erkennen).

Zweite Überlegung:

Wie viele Frames sollen in die Animation integriert werden? Wieder einmal musste probiert werden, um die optimale Anzahl zu finden. Der Schimpanse wurde mit einer Animations-Software erstellt, die für Videoproduktionen verwendet wird. Die Standard Geschwindigkeit eines Videofilms in den USA liegt bei ungefähr 30 Frames pro Sekunde. Aber diese Geschwindigkeit ist absolut ungeeignet für das Internet, weil die Ladezeit einfach viel zu lange wäre. 10 bis 15 Frames pro Sekunde sind da schon vernünftiger, und manchmal genügen sogar noch weniger Frames. Für die Schimpansen-Animation wurde mit 6 Frames die Sekunde gearbeitet, wobei nur jeder fünfte Frame der Animation wiedergegeben wurde.

Die Geschwindigkeit der animierten GIF-Datei musste den Frames angepasst werden. Dreißig Frames die Sekunde entsprechen ungefähr $3,33 \frac{1}{100}$ Sekunden pro Frame. Mit 5 multipliziert (jeder fünfte Frame wurde angezeigt) hat jeder Frame eine Ablaufzeit von etwa $16 \frac{1}{100}$ Sekunden, um den Affen in der richtigen Geschwindigkeit tanzen zu lassen.

Größe der Datei und Anzahl der Frames wurden reduziert. Die nächste Möglichkeit die Dateigröße zu verringern, also Speicherplatz zu sparen, erreicht man durch Farb-reduzierung. Normalerweise ist ein GIF-Bild aus 256 Farben zusammengesetzt. Ihre Datei wird verkleinert, wenn Sie die Anzahl der Farben Ihrer Farbpalette reduzieren. Tabelle 8.1 zeigt, wie Ladezeit durch Farboptimierung verkürzt werden kann (auf einer Datenübertragungsrate von 28.8 Kb pro Sekunde basierend). Bei folgenden Dateigrößen wurde die Option gewählt, nicht zu dithern (keine Erzeugung von Mischfarben, bzw. Grautönen).

Anzahl der Farben	Dateigröße	Ladezeit mit 28.8
256 Farbpalette	54,654 Bytes	23 Sekunden
Auf 128 Farben reduziert	50,290 Bytes	17 Sekunden
auf 64 Farben reduziert	41,637 Bytes	15 Sekunden
auf 32 Farben reduziert	34,772 Bytes	13 Sekunden
auf 16 Farben reduziert	28,324 Bytes	10 Sekunden

Tabelle 8.1: Farbreduzierung in der Schimpansen-Farbpalette

Mit 16 Farben sieht unser Schimpanse noch annehmbar aus, allerdings kann man in einigen Frames kleine Blitzer um den Schimpansen herum erkennen.

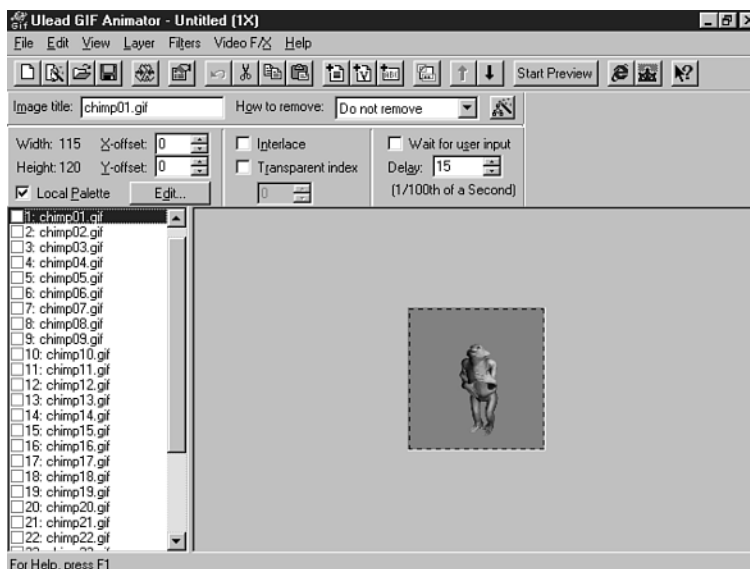


Abbildung 8.4:
Schimpansen-Ani-
mation mit 32
Farben, reduziert
die Dateigröße
um fast 30 Kb.

Die Version mit 32 Farben scheint also sowohl für das Erscheinungsbild, als auch für die Ladezeit am besten. Insgesamt wurde die Dateigröße um 20K verringert und die Ladezeit um 10 Sekunden mit 28.8k verkürzt. Überzeugen Sie sich selbst vom Ergebnis (Bild 8.4).

Übung 8.2: Eine Animation optimieren

Die Frames Ihrer Ball-Animation haben noch eine True-Color-Palette (wirklichkeitsgetreu; Farbdarstellung mit einer Farbtiefe von 24 Bit). Da es eher unwahrscheinlich ist, dass in Ihrer Animation 256 unterschiedliche Farben Verwendung finden, können Sie die Ladezeit erheblich durch Farbreduzierung verringern.

Sehen Sie nun welche Auswirkungen eine Farboptimierung auf Ihre Animation mit 10-Frames in der Größe 300x100-Pixel hat. In Tabelle 8.2 werden Ihnen die Unterschiede der Dateigrößen aufgezeigt, wenn man die Anzahl der Farben in der Farbpalette reduziert. Die Ladezeiten basieren dabei auf einem 28.8 Kb pro Sekunde Modem.

Anzahl der Farben	Dateigröße	Ladezeit mit 28.8
256 Farbpalette	9,476 Bytes	3 Sekunden
reduziert auf 128 Farben	8,765 Bytes	3 Sekunden
reduziert auf 64 Farben	8,418 Bytes	2 Sekunden
reduziert auf 32 Farben	6,971 Bytes	2 Sekunden
reduziert auf 16 Farben	5,552 Bytes	1 Sekunde

Tabelle 8.2: Ergebnisse der Ball-Animation, wenn man die Farbpalette reduziert

Zwei Sekunden Ladezeit erscheint zwar nicht viel, aber wenn Sie verschiedene Grafiken auf Ihrer Seite haben, zählt jedes einzelne Byte. 16 Farben reichen für Ihren springenden Ball vollkommen aus. Natürlich ist die 32-Farben-Version besser, doch ist sie wirklich die 1.5Kb wert? Sie selbst müssen das entscheiden und es kommt auch darauf an, wie wichtig die Animation für Ihre Seite ist. Animationen, die eine Fotoqualität aufweisen sollen, sehen natürlich nicht so toll aus, wenn man die Anzahl der Farben reduziert. In diesem Fall können Sie die Dateigröße nur verringern, wenn Sie die Frame-Anzahl oder die Abmessungen Ihrer Animation reduzieren.

Folgende Schritte sind im Ulead GIF Animator zu beachten, wenn Sie die Anzahl der Farben Ihrer Animation optimieren wollen:

1. Wählen Sie FILE | OPTIMIZATION WIZARD (Optimierungs-Assistent) oder drücken Sie auf F11. Der erste Dialog des Optimierungs-Assistenten erscheint.

2. Anschließend werden Sie gefragt, ob Sie eine Super-Palette für Ihre Animation erstellen wollen. Bestätigen Sie dies, denn mit der Super Palette werden die geeignetsten Farben für Ihre gesamte Animation ausgesucht. Wählen Sie keine Super-Palette, wenn Sie drastische Farbänderungen zwischen den einzelnen Bildern in Ihrer Datei haben (Blinkeffekte, starke Farbübergänge, usw.). In der Regel ist die Dateigröße geringer, wenn Sie die Super-Palette verwenden. Nachdem Sie YES angeklickt haben, klicken Sie NEXT an, um fortzufahren.
3. Der Optimierungs-Assistent fragt Sie, wie viele Farben Sie in der Super-Palette haben möchten. Geben Sie 32 in das Farbfeld ein. Wählen Sie als nächstes die YES für die Option Dithern in der Super-Palette der photo-orientierten Bilder, dann NEXT.
4. Sie haben zwei Optionen zur Bestimmung Ihrer Dateigröße. Die erste fragt Sie, ob Sie die überflüssigen Pixel entfernen wollen. Wenn ein Pixel in verschiedenen Frames die gleiche Farbe hat, warum ihn dann wieder zeichnen anstatt ihn zu entfernen? Wählen Sie YES (Standardeinstellung).
5. Der Assistent zeigt Ihnen noch einmal sämtliche Entscheidungen, die Sie getroffen haben. Ist alles so, wie Sie es wollten, klicken Sie FINISH zum Optimieren Ihrer Animation an. Der in Bild 8.5 abgebildete Bildschirm zeigt Ihnen die Speicher- und Ladezeiten an. Mit Anklicken des PREVIEW-Buttons, können Sie Ihre Animation sehen. Das Preview-Fenster wird geschlossen, wenn Sie den sich oben rechts befindenden X-Button anklicken.
6. Klicken Sie nun auf den SAVE AS-Button (Speichern unter) des GIF Optimierungs-Dialogs. Speichern Sie die Animation als `bouncing-32.gif` ab.

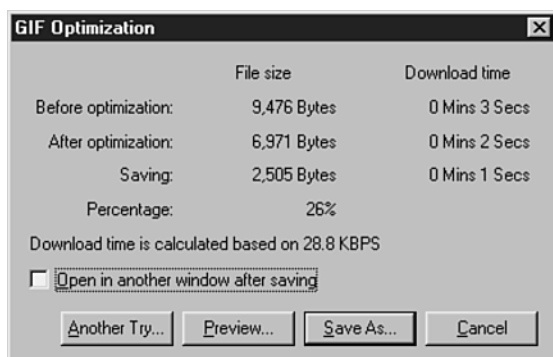


Abbildung 8.5:
Der Assistent zeigt das Ergebnis der Optimierung und was gespeichert wurde.

Werkzeuge, die Ihnen beim Erstellen Ihrer Bilder helfen

Baby Cha und der Schimpanse sind Beispiele dafür, was Sie heutzutage mit Bildbearbeitungs-Software machen können. Sie brauchen jedoch nicht wirklich teure Software zum Erstellen von Animationen. Einige sehr gute Bild-Editoren sind in Form von preiswerter Shareware zu erhalten. Sie können Sie vom Internet herunterladen und deren Features beurteilen, bevor Sie sie kaufen. Sollten Sie ein Standardprodukt mit Benutzerhandbuch bevorzugen, gibt es auch hier preisgünstige und leistungsfähige Produkte.

Wie schon erwähnt, ist Paint Shop Pro 5.0. eines der beliebtesten Bildbearbeitungsprogramme für Windows 95/98 und Windows NT. Dieses Shareware Programm verfügt ebenfalls über einen Compiler für animierte GIFs, den Animation Shop, der Ihnen Schritt für Schritt Anweisungen zum Erstellen animierter GIF Dateien gibt. Die Features sind denen des Ulead Animators sehr ähnlich. Die aktuelle Version von Paint Shop Pro (und Animation Studio) können Sie unter <http://www.jasc.com/psp5.html> herunterladen.

Ulead PhotoImpact (<http://www.ulead.com/pi/runme.htm>) ist ein kommerzieller Grafikeditor, der ebenfalls über viele Editier- und Animationsfähigkeiten verfügt. Das Begleitprogramm von Ulead PhotoImpact, GIF Animator, haben Sie ja schon in diesem Kapitel kennen gelernt. Beide Programme können 30 Tage lang getestet werden. Sie müssen sie nur von der ULead Webseite herunterladen.

CNET verfügt auf seiner Web-Site über eine große Anzahl von Demoprodukten und Shareware-Programmen, die man herunterladen kann. Die Download.com-Site (Bild 8.6) bietet Ihnen einfachen Zugang zu Informationen über alle nur vorstellbaren Software-Programme. Einige kommerzielle Software-Angebote kann man nicht abspeichern oder sind zeitlich begrenzt, andere hingegen sind Shareware-Applikationen, die vor dem Kauf getestet werden können. Auf dieser Site wird angezeigt, wie oft Informationen zu einem bestimmten Programm heruntergeladen wurden. Sie können also mit einem Blick feststellen, welche am beliebtesten sind. Die meisten der angebotenen Produkte verfügen über eine übersichtliche Liste der jeweiligen Features. Bevor Sie also den DOWNLOAD-Button anklicken, können Sie schon einmal eine Übersicht darüber gewinnen.

Für Animations-, Bildbearbeitungs- oder Multimedia-Programme gehen Sie in die Multimedia und Design Kategorie der Download.com-Site (für PC oder Macintosh). PC-Programme findet man unter

<http://www.download.com/pc/fdoor/0,322,0,00.html?st.dl.mfd.platforms.fd>.

Macintosh findet man unter

<http://www.download.com/mac/fdoor/0,322,0,00.html?st.dl.fd.platforms.fd>.

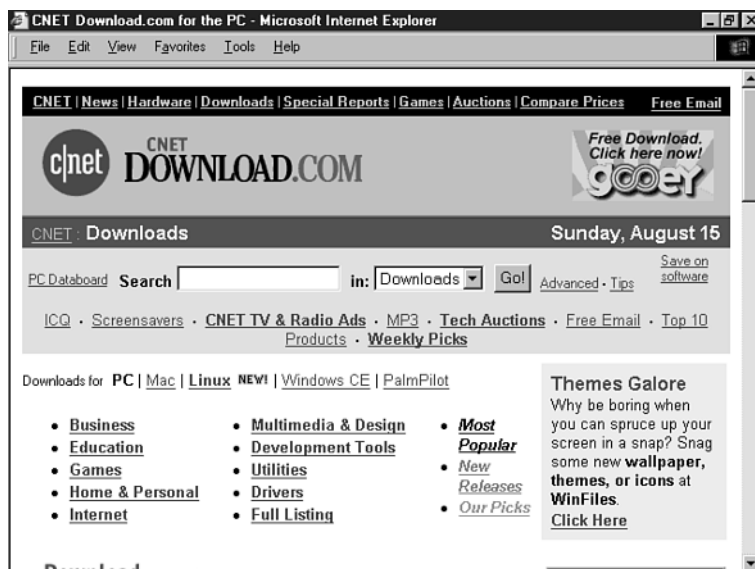


Abbildung 8.6:
CNETs Down-
load.com enthält
eine Vielzahl von
Quellangaben der
Image-Editing-
und Animations-
Software.

Um CNETs Shareware-Produkte herunterzuladen: <http://www.shareware.com/> oder <http://home.cnet.com/category/0-3880.html>. Shareware.com bietet Ihnen über 250 000 verschiedene Shareware Titel an.

Nützliche Software-Features

Nicht alle Grafikprogramme sind gleich und es gibt deren so viele, dass man sich schwer tut, das Beste herauszufinden. Einige Features sind aber wirklich von Vorteil, wenn man Animationen für das Web erstellen will. Was Ihre Software enthalten sollte, sehen Sie in der folgenden Aufzählung:

- ▶ Die Fähigkeit Bilder zu *öffnen*, zu erstellen und abzuspeichern, und dies in einer Vielzahl von Dateiformaten. Am wichtigsten ist das Erzeugen und Abspeichern von GIF und JPEG Bildern für Webseiten.
- ▶ Die Fähigkeit die Anzahl der Farben in einem 256-Farben-Gif Bild zu reduzieren. Wie Sie schon in diesem Kapitel erfahren haben, ist das Abspeichern eines Bildes, das nur 3 oder 4 Farben hat, mit einer 256-Farben-Palette eine enorme Verschwendung von Bytes.
- ▶ Die Fähigkeit mit ausgewählten Bereichen, Objekten und Ebenen zu arbeiten. Mit ausgewählten Bereichen haben Sie die Möglichkeit, mit nur einem Teil des Bildes zu arbeiten, ohne dabei den Rest des Bildes zu beeinflussen. Objekte schweben in

einem Bild und können von Ihnen leicht neu platziert und Ihrer Seite angepasst werden. Mit Ebenen können Sie die Objekte vor- oder hintereinander platzieren.

- Die Fähigkeit ein Bild mit transparenten Stellen abzuspeichern. Beim Erstellen von transparenten GIF-Dateien ist dies besonders wichtig. Diese erscheinen dann, als ob sie auf Ihrer Webseite schweben würden. Es wäre natürlich von Vorteil, wenn Ihr GIF-Animations-Compiler ebenfalls solche Features hätte.

Nachdem Sie Ihren Bild-Editor gefunden haben, können Sie sich ans Werk machen. Dafür benötigen Sie nur noch Gespür für Bewegung, Geduld und viel Kreativität.

Übung 8.3: Erstellen der Frames für die Ball-Animation

Für diese Übung wurde ULead PhotoImpact verwendet. Natürlich können Sie auch ein anderes Grafikprogramm benutzen. Das Konzept an sich ist so ziemlich das Gleiche. Sicherlich ist es eine Herausforderung mit einer neuen Software zu arbeiten, und inzwischen sollten Sie in der Lage sein, folgendes Beispiel mit Ihrem eigenen Bild-Editor zu bewältigen. Sinn und Zweck dieser Übung ist es, Anregungen zu geben und auf Software-Features aufmerksam zu machen, die Ihre eigene Bildbearbeitungssoftware haben sollte. Die springender Ball-Animation sollte aber generell keine größeren Schwierigkeiten darstellen.

Am einfachsten ist es, die Animation mit den Stellen zu beginnen, die in jedem Frame gleich bleiben. Erstellen Sie zuerst ein 300 x 100 Pixel True-Color-Bild auf weißem Hintergrund. (Wenn Sie mit True-Color arbeiten, haben Sie in jedem Bildbearbeitungsprogramm Zugang zu sämtlichen Effekten.) Mit dem Paint-Werkzeug (3. Button von unten der linken Werkzeug-Leiste) zeichnen Sie den Himmel und Boden so, wie er in Bild 8.7 abgebildet ist.

Mit den meisten Bildbearbeitungsprogrammen können Sie die Größe Ihres Pinsels einstellen. In PhotoImpact können Sie nicht nur die Größe des Pinsels bestimmen, sondern auch Farben von der Kontroll-Leiste (oberer Teil des Bildschirms; siehe Bild 8.7) auswählen. Mit dem DELETE-Feature können Sie alles ausradieren bzw. löschen, was Ihnen nicht gefällt. Hier genügt es EDIT | UNDO BEFORE zu wählen.

Bis jetzt gar nicht so schwer, oder? Unser nächster Schritt besteht darin, den Ball zu erstellen. Hierfür müssen Sie einen Kreis zeichnen und ihn mit Farbe füllen. Das ist übrigens ein Feature, das es in den meisten Image-Editoren gibt. Mit PhotoImpact geht das ganz leicht. Erstellen Sie nun ein zweites True-Color-Bild, das 40 x 40 Pixel misst. Die Hintergrundfarbe ist nicht so wichtig, da sie eventuell transparent sein wird, aber lassen Sie sie zunächst einmal weiß. Gehen Sie auf Selection-Tool und ändern Sie die Form des ausgewählten Bereichs in einen Kreis.

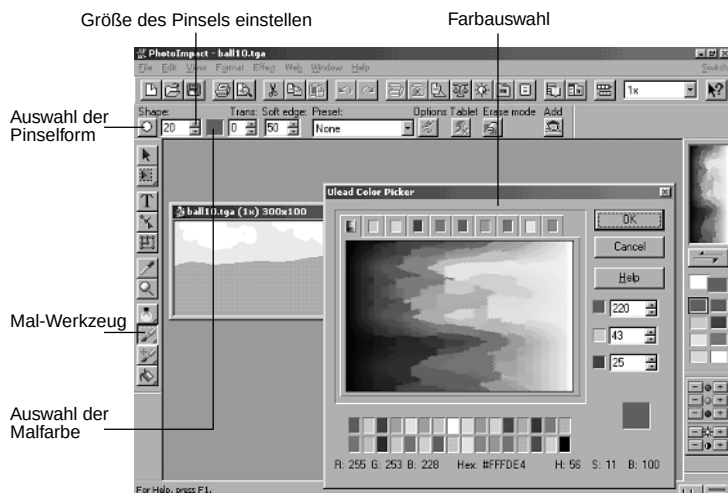


Abbildung 8.7:
Beginnen Sie mit
dem Hintergrund
und den Elementen,
die sich nicht
ändern.

Für das Erstellen einer elliptischen Selektion (ausgewählter Bereich), müssen Sie in der Ecke links oben des kleinen Bildes beginnen. Ziehen Sie Ihre Maus oder Ihren Stift zur Ecke nach rechts unten, und zwar so lange bis der Kreis das gesamte Bild einnimmt. Lassen Sie die Maus los. Der Kreis wird auf einer Seite mit roter Farbe und auf der anderen Seite mit blauer Farbe abgegrenzt. Das bedeutet, dass der gewählte Bereich aktiviert wurde (siehe Bild 8.8).

Der nächste Schritt ist, den Inhalt des Kreises zu füllen. Es mag sein, dass sich die Vorgehensweise bei PhotoImpact von der bei anderen Bildbearbeitungsprogrammen unterscheidet. Üblicherweise werden mit den Befehlen PAINT BUCKET (Farbeimer) oder FILL (Füllen) die Objekte gefüllt. In PhotoImpact gehen Sie auf ELLIPTICAL GRADIENT FILL in der Werkzeug-Leiste (klicken Sie auf FILL und bleiben Sie so lange darauf, bis das Flyout-Menü geöffnet wird. In diesem Menü können Sie dann ELLIPTICAL GRADIENT FILL wählen.) Füllen Sie Ihren Kreis mit zwei Farben (Two-Color Fill method) und verwenden Sie dabei rot und schwarz als Verlaufsfüllung.

Platzieren Sie das Farbfüll-Werkzeug nun an die Stelle des Balles, den Sie hellrot haben möchten (zum Beispiel überhalb und rechts vom Zentrum des Balles). Klicken Sie die Außenlinie des Balles an und ziehen Sie sie so lange, bis der Ballinhalt komplett gefüllt ist (siehe Bild 8.9). Das Glanzlicht wird nämlich von der Größe der elliptischen Selektion beeinflusst, das heißt, je größer die Selektion, desto heller das Glanzlicht. Wenn Sie die Mouse nun loslassen, sehen Sie einen schönen dreidimensionalen Ball.

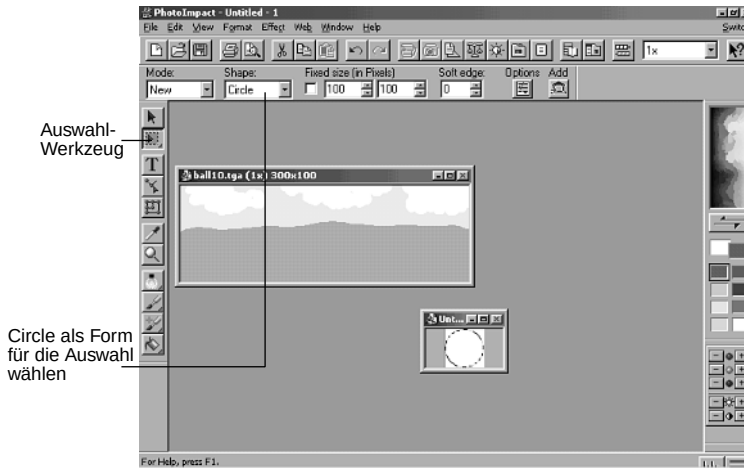


Abbildung 8.8:
Erstellen einer
elliptischen
Selektion in
einem zweiten,
kleinerem
Fenster

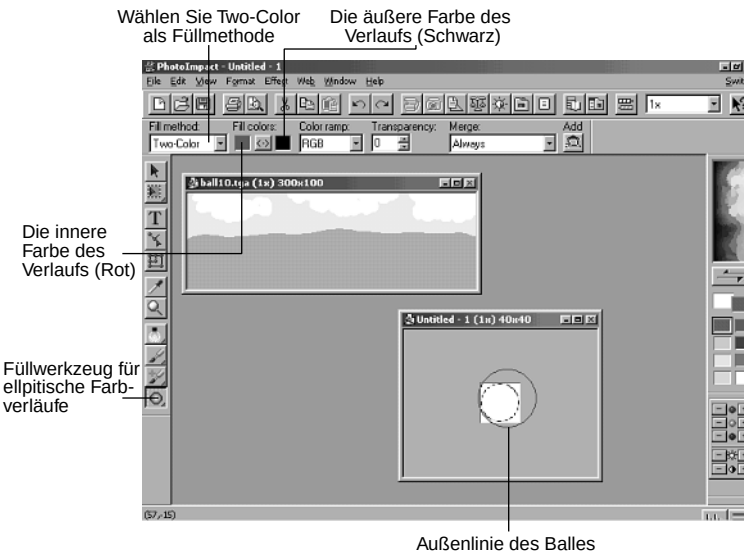


Abbildung 8.9:
Füllen Sie die
elliptische Aus-
wahl mit einem
Verlauf.

Der Ball muss nun in den Hintergrund gebracht werden. Das ist ganz einfach! Mit **Ctrl** + **C** kopieren Sie die elliptische Selektion in Ihr Clipboard. Gehen Sie nun in das Hintergrundbild (hierfür genügt es die Titelleiste anzuklicken, um es zu aktivieren). Mit **Ctrl** + **V** wird der Ball als Objekt in den Hintergrund eingebunden. Mit dem **PICK**-Werkzeug können Sie den Ball platzieren (siehe Bild 8.10.)

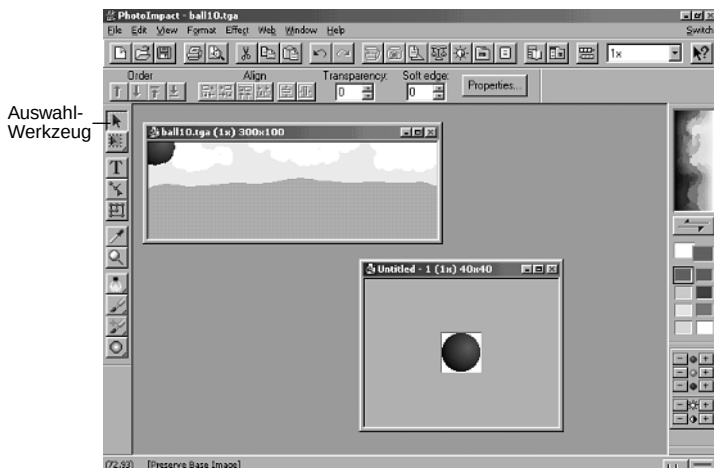


Abbildung 8.10:
Kopieren Sie den
Ball und fügen Sie
ihn in das Hinter-
grundbild ein.

Beim Erstellen einer Animation wird normalerweise jeder Frame in True-Color-Format abgespeichert und das Gif-Animation-Programm wählt dann die beste Farbpalette für alle Frames aus. Das ist leider nicht immer möglich. Wenn Sie nämlich Stellen Ihrer Animation transparent haben möchten, müssen Sie in einigen Programmen zuerst das Bild auf 256 Farben verringern, bevor Sie die Farbe transparent selektieren und in einer GIF-Datei abspeichern können.

Die springender Ball-Animation enthält keine transparenten Stellen. Die Paletten sind außerdem verlässlicher, wenn man mit True-Color-Bildern beginnt. Um den ersten Frame zu speichern, klicken Sie auf File | SAVE AS. Wählen Sie TGA (Targa File Format; siehe Bild 8.11) im SAVE AS TYPE-Feld. Speichern Sie die Datei unter einem Namen ab, an den Sie sich leicht erinnern können (zum Beispiel `ball101.tga`) und stellen Sie sicher, dass sich nach dem Dateinamen genügend Stellen befinden, um auch die Gesamtanzahl der Frames, die Sie erstellen möchten, unterbringen zu können. Beispiel: Wenn Sie eine 10-Frame-Animation erstellen, geben Sie der ersten Datei den Namen `ball101.tga` und nummerieren Sie jede folgende Datei durch (`ball102.tga`, `ball103.tga`,...), wobei `ball110.tga` der letzte Dateiname ist.



Hinweis

In PhotoImpact werden Sie beim Abspeichern Ihres Bildes gewarnt, dass das schwebende Objekt mit dem Hintergrund vermischt wird. Fahren Sie mit YES ruhig fort, denn solange sie Ihre Originaldatei beim Erstellen der restlichen Frames geöffnet haben, wird der Ball weiterhin schweben. Für alle Fälle speichern Sie einen Frame in UFO- (Ulead File for Objects) oder Native-Format, das von Ihrem Image-Editor verwendet wird, wenn er schwebende Objekte unterstützt.

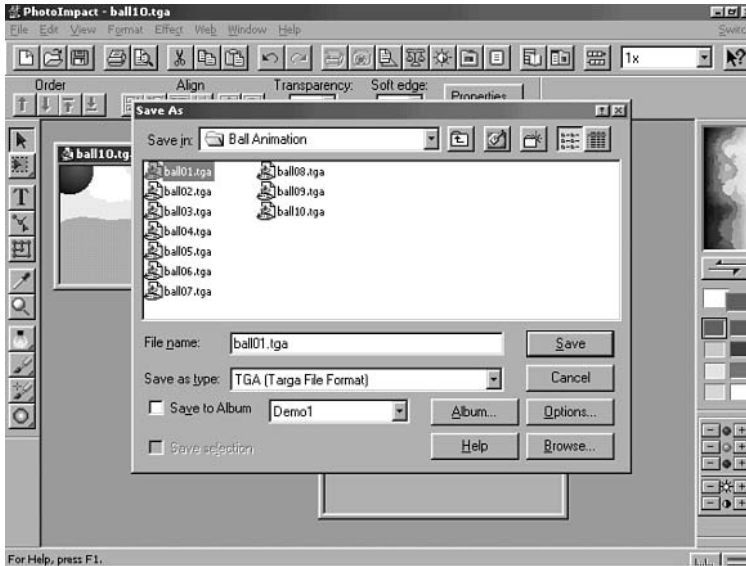


Abbildung 8.11:
Speichern des ersten Frames der Animationen unter einem Dateinamen, der mit einer Nummer endet.

Benutzen Sie das PICK-Werkzeug für den nächsten Frame und platzieren Sie den Ball nach rechts unten (siehe Bild 8.12) Speichern Sie diesen Frame unter ball102.tga.

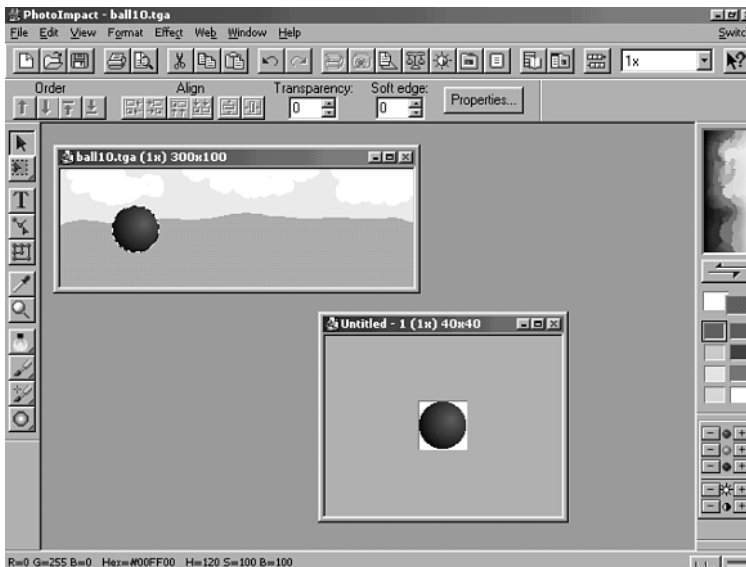


Abbildung 8.12:
Positionieren Sie den Ball mit Hilfe des Pick-Werkzeuges im nächsten Frame und speichern Sie die Datei unter einem neuen Namen ab.

Ahnen Sie schon wie es weitergeht?

In jedem Frame positionieren Sie den Ball anders und simulieren somit eine springende Bewegung – von oben nach unten; von links nach rechts. Erstellen Sie die restlichen Frames Ihrer Animation in der gleichen Art und Weise und nummerieren Sie jede Sequenz so lange durch, bis der Ball aus der rechten Seite des Hintergrundbildes springt. Im letzten Frame wird der Ball mit DELETE gelöscht. Speichern Sie die letzte TGA- Datei ab. In Bild 8.13 sehen Sie zusätzliche Frames. Damit Sie besser erkennen können wie die Frames zueinander in Relation stehen, werden sie mit halber Größe abgespielt.

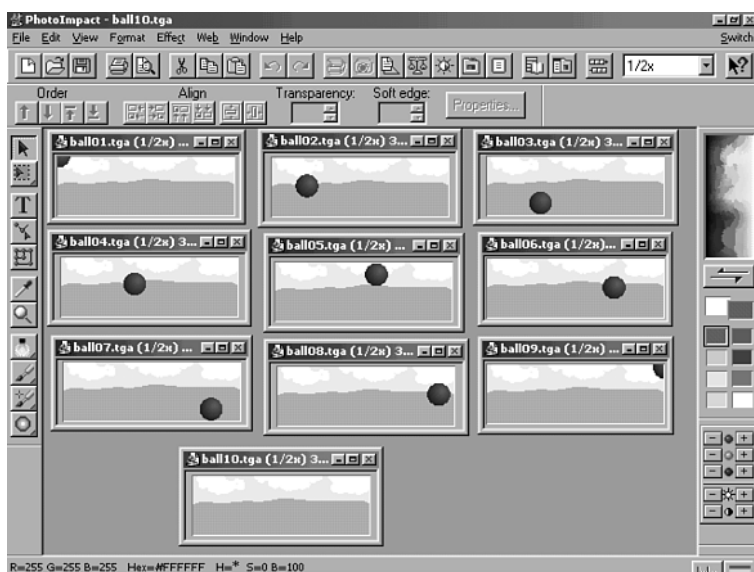


Abbildung 8.13:
Erstellen der restlichen Frames.
Jeder Frame muss in numerischer Reihenfolge abgespeichert werden.

Zusammenfassung

Hoffentlich waren die Übungen in diesem Kapitel nicht zu anstrengend und haben Ihnen gefallen. Die relativ einfachen Beispiele sollten Ihnen einen Überblick über die Basiskonzepte verschaffen (und Sie in Ihrem Lernprozess weiterbringen). Lassen Sie sich nicht entmutigen, falls es beim ersten Mal nicht geklappt hat – man muss eben ein bisschen üben. Lassen Sie nicht locker und geben Sie nicht auf, es wird schon werden! Untersuchen Sie auch die Features und deren Möglichkeiten Ihrer eigenen Software – jedes Programm hat seine eigene Trickkiste, die Ihnen beim Erstellen einer Animation hilft.

Workshop

Die wichtigsten Punkte beim Erstellen einer Animation für das Web werden hier noch einmal zusammengefasst. Außerdem werden Sie einige zusätzliche Übungen zur nächsten Animations-Ebene führen.

Fragen und Antworten

F Was muss ich tun, um am Ende meiner animierten GIF-Datei eine kleine Pause zu setzen? Die Datei sollte nach dieser Pause wieder von vorne abgespielt werden.

A *Mit den meisten Compilern für animierte GIFs kann man die Abspielzeit eines jeden Frames einstellen. Sie müssen also nur die Abspielzeit des letzten Frames Ihrer Animation verlängern. Zum Beispiel müssen Sie für eine 1 Sekunde lange Pause 100/100 Sekunden eingeben. Alle anderen Frames werden mit der ursprünglichen Abspielzeit abgespielt und nur der letzte Frame wird eine Sekunde lang angezeigt, bevor die Animation wieder von vorne beginnt.*

F Kann man einen animierten Hintergrund für die Webseite erstellen?

A *Sicher, allerdings ist das keine sehr gute Idee. Animierte Wolken oder blinkende Sterne sehen bestimmt sehr nett aus, lenken aber von der eigentlichen Information der Webseite ab. Halten Sie sich immer vor Augen, was auf Ihrer Seite wichtig ist und betont werden soll, und verwenden Sie Animationen nur, um dies hervorzuheben.*

F Wie erstellt man Animationen in anderen Dateiformaten, wie etwa Video für Windows (AVI), Quick Time oder MPEG?

A *Beim Erstellen von Animationen und Dateiformaten handelt es sich eigentlich um den gleichen Vorgang. Die Frames müssen einzeln erstellt werden, allerdings benötigen Sie hierfür ein Softwareprogramm, das die Animationen in der gewünschten Form speichert.*

Quiz

1. Was unterscheidet eine animierte GIF-Datei von einer regulären GIF-Datei?
2. Wie fängt man am einfachsten mit der Animation an?
3. Nennen Sie die drei Beispiele, mit denen man die Ladezeit der Animations-Datei verringern kann.

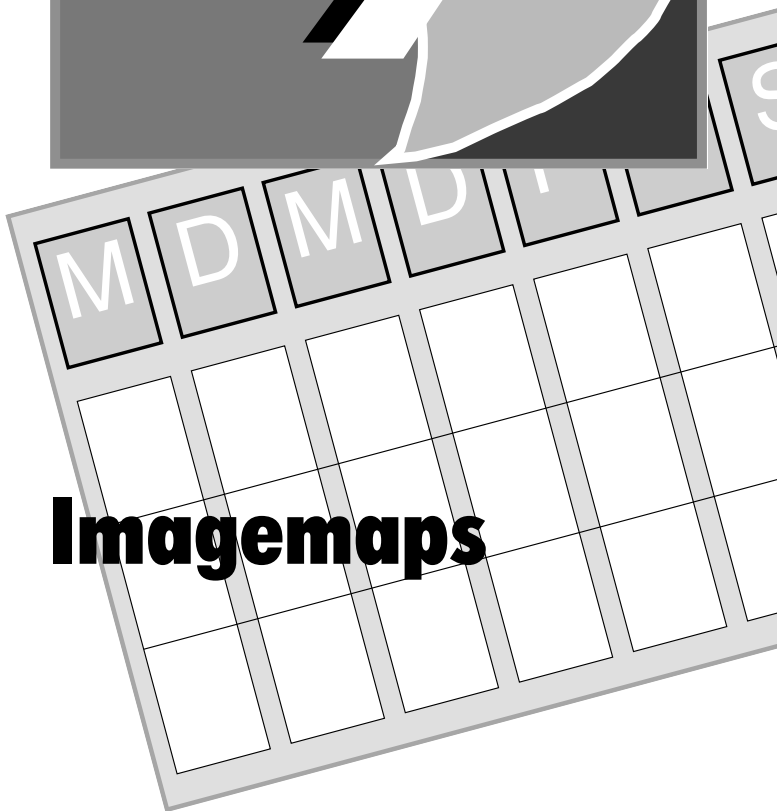
4. Warum sollte man die einzelnen Frames mit nummerierten, aufeinander folgenden Dateinamen abspeichern?
5. Wann sind Animationen für eine Webseite am geeignetsten?

Antworten

1. Eine animierte GIF-Datei enthält mehrere Bilder, die in einer einzigen Datei kompiliert sind. Sie werden in Sequenzen abgespielt und erzeugen die Illusion einer Bewegung.
2. Der einfachste Weg, eine Animation zu erstellen, ist, mit den Stellen des Bildes zu beginnen, die in jedem Frame gleich bleiben.
3. Wenn man die Größe der Datei, die Anzahl der Frames einer Animation oder die Anzahl der Farben einer Animation reduziert, kann man die Ladezeit der Animationsdatei verringern.
4. Einige Animationsprogramme ändern die Reihenfolge der Frames, wenn Sie mehr als einen Frame zur gleichen Zeit wählen. Wenn Sie Ihre Frames durchnummerieren, können Sie sofort feststellen, ob dies der Fall ist.
5. Animationen sind dafür gedacht, die Aufmerksamkeit auf eine wichtige Information oder Stelle Ihrer Webseite zu ziehen.

Übungen

1. Wie man Bewegung simuliert haben Sie ja bereits gelernt. Versuchen Sie es doch mit einem herausfordernden Beispiel. Erstellen Sie ein einfaches Gesicht. Die Augen sollen blinzeln und der Gesichtsausdruck soll zwischen unfreundlich und lächelnd wechseln.
2. Erstellen Sie ein Text-Banner. Animieren Sie den Text durch wechselnde Farben oder wechselnde Position oder beides (von Frame zu Frame).



Imagemaps

**Woche
2**

Imagemaps (Abbildungspläne) sind spezielle Arten von anklickbaren Bildern. Wenn Sie normalerweise ein Bild innerhalb eines Links einbetten, gelangen Sie zu einem einzigen Ort, wenn Sie das Bild irgendwo anklicken. Mit Abbildungsplänen können Sie zu verschiedenen Orten gelangen, je nachdem, welche Stelle innerhalb des Bildes Sie angeklickt haben. In diesem Kapitel werden Sie alles über Abbildungspläne und wie Sie sie erzeugen können lernen, einschließlich:

- ▶ Was ein Abbildungsplan ist
- ▶ Wie Sie serverseitige Abbildungspläne erzeugen
- ▶ Wie Sie clientseitige Abbildungspläne erzeugen
- ▶ Wie Sie beide Typen von Abbildungsplänen einsetzen können

Was ist ein Abbildungsplan?

In Kapitel 7 haben Sie erfahren, wie eine Abbildung zugleich auch als Verknüpfung dienen kann, nämlich einfach, indem Sie die `<img>`-Kennung innerhalb eines `<a>`-Tag verwenden. Auf diese Weise wird die ganze Abbildung zu einer Verknüpfung; Sie könnten die Abbildung anklicken, den Hintergrund oder den Rahmen, alles hätte denselben Effekt.

Bei einem Abbildungsplan (*Imagemap*) aktivieren verschiedene Teile der Abbildung unterschiedliche Verknüpfungen (Abbildung 9.1). Mit Hilfe von Abbildungsplänen könnten Sie eine verknüpfte Landkarte erzeugen, die mit Seiten verbunden ist, die die jeweilige Region beschreiben, die Sie anklicken (siehe Abbildung 17.1). Oder Sie könnten visuelle Metaphern für die von Ihnen präsentierten Informationen erstellen: ein paar Bücher im Regal oder eine Fotografie, bei der jede abgebildete Person einzeln beschrieben wird.



Abbildungspläne sind spezielle Bilder, in denen unterschiedliche Regionen auf unterschiedliche Verknüpfungsorte hinzeigen. Wohin Sie sich auf der Site bewegen, ist abhängig von der Stelle, die Sie auf dem Bild anklicken.

Es gibt zwei verschiedene Imagemaps: Server-Side-Imagemaps und Client-Side-Imagemaps. Server-Imagemaps wurden früher verwendet, aber sie stellten einige Probleme für Web-Autoren dar, über die wir in diesem Kapitel noch sprechen werden. Heutzutage kommen Client-Side-Imagemaps, die von Browsern bearbeitet werden, häufiger vor, und bieten viele Vorteile gegenüber älteren Server-Side-Imagemaps.



Abbildung 9.1:
Abbildungspläne:
unterschiedliche
Orte, unterschied-
liche Verbindun-
gen



Serverseitige Abbildungspläne setzen sich aus zwei Teilen zusammen: einem Bild, das im Browser des Anwenders angezeigt wird, und einem Programm auf dem Webserver, das die Information verarbeitet, wo die Abbildung angeklickt wurde.

Clientseitige Abbildungspläne funktionieren wie serverseitige Abbildungspläne, außer dass kein Programm auf dem Server nötig ist. Die Verarbeitung der Koordinaten und der Links erfolgt vollständig im Browser.

Obwohl serverseitige Abbildungspläne nicht so oft verwendet werden, ist das Kennen lernen beider Imagemaptypen von großem Vorteil. Wenn Ihre Abbildungspläne abwärtskompatibel sein sollen, müssen Sie über beide Methoden, die in diesem Kapitel aufgeführt werden, Bescheid wissen.

Serverseitige Abbildungspläne

Als Abbildungspläne zum ersten Mal im Web erschienen, wurden diese mit speziellen Programmen realisiert, die auf dem Server liefen. Solche Imagemaps werden als serverseitige Abbildungspläne bezeichnet.

Wenn ein Browser ein Link auf einem Abbildungsplan aktiviert, ruft er ein spezielles Abbildungsplan-Programm auf, das auf dem Web-Server gespeichert ist. Zusätzlich dazu, dass der Browser das Abbildungsplan-Programm aufruft, sendet er ihm auch noch

die x- und y-Koordinaten der Position auf dem Bild, an die mit der Maus geklickt wurde. Das Abbildungsplan-Programm sucht dann nach einer speziellen Plandatei, in der die Regionen auf dem Bild bestimmten URLs zugeordnet werden, ermittelt welche Seite geladen werden soll, und lädt dann die entsprechende Seite. Dennoch gibt es Probleme, die mit serverseitigen Abbildungsplänen verbunden sind. Die folgende Liste führt diese auf:

Serverseitige Abbildungspläne gehören zu den ersten Web-Features und werden von den meisten, wenn auch nicht von allen, Browsern unterstützt.

- ▶ Normalerweise, wenn Sie Ihren Cursor über ein Link ziehen, wird die URL, auf den der Link verweist, in der Statusleiste des Web-Browsers angezeigt. Da der Web-Browser jedoch keine Ahnung davon hat, wohin die Teile eines serverseitigen Abbildungsplans zeigen, ist alles, was Sie zu sehen bekommen, wenn Sie Ihren Cursor über einen serverseitigen Abbildungsplan bewegen, entweder die URL des Abbildungsplan-Programms selbst (nicht sehr hilfreich) oder die URL zusammen mit einigen x, y-Koordinaten (auch nicht besonders hilfreich).
- ▶ Es gibt keine Methode, mit der Sie serverseitige Abbildungspläne auf lokale Dateien anwenden oder diese testen können. Abbildungspläne erfordern die Anwendung eines Web-Servers, der das Abbildungsplan-Programm ausführt und die x- und y-Koordinaten verarbeitet.
- ▶ Da jedes Mal, wenn ein User auf eine Seite mit einem Abbildungsplan klickt, ein spezielles Programm vom Server ausgeführt werden muss, reagieren Abbildungspläne viel langsamer auf Mausklicks als normale Links oder verknüpfte Bilder. Dies resultiert oft darin, dass Abbildungspläne ewig lange zu brauchen scheinen, um auf die Anforderung einer neuen Seite zu reagieren.

Clientseitige Abbildungspläne

Obwohl serverseitige Abbildungspläne seit geraumer Zeit gebräuchlich sind, haben die damit verbundenen Probleme zur Entwicklung eines neuen Typs von Abbildungsplan geführt, dem clientseitigen Abbildungsplan. Clientseitige Abbildungspläne beseitigen alle diese Schwierigkeiten, indem sie die Notwendigkeit für ein spezielles Abbildungsplan-Programm auf dem Server auflösen. Stattdessen führen sie die gesamte Abbildungsplan-Verarbeitung lokal auf dem Browser selbst aus. Das Ergebnis ist, dass die meisten Web-Designer heutzutage diese Methode verwenden.



Clientseitige Abbildungspläne werden von allen aktuellen Browsern unterstützt. Der Vorschlag für clientseitige Abbildungspläne hat Eingang in HTML 3.2 Spezifikation und ist ebenfalls Teil von HTML 4.0.

Nun kennen Sie also den grundlegenden Unterschiede zwischen serverseitigen- und clientseitigen Abbildungsplänen. Später in diesem Kapitel werden Sie lernen, wie Sie beide jeweils erstellen und verwenden können.

Abbildungspläne und Textbasierte Browser

Aufgrund der an sich grafischen Natur von Abbildungsplänen funktionieren diese nur mit grafischen Browsern. Wenn Sie versuchen, ein Dokument mit einem Abbildungsplan darin mit einem textbasierten Browser wie Lynx anzusehen, erhalten Sie noch nicht einmal einen Hinweis darauf, dass das Bild überhaupt existiert. (Es sei denn, dass das Bild ein `alt`-Attribut enthält.) Aber auch mit dem `alt`-Attribut werden Sie nicht in der Lage sein, die Abbildung zum Navigieren zu benutzen, wenn Sie keinen grafischen Browser haben.

Wenn Sie sich entschließen, eine Webseite mit einem Abbildungsplan darin zu erstellen, ist es doppelt wichtig, dass Sie auch ein Textgegenstück anbieten, damit die vielen Leser mit textbasierten Browsern Ihre Seite benutzen können. Durch die Verwendung von Abbildungsplänen können Sie Leser mit textbasierten Browsern sehr effizient von Ihren Seiten ausschließen; haben Sie ein Einsehen, und ermöglichen Sie denen, die nicht an hübschen Bildchen, sondern lediglich an Informationen interessiert sind, wenigstens ein paar Wege, um Ihre Inhalte anzusehen.

Erzeugen Sie serverseitige Abbildungspläne

Obwohl Server-Side-Imagemaps Ihre Nachteile haben, ist es trotzdem von Nutzen, wenn man weiß, wie man diese erstellt.

Abgesehen von den verschiedenen Nachteilen, die die Verwendung von serverseitigen Abbildungsplänen mit sich bringt, gibt es da noch einen, der meine Erklärungen weiter kompliziert: Fast jeder Web-Server benutzt eine unterschiedliche Methode, um die Abbildungspläne herzustellen. Die Methoden unterscheiden sich sogar zwischen den einzelnen Servern auf der gleichen Plattform. So haben zum Beispiel der W3C(CERN)-HTTPD-Server und der NCSA-HTTPD-Server nichtkompatible Methoden zur Verarbeitung von Abbildungsplänen. Alle Server verwenden jedoch die gleichen grundlegenden Zutaten für Abbildungspläne:

- ▶ Speziellen HTML-Code, der anzeigt, dass ein Bild ein Plan ist
- ▶ Eine Plandatei auf dem Server, die die Regionen auf dem Bild mit den verknüpften Webseiten korreliert
- ▶ Ein CGI-Script, das alle Komponenten des Abbildungsplans miteinander verbindet

Der folgende Abschnitt erklärt, wie man anklickbare Bilder im Allgemeinen erzeugt; seine Beispiele beziehen sich jedoch auf Server, die sich an die Methoden von NCSA HTTPD halten, wie NCSA selbst oder Apache. Wenn Sie mehr Informationen über Ihren Server benötigen, sehen Sie sich die Dokumentation an, die Sie mit Ihrem Server erhalten haben, oder bitten Sie Ihren Web-Administrator um Rat.

Ein Bild finden

Um einen Abbildungsplan zu erstellen, benötigen Sie eine Abbildung. Die Abbildung, die als Plan dienen soll, ist am brauchbarsten, wenn sie mehrere visuell abgegrenzte Bereiche hat, die einzeln ausgewählt werden können, zum Beispiel Bilder mit mehreren Symbol-Elementen oder Bilder, die leicht in Polygone (Vielecke) zerlegt werden können.

Fotografien ergeben schwierige Abbildungspläne, weil ihre einzelnen »Elemente« dazu neigen, ineinander überzugehen oder ungewöhnliche Formen haben. Die Abbildungen 9.2 und 9.3 zeigen Beispiele von guten und schlechten Bildern für Abbildungspläne.

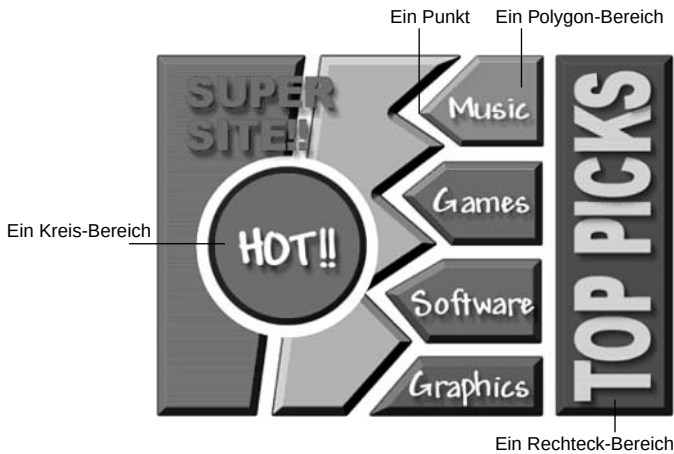


Abbildung 9.2:
Ein gutes Bild für
einen Abbil-
dungsplan



Abbildung 9.3:
Ein weniger gutes
Bild für einen Abbil-
dungsplan

Erzeugen einer Plandatei

Das Herzstück des serverseitigen Abbildungsplans ist eine Plandatei. Um eine Plandatei zu erzeugen, skizzieren Sie zuerst die Regionen in Ihrem Bild, die anklickbar sein sollen, bestimmen dann die Koordinaten dieser Regionen und entscheiden dann, mit welchen HTML-Seiten die kartographierten Regionen verknüpft werden sollen.



Das Format der Plandatei hängt von dem Abbildungsplan-Programm ab, das Sie auf Ihrem Server verwenden. In diesem Abschnitt werde ich über die Abbildungspläne des NCSA-HTTP-Servers und die Plandateien, die er standardmäßig verwendet, reden. Wenn Sie einen anderen Server verwenden, haben Sie vielleicht andere Abbildungsplan-Programme und können zwischen anderen Abbildungsplan-Formaten wählen. Wenn Sie vor diesem Problem stehen, fragen Sie Ihren Web-Administrator, oder lesen Sie sich die Server-Dokumentation aufmerksam durch

Sie können eine Plandatei entweder herstellen, indem Sie Regionen skizzieren und dann die Koordinaten von Hand notieren, oder indem Sie ein Programm anwenden, das Abbildungspläne erzeugt. Die letztere Methode ist einfacher, da das Programm die Plandatei automatisch generieren wird, indem es die Koordinaten der Regionen festhält, die Sie mit der Maus gezeichnet haben.

Des Weiteren können Ihnen die Programme Mapedit und MapThis für Windows und WebMap für den Macintosh dabei helfen, Plandateien im NCSA-Format zu erzeugen. Wenn Sie ein Unix-basiertes System benutzen, können Sie eine Version von Mapedit via FTP beziehen.

In Tabelle 9.1 finden Sie eine Liste aktueller Tools zur Erzeugung von Imagemaps.

Name	Plattform	URL
Imaptool	Linux/ X-Window	Http://www.sci.fi/~uucee/ ownprojects/
LivImage	Windows	Http://www.mediatec.com
Mapedit	Windows/ Unix	Http://www.boutell.com/mapedit/
Poor Person es Image Mapper	Basiert auf dem Web	http://zenith.berkeley.edu/~seidel/ClrHlpr/image- map.html

Wenn Sie Ihre Plandatei in einem anderem Format benötigen, können Sie immer eines der Hilfsprogramme einsetzen, um einen einfachen Plan herzustellen, und dann die Koordinaten, die Sie erhalten, in das Plandatei-Format, das Ihr Server benötigt, umwandeln. Wenn Sie Ihre Plandateien von Hand herstellen wollen, zeige ich Ihnen hier, wie das geht. Zuerst skizzieren Sie die Regionen, die Sie auf dem Bild aktivieren wollen (wie zum Beispiel in Abbildung 9.4).

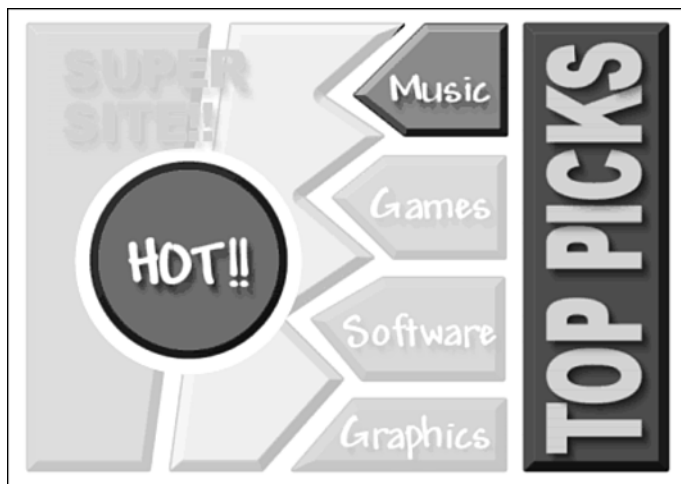


Abbildung 9.4:
Skizzieren der kartogra-
phierten Regionen

Sie müssen als nächstes die Endpunkte dieser Regionen in Koordinaten festlegen – dieser Vorgang ist fast mit dem zu vergleichen, den Sie beim Erstellen von Server-Side oder Client-Side-Imagemaps verwenden. Die meisten Bildbearbeitungsprogramme haben eine Option, die die Koordinaten der Mausposition anzeigen. Verwenden Sie dieses Feature, um sich die entsprechenden Koordinaten merken zu können (Alle Kartenprogramme, die vorhin erwähnt wurden, werden eine Kartendatei für Sie erstellen, aber jetzt sollten Sie die einzelnen Schritte lieber manuell ausführen, um die Vorgänge besser verstehen zu können).

Definieren eines Polygons

In Abbildung 9.5 sehen Sie die x,y-Koordinaten einer polygonförmigen Region. Diese Werte beschreiben die Position des Polygons bezogen auf die obere linke Ecke, die die Koordinaten (0,0) hat. Die erste Zahl im Koordinatenwert gibt den x-Wert an, und definiert die Pixelanzahl vom linken Rand des Bildes aus. Der zweite Wert ist der y-Wert und definiert die Pixelanzahl vom oberen Rand des Bildes.



Der Koordinatenursprung 0,0 befindet sich im linken oberen Eck des Bildes, und der positive y-Wert befindet sich von dort aus abwärts.

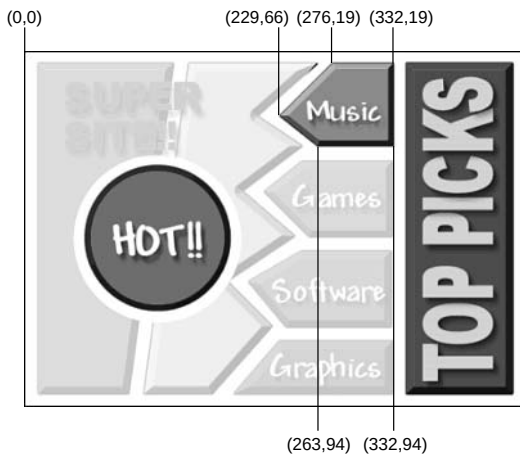


Abbildung 9.5:
Die Koordinaten für ein Polygon finden

Definieren eines Kreises

In Abbildung 9.6 sehen Sie, wie man die Koordinaten für einen Kreis festlegt. Hierzu legen Sie die Koordinaten des Kreismittelpunktes und den Radius in Pixeln fest. Der Mittelpunkt des Kreises ist über die x,y-Koordinaten von der linken oberen Ecke des Bildes aus gerechnet definiert.

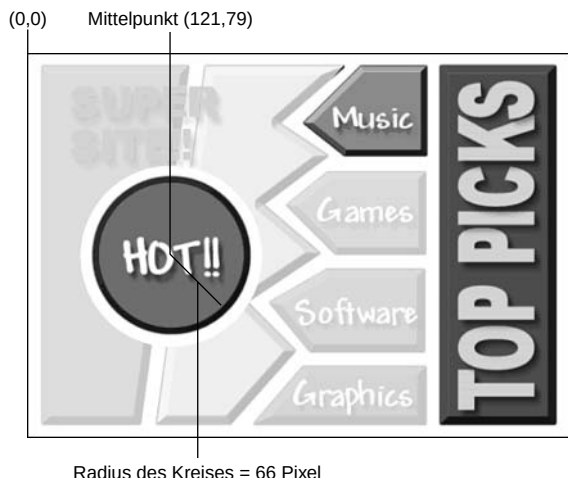


Abbildung 9.6:
Die Koordinaten für einen
Kreis finden

Definieren eines Rechtecks

In Abbildung 9.7 sehen Sie, wie Sie die Koordinaten für ein Rechteck definieren. Hierzu geben Sie die x,y-Koordinaten für die untere rechte und die obere linke Ecke des Rechtecks an.

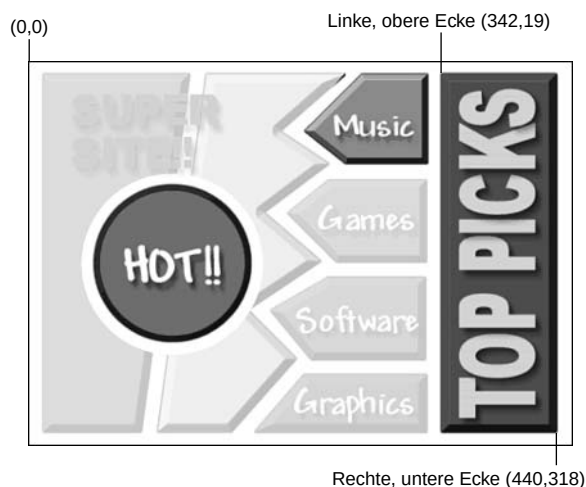


Abbildung 9.7:
Die Koordinaten für ein Recht-
eck finden

Definieren eines Punktes

Ältere Browser ermöglichten es Ihnen, innerhalb einer Imagemap Punkte zu definieren. Landed nun ein Mausklick nicht direkt in einer bestimmten Region, so wird der nächstgelegene Punkt aktiviert. Punkte sind in Photos und anderen Bildern, die keine isolierten Elemente aufweisen sehr hilfreich. Auf der anderen Seite kann man auch eine feinere Abstufung in Bezug auf die Anklickbarkeit erreichen, als nur »nicht innerhalb einer Region«. Punkte definieren Sie, indem Sie deren x,y-Koordinaten bezogen auf die linke, obere Ecke des Bildes, angeben.

Erstellen und Speichern Ihrer Karten-Datei

Nachdem Sie alle anklickbaren Regionen in Ihr Bild eingefügt haben, haben Sie schon fast die Hälfte geschafft. Als nächstes müssen Sie eine URL mit Links zu jeder Region oder zu jedem Punkt erstellen, der ausgewählt werden kann. Sie können viele Regionen haben, die auf dieselbe URL verweisen, aber jede Region darf nur einen Link besitzen.

Mit all Ihren definierten Regionen, Koordinaten und URLs können Sie jetzt eine Plandatei für Ihren Server schreiben. NCSA-HTTP-Plandateien sehen folgendermaßen aus:

```
default URL
circle URL x,y radius
rect URL x,y x,y
poly URL x1,y1 x2,y2 ... xN,yN
point URL x,y
```

Die Plandateien für Ihr spezielles Abbildungsplan-Programm für Ihren Server sehen vielleicht anders aus, haben aber die gleichen grundlegenden Komponenten. Ersetzen Sie die Werte für die Koordinaten durch die, die Sie im vorherigen Abschnitt notiert haben, an jeder der x- oder y-Positionen (oder x1, y1 usw.). *radius* (in der *circle*-Zeile) bezeichnet den Radius der Kreisregion.

Die Reihenfolge der Regionen in der Plandatei ist wichtig; je weiter oben eine Region in der Datei steht, desto höher ist ihre Priorität, wenn sie ausgewählt wird. Wird ein Teil einer Region angewählt, der in überlappenden Bereichen verschiedener Regionen liegt, dann wird die erste in der Plandatei aufgeführte Region aktiviert.

Schließlich kann auch die Plandatei für NCSA-Server eine *default*-Region enthalten, ohne Koordinaten-Punkt und lediglich mit einem URL. Diese Vorgabe wird verwendet, wenn eine Abbildung angewählt wird, die in keiner Region enthalten ist; dieser Eintrag bietet also eine Verbindung für all jene Teile der Abbildung, die in keiner besonderen Region enthalten sind. (Beachten Sie, dass Sie keine Punkte (*points*) einfügen sollten, wenn Sie bei NCSA HTTPD eine *default*-Region in Ihre Plandatei auf-

nehmen. Die Existenz von `point`-Elementen schließt einen `default`-Eintrag aus.) Hier ist ein Beispiel für die Verwendung von `default`.

Wenn Sie die Koordinaten aus Abbildungen 9.5 bis 9.7 in die NCSA HTTPd Karten-Datei einsetzen, könnten die Koordinaten und die URLs wie folgt aussehen:

```
poly http://www.foo.com/mysite/music.html 229,66 276,19, 332,19 333,94, 263,94
circle http://www.foo.com/mysite/music.html 121,79 66
rect http://www.foo.com/mysite/music.html 342,19 440,318
```

Die URLs, die Sie angeben, müssen entweder absolute URLs (beginnend mit `http` oder `ftp` oder einem anderen Protokoll) sein, oder es muss sich dabei um einen vollständigen Pfad-Namen zu den Dateien (auf Ihrem Server) handeln, mit denen Sie eine Verbindung herstellen wollen, das heißt alles, was Sie hinter den Host-Teil einer URL schreiben könnten. Sie können jedoch in Plandateien keine relativen Pfad-Namen verwenden. Hier ein Beispiel für eine NCSA-Plandatei:

```
circle /www/mapping.html 10,15 20
circle /www/mapping.html 346,23 59
poly /www/test/orange.html 192,3 192,170 115,217
rect /www/pencil.html 57,57 100,210
point /www/pencil.html 100,100
point /www/orange.html 200,200
```

Die Plandatei zu erzeugen ist der schwierigste Teil bei der Herstellung eines Abbildungsplans. Nachdem Sie eine Plandatei für Ihr Bild geschrieben haben, müssen Sie sowohl die Plandatei als auch das Abbildungsplan-Programm auf Ihrem Server installieren und dann alles mit Ihren HTML-Dateien verknüpfen, um den Abbildungsplan benutzen zu können.

Installieren der Plandatei und des Abbildungsplan-Programms

Sie müssen zusätzlich zu der Plan-Datei, die Sie gerade erstellt haben, ein Programm auf Ihrem Server installieren. Dieses Programm wird normalerweise in einem speziellen Verzeichnis namens `cgi-bin` gespeichert, das extra zur Verwaltung von Programmen und Scripts angelegt wurde. Bei den meisten Servern ist standardmäßig ein entsprechendes Programm eingerichtet. Wenn Sie nicht den eigenen Server verwenden, wird dieses Programm trotzdem für Sie verfügbar sein. Das Programm, nach dem Sie suchen sollten, wird oft `htimage` oder `imagemap` genannt.



Seien Sie vorsichtig, bei der Kombination NCSA-Server und dem Programm `imagemap`. Ältere Versionen des Programms waren umständlicher und brauchten eine Extra-Konfigurationsdatei; das Programm, das mit der 1.5-Version des Servers geliefert wird, funktioniert viel besser. Wenn Sie

die neueste Version des NCSA-Servers nicht zur Verfügung haben, können Sie die aktuelle Version des Programms unter <http://hoohoo.ncsa.uiuc.edu/docs/tutorials/imapmap.txt> erhalten.

Alles miteinander verknüpfen

Jetzt haben Sie also ein Bild, eine Plan-Datei, die die Koordinaten der anklickbaren Regionen und die Ziele der Links beschreibt, und ein serverseitiges Abbildungsplan-Programm. Das alles können Sie nun verbinden.

Integrieren Sie das Bild in Ihre Webseite. In dieser Seite verwenden Sie die `<a>` und `<img>`-Tags zusammen, um den Effekt eines anklickbaren Bildes zu kreieren. Im folgenden Beispiel wird das NCSA Imagemap-Programm verwendet.

```
<a href="/cgi-bin/imapmap/maps/myimage.map">  
</a>
```

Beachten Sie verschiedene Eigenheiten dieses Links. Erstens wird der Link zum Abbildungsplan-Script (imagemap) so angezeigt, wie Sie es erwarten würden. Diesem wird dann aber der Pfad zur Plandatei am Ende hinzugefügt. Der Pfad zur Plandatei sollte aus einem vollständigen Pfad-Namen vom Wurzelverzeichnis Ihres Webverzeichnisses bestehen (also alles nach dem Host-Namen in Ihrem URL), in diesem Fall `/maps/myimage.map`.

Der zweite Teil des HTML-Codes, der einen serverseitigen Abbildungsplan erstellt, ist das `ismap`-Attribut des `<img>`-Tags. Dies ist ein einfaches Attribut ohne Wert, das den Browser anweist, individuelle Mausclick-Koordinaten zum Abbildungsplan-Programm auf der Serverseite zur Verarbeitung zu schicken.

Und jetzt, nachdem sich alle drei Teile des serverseitigen Abbildungsplan-Programms an Ort und Stelle befinden (die Plandatei, das Abbildungsplan-Programm und der spezielle HTML-Code), sollte der Abbildungsplan funktionieren. Sie sollten jetzt dazu in der Lage sein, Ihre HTML-Datei in den Browser zu laden und den Abbildungsplan dazu zu verwenden, verschiedene Seiten auf Ihrem Server zu erreichen, indem Sie verschiedene Teile des Plans anklicken.



Wenn Sie den NCSA-HTTPD-Server benutzen und nicht die neueste Version von imagemap besitzen, werden Sie die Fehlermeldung »Cannot Open Configuration file« (Konfigurationsdatei kann nicht geöffnet werden) bekommen, wenn Sie versuchen, Teile Ihres Bildes auszuwählen. Wenn Sie diese Fehlermeldungen erhalten, setzen Sie sich mit Ihrem Web-Administrator in Verbindung

Übung 9.1: Eine anklickbare Jukebox

Abbildungspläne können eine ziemlich haarige Angelegenheit sein, besonders da Plandateien recht fehleranfällig sind, wenn Sie die einzelnen Regionen nicht deutlich skizziert und nicht alles am richtigen Ort installiert haben. In dieser Übung nehmen wir eine einfache Abbildung und erstellen eine Plandatei dafür, wobei wir das NCSA-Server-Plandatei-Format benutzen. Auf diese Weise werden Sie ein Gefühl dafür bekommen, wie Plandateien aussehen und wie sie erstellt werden.



Abbildung 9.8:
Das Jukebox-Bild

als Erstes lassen Sie uns die Regionen definieren, die in diesem Bild anklickbar sein sollen. Sicherlich haben Sie schon bemerkt, dass sich dort sechs rechteckige Buttons befinden, die jeweils die unterschiedlichen Musikrichtungen anzeigen. Im Weiteren sehen Sie in der Mitte des Bildes eine Art Haus (das perfekt für das Polygon-Tool ist) und einen Kreis, indem ein Fragezeichen steht. Abbildung 9.9 zeigt ein Beispiel dieser Art von Regionen, die bei diesem Bild Sinnvollerweise verwendet werden können.

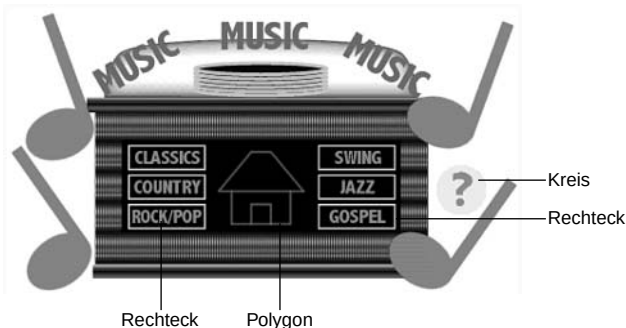


Abbildung 9.9:
Die Jukebox mit
definierter Region

Jetzt haben Sie eine Vorstellung von den Bereichen jeder Region. Lassen Sie uns nun die exakten Koordinaten der Ecken herausfinden. Um dies zu tun, können Sie entweder ein Planprogramm wie Mappedit oder WebMap verwenden (was ich sehr empfehle) oder es von Hand tun. Wenn Sie es tatsächlich von Hand versuchen, geben Ihnen die meisten Bildbearbeitungsprogramme eine Möglichkeit, die x- und y-Koordinaten des Bildes anzugeben, wenn Sie die Maus drüberziehen.

PhotoImpact 4.0 (siehe auch Kapitel 8) zeigt die Cursorposition im unteren linken Eck der Statusleiste an. PhotoImpact 4.0 besitzt einen Imagemap-Assistenten. Sie müssen nur die Selection Tools auswählen, um Rechtecke, Kreise oder Polygone in dem von Ihnen gewünschten Bereich zu erstellen. Wählen Sie dann Web, Image Map Tag aus, und Sie können die Koordinateninformation sehen, eine URL bestimmen und den Imagemaptyp auswählen, den Sie erstellen möchten. PhotoImpact zeigt den Code für NCSA und CERN-Server-Side-Imagemap und auch für clientseitige Imagemaps an. In Abbildung 9.10 sehen Sie ein Beispiel für den Imagemap-Assistenten in PhotoImpact.

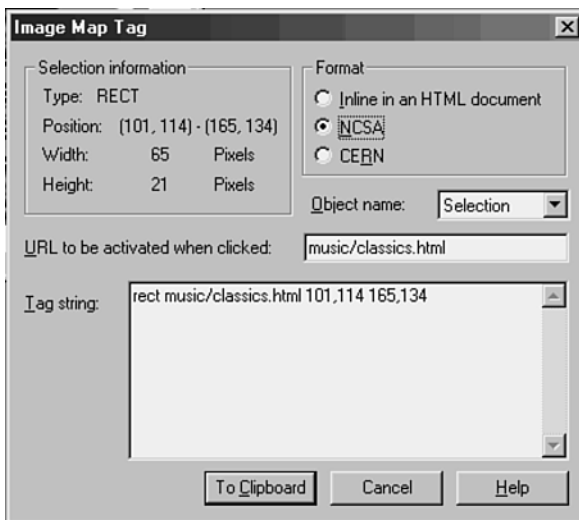


Abbildung 9.10:
Die Image Map Tag Dialogbox in
PhotoImpact 4.0



Sie besitzen kein Bildverarbeitungsprogramm? Dann kann ich Ihnen einen Trick verraten, den Sie mit dem Netscape-Browser anwenden können: Erzeugen Sie eine HTML-Datei mit einem Bild innerhalb eines Links, das auf eine nichtexistierende Datei verweist, und fügen Sie das `ISMAP`-Attribut innerhalb der `<img>`-Kennung ein. Sie brauchen keinen echten Link; nehmen Sie einfach irgendetwas. Der HTML-Code könnte ungefähr so aussehen:

```
<a href="nothing"></a>
```

Wenn Sie jetzt diese HTML-Datei in Ihren Browser laden, wird das Bild dargestellt, als sei es ein Abbildungsplan, und wenn Sie Ihre Maus drüberziehen, werden die x- und y-Koordinaten in der Statuszeile des Browsers dargestellt. Mit diesem Trick können Sie die Koordinaten für jeden Punkt auf dem Bild für die Plandatei herausfinden.

Neben den Regionen und der Liste mit Koordinaten brauchen Sie nun noch Dokumente, zu denen verzweigt werden kann, wenn der passende Bereich angewählt wird. Das kann jedes Dokument sein, es können jedoch auch Scripts sein: Alles, was Sie von einem Browser aus aufrufen können, können Sie nehmen. Für dieses Beispiel habe ich mehrere Dokumente erstellt und diese in das `music`-Verzeichnis auf meinem Web-Server gespeichert. Dies ist die Seite, die wir als Sprungziel benutzen, wenn die anklickbaren Bilder ausgewählt werden. In Abbildung 9.11 wird jede der anklickbaren Bereiche der Imagemap angezeigt. Tabelle 9.2 zeigt die Koordinaten eines jeden Bereichs, sowie die URL, zu der alle Bereiche verweisen, wenn sie angeklickt sind.



Abbildung 9.11:
Acht Hotspots, die in
Tabelle 9.2 definiert wer-
den

Number	Type	URL	Coordinates
1	rect	music/classics.html	101,113 165,134
2	rect	music/country.html	101,139 165,159
3	rect	music/rockpop.html	101,163 165,183
4	poly	music/home.html	175,152 203,118 220,118 247,152 237,153 237,181 186,181 186,153
5	rect	music/swing.html	259,113 323,134

Tabelle 9.1: Anklickbare Bereiche im Jukebox-Bild

Number	Type	URL	Coordinates
6	rect	music/jazz.html	259,139 323,159
7	rect	music/gospel.html	259,163 323,183
8	circle	music/help.html	379,152 21

Tabelle 9.1: Anklickbare Bereiche im Jukebox-Bild (Forts.)

Was nun noch übrigbleibt, ist der Eintrag in der Plandatei für den Klassik- Bereich, mit den Koordinaten und der Zieldatei, die aufgerufen wird, wenn der Bereich ausgewählt wird. Für eine NCSA-Plandatei sähe der Eintrag so aus:

```
rect /music/classics.html 101,113 165,134
```

Beachten Sie, dass die URLs in der Plandatei absolute Pfadnamen sein müssen und zwar sollen diese vom Web-Stammverzeichnis und nicht vom Stammverzeichnis des Betriebssystems aus definiert sein. Es dürfen keine relativen URLs ausgehend vom Verzeichnis, in dem sich die Plandatei befindet, sein. Abbildungspläne funktionieren anders. In diesem Fall befindet sich mein music-Verzeichnis direkt unterhalb des Web-Stammverzeichnisses und die `classics.html`-Datei innerhalb des Verzeichnisses. Die entsprechende URL in der Plandatei ist `/music/classics.html`.

Sie können nun identische Eingaben für andere Regionen in diesem Bild erstellen (Country, Rock/Pop, Home, Swing, Jazz, Gospel und Help). Vergessen Sie nicht eine Standard URL in der Plandatei festzulegen, um nicht gekennzeichnete Bereiche zu erstellen, die dennoch auf Mausclicks reagieren (in diesem Fall heißt die Datei `notaspot.html`). Die gesamte Plandatei sieht wie folgt aus:

```
default /music/notaspot.html
rect /music/classics.html 101,113 165,134
rect /music/country.html 101,139 165,159
rect /music/rockpop.html 101,163 165,183
poly /music/home.html 175,152 203,118 220,118 247,152 237,153 237,181 186,181
186,153
rect /music/swing.html 259,113 323,134
rect /music/jazz.html 259,139 323,159
rect /music/gospel.html 259,163 323,183
circle /music/help.html 379,152 21
```

Speichern Sie Ihre Plandatei in Ihrem Planverzeichnis auf dem Server (oder wo Sie sonst normalerweise Ihre Plandateien speichern). Verwenden Sie hierbei einen Dateinamen wie `jukebox.map`. Schließlich müssen Sie noch eine Webseite erstellen, die das Jukebox-Bild, das `ismap`-Attribut im `<img>`-Tag und den Link, der auf das Imagemap-Programm verweist, enthalten. Im folgenden Beispiel wird das Abbildungsplan-Programm auf meinem Server verwendet:

```
<a href="http://www.lne.com/cgi-bin/imapemap/maps/jukebox.map">
</a>
```

Geschafft! Wenn nun alles miteinander verbunden ist, sollte, wenn Sie das Bild anklicken, die Seite für den Teil des Bildes geladen werden.

Abbildung 9.12 zeigt die komplette Webseite. Beachten Sie, dass ich auch äquivalente Textlinks unterhalb der Abbildungspläne integriert habe. Diese werden in dem Fall hinzugefügt, wenn die Besucher Ihrer Seite reine Text-Browser verwenden oder die Bildanzeige ausgeschaltet haben. Der vollständige Code für die Webseite lautet wie folgt:



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://
www.w3.org/TR/xhtml1/DTD/transitional.dtd">
<html>
<head>
<title>The Really Cool Music Page</title>
</head>
<body bgcolor="#ffffff">
<div align="center">
<h1>The Really Cool Music Page</h1>
<p>Select the type of music you want to hear.<br />
You'll go to a list of songs that you can select from.</p>
<p><a href="http://www.lne.com/cgi-bin/imapemap/maps/jukebox.map">

</a>
</p>
<p><a href="code/music/home.html">Home</a> :
<a href="code/music/classics.html">Classics</a> :
<a href="code/music/country.html">Country</a> :
<a href="code/music/rockpop.html">Rock/Pop</a> :
<a href="code/music/swing.html">Swing</a> :
<a href="code/music/jazz.html">Jazz</a> :
<a href="code/music/gospel.html">Gospel</a> :
<a href="code/music/help.html">Help</a></p>
</div>
</body>
</html>
```




Abbildung 9.12:
Die fertig gestellte Jukebox-Webseite mit serverseitigem Abbildungsplan

Erstellen clientseitiger Abbildungspläne

Da Sie ja schon wissen, wie man Abbildungspläne auf die gute alte Art und Weise erstellt, sehen Sie sich doch einmal an, wie man sie heutzutage erstellt. Wie schon vorher erwähnt, bieten serverseitige Abbildungspläne einige Verbesserungsvorschläge bezüglich der serverseitigen Abbildungspläne an. Die wichtigsten Verbesserungen bestehen darin, dass Sie kein serverseitiges Abbildungsplan-Programm auf Ihrem Server installieren müssen. Neuere Web-Browser verarbeiten den Abbildungsplan lokal auf dem Anwender-Computer.

Wenn Sie einen clientseitigen Abbildungsplan erstellen, stimmen viele der Schritte, wenn es darum geht Koordinaten für eine jede Region auf der Karte zu finden, exakt mit denen für das Erstellen für serverseitige Abbildungspläne überein. Im Gegensatz zum serverseitigen Abbildungsplan, welcher eine separate Datei für das Speichern von Koordinaten und Referenzen für einen jeden Hyperlink verwendet, speichern clientseitige Abbildungspläne alle Koordinateninformationen als Teil eines HTML-Dokuments.

Das <map> und das <area>-Tag

Um einen clientseitigen Abbildungsplan innerhalb eines HTML-Dokument aufzunehmen, verwenden Sie das <map>-Tag, welches wie folgt aussieht:

```
<map name="mapname"> coordinates and links </map>
```

Der Wert, der dem name-Attribut zugewiesen wird, ist der Name der Plan-Definition. Dies ist der Name, der später dazu benutzt werden wird, das anklickbare Bild mit den korrespondierenden Koordinaten und Link-Referenzen zu verbinden. Wenn Sie dann mehrere Abbildungspläne auf derselben Seite haben, haben Sie auch mehrere <map>-Tags mit verschiedenen Namen.

Zwischen den <map> und </map>-Tags geben Sie die Koordinaten für jeden Bereich auf dem Abbildungsplan und die Zielgebiete dieser Regionen ein. Dabei verwenden Sie dieselben Werte und Links, die Sie im Abschnitt über serverseitige Abbildungspläne festgelegt haben. Diesmal werden die Koordinaten jedoch innerhalb eines anderen neuen Tag, nämlich des <area>-Tag, angegeben. Um beispielsweise den Polygon-Bereich von Übung 9.1 zu definieren, würden Sie Folgendes schreiben:

```
<area shape="poly" coords="175,152, 203,118, 220,118,
    247,152, 237,153, 237,181, 186,181, 186,153"
href="music/home.html">
```

Was für eine Form die Region hat, wird durch das shape-Attribut bestimmt, welches die Werte rect, poly und circle annehmen kann. Die Koordinaten für jede der Formen werden durch das coords-Attribut bestimmt. So sieht zum Beispiel das coords-Attribut für die poly-Form folgendermaßen aus, wobei jede x-, y-Kombination einen Punkt auf dem Polygon repräsentiert:

```
<area shape="poly" coords="x1,y1,x2,y2,x3,y3,...,xN,yN" href="URL">
```

Bei rect-Formen, ist x1,y1 die linke obere Ecke des Rechtecks, und x2, y2 die rechte untere:

```
<area shape="rect" coords="x1,y1,x2,y2" href="URL">
```

In einer circle-Form repräsentiert x, y den Kreismittelpunkt und radius gibt die Größe an:

```
<area shape="circle" coords="x,y,radius" href="URL">
```

Das letzte Attribut, das Sie bei jedem <area>-Tag definieren müssen, ist das href-Attribut. Sie können href jede beliebige URL zuweisen, die Sie normalerweise mit einem <a>-Link verbinden würden, einschließlich relativer Pfad-Namen. Außerdem können Sie href mit dem Wert »nohref« versehen, um Bereiche der Abbildung zu definieren, die keine Links zu einer neuen Seite beinhalten.



Wenn Sie clientseitige Abbildungspläne zusammen mit Frames verwenden, können Sie auch das `target`-Attribut innerhalb eines `<area>`-Tag einsetzen, um eine neue Seite in einem bestimmten Fenster zu öffnen, wie im folgenden Beispiel:

```
<area shape="rect" coords="x1,y1,x2,y2" href="URL" target= "window_
name">
```

Sie müssen noch ein zusätzliches Attribut in HTML 4.0 einfügen. In Kapitel 7 haben Sie bereits gesehen, wie man Bildern wechselnden Text zuweist. Das `alt`-Attribut zeigt eine kurze Beschreibung für eine anklickbare Region auf einem clientseitigen Abbildungsplan an, wenn Sie mit Ihrem Cursor darübergehen. Wenn Sie nun das Home-Polygon vom Jukebox-Bild verwenden, erscheint das `alt`-Attribut, wie es in der letzten Zeile des folgenden Beispiels erscheint:

```
<area shape="poly" coords="175,152, 203,118, 220,118,
247,152, 237,153, 237,181, 186,181, 186,153"
href="music/home.html"
alt="Home Page for Music Section">
```

Das usemap-Attribut

Nachdem Ihr clientseitiger Abbildungsplan durch das `<map>`-Tag definiert wurde, bleibt nur noch übrig, die Abbildung auf Ihrer Webseite einzusetzen. Dazu benötigen Sie eine spezielle Version des `<img>`-Tag, die ein Attribut namens `usemap` enthält (dies unterscheidet sich von `ismap` bei serverseitigen Abbildungsplänen). `usemap` sieht folgendermaßen aus, wobei `mapname` der Name eines Abbildungsplanes ist, der durch das `<map name="mapname">`-Tag definiert wird:

```
<map name="mapname"> tag:

```



Im Unterschied zu serverseitigen Abbildungsplänen brauchen Sie das `<img>`-Tag nicht in ein `<a>`-Tag zu stecken. Stattdessen gibt das `usemap`-Attribut dem Web-Browser an, dass die `<img>`-Kennung einen anklickbaren Abbildungsplan enthält.



Der Wert, den `usemap` annimmt, ist eine Standard-URL. Deswegen wird `mapname` mit einem Nummernzeichen (`#`) davor geschrieben. So wie bei Ankeren innerhalb von Webseiten gibt das Nummernzeichen dem Browser an, `mapname` auf der aktuellen Webseite zu suchen. Wenn Sie jedoch einen sehr komplexen Abbildungsplan haben, kann er in einer separaten HTML-Datei gespeichert und durch ein normales URL angegeben werden.

Übung 9.2: Noch ein Besuch bei der anklickbaren Jukebox

Um die Besprechung von Abbildungsplänen abzuschließen, lassen Sie uns überlegen, wie das Abbildungsplan-Beispiel aus der Übung 9.1 geschrieben werden müsste, wenn es sich um einen clientseitigen Abbildungsplan handelt. Da wir bereits im Besitz der Koordinaten und Zielorte sind, brauchen wir eigentlich nur noch die serverseitige Plandatei in clientseitiges HTML umzuwandeln.

Bei der Jukebox sieht das `<map>`-Tag und das damit verbundene `<area>`-Tag folgendermaßen aus:

```
<map name="jukebox">
<area shape="rect" coords="101,113, 165,134"
      href="/music/classics.html"
      alt="Classical Music and Composers">
</map>
```

Das `<img>`-Tag, das sich auf die Plankoordinaten bezieht, ist ebenfalls unterschiedlich. Es verwendet `usemap` anstelle von `ismap` und ist nicht von einem Link umgehen:

```

```

Hierfür müssen Sie ein spezielles Formular des `<img>`-Tag verwenden, das ein Attribut namens `usemap` integriert (für `ismap` von serverseitigen Abbildungsplänen). Schließlich packen wir den ganzen Code zusammen und testen ihn. Hier ist die Beispiel-HTML-Datei, die sowohl die `<map>`-Kennung als auch das Bild selbst enthält. Das Ergebnis wird in Abbildung 9.13 gezeigt.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/
TR/xhtml1/DTD/transitional.dtd">
<html>
<head>
<title>The Really Cool Music Page</title>
</head>
<body bgcolor="#ffffff">
<div align="center">
<h1>The Really Cool Music Page</h1>
<p>Select the type of music you want to hear.<br />
You'll go to a list of songs that you can select from.</p>
<p>

```

```
<map name="jukebox">
<area shape="rect" coords="101,113, 165,134"
href="/music/classics.html"
  alt="Classical music and Composers" />
  <area shape="rect" coords="101,139, 165,159"
href="/music/country.html"
alt="Country and Folk Music" />
  <area shape="rect" coords="101,163, 165,183"
href="/music/rockpop.html"
alt="Rock and Pop from 50's On" />
<area shape="poly" coords="175,152, 203,118, 220,118, 247,152,
237,153, 237,181, 186,181, 186,153"
  href="code/music/home.html"
alt="Home Page for Music Section" />
<area shape="rect" coords="259,113, 323,134"
  href="/music/swing.html"
  alt="Swing and Big Band Music" />
<area shape="rect" coords="259,139, 323,159"
  href="/music/jazz.html"
  alt="Jazz and Free Style" />
<area shape="rect" coords="259,163, 323,183"
  href="/music/gospel.html"
  alt="Gospel and Inspirational Music" />
<area shape="circle" coords="379,152, 21"
  href="/music/help.html"
  alt="Help" />
</map></p>
<p>
<a href="code/music/home.html">Home</a> :
<a href="code/music/classics.html">Classics</a> :
<a href="code/music/country.html">Country</a> :
<a href="code/music/rockpop.html">Rock/Pop</a> :
<a href="code/music/swing.html">Swing</a> :
<a href="code/music/jazz.html">Jazz</a> :
<a href="code/music/gospel.html">Gospel</a> :
<a href="code/music/help.html">Help</a>
</p>
</div>
</body>
</html>
```



Abbildung 9.13:
Die fertige Really
Cool Music Page
mit clientseiti-
gem Abbildungs-
plan

Erzeugen von Webseiten, die beide Typen von Abbildungsplänen unterstützen

Obwohl clientseitige Abbildungspläne in Ihren Webseiten schneller und leichter als serverseitige Abbildungspläne ausgeführt werden können, werden sie nicht von allen Browsern unterstützt. Es gibt vereinzelt noch ältere Browser, die clientseitige Abbildungspläne nicht verarbeiten können. Wenn Sie auf Nummer sicher gehen wollen, können Sie Abbildungspläne erstellen, die in diesen älteren Browsern genauso gut funktionieren wie in den neueren Browsern. Sie sollten deshalb ein serverseitiges Äquivalent erstellen und dann Ihre HTML-Dateien dahingehend modifizieren, dass sie beide Typen von Abbildungsplänen unterstützen. Auf diese Weise werden Ihre Seiten gleich gut mit beiden Versionen der Abbildungspläne funktionieren, aber gleichzeitig die Vorteile der clientseitigen Methode bei den Browsern ausnützen, die sie unterstützen.

Um einen Abbildungsplan zu erzeugen, der die clientseitige Unterstützung ausnutzt, sofern sie verfügbar ist, aber auf die serverseitige Unterstützung zurückgreift, wenn nötig, verwenden Sie die folgende standardmäßige Definition für das Really Cool Music Seitenbeispiel.

```
<a href="http://www.lne.com/cgi-bin/imapemap/maps/jukebox.map">

</a>
```

Danach fügen Sie die Einzelheiten des clientseitigen Abbildungsplans als Teil des

```
 wie folgt ein:
<a href="http://www.lne.com/cgi-bin/imapemap/maps/jukebox.map">

</a>
```

Natürlich müssen Sie, damit dies funktioniert, auch die `Jukebox.map`-Datei auf dem Server installiert haben und die Definition des »Jukebox« `<map>`-Tag irgendwo auf Ihrer HTML-Seite einbauen.

Zusammenfassung

In diesem Kapitel haben Sie gelernt, wie Sie in Ihre Seiten Abbildungspläne einbauen können. Sie sollten jetzt auch den Unterschied zwischen serverseitigen und clientseitigen Abbildungsplänen kennen und wissen, welche in welchen Browsern verfügbar sind. Sie haben auch gelernt, wie Sie die aktiven Regionen bestimmen und die Koordinaten dazu definieren, und wie Sie Plandateien für clientseitige Abbildungspläne erzeugen. Sie sollten jetzt auch wissen, wie Sie anklickbare Bilder, Plandateien und Abbildungsplan-Programme auf dem jeweiligen Server miteinander verbinden.

Da dies ein sehr umfangreiches Kapitel war, zeigt Tabelle 17.2 nochmals eine Zusammenfassung der Tags und Attribute, die Sie in diesem Kapitel kennen gelernt haben.

Kennung	Attribut	Anwendung
<map>		Definiert den Abbildungsplan, wenn es sich um einen clientseitigen Abbildungsplan handelt.
	name	Ein Attribut des <map>-Tags, das den Namen des Abbildungsplans bestimmt.
	usemap	Ein Attribut des -Tags, das benutzt wird, um eine Abbildung mit einem clientseitigen Abbildungsplan zu assoziieren, der durch <map name="plannamen"> bestimmt wird.

Tabelle 9.2: HTML-Kennungen aus diesem Kapitel

Kennung	Attribut	Anwendung
<area>		Die individuellen Regionen innerhalb eines <map>-Elements.
	shape	Gibt den Regionentyp an. Mögliche Werte sind rect, poly, circle:
	coords	Ein Attribut des <area>-Tag, das die Koordinaten der Eckpunkte der Region angibt.
	href	Ein Attribut des <area>-Tag, das den URL der Region angibt.
	nohref	Ein Attribut des <area>-Tag, das eine Region angibt, welche keine Reaktion zeigt, wenn sie angeklickt wird (oder die keine entsprechende URL hat)
	alt	Zeigt den Alternativtext für eine anklickbare Region an. (Jetzt erforderlich für HTML 4.0)

Tabelle 9.2: HTML-Kennungen aus diesem Kapitel

Workshop

Wie immer, dient der Workshop dazu, Ihr heute erlerntes Wissen zu festigen. Und heute haben wir viel gelernt! Einige Fragen und Antworten, die oft über Abbildungspläne (server- oder clientseitig) gestellt werden, gehören zu diesem Workshop.

Fragen und Antworten

F Brauche ich einen Server, um einen Abbildungsplan zu erzeugen? Ich möchte dies alles offline erzeugen und testen, genauso, wie ich das bei meinen regulären HTML-Dateien gemacht habe.

A *Wenn Sie clientseitige Abbildungspläne anwenden, können Sie sie auf Ihrem lokalen System erzeugen und testen (vorausgesetzt natürlich, dass die Zielorte Ihres Abbildungsplans ebenfalls Teil Ihrer lokalen Präsentation sind). Wenn Sie jedoch serverseitige Abbildungspläne verwenden, müssen Sie mit dem Server verbunden sein, wenn es funktionieren soll, da Sie das Abbildungsplan-Programm auf dem Server benötigen.*

F Meine serverseitigen Abbildungspläne funktionieren nicht? Wo liegt das Problem?

A *Es gibt da mehrere Ursachen und Lösungsmöglichkeiten:*

Stellen Sie sicher, dass die URLs in Ihren Plandateien absolute Pfad-Namen sind, ausgehend vom Web-Stammverzeichnis bis zu der Datei, mit der Sie eine Verbindung herstellen wollen. Sie können in Plandateien keine relativen

Pfad-Namen benutzen. Wenn absolute Pfad-Namen nicht funktionieren, versuchen Sie es mit vollständigen URLs (die mit *http* anfangen).

Überprüfen Sie, ob Sie auch da absolute Pfad-Namen (und zwar so, wie sie in der URL erscheinen) benutzt haben, wo Sie den Pfad zu der Plandatei hinter das Abbildungsplan-Programm geschrieben haben.

Wenn Sie den HTTPD von NCSA benutzen, überprüfen Sie, ob Sie die neueste Version von *imagemap* haben. Anfragen, die mit dem *imagemap*-Gateway-Programm verarbeitet werden, sollten nicht nach Konfigurations-Dateien suchen.

F Meine clientseitigen Abbildungspläne funktionieren nicht. Wo liegt das Problem?

A Hier sind ein paar Vorschläge:

Vergewissern Sie sich, dass die Pfad-Namen oder URLs innerhalb der *<area>*-Kennungen auf tatsächlich vorhandene Dateien zeigen.

Vergewissern Sie sich, dass der Plan-Name in der *<map>*-Datei und der Name des Plans im *usemap*-Attribut, das sich im **-Tag befindet, übereinstimmen. Nur der letztere sollte durch ein Nummernzeichen eingeleitet werden.

Quiz

1. Was ist ein Abbildungsplan?
2. Was sind die zwei verschiedenen Arten von Abbildungsplänen und was sind ihre Vor- und Nachteile?
3. Warum ist es eine gute Idee, Text-Links zusätzlich zu Abbildungsplänen anzubieten?
4. Richtig oder falsch? Sie können auch eine relative URL benutzen, wenn Sie eine Ziel-URL in einer Abbildungsplandatei verwenden.
5. Welche drei Dinge benötigen Sie, um einen serverseitigen Abbildungsplan zu erstellen?
6. Welche werden nicht für einen clientseitigen Abbildungsplan benutzt?

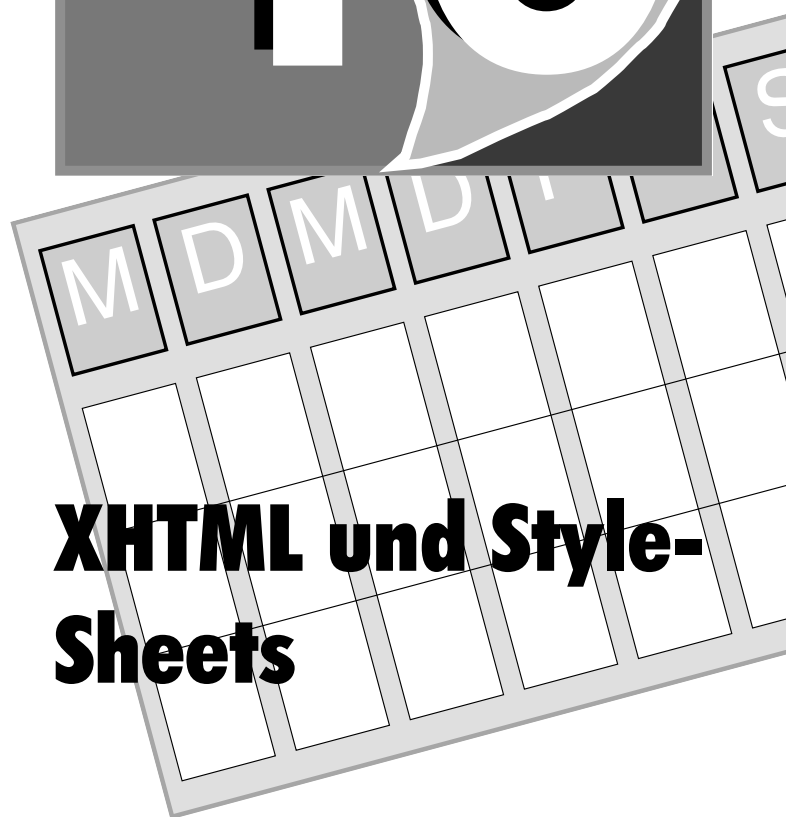
Antworten

1. Ein Abbildungsplan ist ein spezielles Bild, das verschiedene Regionen enthält, die auf bestimmte Locations verweisen.
2. Serverseitige Abbildungspläne werden von einer größeren Zahl von Browsern unterstützt. Aber da die Hotspots von einem Server verarbeitet werden, weiß der Browser nicht, wo jedes Hotspot hinweist. Sie können lokal nicht getestet werden. Sie reagieren auch langsamer auf Mausklicks. Clientseitige Abbildungspläne benötigen ein spezielles Programm auf dem Server und sind schneller, da die Verarbeitung in Web-Browsern erfolgt. Diese Methode wird von den meisten Web-Designern verwendet. Nicht alle Browser unterstützen clientseitige Abbildungspläne (obwohl dies nicht mehr so wichtig ist, da alte Browser nicht mehr so oft verwendet werden)
3. Es ist eine gute Idee, Textversionen von Abbildungsplan-Links zu integrieren, für den Fall, dass Anwender die einen reinen Text-Browser verwenden oder die Anzeige von Bildern abgeschaltet haben, Ihre Seite besuchen. So können diese Besucher die Links auf Ihrer Webseite anklicken und andere Regionen Ihrer Web-Site besuchen.
4. Falsch. URLs in Abbildungsplan-Dateien müssen absolute Pfad-Namen ausgehend vom Web-Stammverzeichnis sein. Die URL darf nicht relativ zum Verzeichnis der Abbildungsplan-Datei aufgebaut werden.
5. Um einen serverseitigen Abbildungsplan zu erstellen, benötigen Sie ein Bild, eine Abbildungsplan-Datei und ein Programm zur Verarbeitung des Abbildungsplans. Beides muss sich auf dem Server befinden, der als Ziel für den Abbildungsplan definiert ist.
6. Sie benötigen keine Abbildungsplan-Datei oder serverseitiges Abbildungsplan-Programm, wenn Sie einen clientseitigen Abbildungsplan erstellen. Stattdessen ist die Information, die den Abbildungsplan definiert, direkt in der Webseite integriert und wird von einem kompatiblen Web-Browser verarbeitet.

Übungen

Erstellen und testen Sie einen einfachen clientseitigen Abbildungsplan, der Links zu Seiten erstellt, die in Unterverzeichnissen einer Web-Site vorhanden sind, oder Links, die zu anderen Sites des WWW verweisen.

Erstellen und testen Sie einen clientseitigen Abbildungsplan für Ihre eigene Homepage oder für die Eingabeseite in eine der Hauptsektionen in Ihrer Web-Site. Denken Sie daran, Alternativen für die zu integrieren, die ältere oder textspezifische Browser verwenden. Wenn Sie sich gerne in Abenteuer stürzen, erstellen und testen Sie die serverseitige Alternative.



**Woche
2**

Während der letzten 3 Tage haben sie einiges über HTML-Tags lesen können, von denen ich häufig gesagt habe, dass sie zugunsten der Style-Sheets von HTML 4.0 verworfen werden. Nun ist es an der Zeit dieses Geheimnis zu lüften und Ihnen zu zeigen, wohin dies alles führen soll.

Heute werden Sie den aktuellen Ansatz des World Wide Web Konsortiums in Bezug auf die Formatierung und Gestaltung von Webseiten kennen lernen.

Wie Sie schon dem Kapitel 3 entnehmen konnten, ist HTML eine Auszeichnungssprache (Markup Language), und hat die Aufgabe, die Struktur einer Seite und nicht ihr Layout zu definieren.

HTML war nie dafür gedacht die Art und Weise, wie eine Seite auszusehen hat (besondere Schriftarten, Farben, Textabstände) zu definieren, sondern die Elemente einer Seite (Überschriften, Text, Grafik, usw...) festzulegen.

Die Erweiterungen der ursprünglichen HTML-Tags (`<font>`, `<color>`, `<margin>`, usw...) hat den Autoren von Webseiten weit mehr als ursprünglich geplant ermöglicht.

Das WWW Konsortium hat CSS und XHTML eingeführt, um auf der einen Seite den Strukturierungsgedanken von HTML wiederherzustellen und gleichzeitig den Webseiten-Autoren die gewünschte Kontrolle über das Design ihrer Seiten zu belassen bzw. diese zu erweitern.

In den nachstehenden Seiten werden folgende Themen behandelt:

- ▶ Unterschiede zwischen HTML und XHTML
- ▶ Das Konzept hinter Cascading Style Sheets
- ▶ Kurze Erläuterungen zu Style Sheets
- ▶ Wie erstellt und fügt man externe (external), eingebettete (embedded) und Inline-Styles (in der Datei definierte Style-Sheets) ein?
- ▶ Die meist benutzten Style-Sheet-Eigenschaften und -Werte
- ▶ Erstellen von Seiten-Layouts, Schriftarten und Farben mit CSS-Eigenschaften

Was ist XHTML und warum es verwenden?

Das World Wide Web Konsortium (W3C) definiert die XHTML wie folgt: »Eine Neuformulierung der HTML 4.0 Spezifikation als XML 1.0 Applikation«. Ich hingegen bezeichne sie als gute Neuigkeiten für Webseiten Autoren. Mit zunehmendem Wachstum des Internets, werden immer mehr Unternehmen mit neuen Internet Programmen auf den Markt kommen. Es ist höchstwahrscheinlich, dass in naher Zukunft etwas voll-

kommen anderes die heutigen Browser ersetzen wird und Sie möchten doch sicher gehen, dass man bis dahin Ihre Seiten lesen kann. Genau dies soll Ihnen XHTML ermöglichen. Durch die Einhaltung bestimmter Regeln zum Erstellen Ihres Dokuments soll dies erreicht werden.

Der <doctype>-Kennzeichner

Erinnern wir uns, dass alle HTML Seiten bestimmte Elemente, sowie abschließende Tags enthalten müssen.

```
<html>
<head>
<title>Basic HTML Pages</title></head>
<body>
The simplest HTML pages contain 3 tags.
</body>
</html>
```

XHTML hingegen, muss ein weiteres Element enthalten: den <doctype>-Kennzeichner.

Dieser Befehl identifiziert den Typ des HTML-Dokuments, den Sie erstellen wollen: Transitional-, Strict- oder Frameset-Dokument.

- Verwenden Sie die Transitional-Version des <doctype>-Tag, wenn Ihr Dokument ein Style Sheet zur Definition von Formateigenschaften einzelner HTML-Tags verwendet, welches zum größten Teil die Formatierung des Dokuments übernimmt, aber auch einige HTML-Formatierungs-Attribute enthält, die den älteren Browsern ermöglichen, das Dokument anzuzeigen.

```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>...</html>
```

- Verwenden sie die Strict-Version des <doctype>-Tag, wenn Ihr Dokument ein Style Sheet verwendet, das die gesamte Formatierung ihres Dokuments übernimmt. Nur CSS-fähige Browser, wie etwa Internet Explorer 4 und Netscape Navigator 4, sind in der Lage, diese Art Dokument anzuzeigen.

```
<!doctype html public "-//w3c//dtd xhtml 1.0 strict//en"
"http://www.w3.org/tr/xhtml1/dtd/strict.dtd">
<html>...</html>
```

- ▶ Verwenden Sie die Frameset-Version des <doctype>-Tag, wenn Ihr Dokument mehrere Frames (Anzeigesegmente) verwendet. Im Kapitel des 12. Tages werden Sie lernen, wie man Frameset-Dokumente erstellt.

```
<!doctype html public "-//w3c//dtd xhtml 1.0 frameset//en" "http://www.w3.org/
tr/xhtml1/dtd/frameset.dtd">
<html>...</html>
```



Die in diesem Buch aufgeführten Beispiele und Übungen folgen dem Transitional XHTML-Standard.

Die HTML-Syntax

HTML ist eine Auszeichnungssprache, die sehr fehlertolerant ist.

Sie kann zum Beispiel erkennen, dass Sie vergessen haben, ein HTML-Tag des Typs innerhalb einer nummerierten Liste () zu schließen. Sobald Sie aber das nächste -Tag eingeben, schließt HTML das vorher gegebene Tag automatisch ab.

Für HTML repräsentiert das folgende Beispiel (mit der korrekten Syntax)

```
<ol>
<li start="3">One ring-y, ding-y</li>
<li>Two ring-y, ding-ys</li>
</ol>
```

das gleiche wie

```
<ol>
<li start="3">One ring-y, ding-y
<li>Two ring-y, ding-ys</li>
</ol>
```

und

```
<ol>
<li start="3">One ring-y, ding-y</li>
<li>Two ring-y, ding-ys</li>
</ol>
```

Für XHTML hingegen sind all diese Beispiele unterschiedlich. Im nachstehenden Text werden Sie herausfinden, warum:

- ▶ **Tags und Attribute sollten Sie nur klein schreiben.** In XHTML sind , und unterschiedliche Tags, die unterschiedlich behandelt werden müssen. Um Probleme mit Anwendungsprogrammen, die XHTML-Dateien verarbeiten, zu vermeiden, sollten Sie deshalb alle HTML-Tags klein schreiben.

- **Setzen Sie Attributwerte in Anführungszeichen.** Achten Sie darauf, dass alle Ihre Attributwerte in Anführungszeichen gesetzt werden. Die Anführungszeichen helfen dem Browser zu erkennen, dass es sich um den Wert eines Attributes handelt und nicht um einen Befehl.
- **Schließen Sie sämtliche nicht leere Elemente ab.** Wie schon oben erwähnt, kann es vorkommen, dass Sie in der HTML vergessen ein Tag abzuschließen, aber der Browser trotzdem in der Lage ist, Ihr Dokument korrekt anzuzeigen. XHTML hingegen verlangt, dass sämtliche Tags abgeschlossen werden. Wie aber beenden Sie `<br />` (Zeilenumbruch) und `<hr />`-Tags (horizontale Linie), die keinen schließenden Steuerbefehl haben? Wie das folgende Beispiel aufzeigen wird, genügt es hier, einen Schrägstrich (/) in das Tag miteinzubeziehen: `<!DOCTYPE html PUBLIC...`

```
<!doctype html public "-//w3c//dtd xhtml 1.0 strict//en" "http://www.w3.org/
tr/xhtml1/dtd/strict.dtd">
<html>
<head>
<title>Line Breaks and Horizontal Rules</title></head>
<body>
<hr />      <!-- öffnet und schließt das hr-Tag -->
The first line<br /> <!-- öffnet und schließt das br-Tag -->
The second line
<hr />      <!-- öffnet und schließt das hr-Tag -->
</body>
</html>
```

- **Verwenden Sie verschachtelte Tags und überlappen Sie diese nicht.** In den folgenden zwei Beispielen werden Sie erkennen, dass für HTML kein Unterschied besteht, für XHTML aber sehr wohl. XHTML verlangt das Schließen von verschachtelten Tags und zwar in der Reihenfolge, in der sie geöffnet wurden. Das erste Beispiel entspricht der richtig verwendeten Syntax, das zweite Beispiel zeigt die falsche Syntax.

```
<b>This text is bold.<i>This is bold and italicized.</i></b><i>This is just
italicized.</i>
<b>This text is bold.<i>This is bold and italicized.</b>This is just
italicized.</i>
```

Entwurf von Style-Sheets

Style-Sheets zu erstellen, sollte für Sie keine Schwierigkeit mehr sein, da Sie nun ja die Regeln von XHTML kennen, um HTML Dokumente zu schreiben. Außerdem wissen Sie wie Sie den `<doctype>` Kennzeichner verwenden, um die Art Ihres HTML-Doku-

ments zu beschreiben. Zuerst muss eine ganz normale Webseite mit ganz gewöhnlichen HTML-Tags erstellt werden. Das Design dieser Standard-Webseite ist so gestaltet, das es in Browsern, die Style Sheets nicht unterstützen, angezeigt werden kann. Hierzu ein einfaches Beispiel:

```
<!doctype html public "-//w3c//dtd xhtml 1.0 strict//en"
"http://www.w3.org/tr/xhtml1/dtd/strict.dtd">
<html>
<head>

<title>Using Style Sheets</title></head>
<body>

<h1> Using Style Sheets </h1>

<p> In this simple example, the heading will be blue, and the
    paragraph will be rendered in a different font. </p>
</body>
</html>
```

Die Webseite im oben aufgeführten Beispiel enthält keine Attribute, die dessen Gestaltung definieren. Laut Text, möchte der Verfasser eine blaue Überschrift und verschiedene Schriftarten für die Absätze haben. Hierfür müssen Style-Regeln kreiert werden, die den Inhalt der Webseite in der gewünschten Art und Weise formatieren.

Style-Regeln verbinden HTML-Befehle (wie zum Beispiel `<h1>` oder `<p>`) mit Eigenschaften (wie etwa `color:blue`) zur Formatierung jedes einzelnen HTML-Tags. Bei Style Sheets, wird ein HTML-Tag als Selektor verwendet. Eigenschaft und Wert dieses Selectors werden in einer so genannten Deklaration vereinigt. Style-Regeln können das Layout eines Tags, sowie auch andere typografische und gestalterische Eigenschaften definieren. Hierzu einige Beispiele:

```
h1 { color: blue }
p { font-family: Arial, Helvetica, sans-serif; color: black }
```



Eine CSS-Style-Regel besteht aus zwei Teilen: dem Selektor und der Deklaration. Ersterer kann zum Beispiel ein HTML-Tag (`<h1>` oder `<p>`) sein. Die Deklaration definiert die Eigenschaft und den Wert des Selektors. Beispiel: `color: magenta;` `color` ist hierbei die Eigenschaft und `magenta` der Wert.

In der ersten Zeile des oben genannten Beispiels wird infolge der Style-Regel die Überschrift erster Ordnung des Textes in blauer Schriftfarbe angezeigt. In der zweiten Zeile, wird der gesamte Inhalt des Textabschnittes (`<p>`-Tag) in Arial, Helvetica oder einer anderen Sans-Serif Schriftart in schwarzer Farbe aufgezeigt.

Da nun der Seiteninhalt (Standard-Webseite) gestaltet und Style-Regeln definiert wurden, die die Gestaltung der Seite beschreiben sollen, müssen jetzt die Style-Regeln in

das Standard-HTML-Dokument eingebunden werden. Dabei kann man auf dreierlei Arten vorgehen: externe-, eingebettete- und Inline-Style-Sheets. Im Unterkapitel »Welches Style-Sheet verwenden?«, werden Sie etwas mehr darüber erfahren. Folgendes Beispiel zeigt Ihnen wie unser einfaches HTML-Beispiel mit einem eingebetteten Style Sheet formatiert wird:

```
<!doctype html public "-//w3c//dtd xhtml 1.0 strict//en" "http://www.w3.org/tr/
xhtml1/dtd/strict.dtd">
<html>
<head>
<title>Using Style Sheets</title>
<style type="text/css">
<!--
h1 { color: blue }
p { font-family: arial, helvetica, sans-serif; color: black }
-->
</style>
</head>
<body>
<h1>Using Style Sheets</h1>
<p> In this simple example, the heading will be blue, and the
    paragraph will be rendered in a different font. </p>
</body>
</html>
```

Im eben aufgeführten Beispiel eines in einem HTML-Dokument eingebetteten Style Sheet wurde das Style Sheet von dem Standard-HTML-Dokument getrennt. Der den Aufbau der Webseite definierende Code erscheint innerhalb des einleitenden und des abschließenden `<style>`-Tag (style = Stil, Format). Browser, die Style Sheets nicht unterstützen, können die Datei noch als Standard HTML-Dokument wiedergeben, während Style-Sheet-fähige Browser den Inhalt der Webseite so wiedergeben, wie dieser mittels Style-Regeln definiert wurde.

Wissenswertes über Style-Sheets

Die erste Spezifikation der Cascading Style Sheets, als CSS1 bekannt, ermöglicht es Ihnen, bereits vieles, das Erscheinungsbild einer Seite betreffend, festzulegen, seien es nun die Schriftart und -farben verschiedener HTML-Elemente, oder Hintergrundfarben, Grafiken, Absätze, Abstände, und noch vieles mehr... Browser, die diese Art Style Sheet unterstützen (wie etwa Netscape Navigator 4, später auch Internet Explorer 4), verwenden die Style-Sheet-Definitionen zur endgültigen Gestaltung ihres Dokuments.

Die nächste Generation der Cascading Style Sheets, CSS2, wurde dann im Mai 1998 offiziell vom W3C empfohlen. Viele CSS2-Befehle werden von den neuesten Versionen des Internet Explorers und auch des Netscape Navigators unterstützt. Schon in naher Zukunft werden Style Sheets Ihnen noch interessantere Möglichkeiten bezüglich der Gestaltung Ihrer Webseiten bieten können. Mit CSS2, XHTML und Style-Sheet-fähigen Browsern werden Sie Webseiten für unterschiedliche Ausgabemedien (Drucker, Bildschirm,...) erstellen können, die jeweils ihren eigenen »Gesetzen« folgen. So werden Sie zum Beispiel in der Lage sein, Style Sheets für Geräte zu erstellen, die über ein Sprachausgabesystem verfügen, und es dem Anwender mit einer Sehstörung ermöglichen, Teile der Webseite zu hören. Außerdem wird es Ihnen möglich sein, eine Webseite in eine Vielzahl von Seiten zu splitten, etwa so wie es in einem Textverarbeitungsprogramm oder in einem Desktop Publishing-Programm der Fall ist. Sie werden die Kontrolle über den Seitenumbruch, Hurenkinder und Schusterjungen und noch vieles mehr haben ... hört sich alles ganz schön aufregend an, was?

Hier kommen wir gleich zu einem weiteren Vorteil von Style-Sheet-Technologien. Mit Style Sheets können Sie Dateien auch auf unkonventionelle Art und Weise anzeigen, wie etwa mit Audioplayer (für Sehbehinderte) oder aber auf eine Weise, die für den derzeitigen Stand der Browser-Technik noch als undurchführbar scheint. Das Standard-HTML-Dokument wird einwandfrei angezeigt, und dies ohne die überflüssigen und oft auch verwirrenden, zusätzlichen Tags, die für die nicht üblichen Layouts in Browsern vorgesehen sind.

Wie HTML auch, werden CSS von den Browsern unterschiedlich unterstützt. Netscape hat eine eigene alternative Style-Sheet-Version kreiert: Netscape verfolgt einen alternativen Ansatz: Layers-Ebenen. Hier werden die Objekte auf den Seiten positioniert. Gleichzeitig wird der kommende Standard unterstützt. Netscape verfügt auch über eine eigene Style-Sheet-Technologie, JavaScript-Style-Sheets und unterstützt gleichzeitig auch Cascading-Style-Sheets. Es handelt sich hierbei um eine neue Technik und gerade deswegen hört sich nicht nur alles sehr verwirrend an, sondern ist es auch.

Doch wie mit allen neu entwickelten Technologien, wird auch diese mit der Zeit verständlicher werden; VHS-Style-Sheets werden immer mehr verwendet werden, während Betamax in Vergessenheit geraten wird. Bis es aber soweit ist, müssen Sie sich darüber im Klaren sein, dass diese Browser CSS-Technologien auf verschiedene Art und Weise handhaben.

Doch zunächst einmal werden wir Ihnen mit dem restlichen Teil des heutigen Kapitels einen Einblick in den Aufgabenbereich der Style Sheets verschaffen. Jedoch muss hinzugefügt werden, dass alles, was Sie heute darüber lernen, nur eine kleine »Kostprobe« von dem ist, was Style Sheets, die in CSS1 und CSS2 zur Verfügung stehen, alles können. Diese Thematik angemessen zu erläutern, würde weit über den Umfang dieses Buches gehen und zweifelsohne dessen Volumen verdoppeln. Dennoch werden

die heute erlernten Konzepte Ihnen gewisse Grundkenntnisse vermitteln und Ihnen den großen Nutzen von Style-Sheets verdeutlichen. Angaben zu zusätzlichen Online-Quellen sind am Ende dieses Kapitels aufgelistet.

Besuchen wir wieder den »Bookworm Bookshop«

Anstatt Sie mit einem Haufen Theorie zu bombardieren, werde ich Ihnen lieber einige CSS-Eigenschaften zeigen, die, wenn auf einer HTML-Webseite angewandt, miteinander in Verbindung stehen. In Kapitel 6 haben Sie einiges über Text- und Schriftformatierungs-Tags einer Seite erfahren, die Ihnen Informationen über »The Bookworm Bookshop« aufzeigten. Jetzt aber werden Sie erfahren, wie man Cascading Style Sheets anwendet, und somit der Webseite ein vollkommen anderes Design geben kann.

Doch zuerst einmal werde ich Ihnen Ihre Erinnerung etwas auffrischen und Ihnen die »The Bookworm Bookshop« Webseite in der bisherigen Version zeigen, bevor Sie all die phantasievollen »Spielereien« am Ende des 6. Tages hinzugefügt haben. Konvertieren Sie diese Seite so, dass sie ein Style Sheet verwendet (wird in den heutigen Übungen aufgezeigt). Speichern Sie die Datei dann als `bookworm.html` ab.

Bild 10.1 zeigt den oberen Teil der Seite. Dieser repräsentiert sämtliche Elementarten, die auf der Seite vorkommen können.



```
<!doctype html public "-//w3c//dtd xhtml 1.0 strict//en"
"http://www.w3.org/tr/xhtml1/dtd/strict.dtd">
<html>
<head>
<title>The Bookworm Bookshop</title>
</head>
<body>
<a name="top"><h1>The Bookworm: A Better Book Store</h1></a>
<blockquote>
"Old books are best--how tale and rhyme<br />
Float with us down the stream of time!"<br />
- Clarence Umy, <cite>Old Songs are Best</cite>
</blockquote>
<p>The Bookworm Bookshop<br />
1345 Applewood Dr<br />
Springfield, CA 94325<br />
(415) 555-0034
```

```

</p>
<a name="contents"><h2>Contents</h2></a>
<ul>
  <li><a href="#about">About the Bookworm Bookshop</a></li>
  <li><a href="#recent">Recent Titles</a></li>
  <li><a href="#upcoming">Upcoming Events</a></li>
</ul>
<hr />
<a name="about"><h2>About the Bookworm Bookshop</h2></a>
<p>Since 1933, The Bookworm Bookshop has offered
rare and hard-to-find titles for the discerning reader.
The Bookworm offers:</p>
<ul>
<li>Friendly, knowledgeable, and courteous help
<li>Free coffee and juice for our customers
<li>A well-lit reading room so you can "try before you buy"
<li>Four friendly cats: Esmerelda, Catherine, Dulcinea and Beatrice
</ul>
<p>Our hours are <strong>10am to 9pm</strong> weekdays,
<strong>noon to 7</strong> on weekends.</p>
<p><a href="#contents">Back to Contents</a> | <a href="#top">Back to Top</a></p>
<hr />
<a name="recent"><h2>Recent Titles (as of 11-Nov-99)</h2></a>
<ul>
<li>Sandra Bellweather, <a href="belladonna.html">
<cite>Belladonna</cite></a>
<li>Johnathan Tin, <a href="20minmeals.html">
<cite>20-Minute Meals for One</cite></a>
<li>Maxwell Burgess, <a href="legion.html">
<cite>Legion of Thunder</cite></a>
<li>Alison Caine, <a href="banquo.html">
<cite>Banquo es Ghost</cite></a>
</ul>
<p><a href="#contents">Back to Contents</a> | <a href="#top">Back to Top</a></p>
<hr />
<a name="upcoming"><h2>Upcoming Events</h2></a>
<ul>
<li><b>The Wednesday Evening Book Review</b> meets, appropriately, on Wednesday
evenings at
7:00 pm for coffee and a round-table discussion. Call the Bookworm for information
on joining
the group and this week es reading assignment.
<li><b>The Children es Hour</b> happens every Saturday at 1pm and includes
reading,
games, and other activities. Cookies and milk are served.
<li><b>Carole Fenney</b> will be at the Bookworm on Friday, September 17, to read
from her book of poems <cite>Spiders in the Web.</cite>

```

```
<li><b>The Bookworm will be closed</b> October 1 to remove a family
of bats that has nested in the tower. We like the company, but not
the mess they leave behind!
```

```
</ul>
```

```
<p><a href="#/contents">Back to Contents</a> | <a href="#/top">Back to Top</a></p>
<hr />
```

```
<address>
```

```
Last Updated: 11-Nov-99<br />
```

```
WebMaster: Laura Lemay lemay@bookworm.com<br />
```

```
&#169; copyright 1999 the Bookworm<br />
```

```
</address>
```

```
</body>
```

```
</html>
```

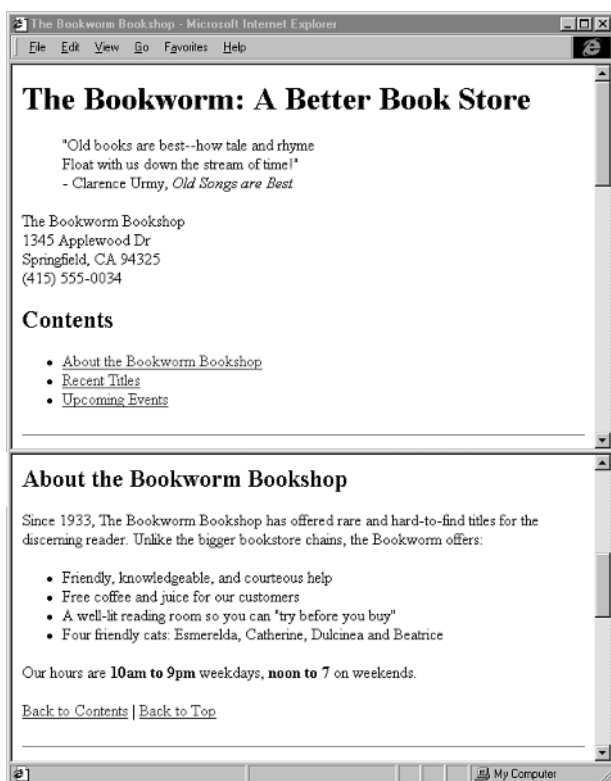


Abbildung 10.1:
»The Bookworm Bookshop« als
eigenständige HTML-Datei

Bild 10.1 zeigt eine Webseite, die auch in einem Browser, der Style-Sheet nicht unterstützt, angezeigt werden kann. Zugegeben, sie sieht nicht gerade außergewöhnlich aus. Vielleicht können wir sie ja mit Farbe, verschiedenen Schriftarten, Textabsätzen usw. etwas lebendiger gestalten. Und genau diese Dinge machen Style Sheets.

Style-Sheet-Ansätze

Wie wendet man CSS-Techniken auf Webseiten an? Im Grunde genommen gibt es da drei Methoden. Das Wunderbare am gesamten Style-Sheet-Konzept ist die gewonnene Flexibilität. Die HTML-Tags und -Attribute, mit deren Hilfe die Regeln auf die HTML-Dokumente angewendet werden, legen weder die Autoren von Webseiten noch die Hersteller von Browsern auf eine einzelne Style-Sheet-Art fest.

Das W3C hat stattdessen eine Reihe von Tags und Attributen definiert, die dazu dienen, Style-Definitionen auf jedes HTML-Tag und in jedem Dokument anwenden zu können. Diese Tags ermöglichen es Ihnen, auf drei verschiedene Arten mit Style-Regeln zu arbeiten: Externe Style Sheets, eingebettete Style Sheets und Inline-Styles. Zunächst einmal werden wir die ersten beiden Methoden erläutern, später dann werden wir Sie mit den in der Datei definierten Style Sheets vertraut machen.

Externe Style-Sheets

Style-Regeln eines externen Style-Sheet sind vom HTML-Web-Dokument getrennt und befinden sich in einer eigenen Datei. Der Vorteil dieser Style Sheets ist, dass Sie die gleichen Style-Regeln in allen Webseiten anwenden können. Auf diesem Weg können Sie Seiten mit einheitlicher Formatierung erstellen. Hinzu kommt, dass Sie den Aufbau Ihrer Web-Site schnell ändern können. Da sämtliche Style-Sheet-Angaben in einem einzigen, separaten Dokument gespeichert sind, können Sie durch ein HTML-Tag des Typs `<link>` alle Styles auf die jeweiligen Seiten verweisen.

So wie HTML-Dokumente auch, ist ein Style-Sheet-Dokument nichts anderes als eine ASCII-Textdatei, allerdings hat es eine spezielle Dateierweiterung. Eine Webseite wird mit der Dateierweiterung `.htm` oder `.html` abgespeichert, ein externes Style-Sheet aber mit der Dateierweiterung `.css`. Die üblichen Style-Regeln Ihrer Web-Dateien werden durch externe Style Sheets definiert. Mit dem HTML-Tag `<link>` können Sie alle Style-Sheets in einer separaten Datei auf jede HTML-Web-Seite verweisen.

Einzelheiten über das Erstellen von `.css`-Dateien erfahren Sie in der nächsten Übung. Bis dahin stellen Sie sich einfach vor, dass Sie ein Style-Sheet erstellt haben und es im gleichen Verzeichnis wie Ihre »Bookworm Bookshop« Webseite abgespeichert haben.

Nennen Sie ihre Datei `mystyle.css`. Der unten aufgeführte Code zeigt Ihnen, wie Sie ein separates Style Sheet in die Kopfzeile Ihrer »Bookworm Bookshop« Webseite einbinden.

```
<!doctype html public "-//w3c//dtd xhtml 1.0 strict//en"
"http://www.w3.org/tr/xhtml1/dtd/strict.dtd">
<html>
<head>
<title>The Bookworm Bookshop</title>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>
```

Das `<link>`-Tag verknüpft die externe Style-Sheet-Datei (`mystyle.css`) mit dem aktuellen HTML-Dokument. Da jede Site den gleichen Befehl erhalten hat, kann ein einheitlicher Stil vom Site-Manager bestimmt werden und auf Dateien angewandt werden, die von einzelnen Autoren einer Organisation erstellt wurden.

Das `rel`-Attribut des `<link>`-Tag (siehe Kapitel 5) hat eine wichtige Funktion. Um das `<link>`-Tag wirksam anzuwenden, bedarf es des Verständnisses von persistenten Styles, Standard-Styles und alternativen Styles.

- ▶ Persistente Styles werden immer angewandt, egal welchen Stil der Anwender wählt.
- ▶ Standard-Styles werden angewandt, wenn eine Seite geladen wird, können aber vom Anwender durch einen anderen Style ersetzt werden.
- ▶ Alternative Styles werden als Optionen gestellt und können vom Autor gewählt werden (im Gegensatz zu Standard-Styles).

Das `rel`-Attribut kontrolliert Teile dieses Prozesses. Falls Sie nun ein `<link>`-Tag eingeben (siehe vorhergehendes Beispiel), müssen Sie den PERSISTENT-Style im gesamten Style Sheet anwenden.

Wird ein `title`-Attribut hinzugefügt, ändert sich der Style automatisch in einen DEFAULT-Style. Hierzu folgendes Beispiel:

```
<link rel="stylesheet" title="mainstyle" href="mystyle.css">
```

Wenn Sie nun das `rel="STYLESHEET"`-Attribut mit dem `rel="ALTERNATE STYLESHEET"` austauschen, wird ein alternatives Style Sheet mit einem anderen Titel erstellt.

Sie können also einen PERSISTENT-Style mit imperativ anzuwendenden Definitionen, egal welche Wahl der Anwender getroffen hat, bestimmen, oder Sie können einen DEFAULT-Style und einen oder mehrere ALTERNATE-Styles bestimmen, die den PERSISTENT-Style ergänzen.

Übung 10.1: Erstellen und Einbinden eines externen Style Sheet

Falls Sie es nicht schon getan haben, öffnen oder erstellen Sie die »Bookworm Bookshop« Webseite (siehe Kapitel 6) und speichern Sie sie als `bookworm.html`. Der Originalcode wurde schon im Unterkapitel »Besuchen wir wieder den Bookworm Bookshop« gezeigt. Um ein externes Style Sheet, das Sie schon bald selbst erstellen werden, einzubinden, befolgen Sie die folgenden Codes im Dokumentkopf:

```
<!doctype html public "-//w3c//dtd xhtml 1.0 strict//en"
"http://www.w3.org/tr/xhtml1/dtd/strict.dtd">
<html>
<head>
<title>The Bookworm Bookshop</title>
<link rel="stylesheet" href="mystyle.css">
</head>
```

Speichern Sie die Seite anschließend mit den neuen, sich im Dokumentkopf befindenden Informationen wieder ab. Sie haben zwar ein Style-Sheet in die Seite eingebunden, aber Sie müssen ein Style-Sheet auch noch erstellen können. Solange sie nicht mehrere Styles in einem Style-Sheet definiert haben, sind Sie auch nicht in der Lage zu beurteilen, ob Ihr Style-Sheet, nachdem es eingebunden wurde, auch funktioniert.

In der nächsten Übung erstellen Sie also eine zweite Datei, die einige Basic-Styles enthalten wird. Wahrscheinlich ergibt das kommende Beispiel für Sie jetzt noch keinen Sinn, doch er wird sich Ihnen schon bald erschließen. Deshalb werde ich Ihnen jetzt nur kurz erklären, was der folgende Code bewirkt.

- ▶ Ein Style Sheet wird erstellt, um die Hintergrundfarbe der Webseite in hellblau zu ändern.
- ▶ Der Textkörper erscheint in Arial, Helvetica oder einer anderen Sans-Serif Schriftart, abhängig davon, welche Schriftart der Anwender in seinem System zur Verfügung hat.
- ▶ Die Schriftfarbe des Textes wird dunkelblau sein.

Die Link-Farben werden ebenfalls wie folgt geändert:

- ▶ Links von noch nicht besuchten Seiten sollen tiefrot sein.
- ▶ Links von bereits besuchten Seiten sollen dunkelbraun sein.
- ▶ Links, die gerade angeklickt werden, sollen rot sein.
- ▶ Wenn mit der Maus ein Link angeklickt wird, soll der Link-Text in goldener Farbe erscheinen.

Erstellen Sie eine neue Text-Datei und geben Sie den im folgenden Beispiel aufgeführten Code ein. Achten Sie darauf, dass Sie die Style-Regeln in geschweiften Klammern ({}) setzen. Außerdem sollten Sie Leerstellen zwischen den Style-Regeln und der öffnenden und schließenden Klammer lassen – zum Beispiel: `h1{color:blue}`. Nachdem Sie den folgenden Text eingegeben haben, speichern Sie die Seite im gleichen Verzeichnis ab wie Ihre »Bookworm Bookshop« -Seite. Benutzen Sie den gleichen Dateinamen wie den, der in der Kopfzeile Ihrer »Bookworm Bookshop« Webseite verwendet wurde (`mystyle.css`). Geben Sie nun folgenden Code ein:

```
body { background-color: #ccffff; font-family: Arial, Helvetica, sans-serif;
color: #330066 }
a:link { color: #ff00ff }
a:visited { color: #660000 }
a:hover { color: #ffcc00 }
a:active { color: #ff0000 }
```

Die erste Style-Regel im vorhergehenden Beispiel definiert die Hintergrundfarbe der Webseite (hellblau oder in diesem Fall `#ccffff`). Der Text dieser Seite wird in Arial, Helvetica oder einer anderen Sans-Serif Schriftart, die sich auf der Festplatte des Lesers befindet, angezeigt. Die Farbe des Textes auf dieser Seite wird in dunkelblau angezeigt.

```
body { background-color: #ccffff; font-family: Arial, Helvetica, sans-serif;
color: #330066 }
```

Die folgenden vier Codezeilen definieren vier Pseudoklassen zur Formatierung der Link-Farben. Links von noch nicht besuchten Seiten (dasselbe Link wie das HTML-Attribut) werden mit der Pseudoklasse `a:link` definiert. Links von besuchten Seiten werden mit der Pseudoklasse `a:visited` definiert. Die Link-Farbe der Links, die gerade angeklickt werden (keine HTML Äquivalenz), werden beim anklicken der Links definiert.

```
a:link { color: #ff00ff }
a:visited { color: #660000 }
a:hover { color: #ffcc00 }
a:active { color: #ff0000 }
```

Gleich werden Sie feststellen können, dass Sie mit nur wenigen Zeilen Code und einem externen Style-Sheet eine Webseite erstellt haben, deren Design vollkommen anders ist. Öffnen Sie nun die `bookworm.html`-Seite in einem Cascading-Style-Sheets-fähigen Web-Browser. Normalerweise sollte diese Seite nun so wie in Bild 10.2. aussehen.

Eingebettete Style Sheets

Eingebettete Style Sheets sind gewöhnliche Webseiten, die Style-Regeln enthalten. Nehmen wir nun an, dass Sie Ihrer Homepage eine schwarze Hintergrundfarbe mit leuchtendem, großem Text und lebendigen Link-Farben geben wollen. Die restlichen

Webseiten sollen in einem gedämpfteren Ton gehalten werden (angenehmer zu lesen) und hellere Hintergrundfarben haben. Für eine einzige Seite extra ein externes Style Sheet zu erstellen, ergibt wenig Sinn. Warum nicht zwei Fliegen mit einer Klappe schlagen? Benutzen Sie deshalb ein eingebettetes Style Sheet für die eine Homepage und ein externes Style Sheet für die anderen Seiten.



Abbildung 10.2:
»Bookworm
Bookshop« mit
eingebundenem
externem Style
Sheet.

Sie könnten nun erwidern: »Moment mal, sind Style Sheets nicht dafür gedacht Style-Regeln zu trennen?« Natürlich sollen sie das. Die Style-Regeln erscheinen in der Kopfzeile, bevor der HTML-Inhalt beginnt. Sie sind also immer noch vom Inhalt getrennt. Bild 10.2 zeigt Ihnen die Kopfzeile der »Bookworm Bookshop« Homepage wie sie mit eingebetteten Styles erscheint. In diesem Fall weisen sie Text-, Hintergrund- und Link-Farben zu.

Um ein eingebettetes Style Sheet zu erstellen, beginnen Sie mit den folgenden Informationen im Dokumentkopf:

```
<!doctype html public "-//w3c//dtd xhtml 1.0 strict//en" "http://www.w3.org/tr/
xhtml1/dtd/strict.dtd">
<html>
<head>
<title>The Bookworm Bookshop</title>
```

Als Nächstes nehmen Sie sich den Bereich vor, in dem Sie Ihre Style-Regeln eingeben. Style-Regeln müssen sich in den öffnenden und schließenden `<style>`-Tags befinden. Das `type`-Attribut des `<style>`-Tag definiert die Seite als eine, die eingebettete Style Sheets verwendet.

```
<style type="text/css">
```

Geben Sie nach dem einleitenden `<style>`-Tag Ihre Style-Regeln an.

Folgendes Beispiel ist dem früher aufgezeigten Beispiel der externen Style Sheets ähnlich. Allerdings befinden sich Style-Regeln der eingebetteten Style Sheets in Kommentar-Tags (`<!--` und `-->`). Falls Sie die Style-Regeln nicht in runde Klammern setzen, können ältere Browser die Style-Tag nicht erkennen und Ihre Style-Regeln würden auf der Webseite erscheinen.

```
<!--
body { color: #000000; background-color: #ffffff }
a:active { color: #666699 }
a:hover { color: #3366ff }
a:link { color: #0066ff }
a:visited { color: #9966cc }
-->
```

Übung 10.2: Erstellen eines eingebetteten Style Sheet

Öffnen Sie die in `bookworm.html` gespeicherte »Bookworm Bookshop«-HTML-Webseite. Benutzen Sie einen anderen Dateinamen, zum Beispiel `bookback.html`, und speichern Sie eine Sicherungskopie dieser Datei ab, bevor Sie folgende Änderungen vornehmen. Diese Übung dient dazu, die `bookworm.html` in eine Webseite, die eingebettete Styles benutzt, zu konvertieren.

Um diese Webseite, die eingebettete Styles verwendet, zu identifizieren, müssen Sie den Kopf der Seite etwas verändern. Vergessen Sie nicht, dass eingebettete Style-Sheet-Codes in einleitenden und abschließenden `<style>`-Tags, und Style-Definitionen in geschweiften Klammern sein müssen.. Entfernen Sie zuerst nachstehend aufgeführte Zeile, die sich auf das externe Style Sheet der Webseite bezog.

```
<link rel="stylesheet" href="mystyle.css">
```

Geben Sie als Nächstes den Code, der das Style Sheet defininiert, in die Kopfzeile ein. Fügen Sie die `<style>`- und Kommentar-Tags hinzu (siehe folgendes Beispiel). Die Kopfzeile sollte nun wie folgt aussehen:

```
<!doctype html public "-//w3c//dtd xhtml 1.0 strict//en"
"http://www.w3.org/tr/xhtml1/dtd/strict.dtd">
<html>
<head>
<title>The Bookworm Bookshop</title>
<style type="text/css">
<!--
-->
</style>
</head>
```

Um die im eingebetteten Style Sheet definierten Styles zu illustrieren, werden wir eine geringfügige Änderung der Styles im vorhergehenden Beispiel vornehmen. Indem Sie `#ccffcc` eingeben, erhalten Sie einen hellgrünen Hintergrund (statt hellblau). Text- und Link-Eigenschaften bleiben die gleichen. Doch dieses Mal gehen die Style-Regeln in Ihre »Bookworm Bookshop«-HTML-Webseite, und zwar innerhalb des `<!--` und `-->` Kommentar-Tag. Geben Sie folgende Style-Regeln in das Kommentar-Tag Ihres Seitenkopfes ein:

```
body { background-color: #ccffcc; font-family: Arial, Helvetica, sans-serif;
color: #330066 }
a:link { color: #cc9900 }
a:visited { color: #660000 }
a:hover { color: #ffcc00 }
a:active { color: #ff0000 }
```

Ihr Seitenkopf müsste nun wie im folgendem Code-Beispiel aussehen:

```
<!doctype html public "-//w3c//dtd xhtml 1.0 strict//en" "http://www.w3.org/tr/
xhtml1/dtd/strict.dtd">
<html>
<head>
<title>The Bookworm Bookshop</title>
<style type="text/css">
<!--
body { background-color: #ccffcc; font-family: arial, helvetica, sans-serif;
color: #330066 }
a:link { color: #cc9900 }
a:visited { color: #660000 }
a:hover { color: #ffcc00 }
a:active { color: #ff0000 }
```

```
-->
</style>
</head>
```

Speichern Sie die Seite wieder als `bookworm.html` ab und öffnen Sie sie anschließend in Ihrem Style-Sheet-kompatiblen Browser. Die Seite hat nun einen hellgrünen Hintergrund, ansonsten sieht sie fast genauso wie die externe Style-Sheet-Version aus (in Bild 10.2. aufgezeigt).

Damit Sie dies alles auch vergleichen können, öffnen Sie dieselbe Webseite in einem CSS-unfähigen Browser. Auf Bild 10.3. sehen Sie die gleiche Seite, allerdings mit NCSA Mosaic 3.0. dargestellt. Da der HTML-Code separat stand, sind alle Informationen noch komplett vorhanden. Allerdings werden die von Ihnen im Style Sheet eingefügte Farben und Schriftarten in diesem Browser nicht angezeigt.

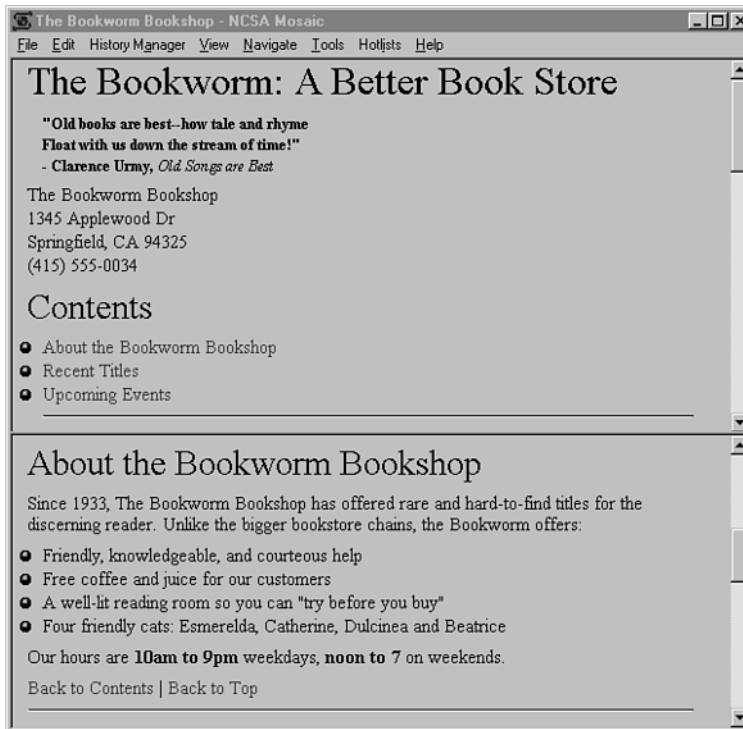


Abbildung 10.3:
»Bookworm Bookshop«-Seite von
einem CSS-unfähigen Browser
angezeigt.

Was bedeutet Cascading?

Was bedeutet es eigentlich, dass dieser Typ von Style Sheets, Cascading Style Sheets genannt wird? Eines ist klar: »Cascading« hat nichts mit Wasserfällen zu tun! »Cascading« steht für die Fähigkeit der Style-Informationen, sich mit mehr als einer Quelle zu verknüpfen. Wie Sie ja schon bereits gelernt haben, können Sie Style-Regeln einer Seite auf vielseitige Art und Weise anwenden. Externe Style Sheets können in mehrere Seiten eingebunden werden und verwenden dabei die gleichen Style-Regeln auf allen Seiten. Eingebettete Style Sheets wenden Style-Regeln nur auf einer einzigen Seite an und Inline-Styles wenden Style-Regeln, wie Sie später noch sehen werden, nur auf Seitenelemente an. Sie könnten auch alle drei Arten auf einer einzigen Seite anwenden.

Der »Cascading« Teil kommt nun ins Spiel, denn es gibt eine geordnete Reihenfolge der Style-Sheets. Regeln, die in späteren Sheets definiert wurden, ersetzen die vorherigen. Um dies alles deutlicher zu machen, werden wir Ihnen aufzeigen was passiert, wenn externe, eingebettete und Inline-Styles auf einer Seite angewandt werden:

- ▶ Styles, die in einem externen Style Sheet definiert wurden, werden als erste auf der Seite angewandt.
- ▶ Styles, die in einem eingebetteten Style Sheet definiert wurden, ersetzen die Styles, die zuerst angegeben wurden (extern).
- ▶ Inline-Styles ersetzen externe und eingebettete Style Sheets.

Die wesentlichen Style-Sheet-Eigenschaften und -Werte

Nun werden Sie einiges über die Haupteigenschaften sowie auch über die Anwendung von Style Sheets erfahren. Es gibt aber deren so viele, dass man sie unmöglich im Kapitel eines Tages aufzählen könnte. Falls Sie mehr darüber wissen möchten, finden Sie im Web unter <http://www.w3.org/TR/REC-CSS1> zusätzliche Informationen über Cascading Style Sheets Level 1 (CSS1). Diese Empfehlung ist vom 1. Januar 1999.

Unter <http://www.w3.org/TR/REC-CSS2> finden Sie zusätzliche Informationen über Cascading Style Sheets Level 2 (CSS2). (Die aktuellste Version dieser Datei stammt vom 12. Mai 1998). Diese eher technische Datei liefert Ihnen ausführliche Informationen über jede Eigenschaft und über die anzuwendenden Codes. Außerdem finden Sie einige Beispiele vor, die Ihnen helfen werden, Seiten zu erstellen, die mit alten und neuen Browsern kompatibel sind.

In den vorhergehenden Beispielen haben wir nur einige Fähigkeiten der Style-Regeln erwähnt, in Wirklichkeit können sie aber noch viel mehr. CSS1 und CSS2 enthalten eine Vielzahl von Tags, mit denen Sie die Kontrolle über jeden einzelnen Aspekt Ihrer Webseite haben.

Die CSS-Technik entwickelt sich jedoch ständig weiter und Browser unterstützen einige CSS-Eigenschaften besser und sicherer als andere. Das ist nun einmal so im Web. Ich möchte Sie daher schon vorher vor unerwarteten Überraschungen und Varianten, die in jedem CSS-fähigen Browser auftauchen können, warnen. Da gibt es nur eines zu tun: ausprobieren und nochmals ausprobieren und zwar solange, bis sie ein akzeptables Ergebnis erzielen.

Kontrolle des Seiten-Layouts mit CSS-Eigenschaften

In Übungen 10.1 und 10.2 haben Sie einfache Style-Regeln eingefügt, die Auswirkungen auf Schriftart und -Farbe Ihrer Webseite hatten. Weitere Style-Regeln, die sich auf Hintergrundfarbe, Schriftart und Farbeigenschaften beziehen, haben Sie im Textkörper der Webseite angegeben. Die Link-Farben wurden erst durch die Definition der vier Pseudoklassen für das `<a>`-Tag bestimmt: `a:link` (für die Link-Farbe), `a:visited` (für die besuchten Links), `a:hover` (für die Farbe der gerade angeklickten Links), `a:active` (für die aktive Link-Farbe).

Es gibt aber weitaus mehr Effekte, die für das Erscheinungsbild Ihres Textes erzielt werden können. So können Sie zum Beispiel Randabstand und Padding bestimmen (etwa so wie in einem Desktop-Publishing- oder Textverarbeitungsprogramm).

Tabelle 10.1 wird Ihnen einige der meist benutzten Eigenschaften aufzeigen.

Eigenschaft	Beschreibung
<code>margin-top</code>	Legt den obere Randabstand eines Elements zu seinem vorausgehenden Element fest. Für die Werte sind numerische und prozentuale Angaben oder die Angabe <code>auto</code> erlaubt.
<code>margin-right</code>	Legt den rechten Randabstand eines Elements fest. Dabei werden die gleichen Werte wie in <code>margin-top</code> verwendet.
<code>margin-left</code>	Legt den linken Randabstand eines Elements fest. (Werte: siehe <code>margin-top</code>)
<code>margin-bottom</code>	Legt den Abstand eines Elements zu seinem nachfolgenden Element fest. (Werte: siehe <code>margin-top</code>)

Tabelle 10.1: Nützliche Seiten-Layout-Eigenschaften

Eigenschaft	Beschreibung
Margin	Diese Angabe ist eine Zusammenfassung folgender Einzelangaben: <code>margin-top</code> , <code>margin-right</code> , <code>margin-bottom</code> , <code>margin-left</code> . Hiermit wird ein einheitlicher Randabstand für oben, unten, links und rechts festgelegt. Erlaubt sind numerische und prozentuale Angaben oder die Angabe <code>auto</code> .
<code>padding-top</code>	Mit der hier beschriebenen Angabe können Sie den Zwischenraum zwischen Elementinhalt und oberer Elementgrenze (sichtbar bei Rahmen, Farben usw.) bestimmen. Erlaubt sind numerische und prozentuale Angaben oder die Angabe <code>auto</code> .
<code>padding-right</code>	Mit der hier beschriebenen Angabe können Sie den Zwischenraum zwischen Elementinhalt und rechter Elementgrenze bestimmen. (Werte: siehe <code>padding-top</code>)
<code>padding-bottom</code>	Mit der hier beschriebenen Angabe können Sie den Zwischenraum zwischen Elementinhalt und unterer Elementgrenze bestimmen. (Werte: siehe <code>padding-top</code>)
<code>padding-left</code>	Mit der hier beschriebenen Angabe können Sie den Zwischenraum zwischen Elementinhalt und linker Elementgrenze bestimmen. (Werte: siehe <code>padding-top</code>)
<code>padding</code>	Mit der hier beschriebenen Angabe können Sie den Zwischenraum zwischen Elementinhalt und Elementgrenze bestimmen. Mit <code>padding</code> bestimmen Sie einen einheitlichen Innenabstand zwischen Elementinhalt und den vier Elementgrenzen oben, links, unten und rechts. (Werte: siehe <code>padding-top</code>)

Tabelle 10.1: Nützliche Seiten-Layout-Eigenschaften

Randabstände und Padding werden mit numerischen Angaben, gefolgt von einer Längeneinheit, mit prozentualen Werten (gegenüber Elementnorm) oder durch den Wert `auto` festgelegt.



Längeneinheiten werden durch relative oder absolute Werte festgelegt. Relative Längeneinheiten sind `EM` (die Höhe der betreffenden Schrift), `EX` (bezogen auf die Höhe des Buchstaben `x` der betreffenden Schrift), `PX` (für Pixel = relative Angabe im Hinblick auf die Auflösungen des Ausgabegerätes). Absolute Werte / Angaben enthalten `PT` für Punkt (= 1/72 inches), `IN` für Inch (= 2,54 cm), `CM` für Zentimeter, `MM` für Millimeter, `PC` für Pica (=12 Punkt)

Prozentuale Werte beziehen sich immer auf einen anderen Wert (zum Beispiel Länge). Prozente werden durch ein `+` oder `-` Zeichen angegeben, dem sofort eine Nummer und Prozentzeichen folgen.

Sie haben sicherlich den Kurzbefehl in Tabelle 10.1 bemerkt. Einige Eigenschaften benutzen eine Vielzahl von Attributen, um deren Erscheinungsbild zu definieren, und deshalb können einige Werte mittels Kurzbefehl sämtliche Eigenschaften miteinander verbinden. So können Sie zum Beispiel den oberen, rechten, unteren und linken Randabstand in einer einzigen Eigenschaft, `margin`, definieren.

`margin` und `padding` können bis zu vier numerische Angaben haben.

Diese sind:

Ein Wert:	wird auf allen Seiten benutzt
Zwei Werte:	Der erste Wert bezieht auf den unteren und oberen Teil / Rand; der zweite Wert bezieht sich auf den rechten und linken Teil.
Drei Werte:	Der erste Wert bezieht sich auf den oberen Teil; der zweite Wert bezieht sich auf den rechten und linken Teil; Der dritte Wert bezieht sich auf den unteren Teil.
Vier Werte:	Jeder Wert bezieht sich jeweils auf den oberen, rechten, unteren und linken Teil.

Übung 10.3: Randabstand und Padding auf einer Seite anwenden

In dieser Übung werden Sie einige Randabstände und Paddings in die »Bookworm Bookshop«-Seite eingeben. Folgende Style-Definitionen müssen Sie in das Style Sheet eingeben:

- ▶ 20 Pixel Rand für die Definition den oberen und unteren Abstand der Seite und 30 Pixel Rand für den rechten und linken Abstand der Seite.
- ▶ 15 Pixel Padding jeweils über und unter jeder Überschrift.

Fügen Sie folgenden Code entweder in das externe Style Sheet (`mystyle.css`) oder in die Webseite mit eingebettetem Style Sheet ein (`bookworm.html`), um Ränder und Padding in der Webseite zu definieren. Wenn Sie keine Sicherheitskopie Ihrer Webseite erstellt haben, können Sie nur mit dem externen Style Sheet arbeiten. Deshalb werde ich diese Seite weiterhin für meine Beispiele verwenden.

Für diese Aufgabe können Sie mit dem Kurzbefehl zwei **Werte** für die Ränder eingeben. Dabei werden die oben/unten, links/rechts Ränder bezüglich des Textkörpers und das obere/untere Padding bezüglich der Überschrift durch die Befehle `<h1>` und `<h2>` bestimmt. Die überarbeitete Version Ihrer Style-Definitionen sieht folgendermaßen aus (in diesem Beispiel habe ich das `<body>`-Tag neu arrangiert, sodass Sie die einzelnen CSS-Regeln besser erkennen können):

```
<!--
body { background-color: #ccffcc;
      font-family: arial, helvetica, sans-serif;
      color: #330066;
      margin: 50px 70px }
a:link { color: #cc9900 }
a:visited { color: #660000 }
a:hover { color: #ffcc00 }
a:active { color: #ff0000 }
h1 { padding-top: 10px;
     padding-bottom: 5px }
h2 { padding-top: 5px;
     padding-bottom: 3px }
-->
```

Nachdem Sie die neue Version Ihres Style Sheets gespeichert haben, öffnen Sie sie in einem CSS-kompatiblen Browser und betrachten Sie das Resultat.

Hintergrund, Farben, Bilder

Seit dem Kapitel des 7. Tags haben Sie einiges an Grundwissen über die Gestaltung von Hintergrundbildern in einer HTML-Datei erworben. Mit `bgcolor` und `background` können Sie jeweils die Hintergrundfarbe eines Dokuments oder die Hintergrundfarbe eines Bildes bestimmen. Einige Browser haben diese Funktion nun erweitert und ermöglichen es den Autoren von Webseiten bei der Gestaltung von Tabellen, die Hintergrundfarbe einzelner Zellen festzulegen.

CSS bietet Ihnen sogar noch mehr Möglichkeiten, denn insgesamt gibt es ja sechs Eigenschaften, die das Erscheinungsbild des Hintergrundes beeinflussen. Das ist unter anderem darauf zurückzuführen, dass Style Sheets elementweise angewandt werden können und deshalb mehrere Hintergründe auf einer Seite möglich sind.

In Tabelle 10.2. finden Sie diese sechs Eigenschaften.

Eigenschaft	Beschreibung
<code>color</code>	Bestimmt die Vordergrundfarbe eines Elements (wird meistens auf den im Element erscheinenden Text angewandt). 16 festgelegte Farben oder eine von mehreren RGB-Werten stehen hier für den Farbwert zur Auswahl.
<code>background-color</code>	Hiermit kann die Hintergrundfarbe eines Elements bestimmt werden. Farbwerte sind hierbei 16 Farbangaben / Farben, eine von mehreren RGB-Werten oder der Wert <code>transparent</code> .

Tabelle 10.2: CSS-Hintergrundeigenschaften

Eigenschaft	Beschreibung
background	Hiermit wird das Hintergrundbild bestimmt. Eine URL oder none sind die zur Auswahl stehenden Werte.
background-repeat	Hiermit wird bestimmt, ob die Hintergrundgrafik wiederholt wird oder nicht. Mögliche Werte sind: repeat, repeat-x, repeat-y, no-repeat.
background-attachment	Mit background-attachement können Sie das Scroll-Verhalten einer Hintergrundgrafik kontrollieren. Erlaubt ist eine der folgenden Angaben. scroll fixed.
background-position	Mit background-position können Sie festlegen, wo sich die Hintergrundgrafik befinden. Die hierbei erlaubten Werte werden im folgenden Text angegeben.
Background	Diese Angabe ist eine Zusammenfassung der folgenden möglichen Einzelangaben: background-image, background-repeat, background-attachement, background-position. Dabei können eine oder mehrere dieser Angaben in einem Style Sheet verwendet werden (Kurzbehl).

Tabelle 10.2: CSS-Hintergrundeigenschaften (Forts.)

Das W3C empfiehlt Ihnen bei der Anwendung eines Hintergrundbildes in der Webseite, ebenfalls eine Hintergrundfarbe zu definieren. Falls das Hintergrundbild einem Anwender unzugänglich sein sollte, so wird ihr oder ihm immer noch eine Hintergrundfarbe angezeigt.

Die background-position-Eigenschaft ist ziemlich komplex und bedarf daher ausführlicher Erklärungen. Diese Eigenschaft akzeptiert zwei, durch eine Leerstelle getrennte Werte und wird mittels einem der drei folgenden Beispiele festgelegt.

- ▶ Mit Schlüsselwort. Das Schlüsselwort top, center oder bottom identifiziert die senkrechte, das Schlüsselwort left, center oder right identifiziert die waagrechte Position des Hintergrunds. Um nun das zentrierte Hintergrundbild einer Seite, horizontal und vertikal zentriert auf eine Webseite zu bekommen, geben Sie die Werte center ein.
- ▶ Mit Längeneinheit. Die Werte werden mit x,y Koordinaten festgelegt, wobei x die horizontale Achse ist (Abstand vom linken Teil der Seite) und y die vertikale Achse (Abstand vom oberen Rand der Seite). Der Wert 20/25 positioniert die obere linke Ecke des Bildes 20 Pixel nach links und 25 Pixel vom oberen Rand oder Element.
- ▶ Mit prozentualem Wert. Die ursprüngliche Position der background-position-Eigenschaft ist 0%0%, wobei dieser Wert der oberen linken Seite eines Elements entspricht. Mit dem 100%100%-Wert, wird das Element auf die untere rechte Ecke der Seite positioniert.

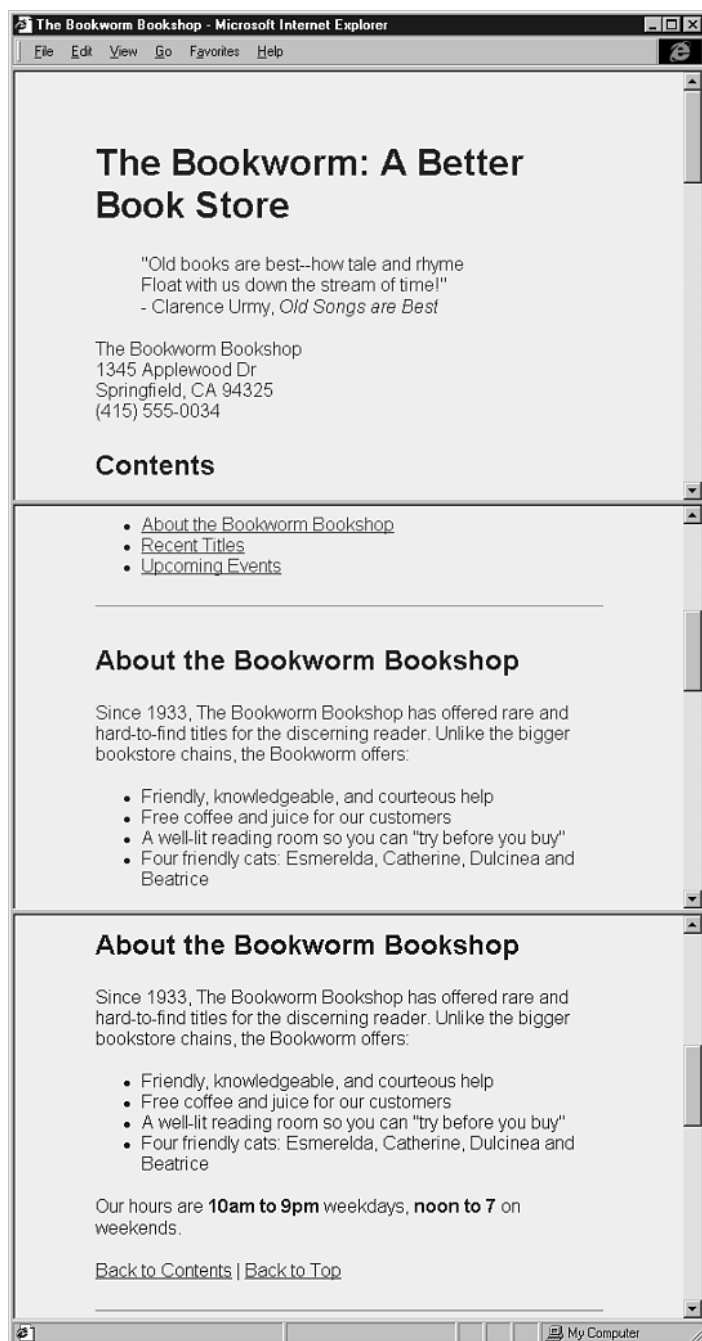


Abbildung 10.4:
Rand und Innenab-
stand in der »Book-
worm Bookshop« Seite

Übung 10.4: Hintergründe und Farben mit CSS auf Elemente angewandt

Mit CSS-Tags haben Sie schon einige Farben in Ihrer »Bookworm Bookshop«-Web-Seite eingefügt. Dann können wir jetzt weitermachen. Die folgenden CSS-Codes werden ein Hintergrundbild in Ihre Webseite einbinden. Die Hintergrundbild-Eigenschaft schreibt die URL des Hintergrundbildes vor und teilt dem Browser mit, das Hintergrundbild wie in Bild 10.5 aufzuzeigen.



Die `background.gif`-Datei des folgenden Beispiels steht in der Web-Site dieses Buches unter <http://www.tywebpub.com/>.

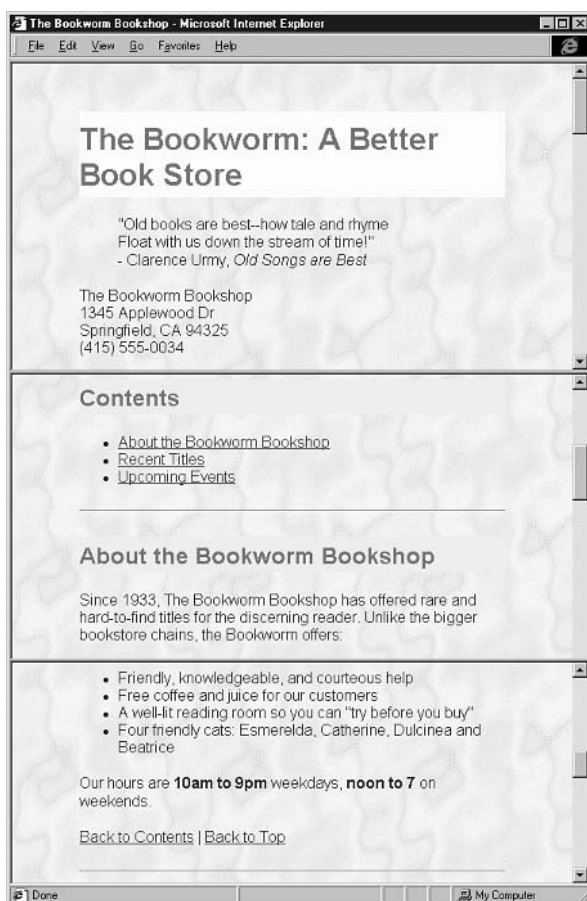


Abbildung 10.5:
Hintergrund- und Farbeigen-
schaften, die in der »Bookworm
Bookshop«-Seite verwendet
wurden.

Hintergrund und Farben können auch woanders als in der Webseite verwendet werden. Sie können zum Beispiel verschiedene Farben und Hintergründe in einzelne Elemente der Webseite einfügen. Der folgende CSS-Code erzielt interessante Effekte bei den Überschriften 1. und 2. Ordnung, die nun in brauner Farbe angezeigt werden sollen. Überschriften 1. Ordnung stehen auf hellgelbem Hintergrund (erzielt durch den Befehl `color #ffffcc`), und Überschriften 2. Ordnung sind auf hellgrünem Hintergrund zu sehen (erzielt durch `color #ccffcc`).

Um Ihnen die Veränderungen in Ihrer Seite klar vor Augen zu führen, haben wir in den folgenden Beispielen alles, was abgeändert wurde, grau schattiert. Nehmen Sie nun die Änderungen Ihrer Style-Definitionen vor und fügen Sie dabei die hellgrau schattierten Zeilen ein.

```
<!--
body { background-color: #ccffcc;
      font-family: arial, helvetica, sans-serif;
      color: #330066;
      margin: 50px 70px;
      background-image: url(background.gif);
      background-repeat: repeat }
a:link { color: #cc9900 }
a:visited { color: #660000 }
a:hover { color: #ffcc00 }
a:active { color: #ff0000 }
h1 { color: #996633;
     padding-top: 10px;
     padding-bottom: 5px;
     background-color: #ffffcc }
h2 { color: #996633;
     padding-top: 5px;
     padding-bottom: 3px;
     background-color: #ccffcc }
-->
```

Umrahmungen einsetzen

CSS verfügen über viele Eigenschaften bezüglich der *Rahmenumgebung* einzelner Elemente Ihrer Seite. In HTML hatten Sie nur einige umrahmte Objekte, wie etwa Bilder und Tabellenzeilen, aber mit CSS können Sie theoretisch jedes beliebige Element Ihrer Seite umrahmen.

Tabelle 10.3 zeigt Ihnen die border-Eigenschaften auf.

Eigenschaft	Beschreibung
<code>border-style</code>	Sie können für einen Rahmen, den Sie für bestimmte oder alle Seiten eines Elements definieren, das Erscheinungsbild des Rahmens festlegen. Dabei sind die Werte dieselben, wie die, die in <code>border-bottom-style</code> festgelegt sind. Mit <code>border-bottom-style</code> , <code>border-left-style</code> , <code>border-right-style</code> oder <code>border-top-style</code> , können Sie den Rahmentyp jedes einzelnen Rahmens bestimmen (oben, rechts, links, unten). Die zu verwendenden Werte sind: <code>none</code> , <code>dotted</code> , <code>dashed</code> , <code>solid</code> , <code>double</code> , <code>groove</code> , <code>ridge</code> , <code>inset</code> , <code>outset</code>
<code>border-color</code>	Mit <code>border-color</code> können Sie alle oder einzelne Rahmenfarben festlegen. Individuelle Festlegung der Rahmenfarben erfolgt durch <code>border-bottom-color</code> , <code>border-left-color</code> , <code>border-right-color</code> oder <code>border-top-color</code> . Dabei gibt es 16 definierte Farbangaben. Weitere zu verwendende Werte sind: einer der RGB-Werte oder der Wert <code>transparent</code> .
<code>border-width</code>	Sie können für die Rahmen rund um ein Element die Dicke festlegen. Mit den Befehlen <code>border-bottom-width</code> , <code>border-left-width</code> , <code>border-right-width</code> oder <code>border-top-width</code> wird die Rahmendicke individuell festgelegt. Werte hierbei sind: <code>thin</code> , <code>medium</code> , <code>thick</code> oder ein Längenwert.
<code>border</code>	Hier handelt es sich wieder um einen Kurzbefehl mit der Eigenschaft gleichzeitig Rahmendicke, -farbe und -typ für alle Rahmen eines Elements festzulegen. Mit den Befehlen <code>border-bottom</code> , <code>border-left</code> , <code>border-right</code> oder <code>border-top</code> können Rahmendicke, -farbe und -typ individuell festgelegt werden.

Tabelle 10.3: Die verschiedenen `border`-Eigenschaften in CSS

Übung 10.5: Umrahmung von Elementen mit CSS

Folgende Codes werden Ihnen die Umrahmung der Überschriften Ihrer Seite ermöglichen. Dadurch werden die Überschriften wie Banner aussehen. Die `border-color`, `border-style`- und `border-width`-Eigenschaften wurden auf jede der Überschriften angewandt. Der obere und linke Rahmen jeder Überschrift wird eine andere Farbe als der untere und rechte Rahmen haben. Hierfür müssen Sie nur zwei Werte für die `border-color`-Eigenschaft eingeben. Rahmen der Überschriften 1. Ordnung erwecken den Anschein, als würden sie nach außen stehen, während Rahmen Überschriften 2. Ordnung den gegenteiligen Effekt haben. Dies wird durch die `border-style:outset`- und `border-style:inset`-Eigenschaften erzielt, sowie durch die Attribute der jeweiligen Befehle.

Die Farben um die Rahmen der Überschriften herum erzielen Sie durch den in hellgrau hervorgehobenen Code der CSS-Sektion in Ihrer »Bookworm Bookshop«-Webseite.

```
<!--
body { background-color: #ccffcc;
      font-family: arial, helvetica, sans-serif;
      color: #330066;
      margin: 50px 70px;
      background-image: url(background.gif);
      background-repeat: repeat }
a:link { color: #cc9900 }
a:visited { color: #660000 }
a:hover { color: #ffcc00 }
a:active { color: #ff0000 }
h1 { color: #996633;
     padding-top: 10px;
     padding-bottom: 5px;
     background-color: #ffffcc;
     border-color: #cccc33 #cc9933;
     border-style: outset;
     border-width: thin }
h2 { color: #996633;
     padding-top: 5px;
     padding-bottom: 3px;
     background-color: #ccffcc;
     border-color: #99cc33 #996633;
     border-style: inset;
     border-width: thin }
-->
```

Wenn sie nun Ihre Webseite in einem CSS-fähigen Browser anzeigen, sollte sie ähnlich wie in Bild 10.6 aussehen. Im Internet Explorer 4 und Netscape Navigator 4 werden Hintergrund und Rahmen etwas anders erscheinen. Während die Rahmen im Internet Explorer etwas kräftiger wirken, sind sie im Netscape dünner und haben den Anschein, als seien sie von der Hintergrundfarbe abgehoben.

Schriftart und Schriftstil

Cascading Style Sheets haben eine große Auswahl an Eigenschaften, die die Schriftart definieren. Mit CSS hat man tatsächlich eine bessere Kontrolle der Schriftart, als mit einem einfachen -HTML-Tag.

Tabelle 10.4 zeigt Ihnen die wesentlichen Eigenschaften für die Kontrolle von Schriftarten.

Eigenschaft	Beschreibung
font-family	Hiermit wird die Schriftformatierung bestimmt. Den Schriftartennamen vergeben Sie mit der Angabe <code>font-family</code> : wie etwa Arial, Times oder Palatino oder geben Sie eine der fünf Schriftarten-Namen an: serif, sans-serif, cursive, fantasy, monospace.
font-size	Mit <code>font-size</code> können Sie die Schriftgröße bestimmen. Erlaubt sind relative, absolute und prozentuale Angaben.
font-style	Mit <code>font-style</code> können Sie den Schriftstil bestimmen. Folgende Angaben sind möglich: italic, oblique, normal.
font-weight	Schriftgewicht ist der Fettigkeitsgrad, in dem die Schrift dargestellt wird. Mit <code>font-weight</code> können Sie das Schriftgewicht bestimmen. Folgende Angaben sind möglich: bold, bolder, lighter.
font-variant	Mit <code>font-variant</code> können Sie die Schriftvariante bestimmen. Folgende Angaben sind möglich: small-caps, normal.
font	Kurzbehl mit der Eigenschaft, folgende Einzelangaben im Style Sheet zusammenzufassen: font-weight, font-size, font-style, font-family.
line-height	Die <code>line-height</code> -Eigenschaft wird im kommenden Unterkapitel »Eigenschaften zur Textausrichtung in CSS« erklärt.

Tabelle 10.4: Schriftart-Eigenschaften in CSS

Im Folgenden finden Sie noch einige Anmerkungen zu der obigen Liste:

- ▶ Wenn Sie die Schriftarten mit der `font-family`-Eigenschaft festlegen, steht Ihnen eine durch Komma getrennte Namensliste zur Verfügung. Sollte der Browser die erste Schriftart nicht zur Verfügung haben, so geht er sofort auf den nächsten, in der Liste aufgeführten Namen über. Das W3C empfiehlt, `font-family` an das Listenende zu stellen, da jeder Browser am Ende einer Schriftliste über eine schon vorher festgelegte `font-family` verfügt.
- ▶ Sollten Sie eine Bezeichnung für Schriftarten verwenden, müssen Sie bei Kursiv- oder Fantasy-Schriften vorsichtig sein. Diese zwei »Familien« hängen von der Schriftart, die der oder die Leser /-in auf dem Betriebssystem hat, ab. Die serif, sans-serif und monospace-Schrift sind mittels Operating-System im Web-Browser installiert, doch für die Kursiv- und Fantasy-Schrift trifft dies nicht unbedingt zu.
- ▶ Wenn Sie die Schriftgröße mit `font-size` bestimmen, sollten Sie wissen, dass bestimmte Werte mittels Schlüsselwort definiert sind. Diese Werte sind: XX-small, X-small, small, medium, larger, X-larger, XX-larger, smaller, larger. Diese Werte stimmen mit bestimmten Schriftgrößen im Browser überein. Relative Schriftgrößen stehen in der Relation zur Schriftart des Elements und können als größer oder kleiner definiert werden.

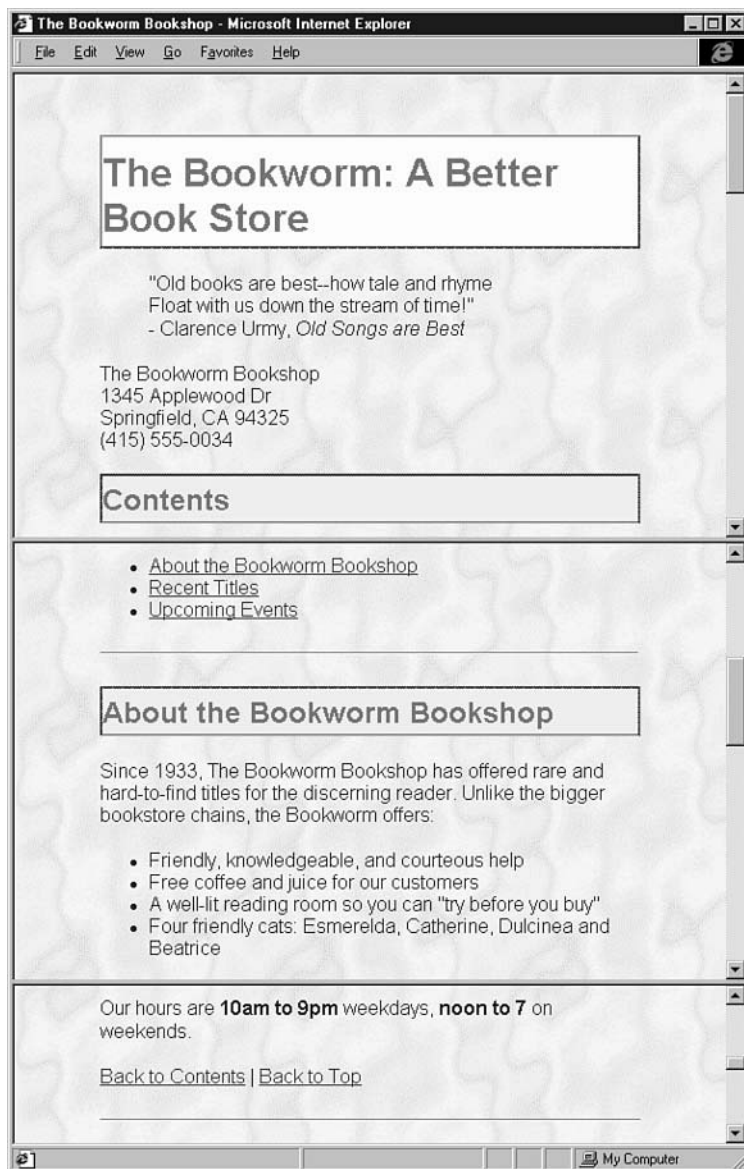


Abbildung 10.6:
Rahmen-Eigen-
schaften im Inter-
net Explorer 4.0.

- Schriftgewicht wird durch numerische Werte auf einer Skala festgelegt. Folgende Angaben sind erlaubt: 100, 200, 300, 400, 500, 600, 700, 800, 900 = extra-dünn (100) bis extra-fett (900). Die Eigenschaft des Schriftgewichts hat einen der folgenden Werte: **bold**, **bolder**, **lighter** oder eben eine der Zahlen.

Alle Werte bis 400 auf der Zahlenskala sind normal, Zahlen bis zu 700 = **bold**, **bold**er und **lighter** bestimmt das Gewicht relativ zum entsprechenden Element.

Folgende Code Beispiele werden Ihnen einige dieser Schriftart-Eigenschaften demonstrieren. Bild 10.7 zeigt Ihnen das Ergebnis.



```
<!doctype html public "-//w3c//dtd xhtml 1.0 strict//en" "http://www.w3.org/tr/
xhtml1/dtd/strict.dtd">
<html>
<head>
<title>CSS Font Properties</title>
<style type="text/css">
<!--
body { background-color: #ffffff }
-->
</style>
<p><span style="font-family: arial">font-family: Arial</span> <br />
<span style="font-family: fantasy">font-family: fantasy</span> </p>
<hr />
<span style="font-size: small">font-size: small</span>
<span style="font-size: medium">font-size: medium</span>
<span style="font-size: xx-large"> font-size: xx-large</span>
<hr />
<span style="font-style: italic">font-style: italic</span>
<hr />
<span style="font-weight: 100">font-weight: 100</span>
<br />
<span style="font-weight: 500">font-weight: 500</span>
<br />
<span style="font-weight: 900">font-weight: 900</span>
<hr />
<span style="font-variant: normal">font-variant: normal</span>
<br />
<span style="font-variant: small-caps">font-variant: small-caps</span>
</body>
</html>
```

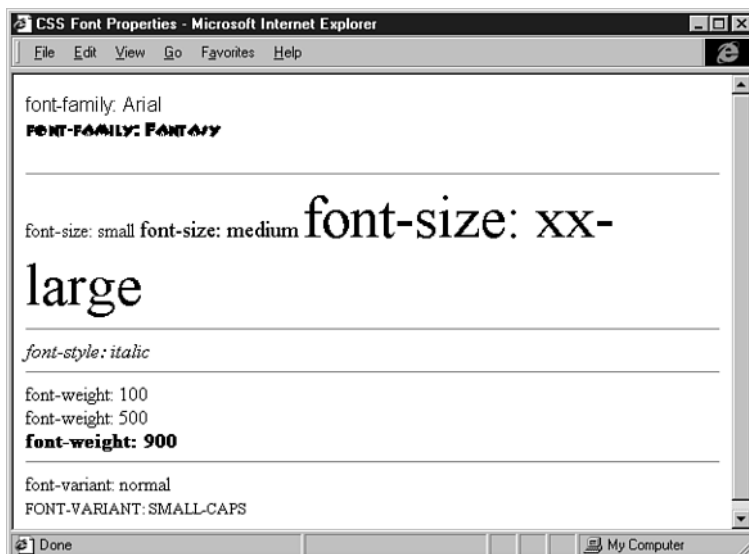


Abbildung 10.7:
Verschiedene
Schriftart-Eigen-
schaften

Eigenschaften zur Textausrichtung in CSS

CSS hat, zusätzlich zu den Schriftart-Eigenschaften, auch zahlreiche Eigenschaften, die dem Verfasser ermöglichen, seinen Text auszurichten. Diese Eigenschaften erlauben es, die typografische Kontrolle eines Word Prozessors und Desktop Publishing Applikationen zu haben, was, bis zur Entwicklung der CSS im Web, vorher nicht möglich war.

Die Haupteigenschaften bezüglich der Textausrichtung finden sie in Tabelle 10.5.

Eigenschaft	Beschreibung
word-spacing	Mit dieser Angabe können Sie den Abstand zwischen Wörtern im Text bestimmen. Mögliche Werte sind hierbei Längenwert oder <code>normal</code> (default). Diese Angabe wird von Netscape und dem MS Internet Explorer in den Versionen 4.0 noch nicht interpretiert.

Tabelle 10.5: Eigenschaften zur Textausrichtung in CSS

Eigenschaft	Beschreibung
LETTER-spacing	Mit dieser Angabe können Sie den Abstand zwischen den Buchstaben bzw. Zeichen im Text bestimmen. Mögliche Werte sind hierbei: Absolute Länge oder <code>normal</code> (default) (vom Internet Explorer 4.0, aber nicht vom Netscape Navigator 4.0 unterstützt).
line-height	Mit <code>line-height</code> können Sie den Abstand der unteren und oberen Basislinie festlegen. Erlaubt ist eine numerische Angabe. Bei numerischen Angaben wird die Zeilenhöhe mit der Nummer der Schriftgröße multipliziert. Beispiel: <code>line-height: 2</code> schafft einen Zeilenabstand, der zweimal so groß ist wie die Schriftart. Absolute Längen können ebenfalls für die Zeilenhöhe definiert werden. Beispiel: <code>line-height: 15px</code> schafft einen Abstand von 15 Pixel zwischen den Zeilen. Prozentuale Werte basieren auf der Höhe des Elements. <code>line-height: 200%</code> ist identisch mit der Bestimmung eines numerischen Wertes der <code>line-height: 2</code> . Sie können ebenfalls die Angabe <code>normal</code> verwenden, die der ursprünglich festgelegten Höhe entspricht (wird vom Netscape Navigator 4.0 unterstützt).
vertical-align	Sie können nebeneinanderstehende Elemente mit unterschiedlicher Höhe, zum Beispiel Tabellenzellen in einer Tabellenzeile, oder Textpassagen mit unterschiedlicher Schriftgröße innerhalb einer Zeile im Verhältnis zueinander ausrichten. Folgende Angaben sind möglich: <code>baseline</code> , <code>middle</code> , <code>sub</code> , <code>super</code> , <code>text-top</code> , <code>text-bottom</code> . Diese Werte werden im Verhältnis zueinander ausgerichtet. <code>top</code> und <code>bottom</code> werden im Verhältnis zur (Basis)Linie ausgerichtet. Eine prozentuale Angabe ist möglich. Die Ausrichtung orientiert sich dann an der elementeigenen Zeilenhöhe (wird vom Internet Explorer 4.0 unterstützt).
text-align	Mit <code>text-align</code> können Sie gewünschte Elemente für Textinhalte ausrichten. Folgende Angaben sind möglich: <code>left</code> , <code>center</code> , <code>right</code> , <code>justify</code> (wird vom Netscape Navigator 4.0 unterstützt, jedoch nicht vom Internet Explorer 4.0).
text-decoration	Mit dieser Angabe können Sie die »Dekoration« der Schrift im Text bestimmen. Erlaubte Angaben sind: <code>underline</code> , <code>overline</code> , <code>line-through</code> , <code>blink</code> , <code>none</code> . Die ersten drei Werte werden sowohl vom Internet Explorer 4.0 als auch vom Netscape Navigator 4.0 unterstützt, <code>blink</code> jedoch nur von Netscape.
text-indent	Mit <code>text-indent</code> können Sie eine Einrückung für die erste Zeile bestimmen. Erlaubt sind prozentuale Werte der Elementdicke oder Werte, die die absolute Länge angeben. Prozentuale Werte basieren auf der Dicke des Elements und funktionieren am besten (vom Netscape Navigator 4.0, aber nicht vom Internet Explorer 4.0 unterstützt).
white-space	Mit <code>white-space</code> können Sie das Verhalten für den Zeilenumbruch festlegen. Folgende Angaben sind möglich: <code>normal</code> , <code>pre</code> , <code>nowrap</code> und entsprechende HTML-Tags.

Tabelle 10.5: Eigenschaften zur Textausrichtung in CSS

Sie sehen also, diese CSS-Eigenschaften werden unterschiedlich vom Internet Explorer 4.0 und Netscape Navigator 4.0 unterstützt. Einige werden von dem einen, aber nicht von dem anderen Browser unterstützt und vice-versa, andere Eigenschaften wiederum werden von beiden Browsern unterstützt. Es muss also nicht unbedingt an Ihren Codes liegen, wenn Sie ab und zu nicht mehr wissen was nun was ist.

Folgendes Beispiel enthält viele Texteingenschaften, die auf verschiedene Textelemente angewandt werden. Beachtenswert ist, auf welche Art und Weise die gleichen Codes in den beiden Haupt-Browsern verwendet werden. Jetzt werden Sie auch sicherlich verstehen, warum man die Seiten in verschiedenen Browsern überprüfen sollte.

Bild 10.8. und 10.9. werden Ihnen zeigen, wie die Versionen 4.0 vom Internet Explorer und vom Netscape Network diese Eigenschaften verschieden wiedergeben.

```
<!doctype html public "-//w3c//dtd xhtml 1.0 strict//en"
"http://www.w3.org/tr/xhtml1/dtd/strict.dtd">
<html>
<head>
<title>CSS Text Properties</title>
<style type="text/css">
<!--
body { background-color: #ffffff }
-->
</style>
<p><span style="word-spacing: normal"> word-spacing: normal </span> <br />
  <span style="word-spacing: 25px"> word-spacing: 25px </span>
</p>
<hr />
<span style="letter-spacing: normal">letter-spacing: normal<br />
</span>
<span style="letter-spacing: 10px">letter-spacing: 10px</span>
<hr />
<span style="line-height: normal">line-height: normal</span>
<br />
<span style="line-height: 10px">line-height: 15px</span>
<br />
<span style="line-height: 150%">line-height: 150%</span>
<hr />
<span style="vertical-align: middle">vertical-align: middle</span>
<span style="vertical-align: sub">sub</span>
<span style="vertical-align: super">super</span>
<hr />
<span style="text-align: left">text-align: left<br />
</span>
<span style="text-align: center">text-align: center<br />
</span>
```

```

<span style="text-align: right">text-align: right<br />
</span>
<hr />
<span style="text-decoration: underline">text-decoration: underline</span><br />
<span style="text-decoration: overline">text-decoration: overline</span><br />
<span style="text-decoration: line-through">text-decoration: line-through</
span><br />
<span style="text-decoration: blink">text-decoration: blink</span>
<hr />
<span style="text-indent: 20px">text-indent: 20px</span><br />
<span style="text-indent: 40px">text-indent: 40px</span><br />
<span style="text-indent: 60px">text-indent: 60px</span><br />
</body>
</html>

```

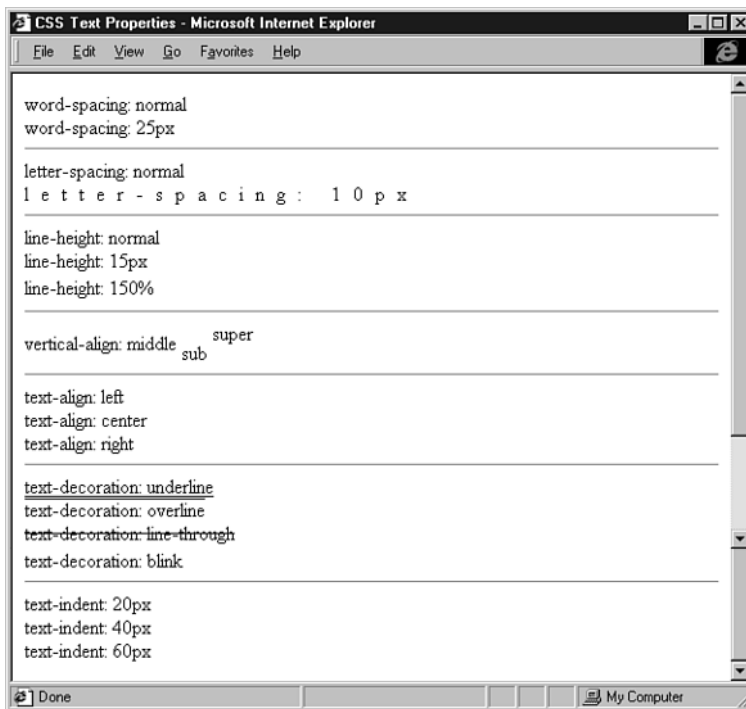


Abbildung 10.8:
 Verschiedene
 Texteeigenschaften
 im Internet
 Explorer 4.0.

Inline Styles

Bevor Sie Text- und Schrift-Eigenschaften in der »Bookworm Bookshop«-Seite anwenden, müssen Sie noch die dritte Anwendungsmethode kennen lernen: Inline-Styles. Diese Methode ermöglicht Ihnen, Style-Regeln an nur ein Element der Websei-

te anzuknüpfen. Falls Sie nur eine Überschrift oder einen Textabschnitt mit einer anderen Farbe versehen oder anders ausrichten möchten, und dies nur auf einer Seite, an einer bestimmten Stelle, so sollten Sie Inline-Styles verwenden.

HTML besitzt mehrere Attribute, die es Ihnen ermöglichen werden, Style-Regeln in HTML-Tags einzubinden. Die wichtigsten Attribute heißen `style` und `class`. Das `<span>`-Tag bezieht sich ebenfalls auf das Style Sheet. Später erhalten Sie diesbezüglich weitere Informationen.

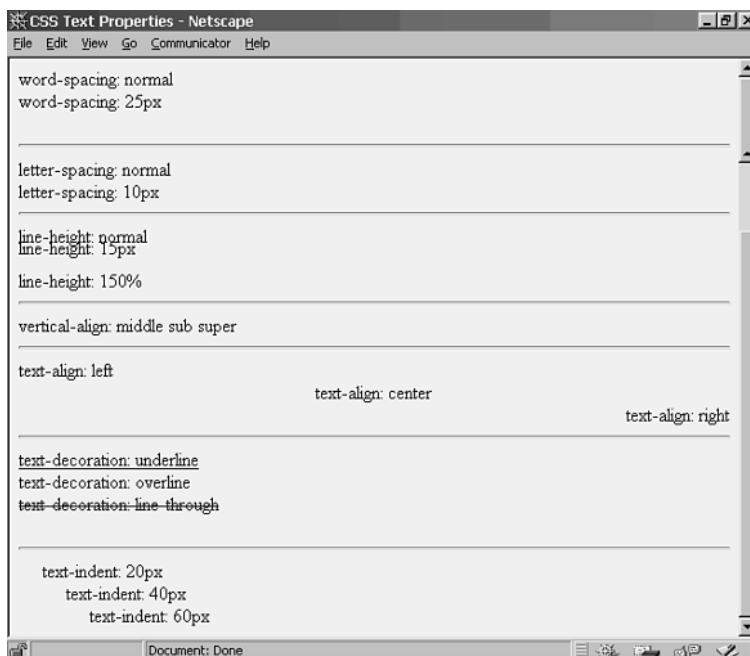


Abbildung 10.9:
Verschiedene
Text-Eigenschaf-
ten im
Netscape 4.

Das Style-Attribut

Im Unterkapitel »Eingebettete Style-Sheets« wurden Sie mit den `<style>`-Tags bekannt gemacht, die ein externes Style Sheet in eine gewöhnliche Webseite einbinden. `style`-Attribute ermöglichen es Ihnen, die Style-Regeln in ein einziges Element der Webseite einzubinden. Während externe und eingebettete Style Sheets ihre Style-Definitionen vom HTML-Inhalt getrennt halten, wird das `style`-Attribut innerhalb des Codes der Web-Datei angewandt.



Sie können das `style`-Attribut jedem HTML-Befehl zuweisen. Ausnahme: `<base>`, `<basefont>`, `<head>`, `<html>`, `<meta>`, `<param>`, `<script>`, `<style>` und `<title>`.

Nehmen wir an, Sie möchten die Farbe eines einzigen Textabschnittes in Ihrer Webseite ändern und benutzen hierfür eine einfache Style-Regel. Normalerweise ist die Textfarbe Ihrer Seite schwarz und in keinsten Weise hervorgehoben. Ein Textabschnitt dieser Seite soll aber in fester, roter Schrift erscheinen.

Die `style`-Attribute definierenden Werte (rot, fett) müssen in Anführungszeichen stehen. Style-Regeln, die diese Werte definieren (`color: red` und `font-weight: bold`) werden durch Strichpunkt innerhalb der Anführungszeichen getrennt.

Folgender Code demonstriert Ihnen, wie die `style`-Attribute bezüglich eines Textabschnittes angewandt werden.

```
<p style="color: red; font-weight: bold">I want this paragraph to be bold and red.</p>
```

Das class-Attribut

`class` ist ein allgemein definierter Style, der Eigenschaften für einige oder alle Elemente in einer Datei festlegt. `class`-Attribute werden in externen Style Sheets oder in der Kopfzeile einer gewöhnlichen Webseite, die eingebettete Styles verwendet, definiert. Der Vorteil dieser Methode ist, dass Sie dieselbe Style-Regel auf verschiedene Elemente einer Webseite einfach anwenden können, ohne die Regel immer wieder eingeben zu müssen.



Sie können das `class`-Attribut jedem HTML-Tag zuweisen. Ausnahme: `<base>`, `<basefont>`, `<head>`, `<html>`, `<meta>`, `<param>`, `<script>`, `<style>`, `<title>`.

Folgendes CSS-Code Beispiel demonstriert zwei, den Textabschnitt betreffende Style-Regeln. Die erste Style-Regel bestimmt die Eigenschaften für die »normalen« Textabschnitte der Seite, während die zweite Style-Regel spezielle Eigenschaften (= `p.bigger`) für einige der Textabschnitte bestimmt. `p` steht für die Benennung des Paragrafen-Tag und `bigger` ist der `class`-Name.

```
<!doctype html public "-//w3c//dtd xhtml 1.0 strict//en"
"http://www.w3.org/tr/xhtml1/dtd/strict.dtd">
<html>
<head>
<title>Need New Glasses</title>
<style type="text/css">
<!--
p { font-family: arial, helvetica, sans-serif; color: black }
p.bigger { font-family: arial, helvetica, sans-serif; font-size: larger }
-->
</style>
</head>
```

Nachdem Sie die Klasse `bigger` in Ihrem Style Sheet definiert haben, müssen Sie die Styles in die Paragraphen einbinden, die Sie gerne größer haben möchten. Um die Klasse `bigger` auf einen bestimmten Paragraphen anwenden zu können, müssen sie folgende Syntax der `class`-Attribute befolgen:

```
<p class="class-name">.
```

Und noch ein Beispiel, das Ihnen den Code *auf* einer einfachen Webseite zeigen wird. Der erste Paragraph erscheint in normaler Schriftgröße, der zweite allerdings benutzt das `bigger`-Style. Bild 10.11 zeigt Ihnen das Ergebnis.



```
<p>Are you having a hard time reading the text on these pages? If you haven't  
had your eyes checked lately, and the following paragraph is still hard to read,  
you might want to have your eyes checked.</p>  
<p class="bigger">You might need new glasses!</p>  
</body>  
</html>
```

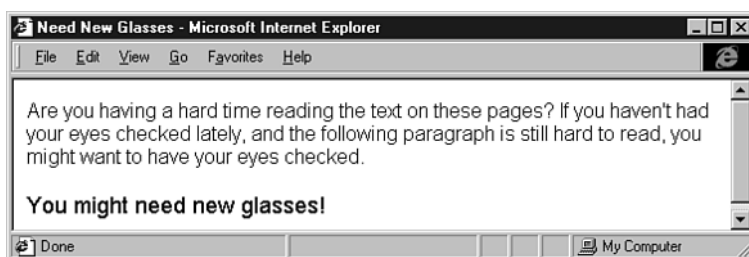


Abbildung 10.10:
Das `class`-Attri-
but auf ein
Seitenelement
angewandt

Das `<span>`-Tag

Die folgende Übung wird Ihnen das umfangreiche Anwendungsgebiet der `<span>`-Eigenschaft vor Augen führen. Diese Eigenschaft bietet die Möglichkeit, einen strukturlosen Textbereich für eine Formatierung zu definieren (deshalb ist diese Eigenschaft auch nicht innerhalb eines bestimmten strukturierenden HTML-Tag enthalten). Sie können diese Eigenschaft sogar auf den ersten Buchstaben oder auf das erste Wort ei-

nes Dokuments anwenden. Der Befehl allein hat keine eigene Bedeutung und bewirkt keinen optisch sichtbaren Effekt. Zum Leben erwacht er erst, wenn Sie im einleitenden `<span>`-Tag das Universalattribut `style=` notieren.

Übung 10.6: Anwendung von Schriftart- und Texteigenschaften mit CSS

Das wird für heute die letzte Übung sein. Sie sollen einige Schriftformatierungs-Eigenschaften den vorhergehenden Beispielen hinzufügen. Die folgenden Beispiele verwenden Fantasy-Schrift (als allgemeinen Schriftartnamen). Typisch ist, dass diese Schrift mit Seiten-Layout oder Grafik-Software installiert wird. Verfügt Ihr Computer nicht über diese Schriftart, so ersetzen Sie `fantasy` mit `fixed`. Der erste Abschnitt Ihrer `bookworm.html`-Seite sieht wie folgt aus:

```
<!doctype html public "-//w3c//dtd xhtml 1.0 strict//en"
"http://www.w3.org/tr/xhtml1/dtd/strict.dtd">
<html>
<head>
<title>The Bookworm Bookshop</title>
<style type="text/css">
<!--
body { background-color: #ccffcc;
      font-family: arial, helvetica, sans-serif;
      color: #330066;
      margin: 50px 70px;
      background-image: url(background.gif);
      background-repeat: repeat }
a:link { color: #cc9900 }
a:visited { color: #660000 }
a:hover { color: #ffcc00 }
a:active { color: #ff0000 }
h1 { color: #996633;
     padding-top: 10px;
     padding-bottom: 5px;
     background-color: #ffffcc;
     border-color: #cccc33 #cc9933;
     border-width: thin;
     border: thin outset;
```

Hier möchten Sie nun die Style-Regel einfügen, die die Schriftart der Überschrift 1. Ordnung (`h1`) in `fantasy` (oder `fixed`) ändert und sie mit der `text-align`-Eigenschaft zentriert.

```
font-family: "fantasy";
text-align: center }
Der code ist noch nicht zu ende:
h2 { color: #996633;
padding-top: 5px;
padding-bottom: 3px;
background-color: #ccffcc;
border-color: #99cc33 #996633;
border-width: thin;
border: thin inset;
```

Wenden Sie nun den Schriftnamen (*fantasy* oder *fixed*) auf die Überschrift 2. Ordnung an und zentrieren Sie diese.

```
font-family: "fantasy";
text-align: center }
```

Der sich gegenwärtig in Ihrer `boodwrm.html`-Datei befindende Code endet mit einer `blockquote`-Style-Regel (`blockquote` = eigener Absatz für Zitate), die folgendermaßen aussehen soll:

```
blockquote { font-family: "book antiqua";
line-height: 12pt;
font-weight: normal;
font-variant: normal;
color: #996633;
word-spacing: 2em;
text-align: center }
```

Nach der `blockquote`-sektion des Codes müssen Sie noch die *fantasy*-Klasse (bzw. *fixed*-Klasse) hinzufügen. Dieses Tag soll einige der inline-Seitenelemente ebenfalls in *fantasy* (oder *fixed*)-Schrift konvertieren. Hierzu müssen Sie folgenden Code eingeben:

```
.fantasy { text-align: left;
font-family: "fantasy";
font-size: 16pt; color: #996600}
```

wie üblich endet diese `style`-Sektion wie folgt:

```
-->
</style>
</head>
```

Aber halt! Sie sind noch nicht fertig! Sie müssen noch die Inline-Styles auf den zu ändernden Abschnitt Ihrer Webseite anwenden. Insgesamt gibt es zwei Sorten von Webseiten, bei denen Sie Inline-Styles anwenden sollten. Der Name »bookstore« erscheint an einigen Stellen der Webseite und Sie möchten, dass er im *fantasy* (oder *fixed*)-Style

angezeigt wird. Die zwei nachstehenden Code-Sektionen demonstrieren, wo diese Änderungen vorgenommen werden sollen.

```
<p><span class="fantasy">The Bookworm Bookshop</span><br />
1345 Applewood Dr<br />
Springfield, CA 94325<br />
(415) 555-0034
</p>
<p>Since 1933, <span class="fantasy">The Bookworm Bookshop</span>
has offered rare and hard-to-find titles for the discerning
reader. The Bookworm offers:</p>
```

Nachstehend finden Sie eine komplette Auflistung des abschließenden Codes der Webseite.

Auf Bild 10.11 sehen Sie das Ergebnis im oberen Teil der Seite im Internet Explorer:

```
<!doctype html public "-//w3c//dtd xhtml 1.0 strict//en"
"http://www.w3.org/tr/xhtml1/dtd/strict.dtd">
<html>
<head>
<title>The Bookworm Bookshop</title>
<style type="text/css">
<!--
body { background-color: #ccffcc;
        font-family: arial, helvetica, sans-serif;
        color: #330066;
        margin: 50px 70px;
        background-image: url(background.gif);
        background-repeat: repeat }
a:link { color: #cc9900 }
a:visited { color: #660000 }
a:hover { color: #ffcc00 }
a:active { color: #ff0000 }
h1 { color: #996633;
      padding-top: 10px;
      padding-bottom: 5px;
      background-color: #ffffcc;
      border-color: #cccc33 #cc9933;
      border-width: thin;
      border: thin outset;
      font-family: "fantasy";
      text-align: center }
h2 { color: #996633;
      padding-top: 5px;
      padding-bottom: 3px;
      background-color: #ccffcc;
```

```

border-color: #99cc33 #996633;
border-width: thin;
border: thin inset;
font-family: "fantasy";
text-align: center }
blockquote { font-family: "book antiqua";
line-height: 12pt;
font-weight: normal;
font-variant: normal;
color: #996633;
word-spacing: 2em;
text-align: center }
.fantasy { text-align: left;
font-family: "fantasy";
font-size: 16pt; color: #996600}
-->
</style>
</head>
<body>
<a name="top"><h1>The Bookworm: A Better Book Store</h1></a>
<blockquote>
"Old books are best--how tale and rhyme<br />
Float with us down the stream of time!"<br />
- Clarence Army, <cite>Old Songs are Best</cite>
</blockquote>
<p><span class="fantasy">The Bookworm Bookshop</span><br />
1345 Applewood Dr<br />
Springfield, CA 94325<br />
(415) 555-0034
</p>
<a name="contents"><h2>Contents</h2></a>
<ul>
<li><a href="#about">About the Bookworm Bookshop</a></li>
<li><a href="#recent">Recent Titles</a></li>
<li><a href="#upcoming">Upcoming Events</a></li>
</ul>
<hr />
<a name="about"><h2>About the Bookworm Bookshop</h2></a>
<p>Since 1933, <span class="fantasy">The Bookworm Bookshop</span>
has offered rare and hard-to-find titles for the discerning
reader. The Bookworm offers:</p>
<ul>
<li>Friendly, knowledgeable, and courteous help</li>
<li>Free coffee and juice for our customers</li>
<li>A well-lit reading room so you can "try before you buy"</li>
<li>Four friendly cats: Esmerelda, Catherine, Dulcinea and Beatrice</li>

```

```

</ul>
<p>Our hours are <strong>10am to 9pm</strong> weekdays,
<strong>noon to 7</strong> on weekends.</p>
<p><a href="#contents">Back to Contents</a> | <a href="#top">Back to Top</a></p>
<hr />
<a name="recent"><h2>Recent Titles (as of 11-Nov-99)</h2></a>
<ul>
<li>Sandra Bellweather, <a href="belladonna.html">
<cite>Belladonna</cite></a></li>
<li>Johnathan Tin, <a href="20minmeals.html">
<cite>20-Minute Meals for One</cite></a></li>
<li>Maxwell Burgess, <a href="legion.html">
<cite>Legion of Thunder</cite></a></li>
<li>Alison Caine, <a href="banquo.html">
<cite>Banquo es Ghost</cite></a></li>
</ul>
<p><a href="#contents">Back to Contents</a> | <a href="#top">Back to Top</a></p>
<hr />
<a name="upcoming"><h2>Upcoming Events</h2></a>
<ul>
<li><b>The Wednesday Evening Book Review</b> meets, appropriately, on Wednesday
evenings at
7:00 pm for coffee and a round-table discussion. Call the Bookworm for information
on joining
the group and this week es reading assignment. </li>
<li><b>The Children es Hour</b> happens every Saturday at 1pm and includes
reading,
games, and other activities. Cookies and milk are served. </li>
<li><b>Carole Fenney</b> will be at the Bookworm on Friday, September 17, to read
from her book of poems <cite>Spiders in the Web.</cite></li>
<li><b>The Bookworm will be closed</b> October 1 to remove a family
of bats that has nested in the tower. We like the company, but not
the mess they leave behind! </li>
</ul>
<p><a href="#contents">Back to Contents</a> | <a href="#top">Back to Top</a></p>
<hr />
<address>
Last Updated: 11-Nov-99<br />
WebMaster: Laura Lemay lemay@bookworm.com<br />
&#169; copyright 1999 the Bookworm<br />
</address>
</body>
</html>

```

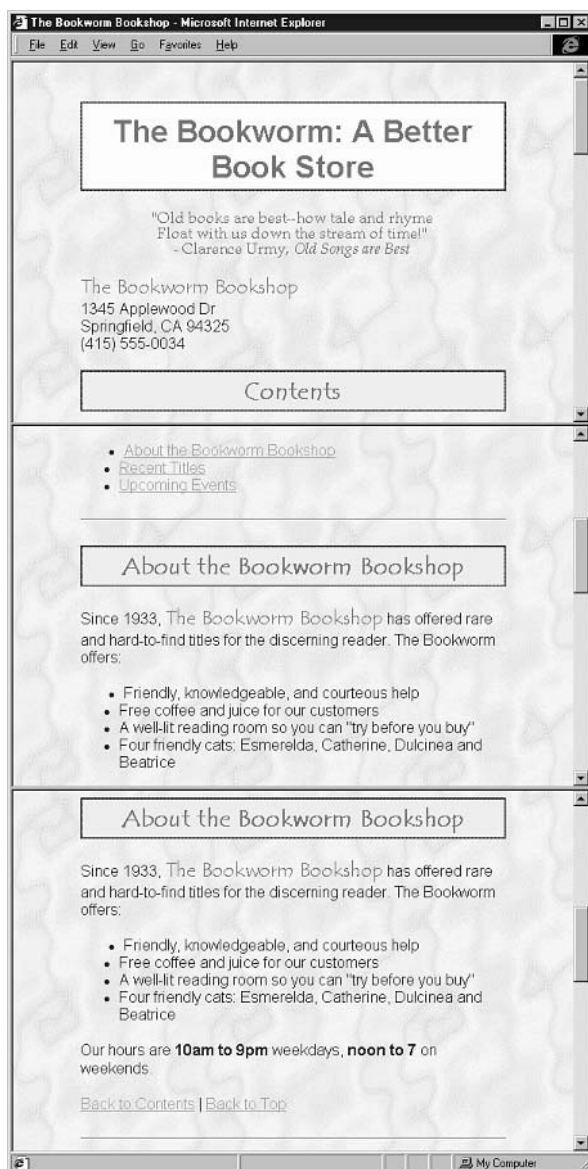


Abbildung 10.11:
Bookworm Bookshop nach Ein-
binden von Schriftart- und Text-
Styles



Falls Sie dieses Beispiel im Netscape Navigator aufgezeigt haben, werden sie wahrscheinlich bemerkt haben, dass das `<span>`-Tag genauso wie das `<br />`-Tag interpretiert worden ist.

Deshalb ist es vielleicht besser, das `<span>`-Tag nicht in die Mitte eines Paragraphen zu platzieren.

Infoquellen über CSS

Das heutige Kapitel hat CSS nur oberflächlich behandeln können. Das liegt hauptsächlich daran, dass diese Thematik sehr umfangreich ist und Browser CSS noch nicht in der Art und Weise verwenden, dass es sich lohnen würde, dieses Thema intensiver zu behandeln.

Weitere Informationen über Cascading Style Sheets finden Sie auf der Webseite des W3C, die detailliertere Angaben über CSS enthält. Die Adresse der CSS-Seite ist: <http://www.w3.org/Style/CSS/>.

Außerdem können Sie auf folgenden Seiten noch weitere nützliche Informationen erhalten:

- ▶ Webreview.com Hat einen CSS-Führer. Ein Teil dieses Führers zeigt auf, wie die Browser CSS-Befehle verschieden ausführen. <http://webreview.com/pub/guides/style/style.html>
- ▶ Cascading Style Sheets Diese Referenz der Web-Design-Group ermöglicht einen guten Einblick in die Style-Sheet-Eigenschaften: <http://www.htmlhelp.com/reference/css/>
- ▶ W3C CSS Home Page: Hier erhalten Sie umfangreiche Informationen, über Links, Beispiele, Werdegang, bis hin zu aufkommenden Trends der Style Sheets oder anderer Web-Technologien. <http://www.w3.org/Style/>

Workshop

Der Workshop steht nun am Ende eines langen und arbeitsreichen Tages. Sollten Sie nun wirklich erpicht sein, Ihr erworbenes Wissen sofort anzuwenden, können Sie gleich damit anfangen. Der Frage-Antwort-Teil wird Ihnen helfen, sich an einige der wichtigsten Dinge, die Sie heute über CSS gelernt haben, zu erinnern.

Frage-Antwort

F Nach welchen Kriterien muss ich Style Sheets auswählen? Ehrlich gesagt, weiß ich nicht so genau, welches ich nun verwenden soll.

A *Keine Sorge, Sie sind nicht der Einzige, der sich bei dieser Wahl schwer tut. Im Grunde genommen kommt es auf den Zweck an, den das Style Sheet erfüllen soll. Wenn Sie nur einige, für Ihre Präsentation nicht ausschlaggebende, visuelle Effekte erzielen wollen, so benutzen Sie die Methode, die Ihnen am geläufigsten ist. Wenn Sie nun aber ein breites Publikum erreichen wollen, so wäre es am besten Cascading Style Sheets zu verwenden, Browser können somit für eine größere Verteilung sorgen.*

F Ich bin mit diesen Hintergrundfarben und Bildeigenschaften noch nicht ganz so vertraut. Warum kann ich nicht einfach Tabellen verwenden und Hintergrundbilder, sowie auch Farben in bestimmte Zellen einfügen?

A *Natürlich können Sie ähnliche Effekte mit Tabellen erzielen. Allerdings tauchen immer wieder Probleme mit Tabellen-Tags auf. Zunächst einmal sind sie keine strukturierten Mark-Up-Befehle, also können sie auch von einigen Browsern, wie zum Beispiel Lynx und akustischen Browsern (für Sehgeschädigte) nicht richtig interpretiert werden. Das bedeutet, dass der Text innerhalb der Tabelle in falscher Reihenfolge aufgezeigt werden kann und in diesen Browsern einfach furchtbar aussieht. Wenn Sie nun Style Sheets für die erwünschten Effekte verwenden, können Sie diese Effekte auf ganz bestimmte Stellen Ihrer Datei anwenden und sichergehen, dass sogar Style-Sheet-unfähige Browser Ihre Datei wenigstens klar und leserlich aufzeigen werden.*

Quiz

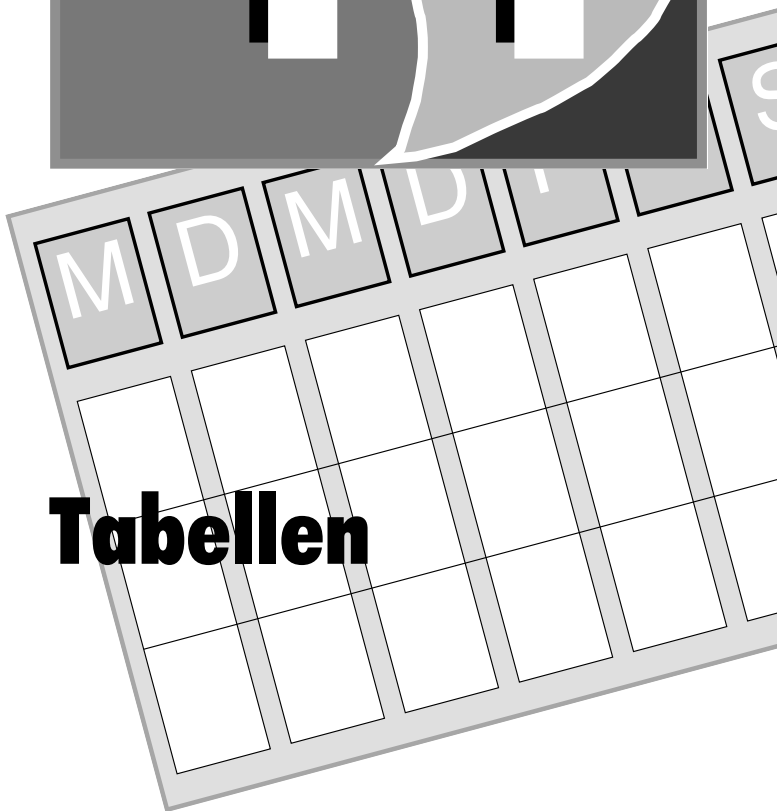
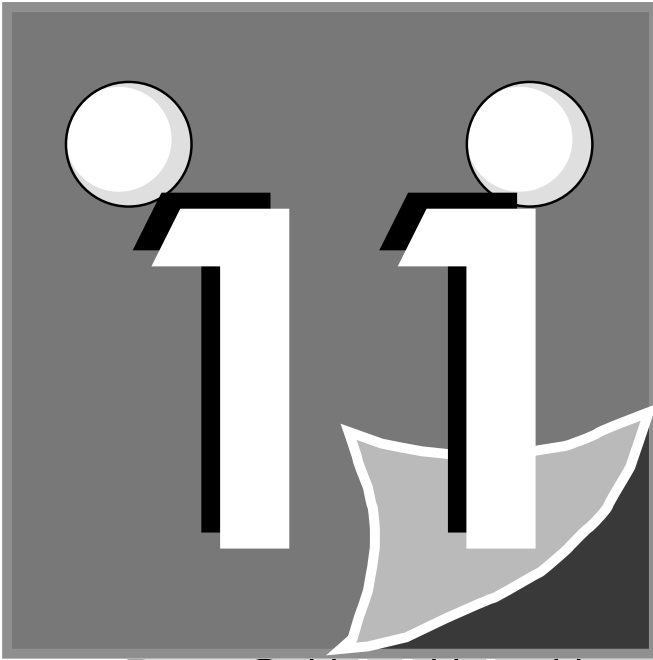
1. Was unterscheidet »strict« und »transitional« HTML-Dateien?
2. Was ist eine CSS-Style-Regel?
3. Welches sind die drei Methoden, mit denen Sie Cascading Style Sheets auf HTML-Elemente anwenden können?
4. Welches ist die beste Methode die gleichen Styles auf mehreren Webseiten anzuwenden? Welche Dateierweiterung muss dabei verwendet werden?
5. Richtig oder Falsch? Externe, eingebettete und Inline-Styles können alle auf einer Webseite angewandt werden.

Antworten:

1. Die strict-HTML-Datei benutzt ein Style Sheet, das die gesamte Formatierung (inklusive Farben und Hervorhebung) einer Datei definiert. Die Transitional-HTML-Datei verwendet ein Style Sheet aber auch integrierte HTML-Attribute, die älteren Browsern ermöglichen soll, die Datei zu formatieren.
2. Eine CSS-Style-Regel definiert den Stil der auf ein HTML-Element anzuwenden ist. Sie besteht aus einem Selector (der ein HTML-Tag sein kann), gefolgt von einer Deklaration, die die Eigenschaft und Wert dieses Selectors definiert.
3. Mittels externer, eingebetteter und Inline-Style-Sheets können Cascading-Style-Sheet-Regeln auf HTML-Elemente angewandt werden.
4. Externe Style Sheets verwendet man am besten, wenn Sie Style-Regeln auf mehreren Seiten anwenden wollen. Das externe Style Sheet wird mit einer .css-Datei-erweiterung gespeichert.
5. Richtig. Die im eingebetteten Style Sheet definierten Eigenschaften ersetzen die der externen Style Sheets. Die Eigenschaften der Inline-Styles ersetzen die externen und eingebetteten Styles.

Übungen

1. Erstellen Sie eine Webseite und wenden Sie einige Style-Regeln, die Sie selbst bestimmen, an. Ihr erstes Beispiel sollte ein externes Style Sheet enthalten, das Sie auf mehr als einer Seite anwenden können.
2. Ändern Sie das oben genannte Beispiel so, dass es ein eingebettetes Style Sheet enthält. Wenn Sie abenteuerlustig genug sind, dann verknüpfen Sie das externe Style Sheet mit Ihrer Webseite. Fügen Sie einige neue Styles hinzu, die die Styles der externen Style Sheets aufheben, und sehen Sie, was passiert!



Tabellen

**Woche
2**

Bis jetzt haben Sie HTML-Befehle verwendet, um Elemente auf Ihren Seiten erstellen und positionieren zu können. Sie können zur Präsentation von Informationen und Inhalt auf Ihren Seiten Absätze und Listen verwenden, aber es gibt noch eine andere Möglichkeit. Wenn Sie nämlich Tabellen verwenden, können Sie Inhalte aller Art auf Ihrer Webseite mit Zeilen und Spalten mit oder ohne Rand layouten. Der in den Tabellen integrierte Inhalt ist nicht auf Text begrenzt. Da Sie wirklich jeden HTML-Inhalt in Tabellen eingeben können (Bilder, Links, Formulare und mehr), bieten Tabellen eine größere Kontrolle über das Erscheinungsbild Ihrer Webseite.

Tabellen wurden offiziell in HTML 3.2 eingeführt. Von da an hatten sie einen enormen Einfluss auf Design und Struktur von Webseiten. HTML 4.0 enthält Veränderungen, die das Laden und Anzeigen von Tabellen im Browser entschieden verbessert haben. Inzwischen ist es möglich, Tabellen zu definieren, die inkrementell angezeigt werden bzw. in nicht grafisch orientierten Browsern (zum Beispiel Textbrowser) einfacher zugänglich sind. Über zusätzliche Elemente lassen sich Tabellen mit Kopf- und Fußzeilen, die dann seitenübergreifend dargestellt werden können, erstellen (wie zum Beispiel bei gedruckten Tabellen).

Heute lernen Sie Folgendes:

- ▶ Wie die Tabellen im Web heute aussehen
- ▶ Wie Sie Tabellen in HTML definieren
- ▶ Wie Sie Titel, Zeilen, Überschriften und Datenzellen erzeugen
- ▶ Wie Sie die Ausrichtung in der Zelle anpassen
- ▶ Wie Sie Zellen erzeugen, die sich über mehrere Zeilen oder Spalten erstrecken
- ▶ Wie Sie den Tabellen Farbe verpassen
- ▶ Wie Sie Tabellen in Ihre Web-Dokumente einsetzen (oder auch nicht)

Eine Bemerkung zur Definition von Tabellen

Als Tabellen von Netscape Anfang 1995 eingeführt wurden, haben Sie in einer rasanten Geschwindigkeit das Webseiten-Design revolutioniert – nicht nur, weil diese für Präsentationsdaten in Tabellenform verwendet wurden, sondern auch, weil sie Web-Designern eine viel bessere Kontrolle über das Seitenlayout und das Platzieren verschiedener HTML-Elemente auf einer Seite gaben.

Jetzt enthalten Tabellenspezifikationen in HTML 4.0 viele Features, die man als Styte Sheets anwendet, was Sie bereits am Tag 10 gelernt haben. Obwohl die Netscape-Spezifikation abgeschlossen ist, definieren Browser-Entwickler wie Microsoft und Netscape immer mehr Tabellenfeatures, die sie in ihre Browser implementieren.

Wenn Sie Ihre Tabellen erstellen, denken Sie daran, dass sie sich immer wieder verändern werden; obwohl man nur ungern daran denkt, dass alles, was man jetzt designt, in der Zukunft von etwas anderem ersetzt sein wird, wird es wahrscheinlich Änderungen für die Zukunft geben. Mit dieser kleinen Warnung im Hinterkopf wollen wir uns nun an die Arbeit machen.

Einfache Tabellen gestalten

Um in HTML Tabellen zu konstruieren, definieren Sie die Teile der Tabelle und legen fest, welche HTML-Elemente wohin gehören. Anschließend setzen Sie dies in HTML-Code um. Danach verfeinern Sie die Erscheinung der Tabelle mit Ausrichtungen, Umrandungen und farbigen Zellen. In diesem Abschnitt werden Sie lernen, wie Sie eine einfache Tabelle mit Überschriften, Daten und Titel erzeugen.

Noch eine Bemerkung: Es macht wirklich keinen Spaß, Tabellen in HTML von Hand zu erstellen. Der Code für Tabellen wurde nicht zur Bearbeitung von Hand konzipiert und ist deshalb ziemlich verwirrend. Sie werden eine ganze Menge experimentieren, testen und zwischen dem Browser und Ihrem Code hin- und herwechseln müssen, um die Tabelle richtig hinzukriegen. HTML-Editoren können dabei gute Dienste leisten, genauso wie der Tabellen-Editor eines Textverarbeitungsprogramms oder einer Tabellenkalkulation Ihnen eine gute Grundidee darüber liefern können, was wohin gehört. Ich schlage aber vor, dass Sie zumindest die ersten Tabellen auf die harte Tour erstellen, sodass Sie eine Idee bekommen, wie die HTML-Tabellen funktionieren.

Tabellenkomponenten

Bevor wir den HTML-Code zum Erzeugen von Tabellen betrachten, hier einige Definitionen, sodass Sie wissen, wovon ich rede:

- ▶ Der *Titel* gibt an, um was es in der Tabelle geht, etwa »Wählerstatistik 1950 bis 1994« oder »Monatliche Ausgaben Ressort Rechenzentrum«. Titel sind optional.
- ▶ *Tabellenüberschriften* bilden die Namen für Spalten oder Zeilen oder beides. Tabellenüberschriften werden in der Regel größer oder hervorgehoben dargestellt, sodass sie sich von der restlichen Tabelle abheben. Tabellenüberschriften sind ebenfalls optional.
- ▶ *Tabellendaten* stellen die eigentlichen Werte in der Tabelle dar. Die Kombination von Tabellenüberschriften und Tabellendaten bildet die gesamte Tabelle.
- ▶ *Tabellenzellen* sind die einzelnen Felder in der Tabelle. Eine Zelle kann normale Tabellendaten oder eine Tabellenüberschrift aufnehmen.

Abbildung 11.1 zeigt eine typische Tabelle und ihre Komponenten.

Name	Height	Weight	Eye Color
Alison	5'4"	140	Blue
Tom	6'0"	165	Hazel
Susan	5'1"	97	Brown

Abbildung 11.1:
Zeigt eine typische Tabelle mit ihren
Komponenten

Das <table>-Tag

Um in HTML eine Tabelle zu erzeugen, verwenden Sie die Tags `<table>...</table>`, die den Code für eine Überschrift und den Inhalt der Tabelle selbst enthalten.

```
<table>
... Tabelleninhalt ...
</table>
```

Um zu demonstrieren, wie ein HTML-Code für eine komplette Tabelle aussieht, sehen Sie hier ein Code-Beispiel, das zur Erstellung der in Abbildung 11.1 gezeigten Tabelle verholffen hat. Seien Sie nicht verunsichert, wenn Sie jetzt noch nicht so genau wissen, um was es sich eigentlich handelt. Merken Sie sich jetzt nur, dass der Code für die Tabelle mit einem `<table>`-Tag und seinen Attributen beginnt, und mit einem `</table>`-Tag endet.

```
<table border="1">
<caption>Vital Statistics</caption>
<tr>
  <th>Name</th>
  <th>Height</th>
  <th>Weight</th>
  <th>Eye Color</th>
</tr>
<tr>
  <td>Alison</td>
  <td>5'4"</td>
  <td>140</td>
  <td>Blue</td>
</tr>
```



```
<tr>
  <td>Tom</td>
  <td>6'0"</td>
  <td>165</td>
  <td>Hazel</td>
</tr>
<tr>
  <td>Susan</td>
  <td>5'1"</td>
  <td>97</td>
  <td>Brown</td>
</tr>
</table>
```

Zeilen und Zellen

Da Sie jetzt schon einige Kenntnisse zum `<table>`-Tag haben, gehen wir zu den Zeilen und den Zellen über. Innerhalb der `<table>...</table>`-Tags definieren Sie den eigentlichen Tabelleninhalt. Tabellen werden in HTML zeilenweise spezifiziert, und jede Zeilendefinition enthält wiederum weitere Definitionen für die einzelnen Zellen in dieser Zeile. Um also eine Tabelle zu definieren, beginnen Sie bei der obersten Zeile, definieren dort die einzelnen Zellen, definieren eine zweite Zeile und ihre Zellen usw. Die Spalten werden automatisch berechnet, basierend darauf, wie viele Zellen in den einzelnen Zeilen enthalten sind.

Jede Tabellenzeile wird durch das `<tr>`-Tag (Table Row = Tabellenreihe) gekennzeichnet und endet mit dem entsprechenden `</tr>`-Tag. Jede Tabellenzeile hat wiederum eine Anzahl von Tabellenzellen, die durch `<th>...</th>` (für Table Header = Tabellenüberschrift) und `<td>...</td>` (für Table Data = Tabellendaten, die eigentlichen Tabellenzellen) angezeigt werden. Sie können beliebig viele Zeilen und innerhalb jeder Zeile beliebig viele Zellen definieren. Achten Sie jedoch darauf, dass jede Zeile dieselbe Anzahl Zellen verwendet, sodass die Spalten übereinstimmen.

Die Zellen innerhalb einer Tabellenzeile werden von einem oder zwei Elementen angegeben:

- ▶ `<th>...</th>`-Elemente werden für Kopfzellen verwendet. Kopfzellen werden in der Regel anders als Tabellenzellen angezeigt, und zwar in fettgedruckter Schrift. Sie sollten mit dem `</th>`-Tag geschlossen werden.
- ▶ `<td>...</td>`-Elemente werden für Datenzellen verwendet. TD steht für Table Data. Das `<td>`-Tag sollte mit dem `</td>`-Tag geschlossen werden.



In den frühen Spezifikationen für Tabellen waren die Abschluss-Kennungen `</tr>`, `</th>` und `</td>` für jede Zeile und Zelle erforderlich. Seither ist die Tabellendefinition dahingehend verfeinert worden, dass jedes dieser Abschluss-Tags optional ist. Gleichzeitig erwarten aber viele Browser doch noch diese Kennungen, sodass eine Tabelle auseinanderfallen kann, wenn Sie diese Abschluss-Kennungen nicht eingefügt haben. Bis die Tabellen konsequent quer durch die Browser-Landschaft verwirklicht werden, ist es sicherer, die optionalen Abschluss-Kennungen weiterhin einzusetzen. Da es immer noch korrekt ist diese zu verwenden, gibt es keinen zwingenden Grund diese wegzulassen.

In den Beispieltabellen, die Sie bis jetzt sehen konnten, erscheint die Kopfzeile in der obersten Zeile und wird mit folgendem Code definiert:

```
<tr>
  <th>Name</th>
  <th>Height</th>
  <th>Weight</th>
  <th>Eye Color</th>
</tr>
  This is followed by three rows of data cells, which are coded as follows:
<tr>
  <td>Alison</td>
  <td>5'4"</td>
  <td>140</td>
  <td>Blue</td>
</tr>
<tr>
  <td>Tom</td>
  <td>6'0"</td>
  <td>165</td>
  <td>Blue</td>
</tr>
<tr>
  <td>Susan</td>
  <td>5'1"</td>
  <td>97</td>
  <td>Brown</td>
</tr>
```

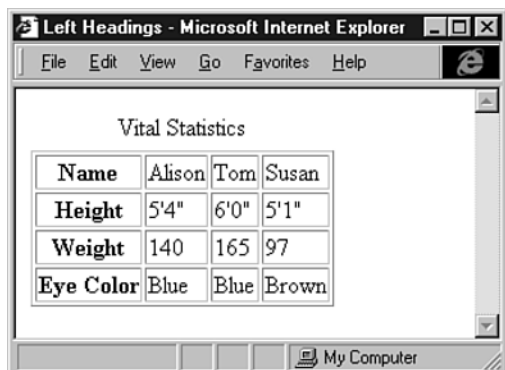
Wie Sie sehen, können Sie die Kopfzeilen am obersten Ende ausrichten, indem Sie das `<th>`-Element innerhalb der ersten Reihe definieren. Ändern wir die Tabelle nun leicht ab. Setzen Sie die Kopfzeilen jetzt an den linken oberen Rand. Um dies zu vervollständigen, setzen Sie jedes `<th>`- in die erste Zelle einer jeden Reihe, und geben Sie anschließend die Daten ein, die zu jeder Kopfzeile gehören. Der neue Code sieht folgendermaßen aus:



```
<tr>
  <th>Name</th>
  <td>Alison</td>
  <td>Tom</td>
  <td>Susan</td>
</tr>
<tr>
  <th>Height</th>
  <td>5'4"</td>
  <td>6'0"</td>
  <td>5'1"</td>
</tr>
<tr>
  <th>Weight</th>
  <td>140</td>
  <td>165</td>
  <td>97</td>
</tr>
<tr>
  <th>Eye Color</th>
  <td>Blue</td>
  <td>Blue</td>
  <td>Brown</td>
</tr>
```



Abbildung 11.2 zeigt das Ergebnis dieser Tabellen.



The screenshot shows a web browser window titled 'Left Headings - Microsoft Internet Explorer'. Inside the browser, there is a table with the title 'Vital Statistics'. The table has four columns: 'Name', 'Height', 'Weight', and 'Eye Color'. There are three rows of data: 'Alison', 'Tom', and 'Susan'. The table is displayed with a border and is centered on the page.

Name	Height	Weight	Eye Color
Alison	5'4"	140	Blue
Tom	6'0"	165	Blue
Susan	5'1"	97	Brown

Abbildung 11.2:
Kleine Tabellen und ihre Überschriften

Leere Zellen

Tabellenkopfzellen und auch Datenzellen können Text, einen HTML-Code, oder beides enthalten. Sie können ebenfalls Links, Listen, Formen und andere Tabellen enthalten. Aber was machen Sie, wenn Sie eine Zelle ohne Inhalt brauchen? Ganz einfach. Definieren Sie eine Zelle mit einem `<th>`- oder einem `<td>`-Tag ohne Inhalt.

```
<table border="1">
<tr>
  <td></td>
  <td>10</td>
  <td>20</td>
</tr>
</table>
```

Manchmal werden solche Zellen auch so dargestellt, als ob sie gar nicht vorhanden wären.

```
<table border="1">
<tr>
  <td><br /></td>
  <td>10</td>
  <td>20</td>
</tr>
</table>
```

Abbildung 11.3 zeigt das Beispiel beider Typen leerer Zellen: die leere Zelle, und die wirklich leere Zelle mit einem eingefügten Zeilenumbruch.

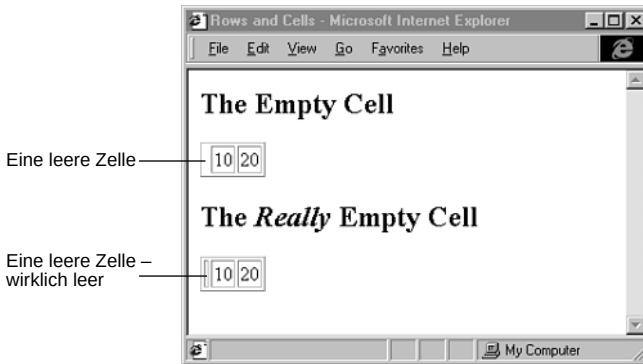


Abbildung 11.3:
Leere und wirklich
leere Zellen

Überschriften

Tabellenüberschriften erklären dem Leser den Inhalt der Tabelle. Sie könnten auch einen normalen Absatz oder eine Überschrift für Ihre Tabelle verwenden, aber es gibt ein `<caption>`-Tag genau für diesen Zweck. Da das `<caption>`-Tag Überschriften als solche kennzeichnet, könnten Werkzeuge zur Verarbeitung von HTML-Dateien sie in eine separate Datei extrahieren, sie automatisch nummerieren oder auf irgendeine andere spezielle Weise behandeln, einfach nur, weil es sich um Überschriften handelt.

Aber was, wenn Sie keine Überschrift wollen? Sie müssen nicht unbedingt eine verwenden. Überschriften sind optional. Wenn Sie einfach nur eine Tabelle ohne Überschrift brauchen, dann lassen Sie die Überschrift einfach weg.

Das `<caption>`-Tag geht direkt von den Tabellenzeilen in das `<table>`-Tag ein und enthält den Tabellentitel. Es wird mit dem `</caption>`-Tag abgeschlossen.

```
<table>
<caption>Vital Statistics</caption>
<tr>
```

Übung 11.1: Erzeugen Sie eine einfache Tabelle

Damit haben Sie die wichtigsten Dinge kennen gelernt, um selbst Tabellen zu erzeugen. Hier wollen wir es an einem einfachen Beispiel versuchen. Wir erzeugen eine Tabelle, die angibt, welche Farbe Sie erhalten, wenn Sie die drei Primärfarben mischen.

Abbildung 11.4 zeigt die Tabelle, die wir in dieser Übung erzeugen wollen.

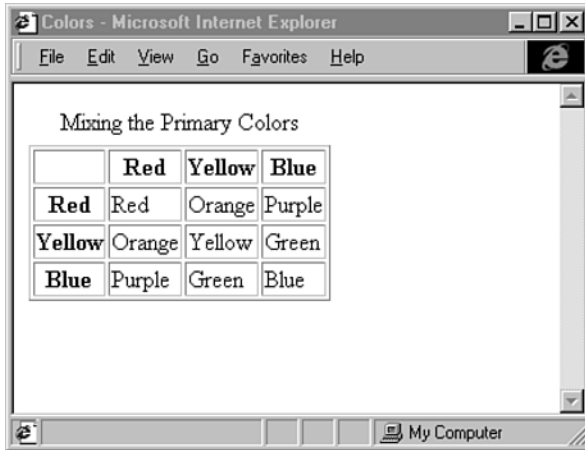


Abbildung 11.4:
Die einfache Farbtabelle

Hier noch ein kurzer Hinweis für Ihre Arbeit mit Tabellen: Da HTML eine Tabelle auf zeilenweiser Basis definiert, kann es manchmal schwierig sein, die Spalten im Auge zu behalten, insbesondere bei sehr komplexen Tabellen. Bevor Sie beginnen, HTML-Code zu schreiben, sollten Sie sich eine Skizze Ihrer Tabelle anlegen, in der Sie festlegen, welche Überschriften verwendet werden und welche Werte in den Zellen enthalten sein sollen. Möglicherweise ist es dabei am einfachsten, ein Textverarbeitungsprogramm mit einem Tabellen-Editor (etwa Microsoft Word) oder eine Tabellenkalkulation zu Hilfe zu nehmen, um ein zufrieden stellendes Layout für Ihre Tabellen zu entwerfen. Wenn das Layout und die Zellenwerte feststehen, schreiben Sie den HTML-Code für diese Tabelle oder konvertieren die erstellte Datei in HTML. Dafür bieten die meisten Standardprogramme inzwischen entsprechende Möglichkeiten an.

Beginnen Sie mit einem einfachen HTML-Framework für die Seite, die eine Tabelle enthält. Wie bei allen HTML-Dateien, können Sie diese Datei in einem Text-Editor Ihrer Wahl erstellen:

```
<html>
<head>
<title>Colors</title>
</head>
<body>
<table border="1">
...hier koennen Sie noch mehr Zeilen und Zellen einfuegen...
</table>
</body>
</html>
```

Nun fügen wir Zeilen innerhalb der `<table>`-Kennungen ein (wo sich oben die Zeile mit »...hier koennen Sie noch mehr Zeilen und Zellen einfuegen...« befindet). Die erste Zeile enthält die drei Überschriften oben in der Tabelle. Die Tabellenzeile wird durch `</tr>`, die einzelnen Zellen werden durch jeweils ein `<th>`-Tag gekennzeichnet.

```
<tr>
  <th>Red</th>
  <th>Yellow</th>
  <th>Blue</th>
</tr>
```



Sie können den HTML-Code beliebig formatieren. Der Browser ignoriert die meisten zusätzlichen Leerzeichen und Zeilenschaltungen. Ich verwende gerne diese Formatierung, wobei der Inhalt der einzelnen Zeilen eingerückt ist und die Zellen-Tags auf separaten Zeilen dargestellt werden, sodass ich Zeilen und Spalten auf Anhieb erkennen kann.

Jetzt fügen wir die zweite Zeile ein. Die erste Zelle in der zweiten Zeile ist die Überschrift Red auf der linken Seite der Tabelle, damit also die erste Zelle in dieser Zeile, gefolgt von den Zellen mit den Tabellendaten:

```
<tr>
  <th>Red</th>
  <td>Red</td>
  <td>Orange</td>
  <td>Purple</td>
</tr>
```

Nun fügen wir die beiden restlichen Zeilen in die Tabelle ein. Sie tragen die Überschriften Yellow und Blue. Damit haben Sie bisher die folgende Tabelle erzeugt:

```
<table border="1">
<tr>
  <th>Red</th>
  <th>Yellow</th>
  <th>Blue</th>
</tr>
<tr>
  <th>Red</th>
  <td>Red</td>
  <td>Orange</td>
  <td>Purple</td>
</tr>
<tr>
  <th>Yellow</th>
  <td>Orange</td>
  <td>Yellow</td>
```

```
<td>Green</td>
</tr>
<tr>
  <th>Blue</th>
  <td>Purple</td>
  <td>Green</td>
  <td>Blue</td>
</tr>
</table>
```

Schließlich fügen wir noch einen einfachen Titel ein. Das `<caption>`-Tag folgt direkt nach dem `<table>`-Tag und kommt unmittelbar vor dem ersten `<tr>`-Tag:

```
<table border="1">
<caption>Mixing the Primary Colors</caption>
<tr>
```

Mit diesem ersten Codekonzept testen Sie Ihre HTML-Datei in Ihrem Browser, der Tabellen unterstützt. Abbildung 11.5 zeigt, wie das im Internet Explorer aussieht.

Hoppla! Was ist mit der obersten Zeile passiert? Die Überschriften sind völlig durcheinander geraten. Die Antwort lautet natürlich, dass Sie am Anfang der ersten Zeile eine leere Zelle benötigen, um die Überschriften korrekt über den entsprechenden Spalten auszurichten. HTML erledigt nicht alles automatisch für Sie (das ist genau der Fehler, dem Sie am Anfang beim Testen Ihrer Tabellen begegnen werden).

Nun fügen wir eine leere Zelle in die erste Zeile ein (in diesem Fall wird das die Zeile `<th><br></th>`):

```
<tr>
  <th><br></th>
  <th>Red</th>
  <th>Yellow</th>
  <th>Blue</th>
</tr>
```



Ich habe hier `<th>` verwendet, aber Sie könnten genausogut `<td>` angeben. Da in der Zelle kein Wert enthalten ist, spielt die Formatierung keine Rolle.

Wenn Sie es jetzt noch einmal versuchen, erhalten Sie ein korrektes Ergebnis, und alle Überschriften sind über den entsprechenden Spalten angeordnet, wie das Originalbeispiel in Abbildung 11.4 zeigt.

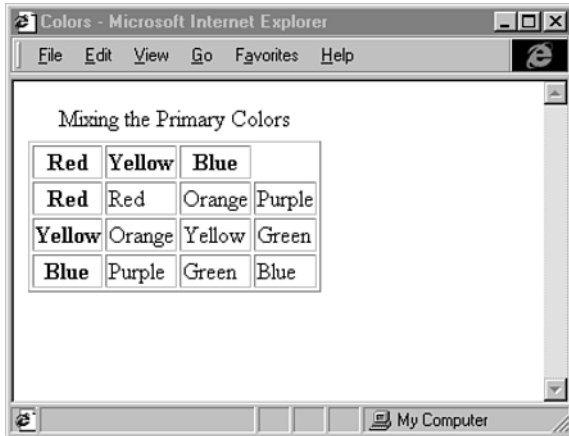


Abbildung 11.5:
Die Farbtabelle

Die Ausrichtung von Tabellen und Zellen

Mit Ihren nun vorhandenen Grundkenntnissen sehen Sie sich jetzt Attribute an, mit denen Sie das Erscheinungsbild Ihrer Tabelle ändern können. Die Attribute, die Sie in dieser Lektion lernen, kontrollieren die Anzeigenbreite Ihrer Tabellen und Zellen, den Abstand zwischen Zelleninhalt und Längs- und Querspalten und die Breite der Rahmen.

Festlegen der Tabellenbreite

Die Tabellen- und Spaltenbreite der Tabelle aus dem vorhergehenden Beispiel wurde direkt durch den Browser festgelegt. In vielen Fällen ist das die beste Möglichkeit, um sicherzugehen, dass Ihre Tabellen in verschiedenen Browsern mit verschiedenen Bildschirmgrößen und -Breiten richtig angezeigt werden können; überlassen Sie einfach dem Browser die Entscheidung.

In anderen Fällen jedoch möchten Sie vielleicht mehr Kontrolle über die Tabellen- und Spaltenbreite haben, vor allem, wenn die Standardeinstellungen des Browsers wirklich seltsam sind. Im folgenden Teil werden Sie einige Möglichkeiten dafür kennen lernen. Das `width`-Attribut des `<table>`-Elements definiert die Breite der Tabelle auf der Seite. `width` kann einen Wert haben, der exakt die Breite der Tabelle anzeigt (in Pixeln). Der Wert kann auch in Prozent der entsprechenden Breite angegeben werden (wie zum Beispiel 50 oder 75 %), was sich allerdings ändern kann, wenn die Größe des Fensters geändert wird. Wenn `width` spezifiziert ist, kann die Spaltenbreite innerhalb der Tabelle komprimiert oder expandiert werden, um mit der erforderlichen Breite übereinzustimmen.

Damit eine Tabelle einer 100%-Bildschirmanzeigenbreite angepasst wird, fügen Sie das `width`-Attribut der Tabelle hinzu, wie es im folgenden Code gezeigt wird. Das Ergebnis sehen Sie in Abbildung 11.6

```
<table border="1" width="100%">
```

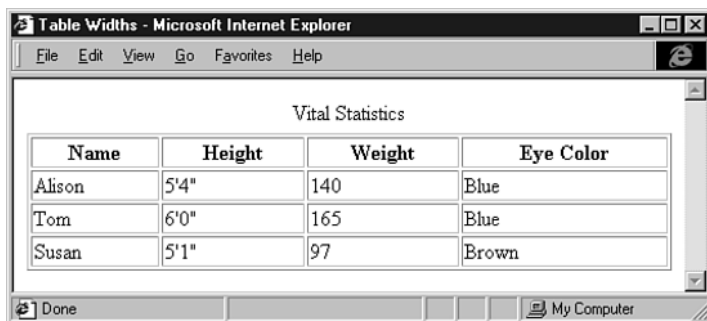


Abbildung 11.6:
Tabellenbreiten im
Internet Explorer«



Zu schmale Spalten für die Daten können unmöglich funktionieren, da der Browser in diesem Fall die von Ihnen gewünschte Breite so gut wie möglich annähert.

Sie sollten Ihre Tabellenbreiten immer in Prozent anstatt in konkreten Pixel-Werten angeben. Da Sie nicht wissen, wie groß das Browserfenster sein wird, erlauben Prozentwerte eine dem Browserfenster angepasste Größe. Die Verwendung konkreter Pixel-Werte könnte Ihre Tabelle in unangepasster Größe erscheinen lassen.

Ändern der Tabellenrahmen

Das `border`-Attribut des `<table>`-Tag ist das gängigste Attribut des `<table>`-Elements. Mit diesem Attribut bestimmen Sie, ob ein Rahmen um die Tabelle erscheinen soll oder nicht, und wenn ja, wie breit der Rahmen sein muss.

Das `border`-Attribut unterlag, seit seiner Aufnahme in HTML, einigen Veränderungen:

- ▶ in HTML 2.0 haben Sie `<table border>` verwendet, um einen Rahmen um die Tabelle zu erstellen. Der Rahmen konnte in einem Grafik-Browser besonders schick und in einem Text-Browser als eine Folge kleiner Längsstriche erscheinen.
- ▶ In HTML 3.2 und neueren Versionen ist die korrekte Verwendung vom `border`-Attribut ein wenig anders: es gibt die Rahmenbreite in Pixeln an. `<table border="1">` erstellt einen 1 Pixel breiten Rand, `<table border="2">` erstellt einen 2 Pixel breiten Rand usw. HTML 3.2 und neuere Browser zeigen höchstwahrscheinlich die alte HTML 2.0- Form von `<table border>` an, und zwar ohne Wert, mit einem 1-Pixel-Rand (wie Sie in `<table border="1">` spezifiziert haben).

- Um einen Rahmen zu erstellen, der keine Breite hat und nicht angezeigt wird, geben Sie `<table border="0">` ein. Rahmenlose Tabellen sind vor allem bei Tabellenstrukturen für Layout-Zwecke geeignet, wenn Sie nicht unbedingt die Tabellenbegrenzung auf Ihrer Webseite anzeigen wollen.

Sie können die Rahmenbreite um eine Tabelle herum auch ändern. Wenn `border` einen Nummernwert hat, wird die Rahmenbreite mit der Pixel breite bestimmt. Der Standardcode ist `border="1"`. `border="0"` unterdrückt die Rahmenanzeige (als ob Sie das `border`-Attribut ganz ausgelassen hätten).

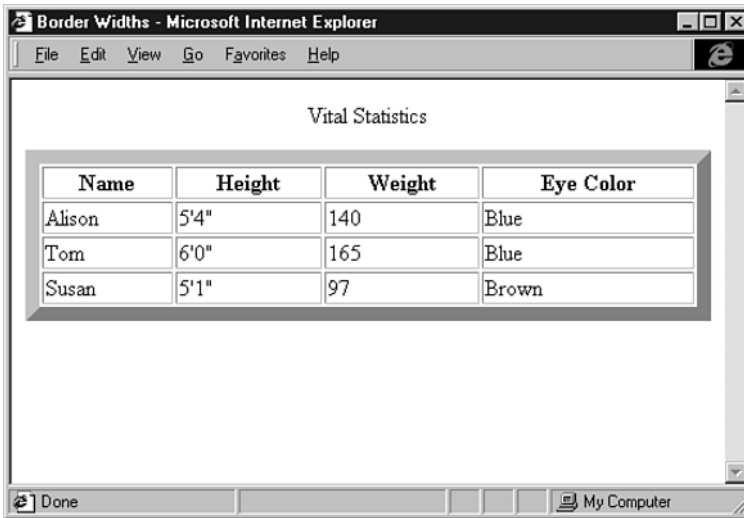


Abbildung 11.7:
Rahmenbreite

Abbildung 11.7 zeigt eine Tabelle mit einer 10 Pixel Rahmenbreite und Rahmendefinitionen.

```
<table border="10" width="100%">
```

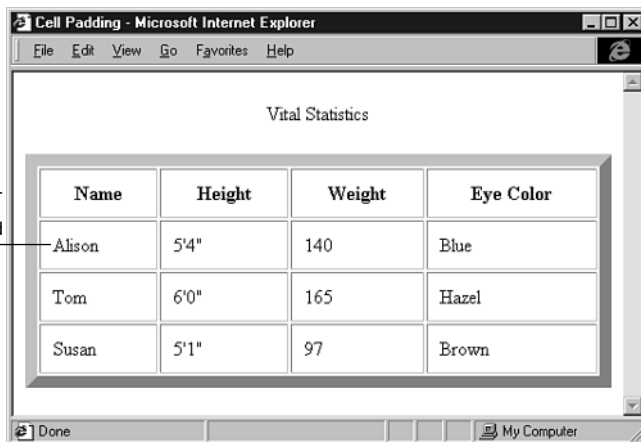
Zelleninnenabstände

Die Zelleninnenabstände (Cell Padding) beschreiben den Raum zwischen den Kanten der Zellen und dem Zellinhalt. Standardmäßig zeichnet Netscape seine Tabellen mit einem Innenabstand von 1 Pixel. Mit dem `cellpadding`-Attribut für das `<table>`-Tag können Sie diesen Wert überschreiben. Abbildung 13.31 zeigt ein Beispiel für eine Tabelle mit einem Innenabstand von 10 Pixel. Das `cellpadding`-Attribut mit einem Wert von 0 bewirkt, dass die Kanten der Zellen den Zellinhalt berühren (das sieht nicht besonders gut aus).

Hier ist der überarbeitete Code für Ihr `<table>`-Tag, welches das Cell-Padding auf 10 Pixel erhöht. Das Ergebnis sehen Sie in Abbildung 11.8:

```
<table border="10" width="100%" cellpadding="10">
```

Über das cellpadding-Attribut wird der Abstand zwischen dem Zellinhalt und den Zellrändern vergrößert.



Name	Height	Weight	Eye Color
Alison	5'4"	140	Blue
Tom	6'0"	165	Hazel
Susan	5'1"	97	Brown

Abbildung 11.8:
Zelleninnenab-
stand

cellpadding-Attribute mit dem Wert 0 bewirken, dass die Abgrenzungen der Zellen den Zelleninhalt berühren (was nicht gerade besonders gut aussieht).

Zellenabstände

Zellenabstände ähneln den Zelleninnenabständen, jedoch mit dem Unterschied, dass die Zellenabstände den Raum zwischen den Zellen beschreiben – das heißt, die Breite der schattierten Linien, die die Zellen voneinander abtrennen. Der Wert dafür wird im cellspacing-Attribut für das `<table>`-Tag spezifiziert. Der Standardabstand beträgt 2.

Der Zellenabstand bezieht sich auch auf den Teil der Umrandung der Tabelle, der sich an der Innenseite des Tabellenrahmens befindet (Der wiederum vom `border`-Attribut bestimmt wird). Experimentieren Sie damit, und Sie werden den Unterschied bemerken. So zeigt Abbildung 11.9 ein Beispiel einer Tabelle mit einem Zellabstand von 8 und einem Rahmen von 4.

```
<table border="4" width="100%" cellpadding="10" cellspacing="8">
```

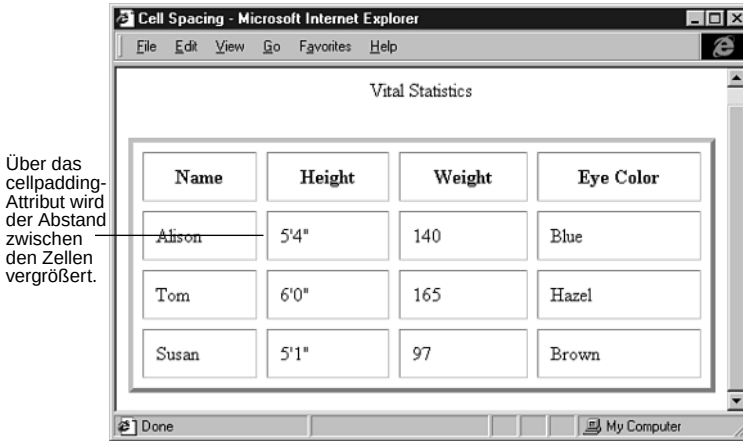


Abbildung 11.9:
Abstand der Zellen voneinander und die Rahmen

Spaltenbreiten

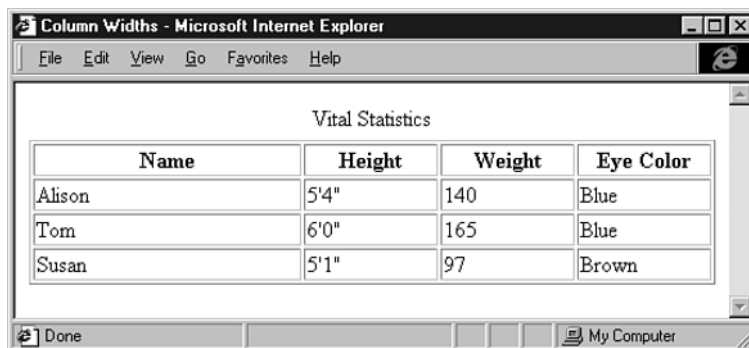
Sie können das `width`-Attribut auch für einzelne Zellen benutzen (`<th>` oder `<td>`), um die Breite der einzelnen Spalten anzugeben. Sie können, wie wir schon vorhin feststellen konnten, genauso wie bei Tabellenbreiten, die Spaltenbreite entweder in Pixeln oder in Prozent festlegen (die Prozente sind auf die ganze Tabellenbreite bezogen). Wie bei Tabellenbreiten, sollte Sie die Prozentangabe den Pixeln vorziehen, damit Ihre Tabelle unabhängig von der Fenstergröße angezeigt wird.

Spaltenbreiten sind vor allem dann nützlich, wenn Sie mehrere Spalten mit gleicher Breite – unabhängig von ihrem Inhalt – erstellen müssen (zum Beispiel für Seitenlayouts).

In Abbildung 11.10 sehen Sie die Originaltabelle aus Abbildung 11.1. Diesmal nimmt die Tabelle jedoch 100% der Bildschirmanzeigenbreite ein. Die erste Spalte beträgt 40% der Tabellenbreite, die übrigen drei Spalten nehmen jeweils 20% der Tabellenbreite ein.

Um das Ganze zu vervollständigen, werden die Spaltenbreiten wie folgt mit den Kopfzellen in Verbindung gebracht:

```
<table border="1" width="100%">
<caption>Vital Statistics</caption>
<tr>
  <th width="40%">Name</th>
  <th width="20%">Height</th>
  <th width="20%">Weight</th>
  <th width="20%">Eye Color</th>
</tr>
```



Column Widths - Microsoft Internet Explorer

File Edit View Go Favorites Help

Vital Statistics

Name	Height	Weight	Eye Color
Alison	5'4"	140	Blue
Tom	6'0"	165	Blue
Susan	5'1"	97	Brown

Done My Computer

Abbildung 11.10:
Spaltenbreiten

Was geschieht, wenn die Tabelle 80% der Bildschirmanzeigenbreite einnimmt, aber dieselben Kopfzellen wie im vorhergehenden Beispiel haben (40 Prozent, 20 Prozent, 20 Prozent, 20 Prozent)? Ändern Sie den Code leicht ab, indem Sie die Tabellenbreite auf 80%, wie im folgenden Beispiel, reduzieren. Wenn Sie die neue Tabelle nun in Ihrem Browser öffnen, werden Sie feststellen, dass die Tabelle 80% der Anzeigenbreite einnimmt. Die vier Spalten betragen jetzt 40%, 20%, 20% und 20% der Tabellenbreite, wohlgemerkt. Wenn man es ganz genau ausdrückt, messen die Spalten 32%, 16%, 16%, 16% der Gesamtanzeigenbreite.

```
<table border="1" width="80%">
<caption>Vital Statistics</caption>
<tr>
  <th width="40%">Name</th>
  <th width="20%">Height</th>
  <th width="20%">Weight</th>
  <th width="20%">Eye Color</th>
</tr>
```

Zeilenumbrüche in den Text einfügen

Zeilenumbrüche sind besonders nützlich, wenn es sich um eine Tabelle handelt, bei der die meisten Zellen klein sind und nur eine oder zwei Zellen mehr Daten enthalten. So lange die Bildschirmbreite ausreicht, bringt der Browser normalerweise einfach lange Zeilen hervor, was in einigen Tabellen ziemlich seltsam aussehen kann (vgl. Abbildung 11.11).

```
<tr>
  <td>TC</td>
  <td>7</td>
  <td>Suspicious except when hungry, then friendly</td>
</tr>
```

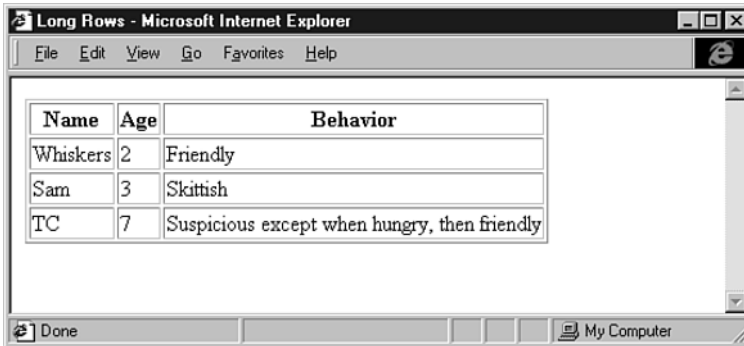


Abbildung 11.11:
Eine Tabelle mit
einer langen Zeile

Durch das Einfügen von Zeilenumbrüchen können Sie diese Zeile in eine schmalere Spalte zwängen, sodass das ganze mehr wie die Tabelle in Abbildung 11.12 aussieht. Sehen Sie Änderung in der letzten Zeile des abgeänderten Codes:

```
<tr>
  <td>TC</td>
  <td>7</td>
  <td>Suspicious except<br />
    when hungry, <br />
    then friendly</td>
</tr>
```

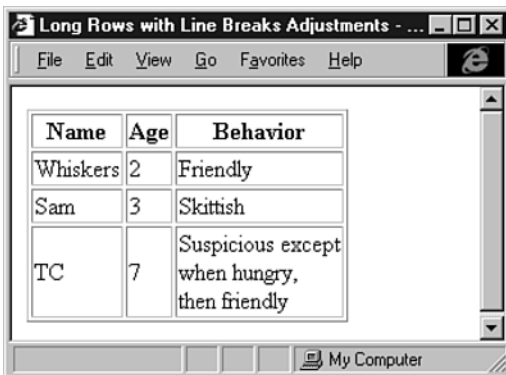


Abbildung 11.12:
Die lange Zeile wurde mit

 umbrochen.

Andererseits wollen Sie vielleicht eine Tabelle erzeugen, bei der eine Zelle gegen Ihren Willen umbrochen wird, in der Sie alle Daten auf einer einzigen Zeile darstellen wollen. (Dies kann bei Formularelementen innerhalb von Tabellenzellen besonders wichtig werden, wenn Sie das Eingabefeld neben seiner Bezeichnung anordnen wollen.) In diesem Fall können Sie dem <th>- oder <td>-Tag das NOWRAP-Attribut hinzufügen, so-

dass der Browser alle Daten dieser Zelle auf einer Zeile lässt. Sie können dann immer noch `<br>`-Tags von Hand zu dieser Zelle hinzufügen und einen Zeilenumbruch an der Stelle bewirken, an der Sie ihn brauchen.



Das `nowrap`-Attribut gilt in HTML 4.0 als verworfen und wird durch Style Sheets ersetzt.

Seien Sie vorsichtig, wenn Sie Zeilenumbrüche in Tabellenzellen mit dem `nowrap`-Attribut festkodieren. Denken Sie daran, Ihre Tabelle wird in den unterschiedlichsten Bildschirmanzeigengrößen angesehen. Versuchen Sie die Anzeigengröße des Fensters zu ändern, in dem Ihre Tabelle erscheinen soll, und testen Sie, ob Ihre Tabelle auch bei unterschiedlichen Fensterbreiten funktioniert. In den meisten Fällen sollten Sie Ihrem Browser die Tabellenformatierung überlassen und nur, wenn es wirklich notwendig ist, geringfügig eingreifen.

Ausrichtung von farbigen Tabellen und Zellen

Nachdem Sie Ihre Grundtabellen mit Querspalten, Kopfzeilen und Daten gestaltet haben, können Sie ihr Aussehen verbessern. Sie können zum Beispiel Farben in Rahmen und Zellen einfügen, und Ihre Tabelle sieht schon viel schicker aus.

Ändern der Hintergrundfarben von Tabellen und Zellen

Um die Hintergrundfarbe einer Tabelle, Zeile oder Zelle innerhalb einer Zeile zu verändern, wenden Sie das Attribut `bgcolor` am jeweiligen Tag an, also an `<table>`, `<tr>`, `<th>` oder `<td>`. So wie bei `<body>` wird der Wert von `bgcolor` durch eine Farbe bestimmt, die entweder als drei hexadezimal Zahlen oder in vielen Browsern, darunter der Internet Explorer und der Netscape Navigator/Communicator, mit einem der Farbnamen bezeichnet wird: Black (Schwarz), Green (Grün), Maroon (Kastanienbraun), Olive (Olivgrün), Navy (Marineblau), Purple (Violett), Gray (Grau), Red (Rot), Yellow (Gelb), Blue (Blau), Lime (Zitronengelb), Aqua (Blaugrün), Fuchsia (Fuchsia) oder Silver (Silber). Das `bgcolor`-Attribut ist inzwischen Teil der Vorschläge für die HTML-4.0-Spezifikation, wurde aber zugunsten der Style Sheets verworfen.

Jede Hintergrundfarbe verdrängt die Hintergrundfarbe des umliegenden Elements. Beispielsweise verdrängt ein Tabellenhintergrund den Seitenhintergrund, ein Zeilenhintergrund verdrängt den Tabellenhintergrund, und eine Zellfarbe verdrängt alle anderen Farben. Wenn Sie eine Tabelle innerhalb einer Zelle verschachteln, nimmt diese verschachtelte Tabelle die Hintergrundfarbe der umgebenden Zelle an.

Wenn Sie die Farbe einer Zelle ändern, sollten Sie ebenfalls die Farbe des sich darin befindlichen Textes ändern (mit `<font color ...>`), sodass er lesbar bleibt.

Wenn Sie möchten, dass Ihre Seiten mit Browsern, die älter als der Internet Explorer 4.0 der und Netscape Navigator 4.0 sind, kompatibel sind, verwenden Sie `<font color...>`. Verwenden Sie in Browsern, die Cascading Style Sheets unterstützen, zum Beispiel Internet Explorer 4.0 (oder neuere Versionen) und Netscape 4.0 (oder neuere Versionen) die CSS *color-Eigenschaft*.



Damit die Hintergrundfarbe einer Tabellenzelle dargestellt wird, muss die Zelle irgendeinen Inhalt enthalten. Es reicht aus, einfach ein `<br>`-Tag in die leere Zelle zu stecken.

Hier folgt ein Beispiel, bei dem die Hintergrund- und Zellfarben innerhalb einer Tabelle verändert wurden. Und zwar habe ich hier mit einer HTML-Tabelle ein kariertes Muster für ein Damebrett erzeugt. Die Tabelle selbst ist weiß und enthält abwechselnd schwarze Zellen. Die Damefiguren sind Abbildungen (hier: rote und schwarze Kreise).



Wenn Sie Bilder in Ihre Tabellen einfügen möchten, dann können Sie die weißen Spalten, die bei der Tabellenerstellung im Original – HTML- Code erscheinen, einfach ignorieren, da diese nicht in der fertigen Tabelle auftreten. Bei Netscape jedoch gibt es eine Ausnahme zu dieser Regel, und zwar dann, wenn Sie Bilder in Tabellenzellen einfügen. Nehmen wir an, Sie hätten Ihren Code mit dem `<img>`-Tag in einer separaten Zeile, wie im folgenden Beispiel, erstellt:

```
<td>
  
</td>
```

Bei diesem Stückchen Quelltext ist der Absatzumbruch zwischen dem `<td>`-Tag und dem `<img>`-Tag problematisch; Ihr Bild wird innerhalb der Zelle nicht korrekt platziert (dies äußert sich besonders in Zellen mit zentrierter Absatzausrichtung). Diese Unschönheit des Netscape Browsers besteht selbst in der neuesten Version des Navigators noch. Um dieses Problem zu korrigieren, platzieren Sie das `<td>` und das `<img>`-Tag in derselben Zeile, wie das im folgenden Code-Ausschnitt der Fall ist:

```
<td></td>
```

Ich habe die Regel aus der vorhergehenden Bemerkung im folgenden Beispiel befolgt. Das Ergebnis im Internet Explorer sieht man in Abbildung 11.13

```
<head>
<title>Checkerboard</title>
</head>
```

```

<body>
<table bgcolor="#ffffff" width="50%">
<tr align="center">
<td bgcolor="#000000" width="33%"></td>
<td width="33%"></td>
<td bgcolor="#000000" width="33%"></td>
</tr>
<tr align="center">
<td></td>
<td bgcolor="#000000"><br />
</td>
<td></td>
</tr>
<tr align="center">
<td bgcolor="#000000"><br />
</td>
<td><br />
</td>
<td bgcolor="#000000"><br />
</td>
</tr>
</table>
</body>
</html>

```

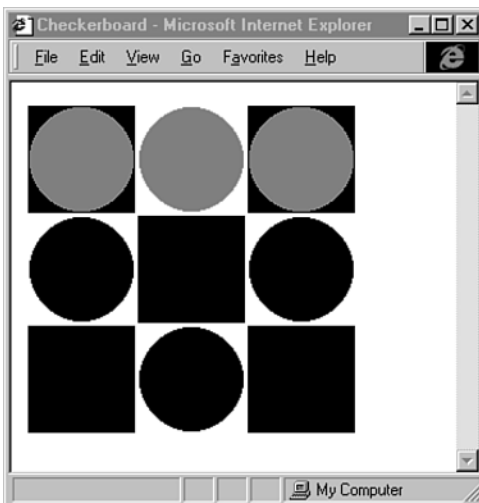


Abbildung 11.13:
 Farbige Tabellenzellen

Ändern von Rahmenfarben

Der Internet Explorer ermöglicht eine Farbänderung der Elemente im Tabellenrahmen bei Verwendung der `bordercolor`-, `bordercolorlight`- und `bordercolordark`-Attribute. Jedes dieser Attribute erhält entweder eine Farbnummer oder eine Farbbezeichnung und kann in `<table>`, `<td>` oder `<th>` verwendet werden. Wie bei Hintergrundfarben, überdecken die Rahmenfarben jeweils die Farben des umschließenden Elements. Alle drei Attribute benötigen das abschließende `<table>`-Tag zur Erstellung des `border`-Attributs.

Diese Erweiterungen werden nur (momentan) vom Internet Explorer unterstützt, außer `bordercolor`, was auch in Netscape Navigator 4.0 unterstützt wird. Alle drei Attribute gelten als verworfen und wurden von Style Sheets ersetzt.

- ▶ `bordercolor` legt die Rahmenfarbe fest, die den 3D-Look des Standardrahmens überdeckt.
- ▶ `bordercolordark` legt die dunklen Komponenten des 3D-Looks des Rahmens fest und platziert diese außen rechts im Tabellenrahmen.
- ▶ `bordercolorlight` legt die hellen Komponenten des 3D-Looks des Rahmens fest und platziert diese innen links im Tabellenrahmen.

In Abbildung 11.14 sehen Sie ein Beispiel einer Tabelle mit einem 10 Pixel Rahmen. Um die Internet Explorer-Attribute zu demonstrieren, wurden `bordercolorlight` und `bordercolordark` hinzugefügt, um dem dickeren Tabellenrahmen einen 3D-Look zu geben. Die erste Zeile des Codes wurde wie im folgenden Beispiel geändert:

```
<table border="10" bordercolorlight="red" bordercolordark="black"
bgcolor="#ffffff" width="50%">
```

Diese Code-Zeile ist etwas lang, finden Sie nicht? Das Lesen des Codes wird Ihnen leichter fallen, wenn Sie jedes Attribut in eine separate Zeile setzen, wie Sie im folgenden Beispiel sehen können. Dieser Code funktioniert genauso wie der andere. Denken Sie nur daran, dass die abschließende Klammer (`>`) nur einmal, nach dem letzten Attribut, erscheinen darf.

```
<table border="10"
bordercolorlight="red"
bordercolordark="black"
bgcolor="#ffffff"
width="50%">
```

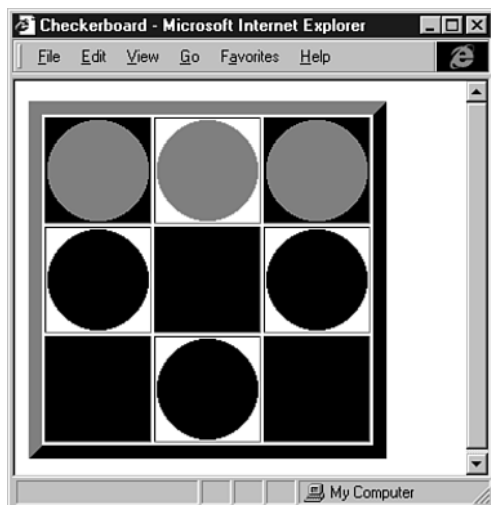


Abbildung 11.14:
Rahmenfarben

Ausrichtung des Tabelleninhalts

Sie können Ihre Tabellen noch verbessern, indem Sie die Ausrichtung des Tabellen- und Zelleninhalts genauer präzisieren. Das `align`-Attribut richtet den Inhalt horizontal, das `valign`-Attribut richtet den Inhalt vertikal aus. Beide Attribute wurden in HTML 3.2 eingeführt, aber in HTML 4.0 verworfen; in den folgenden Abschnitten wird erklärt, wie diese Attribute für die Tabellenerstellung verwendet werden.

Tabellenausrichtung

Standardmäßig werden Tabellen zum linken Seitenrand hin ausgerichtet, wobei der übrige Text über oder unter der Tabelle dargestellt wird.

Mit dem `align`-Attribut können Sie Tabellen zum linken bzw. rechten Seitenrand hin ausrichten und Text um diese herumfließen lassen, wie Sie das schon bei Abbildungen getan haben. `align=left` richtet die Tabelle zum linken Seitenrand hin aus, und der der Tabelle folgende Text wird in den Freiraum zwischen der Tabelle und dem rechten Seitenrand gepackt. `align=right` tut das gleiche, wobei die Tabelle zum rechten Seitenrand hin ausgerichtet wird.

Im Beispiel, das in Abbildung 11.15 gezeigt wird, wird eine Tabelle, die 70% der Seite einnimmt, mit folgendem Code nach links ausgerichtet:

```
<table border="1" align="left" width="70%">
```

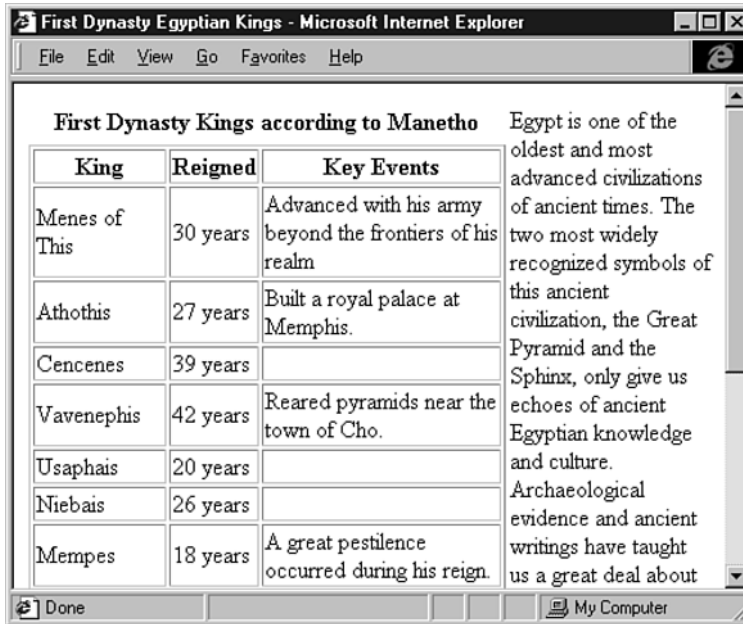


Abbildung 11.15:
Eine Tabelle mit
herumfließen-
dem Text.

Wenn Sie das Herumfließen des Textes unterbrechen möchten, können Sie das so wie bei Abbildungen tun, indem Sie die Zeilenumbruch-Kennung (
) mit dem clear-Attribut einsetzen.

Das Zentrieren von Tabellen ist etwas schwieriger. Erst ab dem Internet Explorer 4.0 und dem Netscape Navigator 4 wird align=center bei Tabellen unterstützt. Sie können jedoch die <center>- oder <div align=center>-Kennungen verwenden, um Tabellen auf der Seite zu zentrieren (Sie haben diese beiden Tags schon in Kapitel 6, »Noch mehr Text-formatieren mit HTML«, kennen gelernt). Mit den jüngsten Versionen der beiden Browser wird <table align=center> korrekt unterstützt. Wie andere Formatierungs-Attribute wurde das align-Attribut in HTML 4.0 verworfen.

Zellenausrichtung

Wenn Sie die Zeilen und Zellen an Ort und Stelle und die Tabelle auch richtig auf der Seite ausgerichtet haben, richten Sie die Daten innerhalb der Zellen so aus, wie es dem Tabelleninhalt am besten entspricht. HTML-Tabellen bieten mehrere Optionen für die Ausrichtung von Zellen, sowohl horizontal als auch vertikal. Abbildung 11.16 zeigt eine Tabelle (eine echte HTML-Tabelle!) mit den verschiedenen Möglichkeiten der Ausrichtung.

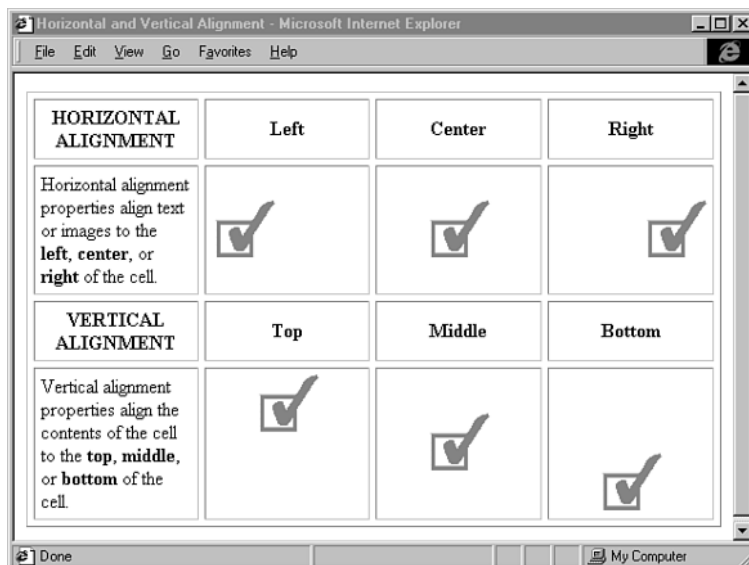


Abbildung 11.16:
Zellenausrichtung

Die horizontale Ausrichtung (das Attribut `align`) definiert, ob die Daten innerhalb einer Zelle linksbündig (`left`), rechtsbündig (`right`) oder zentriert (`center`) dargestellt werden.

Die vertikale Ausrichtung (das Attribut `valign`) definiert die vertikale Datenausrichtung innerhalb der Zellen, das heißt ob die Daten oben in der Zelle (`top`), unten (`bottom`) oder vertikal zentriert (`middle`) angeordnet sind. Der Netscape Navigator implementiert auch `valign=baseline`, welches so ähnlich wie `valign=top` ist, außer dass es die Grundlinie der ersten Textzeile ausrichtet (abhängig vom Zelleninhalt kann das möglicherweise ein anderes Ergebnis als `align=top` bringen).

Standardmäßig werden die Überschriftszellen sowohl horizontal als auch vertikal ausgerichtet, und die Datenzellen werden vertikal zentriert aber linksbündig angezeigt.

Sie können die Standardwerte für jeweils eine ganze Zeile durch die Angabe der Attribute `align` oder `valign` im Tag `<tr>` überschreiben, wie etwa im folgenden Beispiel:

```
<tr align="center" valign="top">
```

Die Zeilenausrichtung einzelner Zellen überschreiben Sie durch die Angabe von `align` in den Tags `<td>` oder `<th>`:

```
<tr align="center" valign="top">
  <td>14</td>
  <td>16</td>
  <td align="left">No Data</td>
  <td>15</td>
</tr>
```

Das folgende Beispiel zeigt die verschiedenen Zellausrichtungen und ihre Darstellung im Internet Explorer (Abbildung 11.17).



```
<html>
<head>
<title>Cell Alignments</title>
</head>
<body>
<table border="1">
<tr>
<th width="25%"><br /></th>
<th width="25%">Left</th>
<th width="25%">Centered</th>
<th width="25%">Right</th>
</tr>
<tr>
<th>Top</th>
<td align="left" valign="top"></td>
<td align="center" valign="top"></td>
<td align="right" valign="top"></td>
</tr>
<tr>
<th>Centered</th>
<td align="left" valign="middle"></td>
<td align="center" valign="middle"></td>
<td align="right" valign="middle"></td>
</tr>
<tr>
<th>Bottom</th>
<td align="left" valign="bottom"></td>
<td align="center" valign="bottom"></td>
<td align="right" valign="bottom"></td>
</tr>
</table>
```

```
</table>
</body>
</html>
```

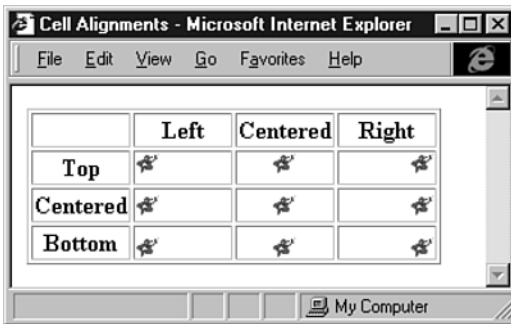


Abbildung 11.17:
Optionen für die Ausrichtung

Überschriftenausrichtung

Das `align`-Attribut zur Überschrift- es handelt sich um eine Option – legt die Ausrichtung der Überschrift fest. Je nach Browser kann `align` verschiedene Bedeutungen haben.

In den meisten Browsern nimmt `align` einen von zwei Werten an: `top` (oben) oder `bottom` (unten). Dies stellt die korrekte Verwendung des `align`-Attributs dar. Standardmäßig wird die Überschrift oben angezeigt (`align=top`). Sie können das Attribut `align=bottom` auf die Überschrift anwenden, wenn Sie sie unten darstellen wollen, etwa folgendermaßen:

```
<table>
<caption align=bottom>Drehmomentbegrenzungen verschiedener Obstsorten</caption>
```

Beim Internet Explorer sind die Tabellenüberschriften jedoch anders. Dort benutzen Sie das Attribut `valign`, um die Überschrift nach oben oder unten auszurichten, und `ALIGN` hat drei verschiedene Werte: `left`, `right` und `center`, mit denen die Überschrift horizontal ausgerichtet wird.

Um ähnliche Ergebnisse im Netscape Navigator zu erzielen, verwenden Sie `align=bottom` oder `align=top` und anschließend das `<div>`-Tag mit dessen `align`-Attribut, um den Text der Überschrift linksbündig, rechtsbündig oder zentriert auszurichten. Dies funktioniert auch im Internet Explorer 4.0.

Wenn Sie zum Beispiel eine Überschrift am unteren Rand einer Tabelle im Internet Explorer platzieren wollen, die rechtsbündig ausgerichtet ist, können Sie folgenden Code verwenden:

```
<caption valign="bottom" align="right">This is a caption</caption>
oder Sie verwenden das <div>-Tag, das auch im Netscape Navigator funktioniert:
<caption align="bottom"><div align="right">This is a caption</div></caption>
```

Wenn Sie nicht gerade eine sehr kurze Tabelle erzeugen, dann sollten Sie Ihre Überschrift in der Standardposition lassen – zentriert, am oberen Tabellenrand – sodass Ihre Leser sie zuerst sehen und wissen, worum es geht. Warten Sie also nicht, bis die Leser die gesamte Tabelle schon durchgesehen haben (obwohl sie dann meist ohnehin schon wissen, um was es geht).

Zellen, die sich über mehrere Zeilen oder Spalten erstrecken

Die Tabellen, die wir bisher erzeugt haben, zeigten alle einen Wert pro Zelle an und enthielten zum Teil auch leere Zellen. Sie können jedoch auch Zellen erzeugen, die sich über mehrere Zeilen oder Spalten in der Tabelle erstrecken. Diese Zellen können Überschriften enthalten, die Unterüberschriften in der nächsten Zeile oder Spalte anzeigen. Sie können auf diese Weise aber zum Beispiel auch spezielle Effekte innerhalb des Tabellenlayouts erzeugen. Abbildung 11.18 zeigt eine Tabelle mit Beispielen für Zellen, die sich über mehrere Spalten und Zeilen erstrecken.

		Handedness	
		Right	Left
Gender	Male	96	8
	Female	80	14

Abbildung 11.18:
Tabellen mit Zellen, die sich über mehrere Zeilen oder Spalten erstrecken

Um eine Zelle zu erzeugen, die sich über mehrere Zeilen oder Spalten erstreckt, fügen Sie den Tags `<th>` oder `<td>` die Attribute `rowspan` oder `colspan` hinzu, zusammen mit der entsprechenden Anzahl der Zeilen oder Spalten. Die Daten innerhalb dieser Zellen werden dann über die gesamte Breite oder Länge der kombinierten Zellen angezeigt, wie im folgenden Beispiel gezeigt:

```
<html>
<head>
<title>Row and Column Spans</title>
</head>
<body>
<table border="1">
<tr>
<th colspan="2">Gender</th>
</tr>
<tr>
<th>Male</th>
<th>Female</th>
</tr>
<tr>
<td>15</td>
<td>23</td>
</tr>
</table>
</body>
</html>
```

Abbildung 11.19 zeigt, wie diese Tabelle aussehen könnte.

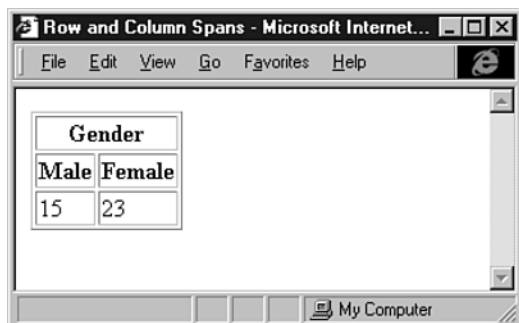


Abbildung 11.19:
Eine Zelle erstreckt sich über mehrere
Spalten

Beachten Sie, dass Sie die Zelle in den nächsten Zeilen nicht erneut als leer definieren müssen, wenn sie sich über mehrere Zeilen erstreckt. Ignorieren Sie das einfach, und gehen Sie weiter zur nächsten Zelle in der Zeile. Die Anordnung wird automatisch angepasst.

Zellen erstrecken sich immer nach unten und nach rechts. Um also eine Zelle über mehrere Spalten zu erzeugen, fügen Sie der Zelle ganz links das Attribut `colspan` hinzu, und für Zeilen, die sich über Zeilen erstrecken, das Attribut `rowspan` zur obersten Zelle.

Das folgende Ein- und Ausgabe-Beispiel zeigt eine Zelle, die sich über mehrere Zeilen erstreckt (die Zelle mit dem Wort »Piston«). Abbildung 11.20 zeigt das Ergebnis im Internet Explorer.



```
<html>
<head>
<title>Ring Clearance</title>
</head>
<body>
<table border="1">
<tr>
<th colspan="2"> </th>
<th>Ring<br />
Clearance</th>
</tr>
<tr align="center">
<th rowspan="2">Piston</th>
<th>Upper</th>
<td>3mm</td>
</tr>
<tr align="center">
<th>Lower</th>
<td>3.2mm</td>
</tr>
</table>
</body>
</html>
```

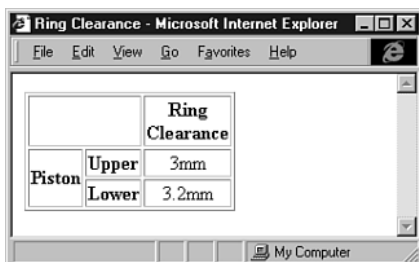


Abbildung 11.20:
Zellen, die sich über mehrere Zeilen und Spalten erstrecken

Übung 11.2. Eine Tabelle mit Kundendienst-Spezifikationen

Haben Sie schon genug von Tabellen? Wir wollen noch ein letztes Beispiel betrachten, das alle Features anwendet, die Sie bisher kennen gelernt haben: Tabellen mit Überschriften und normalen Zellen, Ausrichtungen und Zellen, die sich über mehrere Zeilen oder Spalten erstrecken. Dabei handelt es sich um eine sehr komplexe Tabelle, deshalb werden wir schrittweise erklären, wie sie erzeugt wird.

Abbildung 11.21 zeigt eine Tabelle, die Kundendienst- und Reparatur-Spezifikationen aus dem Service-Handbuch für ein Auto darstellt.

Service Data - Microsoft Internet Explorer

File Edit View Go Favorites Help

Drive Belt Deflection

		Used Belt Deflection		Set deflection of new belt
		Limit	Adjust Deflection	
Alternator	Models without AC	10mm	5-7mm	5-7mm
	Models with AC	12mm	6-8mm	
Power Steering Oil Pump		12.5mm	7.9mm	6-8mm

Done My Computer

Abbildung 11.21:
Die wirklich komplexe Kundendienst-Tabelle

Service Data - Microsoft Internet Explorer

File Edit View Go Favorites Help

Drive Belt Deflection

		Used Belt Deflection		Set deflection of new belt
		Limit	Adjust Deflection	
Alternator	Models without AC	10mm	5-7mm	5-7mm
	Models with AC	12mm	6-8mm	
Power Steering Oil Pump		12.5mm	7.9mm	6-8mm

Done My Computer

Spalte 1 Spalte 2 Spalte 3 Spalte 4 Spalte 5

Zeile 1
Zeile 2
Zeile 3
Zeile 4
Zeile 5

Abbildung 11.22:
Fünf Spalten,
fünf Zeilen

Diese Tabelle enthält fünf Zeilen und Spalten. Können Sie sie erkennen? Einige Zellen erstrecken sich über mehrere Zeilen und Spalten. Abbildung 11.22 zeigt dieselbe Tabelle, dargestellt in einem Raster, sodass Sie die Zeilen und Spalten besser erkennen.

Bei solchen Tabellen ist es sinnvoll, ein Raster anzuzeigen, um sich besser vorstellen zu können, welche Zellen sich über mehrere Spalten und Zeilen erstrecken und in welche Zeile sie ursprünglich gehören. Sie wissen ja, die Zellen erstrecken sich von oben nach unten und von links nach rechts.

Fertig? Wir wollen mit dem Gerüst für die Tabelle beginnen, wie Sie es aus diesem Kapitel bereits kennen:

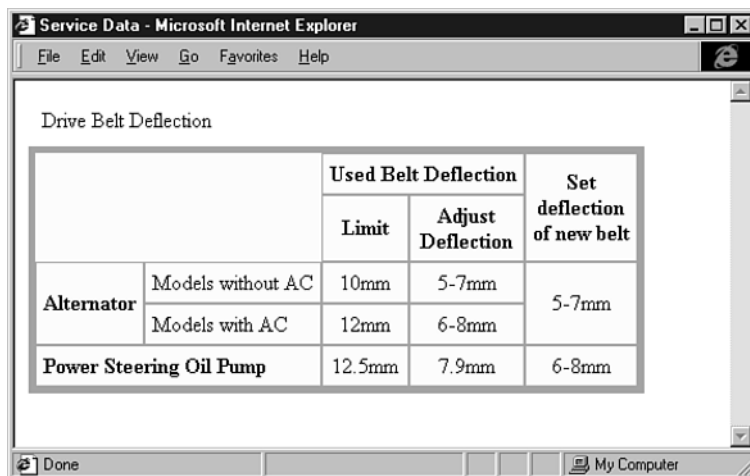
```
<html>
<head>
<title>Service Data</title>
</head>
<body>
<table border="1">
<caption>Drive Belt Deflection</caption>
</table>
</body>
</html>
```

Um das Aussehen Ihrer Tabelle zu verbessern, versehen Sie alle Zellen mit einem hellgelben (#ffffcc) Hintergrund. Dazu verwenden Sie das bgcolor-Attribut verwenden. Die Rahmenbreite wird mit 5 Pixeln vergrößert und erhält eine tiefgoldene Farbe (#cc9900), indem Sie das bordercolor-Attribut verwenden, das mit Netscape und Internet Explorer kompatibel ist. Die Trennlinien zwischen den Zellen erscheinen deutlicher, wenn Sie cellspacing auf 0 stellen und die weiße Fläche zwischen Zellinhalt und – rahmen vergrößern, indem Sie cellpadding auf 5 stellen. Der neue Tabellencode sieht folgendermaßen aus:

```
<table border="5"
  bgcolor="#ffffcc"
  bordercolor="#cc9900"
  cellspacing="0"
  cellpadding="5">
```

Anschließend erstellen Sie die erste Zeile. Anhand des Rasters in Ihrem Bild der Tabelle können Sie erkennen, dass die erste Zelle leer ist und sich über zwei Zeilen und zwei Spalten (siehe auch Abbildung 11.23) erstreckt. Hierfür verwenden Sie den folgenden HTML-Code:

```
<tr>
<th rowspan="2" colspan="2"></th>
```



		Used Belt Deflection		Set deflection of new belt
Alternator	Models without AC	Limit	Adjust Deflection	
	Models with AC	10mm	5-7mm	5-7mm
Power Steering Oil Pump		12mm	6-8mm	5-7mm
		12.5mm	7.9mm	6-8mm

Abbildung 11.23:
Die erste Zeile

Die zweite Zeile in dieser Zeile ist die Überschrift `Used Belt Deflection`, die sich über zwei Spalten erstreckt (für die beiden Zellen, die darunter angezeigt werden). Der Code für diese Zeile sieht also wie folgt aus:

```
<th colspan="2">Used Belt Deflection</th>
```

Damit haben Sie zwei Zellen, die sich über je zwei Spalten erstrecken, und in dieser Zeile ist nur noch eine Spalte übrig. Diese jedoch erstreckt sich in die darunterliegende Zeile:

```
<th rowspan="2">Set deflection of new belt</th>
</tr>
```

Jetzt gehen wir weiter zur zweiten Zeile. Das ist aber nicht diejenige, die mit der Überschrift `Alternator` beginnt. Beachten Sie, dass die erste Zeile in der ersten Zeile die Attribute `rowspan` und `colspan` mit einem Wert von 2 verwendet. Das bedeutet, dass sie in dieser zweiten Zeile fortgesetzt wird und zwei Zellen aufnimmt. Für diese Zeile muss also keine Neudefinition erfolgen. Gehen Sie einfach weiter zur nächsten Zeile im Raster. Die erste Zeile in dieser Zeile trägt die Überschrift `Limit`, die zweite Zeile `Adjust Deflection`.

```
<tr>
  <th>Limit</th>
  <th>Adjust Deflection</th>
</tr>
```

Aber was ist mit der letzten Zeile? So wie die erste Zeile hatte die Zeile in der darüberliegenden Zeile den `rowspan`-Wert zwei und nimmt jetzt den Platz in dieser Zeile ein. Die einzigen Werte, die Sie für diese Zeile benötigen, sind also bereits definiert.

Haben Sie mir folgen können? Sie sollten jetzt einfach alles mit Ihrem Browser ausprobieren, um sicherzustellen, dass alles einwandfrei funktioniert. Es wird ein bisschen witzig aussehen, weil wir in die linke Seite der Tabelle noch nichts eingegeben haben. Sie sollten es jedoch versuchen. Abbildung 11.22 zeigt, was wir bisher geschafft haben.

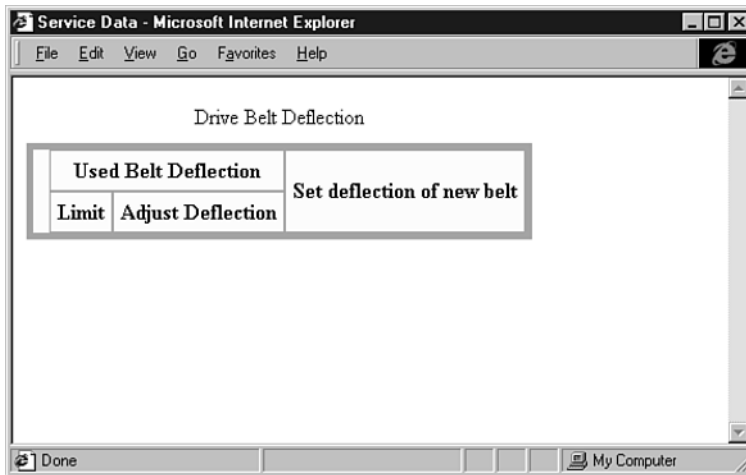


Abbildung 11.24:
Die Tabelle, so
wie sie bisher aus-
sieht

Nächste Zeile! Sehen Sie gegebenenfalls noch einmal im Raster nach. Hier trägt die erste Zelle die Überschrift **Alternator**, und sie erstreckt sich auf diese und die darunterliegende Zeile. Wird es Ihnen jetzt zuviel?

```
<tr>
  <th rowspan="2">Alternator</th>
```

Die nächsten drei Zeilen sind relativ einfach, weil sie nicht in irgendwelchen Zeilen oder Spalten fortgesetzt werden. Hier ihre Definitionen:

```
<td>Models without AC</td>
<td>10mm</td>
<td>5-7mm</td>
```

Die letzte Zeile sieht wie die erste aus:

```
<td rowspan="2">5-7mm</td>
</tr>
```

Damit sind wir in Zeile vier. Aufgrund der `rowspan` aus der vorherigen Zeile müssen hier nur drei Zellen definiert werden: die Zelle für Modelle mit AC (Klimaanlage) und zwei Zellen für die Nummern:

```
<tr>
  <td>Models with AC</td>
  <td>12mm</td>
  <td>6-8mm</td>
</tr>
```



In dieser Tabelle habe ich die Alternator-Zelle zur Überschrift gemacht, die AC-Zellen enthalten einfach Daten. Das war nur eine ästhetische Entscheidung meinerseits. Ich hätte auch einfach drei Überschriften verwenden können.

Jetzt zur letzten Zeile – das sollte Ihnen nicht schwerfallen. Die erste Zelle (Power Steering Oil Pump) erstreckt sich über zwei Spalten (Alternator und With/With-out AC). Die übrigen drei bestehen aus nur jeweils einer Zelle.

```
<tr>
  <th colspan="2">Power Steering Oil Pump</th>
  <td>12.5mm</td>
  <td>7.9mm</td>
  <td>6-8mm</td>
</tr>
```

Das ist alles! Damit haben Sie alle Zeilen und Spalten entworfen. Das war das schwierigste. Beim Rest handelt es sich nur noch um die Feinarbeiten. Jetzt probieren wir das Ganze aus, um sicherzustellen, dass sich nicht irgendwelche Fehler eingeschlichen haben (siehe Abbildung 11.25).

Drive Belt Deflection				
		Used Belt Deflection		Set deflection of new belt
		Limit	Adjust Deflection	
Alternator	Models without AC	10mm	5-7mm	5-7mm
	Models with AC	12mm	6-8mm	
Power Steering Oil Pump		12.5mm	7.9mm	6-8mm

Abbildung 11.25:
Die Tabelle:
der nächste
Schritt

Damit haben Sie alle Zeilen und Spalten entworfen und können nun innerhalb der Zellen die Ausrichtungen anpassen. Zumindest die Zahlen sollten zentriert dargestellt werden. Da sie einen Großteil der Tabelle einnehmen, wollen wir die Zentrierung für alle Zeilen zum Standard machen:

```
<tr align="center">
```

Die Beschriftungen links (Alternator, Models With/Without AC und Power Steering Oil Pump) sehen etwas seltsam aus, wenn sie zentriert dargestellt werden. Jetzt sollen sie linksbündig ausgerichtet werden:

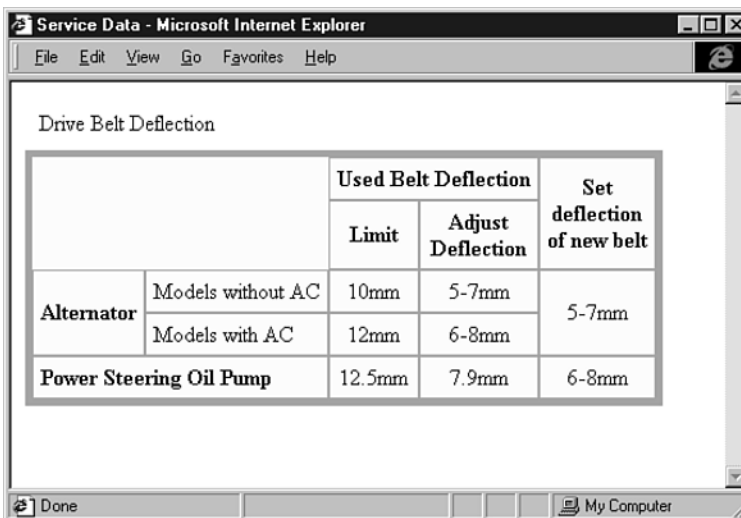
```
<th rowspan="2" align="left">Alternator</th>
<td align="left">Models without AC</td>
<td align="left">Models with AC</td>
<th colspan="2" align="left">Power Steering Oil Pump</th>
```

Zum Abschluss habe ich in die längeren Überschriften Zeilenumbrüche eingefügt, so dass die Spalten etwas näher beieinander angezeigt werden. Da der Text in den Überschriften relativ kurz ist, musste ich mir nicht viele Gedanken darüber machen, wie die Spalten aussehen, wenn der Text zu nahe aneinanderrückt. Hier die veränderten Zeilen:

```
<th rowspan="2">Set<br />deflection<br />of new belt</th>
<th>Adjust<br />Deflection</th>
```

Richten Sie zum Schluss die Überschrift auf der linken Tabellenseite aus:

```
<caption align="left">Drive Belt Deflection</caption>
```



		Used Belt Deflection		Set deflection of new belt
		Limit	Adjust Deflection	
Alternator	Models without AC	10mm	5-7mm	5-7mm
	Models with AC	12mm	6-8mm	
Power Steering Oil Pump		12.5mm	7.9mm	6-8mm

Abbildung 11.26:
Die endgültige
Kundendienst-
tabelle

Voilà – hier die endgültige Tabelle, und alles steht an seinem richtigen Platz! Abbildung 11.26 zeigt das Endergebnis.



Wenn Sie ein bisschen Zeit haben, sollten Sie Ihren Text-Editor aufrufen und es selbst einmal ausprobieren, Schritt für Schritt. Danach wird alles einfacher.

Hier der vollständige Text für das Tabellenbeispiel:

```
<html>
<head>
<title>Service Data</title>
</head>
<body>
<table border="5"
  bgcolor="#ffffcc"
  bordercolor="#cc9900"
  cellspacing="0"
  cellpadding="5">
<caption align="left">Drive Belt Deflection</caption>
<tr>
  <th rowspan="2" colspan="2"></th>
  <th colspan="2">Used Belt Deflection</th>
  <th rowspan="2">Set<br />deflection<br />of new belt</th>
</tr>
<tr>
  <th>Limit</th>
  <th>Adjust<br />Deflection</th>
</tr>
<tr align="center">
  <th rowspan="2" align="left">Alternator</th>
  <td align="left">Models without AC</td>
  <td>10mm</td>
  <td>5-7mm</td>
  <td rowspan="2">5-7mm</td>
</tr>
<tr align="center">
  <td align="left">Models with AC</td>
  <td>12mm</td>
  <td>6-8mm</td>
</tr>
<tr align="center">
  <th colspan="2" align="left">Power Steering Oil Pump</th>
  <td>12.5mm</td>
  <td>7.9mm</td>
  <td>6-8mm</td>
</tr>
```

```
</tr>
</table>
</body>
</html>
```

Tabellenverbesserungen für Fortgeschrittene

Ob Sie es glauben oder nicht, nach all der getanen Arbeit werden Sie nun endlich die Tabellenelemente kennen lernen, die in HTML 4.0 eingeführt wurden. Es gibt große Verbesserungen, was das Definieren von Zeilen und Spalten in Tabellen betrifft, die ich in den folgenden Abschnitten besprechen werde.

Gruppieren und Ausrichten von Spalten

Einer der größten Fortschritte in HTML 4.0 ist die Möglichkeit, Tabellen inkrementell laden zu können. Das heißt Teile der Tabelle können angezeigt werden, auch wenn noch nicht die gesamten Tabellendaten geladen wurden. Man kann nun auch Tabellenspalten mit den `<colgroup>`- und `<col>`-Elementen definieren. Diese Elemente ermöglichen dem Autor strukturelle Unterteilungen der Tabellenspalten, deren Aussehen mit Style-Sheet-Eigenschaften verbessert werden kann.

Das `<colgroup>...<colgroup>`-Element wird zum Gruppieren einer oder mehrerer Spalten verwendet. Das abschließende `</colgroup>`-Tag ist nicht zwingend notwendig. Dieses Element hat zwei Attribute:

- ▶ `span` definiert die Spaltenanzahl der Spaltengruppe. Der Wert muss ein Integer, der größer als 0 ist, sein. Wenn `span` nicht definiert ist, wird die Standardeinstellung des `<colgroup>`-Elements aktiv und erstellt eine Spaltengruppe, die eine Spalte enthält. Wenn das `<colgroup>`-Element ein oder mehrere `<col>`-Elemente enthält (die etwas später erklärt werden), wird das `span`-Attribut jedoch ignoriert.
- ▶ `width` definiert die Spaltenbreite einer jeden Spalte aus der Spaltengruppe. Die Breite kann in Pixeln, Prozentsätzen und Relativwerten definiert werden. Sie können auch einen speziellen Breitenwert mit `"0*"` (nach der 0 folgt ein Sternchen) erstellen. Dieser Wert spezifiziert die Spaltenbreite einer jeden Spalte in der Spaltengruppe. Somit ist der Spaltenwert der Minimalwert, der für das richtige Anzeigen des Inhalts einer jeden Spaltenzelle verantwortlich ist. Wenn sie den `"0*"`-Wert jedoch spezifizieren, können Browser keine Inkrementellspalte erstellen.

Angenommen, Ihre Tabelle ist 450 Pixel breit und enthält sechs Spalten. Sie möchten, dass jede der Spalten 75 Pixel breit ist. Der Code würde folgendermaßen aussehen:

```
<table border="1" width="450">
<colgroup span="6" width="75">
</colgroup>
```

Nehmen wir an, Sie möchten die Spalten jetzt ändern. Verwenden Sie dieselbe 450 Pixel breite Tabelle, erstellen Sie die beiden ersten Spalten jeweils mit einer Breite von 25 Pixeln, und die letzten beiden Spalten mit einer Breite von 100 Pixeln. Dazu benötigen Sie, wie im folgenden Beispiel gezeigt wird, zwei `<colgroup>`-Elemente:

```
<table border="1" width="450">
<colgroup span="2" width="25">
</colgroup>
<colgroup span="4" width="100">
</colgroup>
```

Was ist, wenn Ihnen diese Spalten als Spaltengruppe, in der alle Spalten dieselbe Breite oder dasselbe Aussehen haben, gar nicht gefallen? Das ist nun eine Aufgabe für das `<col>`-Element. Während `<colgroup>` die Struktur der Tabellenspalten definiert, definiert `<col>` ihre Attribute. Um dieses Element zu verwenden, beginnen Sie die Spaltendefinition mit einem `<col>`-Tag. In diesem Fall ist das abschließende Tag verboten.

Kommen wir zurück zu unserer 450 Pixel breiten Tabelle, in der Sie jetzt die beiden Tabellen in der ersten Gruppe mit jeweils 75 Pixeln formatieren möchten. In der zweiten Spaltengruppe haben Sie 50, 75, 75, 100 Pixel breite Spalten. Im folgenden Beispiel sehen Sie, wie Sie die zweite Spaltengruppe mit dem `<col>`-Tag formatieren:

```
<table border="1" width="450">
<colgroup span="2" width="75">
</colgroup>
<colgroup>
  <col span="1" width="50">
  <col span="2" width="75">
  <col span="1" width="100">
</colgroup>
```

Lassen Sie uns das ganze mal praktisch anwenden. Im folgenden Beispiel sehen Sie eine Tabelle, die den Stundenplan für verschiedene Naturwissenschaftliche Fächer zeigt. Definieren Sie als Erstes eine Tabelle, die ein 1 Pixel breite Rahmen umgibt, und die 100% der Browserfensterbreite einnimmt.

Definieren Sie als nächstes die Spaltengruppen in der Tabelle. In der ersten Spaltengruppe sollen die verschiedenen Fachnamen angezeigt werden. Die zweite Spaltengruppe besteht aus zwei Spalten, die die Nummer der jeweiligen Unterrichtsräume und die Uhrzeit der Unterrichtsstunden anzeigt. Das `align`- und das `valign`-Attribut, die Sie bereits kennen gelernt haben, wurden von HTML 4.0 für `col`- und `colgroup`-Elemente nicht verworfen. So können Sie den Vorteil hier nutzen. Die erste Spaltengrup-

pe besteht aus einer Zellenspalte, die 20% der Tabelle einnimmt. Der Inhalt der Zellen wird nach oben hin vertikal ausgerichtet und horizontal zentriert. Die zweite Spaltengruppe besteht aus zwei Spalten, die jeweils 40% der Tabelle einnehmen. Der Inhalt dieser Spalten wird zu den Zellen nach oben hin vertikal ausgerichtet.

Geben Sie schließlich die Tabellendaten, genauso wie Sie es bis jetzt gehandhabt haben, ein. Sie sehen im folgenden Beispiel, wie der komplette Code für den Stundenplan aussieht. Das Ergebnis im Internet Explorer sehen Sie in Abbildung 11.27:



```
<html>
<head>
<title>Grouping Columns</title>
</head>
<body>
<table border="1" width="100%" summary="grouping columns">
<caption><b>Science and Mathematic Class Schedules</b></caption>
<colgroup width="20%" align="center" valign="top"></colgroup>
<colgroup span="2" width="40%" valign="top"></colgroup>
<tr>
<th>Class</th>
<th>Room</th>
<th>Time</th>
</tr>
<tr>
<td>Biology</td>
<td>Science Wing, Room 102</td>
<td>8:00 AM to 9:45 AM</td>
</tr>
<tr>
<td>Science</td>
<td>Science Wing, Room 110</td>
<td>9:50 AM to 11:30 AM</td>
</tr>
<tr>
<td>Physics</td>
<td>Science Wing, Room 107</td>
<td>1:00 PM to 2:45 PM</td>
</tr>
<tr>
<td>Geometry</td>
<td>Mathematics Wing, Room 236</td>
<td>8:00 AM to 9:45 AM</td>
```

```

</tr>
<tr>
<td>Algebra</td>
<td>Mathematics Wing, Room 239</td>
<td>9:50 AM to 11:30 AM</td>
</tr>
<tr>
<td>Trigonometry</td>
<td>Mathematics Wing, Room 245</td>
<td>1:00 PM to 2:45 PM</td>
</tr>
</table>
</body>
</html>

```



Class	Room	Time
Biology	Science Wing, Room 102	8:00 AM to 9:45 AM
Science	Science Wing, Room 110	9:50 AM to 11:30 AM
Physics	Science Wing, Room 107	1:00 PM to 2:45 PM
Geometry	Mathematics Wing, Room 236	8:00 AM to 9:45 AM
Algebra	Mathematics Wing, Room 239	9:50 AM to 11:30 AM
Trigonometry	Mathematics Wing, Room 245	1:00 PM to 2:45 PM

Abbildung 11.27:
»Der Stundenplan mit formatierten Spaltengruppen«.

Gruppieren und Ausrichten von Zeilen

Da Sie jetzt wissen, wie man Spalten gruppiert und formatiert, lassen Sie uns jetzt zu den Zeilen kommen; Tabellenkopf, Tabellenfuß und Tabellenrumpf. So gibt es Vorteile beim Betrachten der Seite in einem HTML 4.0 Compliant Browser. Erstens kann der Tabellenrumpf unabhängig von Kopf und Fuß der Tabelle scrollen. Ein weiterer

Vorteil ist, dass beim Ausdruck der Tabelle, die auf mehrere Seiten verteilt sein kann, der Tabellenkopf und die Fußzeile der Tabelle auf jeder Seite erneut ausgedruckt wird. Sie können auch Cascading-Style-Sheet-Eigenschaften anwenden, um Tabellenkopf und -fuß hervorzuheben und dem Tabellen-Body ein anderes Aussehen zu verleihen. Die Tabellenkopf, Fuß- und Rumpfabschnitte werden mit `thead`, `tfoot` und `tbody`-Elementen jeweils definiert. Jedes dieser einzelnen Elemente muss dieselbe Spaltenanzahl haben.

Das `<thead>...</thead>`-Element definiert den Tabellenkopf, der Informationen über die Rumpfspalten in der Tabelle enthalten sollte. Es handelt sich hierbei um denselben Informationstyp, den Sie heute schon in die Kopfzellen eingegeben haben. Das einleitende `<thead>`-Tag ist immer erforderlich, wenn Sie einen Kopfabschnitt in Ihre Seite einfügen möchte, während das abschließende `</thead>`-Tag optional ist.

Der Tabellenkopf erscheint direkt nach dem `<table>`-Element oder nach dem `<colgroup>`-Element, wie es im folgenden Beispiel gezeigt wird. Der Tabellenkopf muss mindestens eine Spaltengruppe enthalten, die vom `<tr>`-Element definiert wird. Der Tabellenkopf wird folgendermaßen formatiert:

```
<table border="1" width="100%">
<caption><b>Science and Mathematic Class Schedules</b></caption>
<colgroup width="20%" align="center" valign="top">
  <colgroup span="2" width="40%" valign="top">
<thead>
  <tr>
    <th>Class</th>
    <th>Room</th>
    <th>Time</th>
  </tr>
</thead>
```

Das `<tfoot>...</tfoot>`-Element definiert den Tabellenfuß. Das einleitende `<tfoot>`-Tag ist immer obligatorisch, wenn der Tabellenfuß definiert wird. Der Tabellenfuß erscheint direkt nach dem Tabellenkopf (wenn einer vorhanden ist), oder nach dem `<table>`-Element (wenn kein Tabellenkopf vorhanden ist). Der Tabellenfuß muss mindestens eine Querspaltengruppe enthalten, die vom `<tr>`-Element definiert wird. Ein gutes Informationsbeispiel, das Sie in den Tabellenfuß platzieren könnten, ist eine Querspalte, die Zahlenspalten der Tabelle enthält.

Sie müssen den Tabellenfuß vor dem Tabellenrumpf erstellen. Der Grund dafür ist, dass der Browser die Fußtabelle anzeigen muss, bevor er alle Daten in den Tabellenrumpf platziert. Geben Sie im folgenden Beispiel die gleiche Information im Tabellenkopf und im Tabellenfuß ein. Der Code sieht folgendermaßen aus:

```
<tfoot>
  <tr>
    <th>Class</th>
```

```
<th>Room</th>
<th>Time</th>
</tr>
</tfoot>
```

Nachdem Sie Tabellenkopf und -fuß erstellt haben, definieren Sie die Zeilen im Tabellenrumpf. Eine Tabelle kann mehr als einen Rumpf enthalten, und jeder Rumpf eine oder mehrere Datenzeilen. Sind Sie verwirrt, fragen Sie sich, wie das nur funktionieren soll und wie Sie es verwenden sollen? Ich zeige Ihnen ein Beispiel, das demonstriert, warum das Ganze auch irgendwie cool ist.

Das `<tbody>...</tbody>`-Element definiert ein oder mehrere Tabellenbodys in Ihrer Tabelle.

Das einleitende `<tbody>`-Tag ist obligatorisch, wenn mindestens einer der folgenden Fälle eintrifft:

- ▶ Die Tabelle enthält Kopf-oder Fußabsätze
- ▶ Die Tabelle enthält mehr als einen Tabellenbody

Das folgende Beispiel zeigt zwei Tabellenrumpfe. Jeder Tabellenrumpf besteht aus drei Zeilen, die jeweils drei Zellen enthalten. Der Rumpf befindet sich im Code hinter dem Tabellenfuß, wie Sie im folgenden Beispiel erkennen können:

```
<tbody>
  <tr>
    <td>Biology</td>
    <td>Science Wing, Room 102</td>
    <td>8:00 AM to 9:45 AM</td>
  </tr>
  <tr>
    <td>Science</td>
    <td>Science Wing, Room 110</td>
    <td>9:50 AM to 11:30 AM</td>
  </tr>
  <tr>
    <td>Physics</td>
    <td>Science Wing, Room 107</td>
    <td>1:00 PM to 2:45 PM</td>
  </tr>
</tbody>
<tbody>
  <tr>
    <td>Geometry</td>
    <td>Mathematics Wing, Room 236</td>
    <td>8:00 AM to 9:45 AM</td>
  </tr>
```

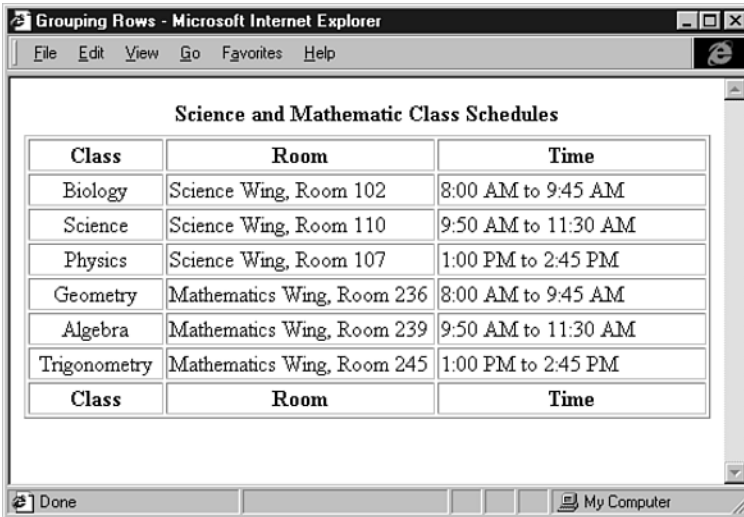


```

<tr>
<td>Algebra</td>
<td>Mathematics Wing, Room 239</td>
<td>9:50 AM to 11:30 AM</td>
</tr>
<tr>
<td>Trigonometry</td>
<td>Mathematics Wing, Room 245</td>
<td>1:00 PM to 2:45 PM</td>
</tr>
</tbody>
</table>

```

Fügen Sie nun alles zusammen, und Sie erhalten eine Tabelle, wie sie in Abbildung 11.28 gezeigt wird.



Class	Room	Time
Biology	Science Wing, Room 102	8:00 AM to 9:45 AM
Science	Science Wing, Room 110	9:50 AM to 11:30 AM
Physics	Science Wing, Room 107	1:00 PM to 2:45 PM
Geometry	Mathematics Wing, Room 236	8:00 AM to 9:45 AM
Algebra	Mathematics Wing, Room 239	9:50 AM to 11:30 AM
Trigonometry	Mathematics Wing, Room 245	1:00 PM to 2:45 PM
Class	Room	Time

Abbildung 11.28:
»Der Stundenplan mit einem Kopf, zwei Bodys und einem Fuß«

Die FRAME – und RULE-Attribute

Wenn Sie das vorhergehende Beispiel betrachten, ist nicht ganz klar, wo die Spaltengruppen erscheinen. Eine einfache Methode herauszufinden, wo sie liegen, ist die Verwendung des Attributs `frame` und `rules` des `<table>`-Elements.

Das `frame`-Attribut bestimmt die Anzeige des äußeren Rahmens der Tabelle. Sie können einen von mehreren möglichen Werten angeben, um zu definieren, welche Seiten des Tabellenrahmens sichtbar sind:

void	Der Standardwert. Keine Seiten des äußeren Rahmens sind sichtbar.
above	zeigt nur die obere Rahmenseite an
below	zeigt nur die untere Rahmenseite an
hsides	zeigt die oberen und unteren Rahmenseiten an
lhs	zeigt die linke Rahmenseite an
rhs	zeigt die rechte Rahmenseite an
vsides	zeigt die rechte und die linke Rahmenseite an
box	zeigt alle vier Rahmenseiten an
border	zeigt alle vier Rahmenseiten an

Das `rules`-Attribut ist dem `frame`-Attribut ähnlich, außer dass es die Trennlinien zwischen der Zelle mit einer Tabelle definiert. Folgende Werte gehören zum `rules`-Attribut:

none	Der Standardwert. Es werden keine Trennlinien zwischen den Zellen erstellt.
groups	Trennlinien erscheinen zwischen Spaltengruppe, die mit <code><thead></code> , <code><tfoot></code> und <code><tbody></code> definiert werden und zwischen Spaltengruppen, die mit <code><col></code> und <code><colgroup></code> definiert werden.
rows	Trennlinien erscheinen nur zwischen Zeilen
cols	Trennlinien erscheinen nur zwischen Spalten
all	Trennlinien erscheinen zwischen Zeilen und Spalten

Nun lassen Sie Ihre Spaltengruppen ruhig nach etwas mehr aussehen. Erstellen Sie einen Rahmen um die Stundenplantabelle, aber setzen Sie den Rahmen nur zum oberen und unteren Tabellenteil, indem Sie `frame="hsides"` im `<table>`-Tag anwenden.

Trennen Sie den Tabellenkopf- und fuß von den anderen Tabellenrümpfen (ein Rumpf gilt dem Naturwissenschaftlichen Fach, der andere der Mathematik). Trennen Sie auch die Fach-Spaltengruppen von den Raum/Zeit-Spaltengruppen. Verwenden Sie dazu `rules="groups"` im `<table>`-Element.

Sie müssen nur eine Zeile in Ihrem Code verändern, um diese Aufgabe bewerkstelligen zu können. Die überarbeitete Tabellendefinition sehen Sie wie folgt in Abbildung 11.29. Diese Abbildung zeigt das Ergebnis im Internet Explorer.

```
<table border="1" width="100%" frame="hsides" rules="groups">
```

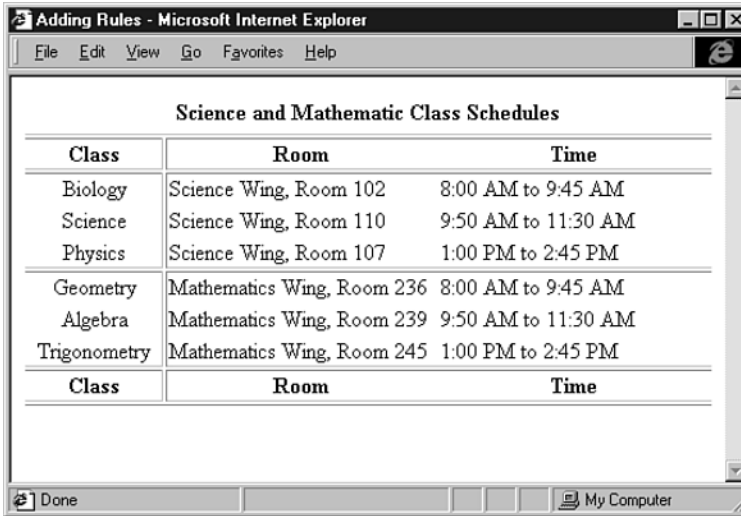


Abbildung 11.29:
Der Stundenplan
mit zusätzlichen
Rules

Weitere Tabellenelemente und Attribute

Tabelle 11.1 zeigt Ihnen einige der zusätzlichen Elemente und Attribute, die man bei der Tabellenerstellung einsetzen kann.

Attribut	Element	Verwendung
Char	Siehe »Verwendung«	Legt ein Zeichen fest, das als Achse dienen soll, um die herum der Inhalt der Zelle ausgerichtet werden soll. Sie können so zum Beispiel eine Dezimalstelle bei Zahlenwerten ausrichten. <char> kann in <colgroup>, <col>, <tbody>, <thead>, <tfoot>, <tr>, <td> und <th>-Elementen angewendet werden.
Charoff	Siehe »Verwendung«	Gibt an, an welcher Position das Zeichen, das im char-Attribut spezifiziert ist, frühestens vorkommen kann. charoff kann in <colgroup>, <col>, <tbody>, <thead>, <tfoot>, <tr>, <td> und <th>-Elementen angewandt werden.
Summary	<table>	Liefert eine genauere Beschreibung des Tabelleninhalts und wird vor allem bei Browsern verwendet, die keinen Text anzeigen.

Tabelle 11.1: Weitere Tabellenelemente und Attribute

Zusammenfassung

In diesem Kapitel haben Sie alles über Tabellen gelernt. Tabellen ermöglichen es Ihnen, Ihre Daten in Zeilen und Spalten anzuordnen, sodass Ihre Leser wichtige Informationen auf einen Blick erkennen können.

Bei der Arbeit mit den Tabellen in diesem Kapitel haben Sie viel über Überschriften und Daten, Titel, Zeilen- und Spaltendefinitionen, Ausrichtungsmöglichkeiten und Zellen, die sich über mehrere Zeilen und Spalten erstrecken, erfahren. Mit Hilfe dieser Funktionen sind sie in der Lage, Tabellen für fast alle Anforderungen zu erzeugen.

Die folgenden Punkte sollten Sie beachten, wenn Sie eine Tabelle erzeugen:

- ▶ Erstellen Sie sich eine Skizze der Zeilen und Spalten Ihrer Tabelle. Markieren Sie, welche Zellen sich über mehrere Zeilen und Spalten erstrecken.
- ▶ Beginnen Sie mit einem grundlegenden Gerüst, und gestalten Sie die Zeilen, Überschriften und Daten zeilen- und zellenweise in HTML. Ziehen Sie die Zellen gegebenenfalls über mehrere Zeilen oder Spalten. Testen Sie Ihren Code dabei häufig mit einem Browser, um sicherzustellen, dass er korrekt funktioniert.
- ▶ Passen Sie die Ausrichtung der Zeilen so an, wie sie der Mehrzahl der Zellen entspricht.
- ▶ Passen Sie die Ausrichtung den einzelnen Zellen an.
- ▶ Fügen Sie gegebenenfalls Zeilenumbrüche ein.
- ▶ Testen Sie Ihre Tabelle mit mehreren Browsern. Andere Browser verwenden möglicherweise ein anderes Tabellenlayout und können Ihren HTML-Code nicht korrekt interpretieren und reagieren anders auf mögliche Fehler.

In Tabelle 11.2 sehen Sie eine kurze Zusammenfassung der HTML-Elemente, die Sie heute kennen gelernt haben. Die gängigen HTML 4.0-Attribute, die in einem oder mehreren Elementen angewendet werden, finden Sie in Tabelle 11.3

Tag	Verwendung
<code><table>...</table></code>	Kennzeichnet eine Tabelle.
<code><caption>...</caption></code>	Erstellt eine (optionale) Überschrift für die Tabelle
<code><colgroup>...</colgroup></code>	Schließt eine oder mehrere Spalten in einer Gruppe zusammen
<code><col></code>	Definiert die Attribute einer Tabellenspalte
<code><thead>...</thead></code>	Definiert eine Zeilengruppe, die die Tabellenüberschrift festlegt. Eine Tabelle kann nur eine Überschrift enthalten.

Tabelle 11.2: Gängige HTML 4.0 Tabellenelemente

Tag	Verwendung
<code><tfoot>...</tfoot></code>	Erstellt eine Zeilengruppe, die den Tabellenfuß definiert. Eine Tabelle kann nur einen Tabellenfuß enthalten. muss vor dem ersten Tabellenrumpf definiert werden.
<code><tbody>...</tbody></code>	Definiert eine oder mehrere Zeilengruppen im Tabellenrumpf. Tabellen können mehr als einen Rumpf enthalten
<code><tr>...</tr></code>	Definiert eine Tabellenzeile, die Überschriften- und Datenzellen enthalten kann.
<code><th>...</th></code>	Definiert eine Tabellenzeile mit einer Überschrift. Der Text in Überschriftszellen wird in der Regel fett und sowohl horizontal als auch vertikal in der Zelle zentriert dargestellt.
<code><td>...</td></code>	Definiert eine Tabellenzeile mit Daten. Der Text in Tabellenzellen wird in Normalschrift dargestellt, linksbündig ausgerichtet und innerhalb der Zelle vertikal zentriert.

Tabelle 11.2: Gängige HTML 4.0 Tabellenelemente

Da verschiedene Dateiattribute zu einem oder mehreren der oben aufgelisteten Elementen gehören, liste ich diese extra auf. In Tabelle 11.3 sehen Sie eine kurze Zusammenfassung der HTML-Attribute, die Sie heute kennen gelernt haben, und die gängige HTML-Attribute sind.

Attribut	Element	Verwendung
Align	<code><tr></code>	Erlaubte Werte sind <code>left</code> , <code>center</code> und <code>right</code> , die die horizontale Ausrichtung der Zellen innerhalb einer Querspalte definieren (überschreibt die Standardausrichtung von Überschrift- und Datenzellen).
	<code><th></code> oder <code><td></code>	Überdeckt die Standardausrichtung und die Ausrichtung, die im <code><tr></code> -Tag festgelegt wurde. Erlaubte Werte sind <code>left</code> , <code>center</code> und <code>right</code> .
	<code><thead></code> , <code><tbody></code> , <code><tfoot></code>	Richtet den Inhalt in den Zellen des Kopfs, des Rumpfes und des Fußes der Tabelle aus. Erlaubte Werte sind <code>left</code> , <code>center</code> und <code>right</code> .
	<code><col></code>	Richtet alle Zellen in einer Spalte aus. Erlaubte Werte sind <code>left</code> , <code>center</code> und <code>right</code> .
	<code><colgroup></code>	Richtet alle Zellen in einer Spaltengruppe aus. Erlaubte Werte sind <code>left</code> , <code>center</code> und <code>right</code> .

Tabelle 11.3: Gängige HTML 4.0 Tabellen-Attribute

Attribut	Element	Verwendung
	<code><table></code>	In HTML 4.0 verworfen. Erlaubte Werte sind <code>left</code> , <code>center</code> und <code>right</code> . <code>align="center"</code> wird nicht in Browsern für HTML 3.2 und älter unterstützt. Richtet die Tabelle aus und gibt an, dass Text, der die Tabelle umgibt, mit Zeilenumbruch formatiert wird.
	<code><caption></code>	In HTML 4.0 verworfen. Legt fest, wo die Überschrift der Tabelle angezeigt werden soll. Die erlaubten Werte sind in den meisten Browsern <code>top</code> und <code>bottom</code> . HTML 4.0-Browser unterstützen auch <code>left</code> und <code>right</code> . Im Internet Explorer sind die erlaubten Werte <code>left</code> , <code>right</code> und <code>center</code> und geben die horizontale Ausrichtung der Überschrift an.
<code>bgcolor</code>	Alle	(HTML 3.2, in HTML 4.0 verworfen). Ändert die Hintergrundfarbe des Tabellenelements. Zellenfarben überdecken die Farbdefinition der Zeilen, die wiederum die Tabellenfarben überschreiben. Der Wert kann eine Hexadezimalzahl oder eine Farbbezeichnung sein.
<code>border</code>	<code><table></code>	Gibt an, ob eine Tabelle einen Rahmen erhält. Die Standardeinstellung ist ohne Rahmen. Wenn <code>border</code> einen Wert hat, ist der Wert die Breite des schattierten Rahmens, der die Tabelle umgibt.
<code>bordercolor</code>	<code><table></code>	(Internet Explorer und Netscape-Erweiterungen). Kann mit allen Tabellenelementen verwendet werden, um die Farbe des Rahmens, das dieses Element umgibt, zu ändern. Der Wert kann eine Hexadezimalzahl oder eine Farbbezeichnung sein.
<code>bordercolorlight</code>	<code><table></code>	(Internet-Explorer-Erweiterung) Dasselbe wie <code>bordercolor</code> . Definiert allerdings nur die helle Komponente eines 3D-Rahmens.
<code>bordercolordark</code>	<code><table></code>	(Internet-Explorer-Erweiterung) Dasselbe wie <code>bordercolor</code> . Definiert allerdings nur die dunkle Komponente eines 3D-Rahmens.
<code>cellspacing</code>	<code><table></code>	Definiert den Abstand zwischen den Zellen in einer Tabelle.

Tabelle 11.3: Gängige HTML 4.0 Tabellen-Attribute

Attribut	Element	Verwendung
cellpadding	<table>	Definiert den Abstand zwischen den Zellrändern und deren Inhalt.
char		Spezifiziert ein Zeichen, das als Achse dienen soll, um den Inhalt der Zelle auszurichten. Sie können so zum Beispiel eine Dezimalstelle bei Zahlenwerten ausrichten. Char kann in <colgroup>, <col>, <tbody>, <thead>, <tfoot>, <tr>, <td> und <th>-Elementen angewendet werden.
charoff		Gibt an, an welcher Position das Zeichen, das im char-Attribut spezifiziert ist, frühestens vorkommen kann. charoff kann in <colgroup>, <col>, <tbody>, <thead>, <tfoot>, <tr>, <td> und <th>-Elementen angewendet werden.
frame	<table>	Definiert, welche Seiten des die Tabelle umgebenden Rahmens sichtbar sind. Erlaubte Werte sind void, above, below, hside, lside, rside, vside, box und border.
height	<th> oder <td>	In HTML 4.0 verworfen. Gibt die Zellenhöhe in Pixeln oder in Prozent an.
nowrap	<th> oder <td>	In HTML 4.0 verworfen. Stellt sicher, dass der Browser keine Zeilenumbrüche für den Zellinhalt erstellt.
rules	<table>	Definiert, welche Trennlinien zwischen den Zellen in der Tabelle erscheinen. Erlaubte Werte sind none, groups, rows, cols und all.
width	<table>	Gibt die Tabellenbreite in exakten Pixelwerten oder in Prozent der Seitenbreite an (zum Beispiel 50%).
span	<colgroup>	Definiert die Spaltenanzahl in einer Spaltengruppe. Der Integer muss größer als 0 sein.
	<col>	Definiert die Spaltenanzahl, in denen Zellen vorhanden sind. Der Integer muss größer als 0 sein.
width	<colgroup>	Definiert die Zellenbreite aller Zellen einer Spaltengruppe.

Tabelle 11.3: Gängige HTML 4.0 Tabellen-Attribute

Attribut	Element	Verwendung
	<code><col></code>	Definiert die Zellenbreite aller Zellen in einer Zeile.
<code>colspan</code>	<code><th></code> oder <code><td></code>	Legt fest mit wie vielen Zellen zur rechten Seite hin die Zelle zu einer Zelle zusammengefasst werden soll.
<code>rowspan</code>	<code><th></code> oder <code><td></code>	Legt fest mit wie vielen Zellen unterhalb die Zelle zu einer Zelle zusammengefasst werden soll.
<code>valign</code>	<code><tr></code>	Gibt die vertikale Ausrichtung der Zellen innerhalb der Zeile an (überdeckt die Standardeinstellungen). Erlaubte Werte sind <code>top</code> , <code>middle</code> und <code>bottom</code> .
	<code><th></code> oder <code><td></code>	Überdeckt sowohl die vertikale Ausrichtung der Zeile als auch die Standardzellenausrichtung. Erlaubte Werte sind <code>top</code> , <code>middle</code> und <code>bottom</code> . In Netscape kann <code>valign</code> auch den Wert <code>baseline</code> haben.
	<code><thead></code> , <code><tfoot></code> , <code><tbody></code>	Definiert die vertikale Ausrichtung der Zellen im Kopf, Rumpf bzw. Fuß der Tabelle.
	<code><colgroup></code> , <code><col></code>	Definiert die vertikale Ausrichtung aller Zellen in einer Spaltengruppe. Definiert die vertikale Ausrichtung aller Zellen in einer einzelnen Spalte.
<code>width</code>	<code><th></code> oder <code><td></code>	In HTML 4.0 verworfen. Gibt die Zellenbreite in exakten Pixeln oder in Prozent der Tabellenbreite an (zum Beispiel 50%).

Tabelle 11.3: Gängige HTML 4.0 Tabellen-Attribute

Fragen und Antworten

- F** Es ist sehr kompliziert, eine Tabelle anzulegen, insbesondere wenn sich die Zellen über mehrere Zeilen und Spalten erstrecken. Das letzte Beispiel war fürchterlich.
- A** Sie haben recht. Es ist extrem schwierig, Tabellen manuell zu entwerfen. Wenn Sie jedoch Tools verwenden, um HTML-Code zu erzeugen, dann macht diese Tabellendefinition mehr Spaß.

F Meine Tabellen funktionieren in Netscape ausgezeichnet, aber in anderen Browsern werden sie völlig durcheinander gewürfelt angezeigt. Was habe ich falsch gemacht?

A Haben Sie alle `<tr>`-, `<th>`- und `<td>`-Tags geschlossen? Stellen Sie sicher, dass Sie die entsprechenden `</tr>`-, `</th>`- und `</td>`-Tags angegeben haben. Die abschließenden Tags sind zwar offiziell optional, andere Browser benötigen sie jedoch oft, um das Layout der Tabelle verstehen zu können.

F Kann man Tabellen verschachteln, also innerhalb einer Tabellenzelle eine weitere Tabelle definieren?

A Natürlich! Wie ich in diesem Kapitel erklärt habe, können Sie in eine Tabellenzelle beliebigen HTML-Code einfügen, und dieser wiederum kann andere Tabellen enthalten.

F Warum benutzt fast die ganze Welt `align` dazu, einen Titel oben oder unten auf einer Seite zu platzieren, aber der Internet Explorer tut etwas total anderes?

A Das weiß ich auch nicht. Und, schlimmer noch, Microsoft behauptet, dass man diese Definition von HTML 3.0 bekommen habe, aber keine der Versionen von HTML 3.0 oder der neueren Tabellenspezifikation definiert es auf diese Weise. HTML 4.0 verspricht diesen Attributen die Möglichkeit zum links- und rechtsbündigen Ausrichten hinzuzufügen. Der Internet Explorer verfügte allerdings über diese Ausrichtungsmöglichkeiten bevor HTML diese überhaupt in Betracht gezogen hatte.

Quiz

1. Was sind die Hauptelemente einer Tabelle und welche Tags identifizieren sie?
2. Welches Attribut ist das gängigste Attribut des Tabellen-Tag, und was tut es?
3. Welche Attribute definieren den Abstand der Zellenränder zum Zelleninhalt und den Abstand zwischen den einzelnen Zellen?
4. Welche Attribute werden verwendet, um Zellen zu erstellen, die größer als eine Spalte sind?
5. Welche Elemente werden verwendet, um den Kopf, den Fuß und den Rumpf einer Tabelle zu erstellen?

Antworten

1. Die Hauptelemente einer Tabelle (`<table>`-Tag) sind der Rahmen (mit dem `border`-Attribut definiert), die Überschrift (mit dem `<caption>`-Tag definiert), Kopfzellen (`<th>`), Datenzellen (`<td>`) und Tabellenzeilen (`<tr>`).
2. Das `border`-Attribut ist das gängigste Attribut für das `<table>`-Tag. Es legt fest, ob der Rahmen um die Tabelle angezeigt wird, und wie breit dieser Rahmen ist.
3. `cellpadding` definiert den Abstand zwischen Zellenrändern und ihrem Inhalt. `cellspacing` definiert den Abstand zwischen den Zellen.
4. Das `rowspan`-Attribut erstellt eine Zelle, die über mehrere Reihen geht. Das `colspan`-Attribut erstellt eine Zelle, die über mehrere Spalten geht.
5. `<thead>`, `<tbody>` und `<tfoot>` definieren Kopf, Rumpf und Fuß einer Tabelle.

Übungen

1. Hier ist eine kleine Herausforderung für Sie. Versuchen Sie, eine verschachtelte Tabelle zu erstellen (eine Tabelle, die eine Tabelle enthält). Erstellen Sie eine einfache Tabelle, die drei Zeilen und vier Spalten enthält. Erstellen Sie innerhalb der Zelle, die in der zweiten Spalte in der zweiten Zeile erscheint, eine zweite Tabelle, die zwei Zeilen und zwei Spalten enthält.
2. Verändern Sie die Tabelle aus Abbildung 11.28, sodass die Trennlinien in der Tabelle nur vertikal zwischen den Zellen erscheinen.



Frames und ver- knüpfte Fenster

**Woche
2**

Stellen Sie sich Folgendes vor: Sie navigieren auf einer sehr großen Web-Site, die viele unterschiedliche Webseiten umfasst. Je tiefer Sie in diese Site vordringen, umso mehr verlieren Sie die Orientierung und umso schwieriger wird es, den Weg zurück zum Anfang zu finden. Insgeheim sagen Sie sich, dass es eigentlich einfacher sein sollte, sich in der Site zurechtzufinden. Und jetzt aufgepasst, ich habe nämlich gute Neuigkeiten für Sie! Mit Frames tun sich die Besucher einer Site mit der Navigation um einiges leichter. Frames teilen das Browser-Fenster in eine Vielzahl von Sektionen, wobei in jeder Sektion eine Webseite aufzeigt werden kann.

In diesem Kapitel werden Sie mit den folgenden Themen vertraut gemacht:

- ▶ Was Frames sind, welche Vorteile sie Ihnen bezüglich des Layouts geben und wie sie unterstützt werden
- ▶ Wie Sie mit verknüpften Fenstern arbeiten
- ▶ Wie Sie mit Frames arbeiten
- ▶ Wie Sie komplexe Framesets erstellen

Frames und wie sie unterstützt werden

Mit Ausnahme der Cascading-Style-Sheets-Eigenschaften, werden die meisten der Features und der in den vorherigen Kapiteln erwähnten Tags von fast jedem Browser unterstützt. Die Erscheinung der Seite wird vielleicht nicht ganz Ihren Erwartungen entsprechen, wenigstens können aber Leute mit älteren Web-Browsern noch den Text und die auf der Seite enthaltenen Verknüpfungen erkennen.

In diesem Kapitel werden Sie jedoch einige neue Tags kennen lernen – nämlich solche, die Frames erzeugen, Webseiten, die mit Frames erzeugt wurden, können wegen der Beschaffenheit dieser Kennungen in anderen Browsern überhaupt nicht dargestellt werden. Die Tatsache, dass Frames in anderen Web-Browsern überhaupt nicht dargestellt werden können, hat sie zu einem der am heißesten diskutierten Themen in der »Netscape gegen den Rest der Welt«-Debatte gemacht.

Frames wurden zuerst im Netscape Navigator 2.0 eingeführt. Microsoft Internet Explorer 2.0 folgte, fügte jedoch noch etwas dazu. Zusätzlich zur Unterstützung der Frames, die im Netscape entworfen wurden, werden im Internet Explorer so genannte Inline Frames (auch schwebende Frames genannt) unterstützt.

Obwohl die beiden Hauptbrowser Frames unterstützten, waren diese kein offizieller Teil der HTML 3.2 Spezifikation. Deshalb unterstützten nur einige andere HTML 3.2 Browser Frames. Immer mehr Webseiten-Designer begannen aber mit Frames zu ar-

beiten, da sich durch sie erweiterte Navigationssysteme für Web-Sites entwickeln ließen. Deswegen sind in HTML 4.0 heute beide Frame-Arten (Netscape Frames und Internet Explorer Inline Frames) Teil der offiziellen Spezifikation.

Mit Frames bietet sich eine ganz andere Layout Kontrolle und es können ganz andere Möglichkeiten als die, die wir bisher kennen gelernt haben, wahrgenommen werden. Sehen Sie auf Bild 12.1, wie eine einfache Webseite mit Frames aussieht:

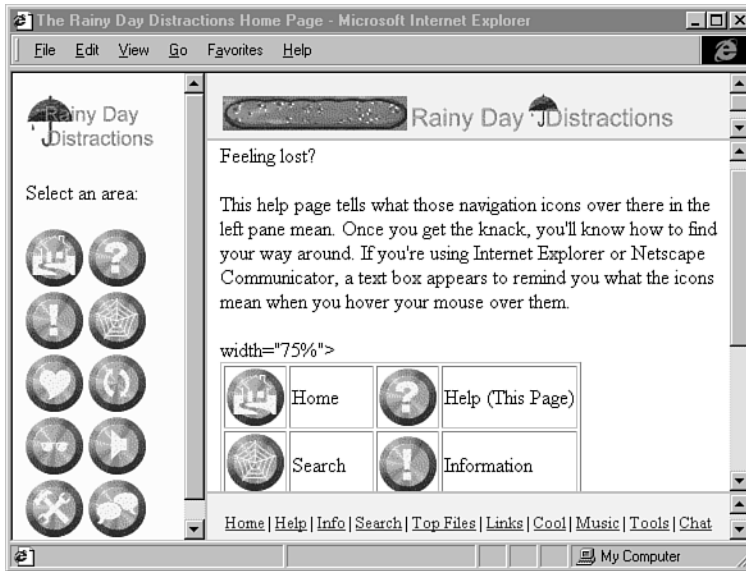


Abbildung 12.1:
Eine Beispiel-
Webseite mit
Frames

Auf diesem einzelnen Bildschirm hat Netscape Informationen integriert, die vorher viele verschiedene Bildschirmflächen benötigt hätten. Da die Informationen, die auf der Seite abgebildet sind, in verschiedene Bereiche oder Frames eingeteilt sind, kann der Inhalt eines einzigen Frame außerdem aktualisiert werden, ohne dass die Inhalte anderer Frames davon beeinträchtigt würden. Wenn Sie zum Beispiel eine beliebige Verknüpfung, die mit den Fotos im linken Rahmen verbunden ist, anklicken, wird der Inhalt des großen Rahmens auf der rechten Seite automatisch aktualisiert, um persönliche Angaben des ausgewählten Mitarbeiters darzustellen. Dabei werden die Inhalte des linken und des unteren Frame nicht beeinflusst.

Mit verknüpften Fenstern arbeiten

Bevor Sie sich ansehen, wie Frames auf eine Seite eingefügt werden, müssen Sie zuerst etwas über ein neues Attribut des `<a>`-Tag lernen, das `TARGET`-Attribut. Dieses neue Attribut wird folgendermaßen verwendet:

```
TARGET="window_name"
```

Normalerweise, wenn Sie eine Hyperverknüpfung anklicken, ersetzt der Inhalt der neuen Seite die aktuelle Seite im Browser-Fenster. In einer Umgebung, die ohnehin aus Fenstern besteht, gibt es aber keinen technischen Grund dafür, dass der Inhalt der neuen Seite nicht in einem neuen Fenster dargestellt wird, wobei der Inhalt der aufrufenden Seite weiterhin auf dem Bildschirm in seinem eigenen Fenster verbleibt.

Das `TARGET`-Attribut erlaubt Ihnen, genau dies zu tun, indem es den Web-Browser anweist, die Informationen, auf die eine Hyperverknüpfung hinweist, in einem neuen Fenster namens `window_name` darzustellen. Sie können das neue Fenster beliebig benennen, mit der einzigen Ausnahme, dass die Namen nicht mit einem Unterstrich (`_`) beginnen dürfen. Namen, die mit einem Unterstrich anfangen, sind nämlich für spezielle `TARGET`-Werte reserviert, über die Sie später in der Sektion »Magische `TARGET`-Namen« etwas lernen werden.

Wenn Sie das `Target`-Attribut innerhalb eines `<a>`-Tag verwenden, prüft Netscape zuerst, ob ein Fenster mit dem Namen `window_name` existiert. Wenn es existiert, wird der gegenwärtige Inhalt von `window_name` durch das Dokument ersetzt, auf das die Verknüpfung zeigt. Wenn andererseits momentan kein Fenster mit dem Namen `window_name` existiert, wird ein neues Browser-Fenster geöffnet, dem der Name `window_name` gegeben wird. Das Dokument, auf das die Hyperverknüpfung zeigt, wird dann in das neu geöffnete Fenster geladen.

Übung 12.1 Arbeiten mit Fenstern

Framesets sind auf das `target`-Attribut angewiesen, um Seiten in spezifische Frames innerhalb eines Framesets zu laden. In der folgenden Übung verwendet jedes der Hyperlinks das `target`-Attribut, um eine Webseite in verschiedenen Browser-Fenstern zu öffnen. Die Konzepte, die Sie hier kennen lernen, werden Ihnen helfen zu verstehen, wie Hyperlinks und das `target`-Attribut und Framesets zusammenarbeiten.

In dieser Übung werden Sie vier verschiedene HTML-Dokumente erzeugen, die Hyperverknüpfungen mit `target`-Attribut enthalten. Diese Hyperverknüpfungen werden dazu eingesetzt, zwei neue Fenster namens `yellow-page` und `blue-page` zu öffnen, wie

es in Abbildung 12.2 gezeigt wird. Das obere Fenster ist das ursprüngliche Web-Browser-Fenster (red-page), die yellow-page ist unten links und die blue-page befindet sich unten rechts.

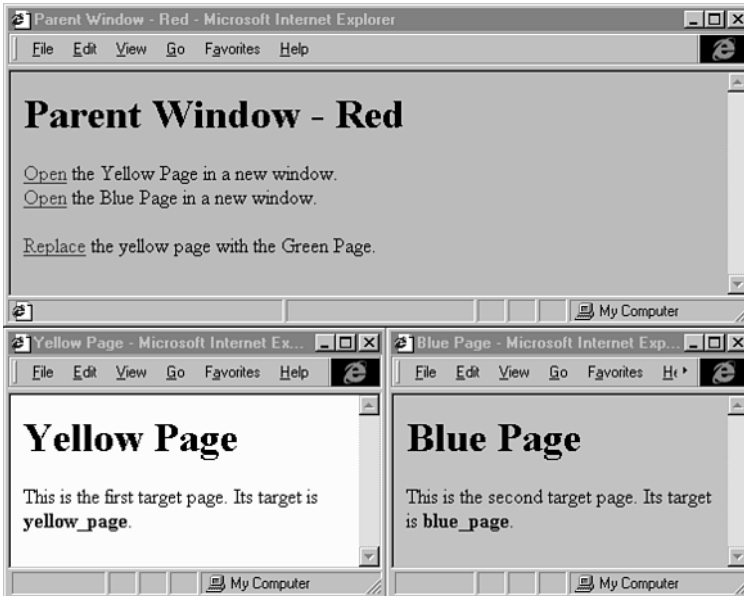


Abbildung 12.2: Hyperverknüpfungen können so konstruiert sein, dass für jede Seite, auf die sie verweisen, ein neues Fenster geöffnet wird

Zuerst erstellen Sie das Dokument, das im Haupt-Browser-Fenster dargestellt werden soll, welches in Abbildung 12.13 gezeigt ist, indem Sie Ihren Text-Editor öffnen und den folgenden Code eingeben:



```
<html>
<head>
<title>Parent Window - Red</title>
</head>
<body bgcolor="#ff9999">
<h1>Parent Window - Red</h1>
<p><a href="yellow.html" target="yellow_page">Open</a> the Yellow Page in a new
window. <br>
  <a href="blue.html" target="blue_page">Open</a> the Blue Page in a new
window. </p>
<p><a href="green.html" target="yellow_page">Replace</a> the yellow page with the
Green Page.</p>
```

```
</body>
</html>
```

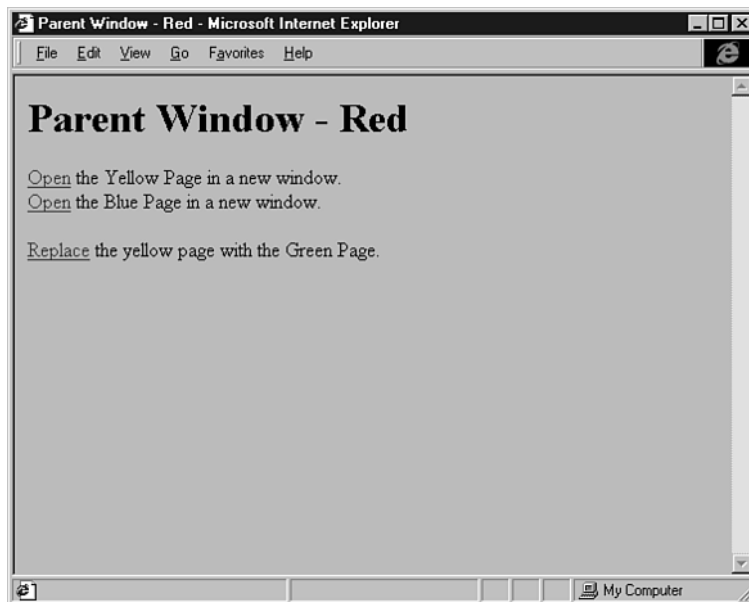


Abbildung 12.3:
Das Ziel-
Ursprungs-
Fenster

Mit diesem Code erstellen Sie eine hellrote Seite, die auf die anderen drei Seiten verweist. Speichern Sie diese HTML Quelle als `parent.html`.

Erstellen Sie als nächstes ein Dokument namens `yellow.html` mit folgendem Code. Sehen Sie das Ergebnis in Abbildung 12.4.



```
<html>
<head>
<title>Yellow Page</title>
</head>
<body bgcolor="#ffffcc">
<h1>Yellow Page</h1>
```



```
<p>This is the first target page. Its target is <b>yellow_page</b></p>
</body>
</html>
```



Abbildung 12.4:
Yellow.html wird hier
im Web-Browser-Fens-
ter namens yellow-
page dargestellt

Nachdem Sie yellow.html gespeichert haben, erschaffen Sie ein anderes Dokument mit dem Namen blue.html, siehe hierzu Abbildung 12.5. Und zwar tun Sie dies durch Eingabe des folgenden Codes:



```
<html>
<head>
<title>Blue Page</title>
</head>
<body bgcolor="#99ccff">
<h1>Blue Page</h1>
<p>This is the second target page. Its target is <b>blue_page</b>.</p>
</body>
</html>
```

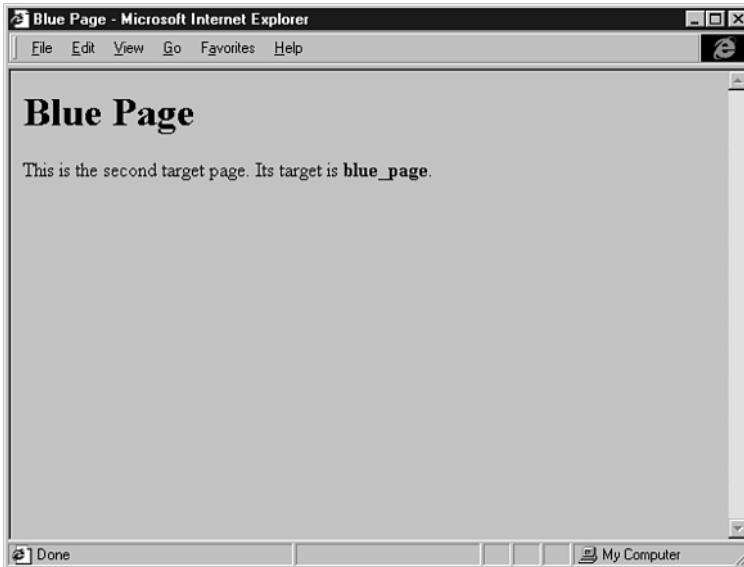


Abbildung 12.5:
Blue.html wird
hier im Web-
Browser-Fenster
namens blue-page
dargestellt.

Und schließlich erstellen Sie ein viertes Dokument namens `green.html`, das so aussieht:



```
<html>
<head>
<title>Green Page</title>
</head>
<body bgcolor="#ccffcc">
<h1>Green Page</h1>
<p>This is the third target page. Its target is <b>yellow_page</b>. It should
replace the yellow page in the browser.</p>
</body>
</html>
```

Um die Übung abzuschließen, laden Sie `parent.html` (red-page) in Ihren Web-Browser und klicken die oberen beiden Hyperverknüpfungen an. Dies geschieht, weil das erste Hyperlink eine `target="yellow_page"`-Anweisung enthielt, wie auch der folgende Code, der `parent.html` demonstrieren wird:

```
<p><a href="yellow.html" target="yellow_page">Open</a> the Yellow Page in a  
new window.<br />
```

Gehen Sie nun zur roten Seite zurück und klicken Sie das zweite Link an. Die blaue Seite wird in einem dritten Browser-Fenster geöffnet. Wahrscheinlich werden die neuen Fenster nicht so angeordnet sein, wie die in Bild 12.2; normalerweise überlappen sie sich. Die folgende `target="blue_page"`-Anweisung in der `parent.html`-Seite bewirkt, dass sich das neue Fenster öffnet.

```
<a href="blue.html" target="blue_page">Open</a> the Blue Page in a new  
window.</p>
```

Die vorherigen zwei Beispiele öffneten jede der Webseiten in einem neuen Browser-Fenster. Das dritte Link jedoch verwendet die `Target="yellow_page"`-Anweisung, um die grüne Seite in dem `yellow_page` Fenster zu öffnen. Dies wird mit folgendem Code in `parent.html` erreicht:

```
<p><a href="green.html" target="yellow_page">Replace</a> the yellow page  
with the Green Page.</p>
```



Abbildung 12.6:
green.html im Web-Browser-
Fenster, mit dem Namen
yellow_page
angezeigt.

Als Sie das Link für die `yellow_page` angeklickt haben, wurde das `yellow_page`-Fenster geöffnet. Die grüne Seite soll nun die Seite ersetzen, die sich bereits in diesem Fenster befindet. Wenn Sie den dritten Hyperlink der roten Seite anklicken, können Sie dies überprüfen. Sie werden den Inhalt der gelben Seite (mit dem `yellow_page`-Zielnamen) durch die grüne Seite (`green.html`) ersetzen (siehe Bild 12.6).

Die `<base>`-Kennung

Wenn Sie das `target`-Attribut mit Verknüpfungen verwenden, werden Sie manchmal in eine Situation kommen, in der alle oder die meisten der Hyperverknüpfungen auf einer Webseite auf dasselbe Fenster zeigen – dies passiert vor allem, wenn Sie Frames verwenden, wie Sie noch im folgenden Abschnitt feststellen werden.

In solchen Fällen können Sie, anstatt ein `target`-Attribut bei jedem `<a>`-Tag einzufügen, eine andere Kennung einsetzen, die `<base>` genannt wird, mit der Sie ein globales Ziel für alle Verknüpfungen auf einer Webseite definieren. Die `<base>`-Kennung erscheint in der folgenden Form:

```
<base target="window_name">
```

Wenn Sie die `<base>`-Kennung zwischen den `<head>...</head>`-Kennungen eines Dokuments verwenden, wird jede `<a>`-Kennung, die kein korrespondierendes `TARGET`-Attribut hat, das Dokument, auf das sie zeigt, in dem Fenster abbilden, das durch `<base target="yellow_page">` bestimmt wird. Wenn zum Beispiel die Kennung `<base target="window_name">` im HTML-Quellcode für `target1.html` enthalten gewesen wäre, hätte man die drei Hyperverknüpfungen auf folgende Weise schreiben können:

```
<html>
<head>
<title>Parent Window - Red</title>
<base target="yellow_page"> <!-- hier steht base target="value" -->
</head>
<body bgcolor="#ff9999">
<h1>Parent Window - Red</h1>
<p>
<a href="yellow.html">Open</a> <!-- hier muss kein target mehr definiert werden -->
  the Yellow Page in a new window.<br />
<a href="blue.html" target="blue_page">Open</a> the Blue Page in a new
  window. </p>
<p><a href="green.html">Replace</a> <!-- hier muss kein target mehr definiert
  werden -->
  the yellow page with the Green Page.</p>
</body>
</html>
```

In diesem Fall werden `yellow.html` und `green.html` in das Standard-Fenster geladen, das vom `<base>`-Tag (`yellow_page`) festgelegt wurde; `blue.html` überschreibt diese Voreinstellung, indem es sein eigenes Ziel-Fenster definiert.

Sie können das Voreinstellungs-Fenster, das durch das `<base>`-Tag festgelegt wurde, durch die Verwendung eines (von zwei) speziellen Fenster-Namen überschreiben. Wenn Sie `target="_blank"` in einer Hyperverknüpfung einsetzen, wird ein neues Browser-Fenster ohne Namen geöffnet. Sie können auch alternativ `target="_self"` einsetzen, sodass das aktuelle Fenster anstelle des durch das `<base>`-Tag definierte benutzt wird.



Folgendes sollten Sie sich einprägen: Wenn Sie kein Ziel über das `target`-Attribut des `<base>`-Tag und auch keines über das `<a>`-Tag eines Link bieten, dann wird das neue Dokument in denselben Frame geladen, in dem sich auch das Dokument mit dem Link befindet.

Frames anwenden

Die Einführung von Frames durch Netscape 2.0 leitet eine neue Ära für Web-Publizisten ein. Mit Frames können Sie Webseiten erstellen, die völlig anders aussehen und sich anders anfühlen als herkömmliche Webseiten – zum Beispiel Seiten mit verschachtelten Tabellen, Logos, Fußnoten und Rollbalken, um nur einige der Merkmale von Frames zu nennen.

Gleichzeitig verändern Frames die Bedeutung einer »Seite« für den Browser und den Leser. Im Unterschied zu all den vorigen Beispielen, die eine einzelne HTML-Seite benutzen, um einen Bildschirm voller Informationen zu zeigen, besteht bei Webseiten, die mit Frames erstellt wurden, ein einzelner Bildschirm tatsächlich aus einer Anzahl unterschiedlicher HTML-Dokumente, die miteinander in Verbindung stehen. Abbildung 14.8 zeigt, wie ein Minimum von vier unterschiedlichen Dokumenten notwendig ist, um den Bildschirm hervorzubringen, der bereits in Abbildung 12.1 gezeigt wurde.

Das erste HTML-Dokument, das Sie erzeugen müssen, wird Frame-Definitionsdocument genannt. In diesem Dokument geben Sie den HTML-Code an, der das Layout jedes einzelnen Frame bestimmt, und Sie bestimmen auch die Namen der einzelnen HTML-Dokumente, welche die physische Information enthalten, die dargestellt werden soll.. Die drei verbleibenden HTML-Dokumente enthalten die normalen HTML-Tags, die den physischen Inhalt jedes einzelnen Frame-Bereichs beschreiben. Dies sind die Dokumente, auf die sich das Frame-Definitionsdocument bezieht.



Das Frame-Definitionsdocument ist die Seite, die das Layout für jeden Frame und die Namen der HTML-Dokumente enthält, die diese Frames füllen werden.

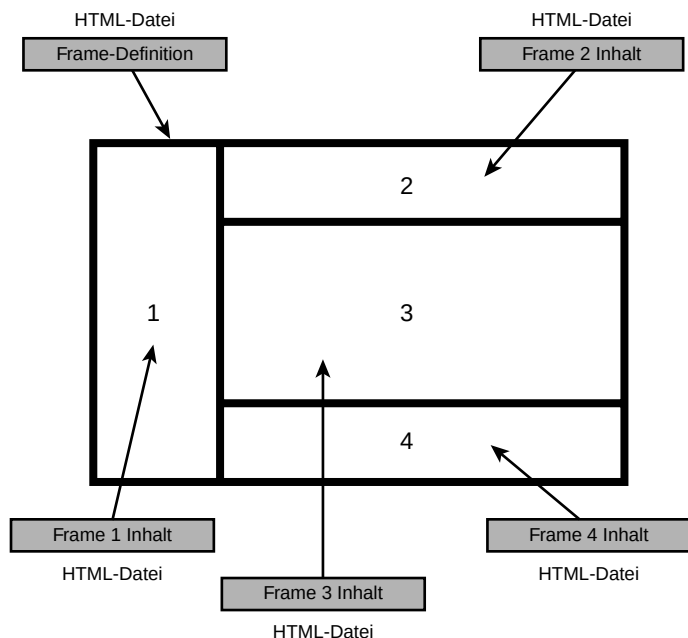


Abbildung 12.7:
Für jeden einzelnen Frame mussten separate HTML-Dokumente erzeugt werden.

Die <frameset>-Kennung

Um ein Frame-Definitionsdocument zu erstellen, benutzen Sie die <frameset>-Kennung. Wenn sie in einem HTML-Dokument verwendet wird, ersetzt die <frameset>-Kennung wie hier die <body>-Kennung:

```
<html>
<head>
<title>Page Title</title>
</head>
<frameset>
  ... hier fügen Sie Ihr Frameset ein...
</frameset>
</html>
```

Es ist wichtig, von Anfang an zu verstehen, wie sich ein Frame-Definitionsdocument von einem normalen Dokument unterscheidet. Wenn Sie ein <frameset>-Tag in einem HTML-Dokument verwenden, können Sie nicht gleichzeitig eine <body>-Kennung anwenden. Im Grunde genommen schließen diese beiden Kennungen einander aus. Außerdem sollten auch keine anderen Formatierungs-Tags, Hyperverknüpfungen oder Dokumententext in einem Frame-Definitionsdocument enthalten sein, mit einer

speziellen Ausnahme (die `<noframe>`-Kennung), über die Sie im Abschnitt »Die `<noframe>`-Kennung« noch mehr erfahren werden. Die `<frameset>`-Tags enthalten ausschließlich die Definitionen der Frames, die in einem Dokument verwendet werden, also die Struktur der Seite.

Das `<frameset>`-Tag wird in der für HTML 4.0 vorgeschlagenen Spezifikation mit zwei möglichen Attributen, `cols` und `rows`, unterstützt.



Ein *Frameset* sind mehrere Frames, die von `<frameset>`-Kennungen im Frame-Definitionsdocument eingeschlossen werden.

Das cols-Attribut

Wenn Sie ein `<frameset>`-Tag definieren, müssen Sie eines von zwei Attributen als Teil der Kennungsdefinition einfügen. Das erste dieser Attribute ist `COLS`, das folgendermaßen angewendet wird:

```
<frameset cols="spaltenbreite, spaltenbreite, ...">
```

Das Attribut `COLS` (Spalten) veranlasst Netscape, den Bildschirm in mehrere vertikale Frames aufzuteilen, deren Breite durch die Werte der Spaltenbreite angegeben werden. Diese Werte werden durch Kommata getrennt und können auf drei Arten definiert werden: als genauer Pixel-Wert, als Prozentsatz der totalen Breite des `<frameset>` oder durch ein Sternchen (*). Wenn Sie * benutzen, benutzt Netscape so viel Platz wie möglich für den aufgeführten Frame.

Wenn es in einer vollständigen Frame-Definition enthalten ist, bringt das folgende `<frameset>`-Tag einen Bildschirm mit drei vertikalen Frames hervor (vgl. Abbildung 12.8). Der erste Frame ist 100 Pixel breit, der zweite 50 Prozent der Bildschirmbreite und der dritte benutzt den restlichen Platz.



```
frameset cols="100,50%,*"Input
<html>
<head>
<title>Three Columns</title>
</head>
<frameset cols="100,50%,*">
  <frame src="leftcol.html">
  <frame src="midcol.html">
  <frame src="rightcol.html">
```

```
</frameset>
</html>
```

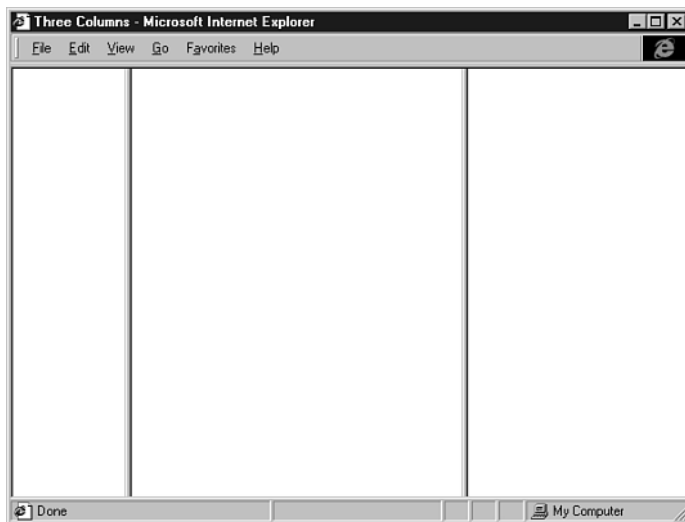


Abbildung 12.8:
Das Attribut `cols` legt fest, wie viele vertikale Frames oder Spalten in einem Frameset enthalten sind.



Da Sie Webseiten gestalten, die auf verschiedenen großen Bildschirmen betrachtet werden, sollten Sie absolute Frame-Größen sparsam verwenden. Wenn Sie doch einmal absolute Größen angeben, stellen Sie sicher, dass einer der anderen Frames mit einem `*` definiert wurde, um die verbleibende Bildschirmbreite einzunehmen.



Wenn Sie einen Frameset mit drei gleich großen Spalten erstellen wollen, verwenden Sie `cols="*, *, *"`. Auf diese Weise brauchen Sie sich nicht mit Prozentsätzen abzugeben, da Netscape jedem Frame, dessen Breite durch `*` bestimmt wird, einen gleichgroßen Platz zuweist.

Das rows-Attribut

Das `rows`-Attribut (Reihe, Zeile) arbeitet genauso wie das `cols`-Attribut, außer dass es den Bildschirm in horizontale Frames statt in vertikale einteilt. Um zum Beispiel den Bildschirm in zwei gleich hohe Frames einzuteilen, wie es in Abbildung 12.9 gezeigt wird, könnten Sie eine der beiden folgenden Methoden anwenden:



```
<html>
<head>
<title>Two Rows</title>
</head>
<frameset rows="50%,50%">
  <frame src="toprow.html">
  <frame src="botrow.html">
</frameset>
</html>
```

Alternativ können Sie auch die folgende Zeile Code verwenden:



```
<frameset rows="*,*">
```

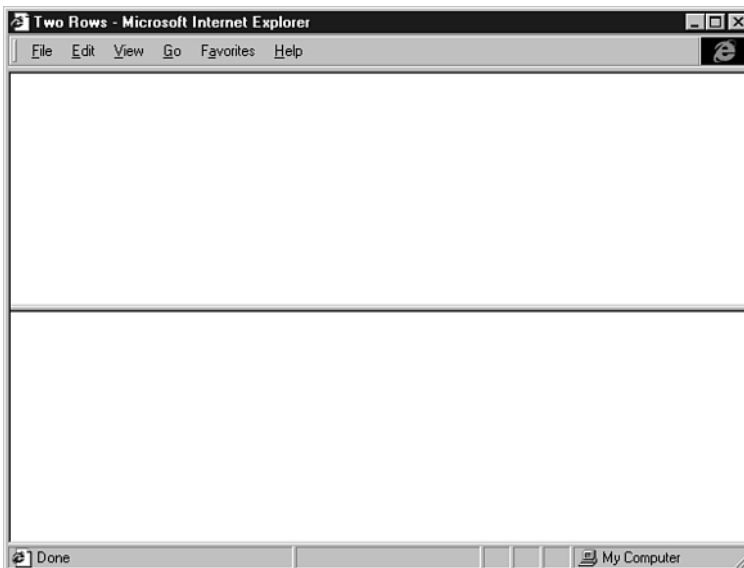


Abbildung 12.9:
Das rows-Attribut bestimmt die Anzahl der horizontalen Frames oder Reihen in einem Frameset.



Wenn Sie eines der vorhergehenden Beispiele selbst ausprobieren, werden Sie feststellen, dass das `<frameset>`-Tag anscheinend nicht funktioniert. Das liegt daran, dass momentan noch keine Inhalte für die Reihen oder Spalten im Frameset definiert worden sind. Um Inhalte festzulegen, wenden Sie die `<frame>`-Kennung an, die im nächsten Abschnitt besprochen wird.

Die `<frame>`-Kennung

Nachdem Sie Ihren grundlegenden Frameset festgelegt haben, müssen Sie mit jedem Frame noch ein HTML-Dokument verbinden. Dazu wenden Sie die `<frame>`-Kennung mit der folgenden Syntax an:

```
<frame src="document url">
```

Für jeden Frame, der innerhalb der `<frameset>`-Kennung definiert wird, müssen Sie ein entsprechendes `<frame>`-Tag ergänzen:



```
<html>
<head>
<title>The frame Tag</title>
</head>
<frameset rows="*,*,*">
  <frame src="document1.html">
  <frame src="document2.html">
  <frame src="document3.html">
</frameset>
</html>
```

In diesem Beispiel wurde ein Frameset mit drei gleich hohen horizontalen Frames definiert (vgl. Abbildung 12.10). Der Inhalt von `document1.html` wird im ersten Frame dargestellt, der von `document2.html` im zweiten und der von `document3.html` im dritten Frame.



Wenn Sie Frame-Definitionsdokumente erstellen, kann es hilfreich sein, die `<frame>`-Tags einzurücken, sodass sie sich von den `<frameset>`-Kennungen in Ihrem Quellcode abheben. Sie beeinflussen damit das Aussehen der resultierenden Webseiten nicht, gestalten aber dadurch den HTML-Quellcode übersichtlicher.

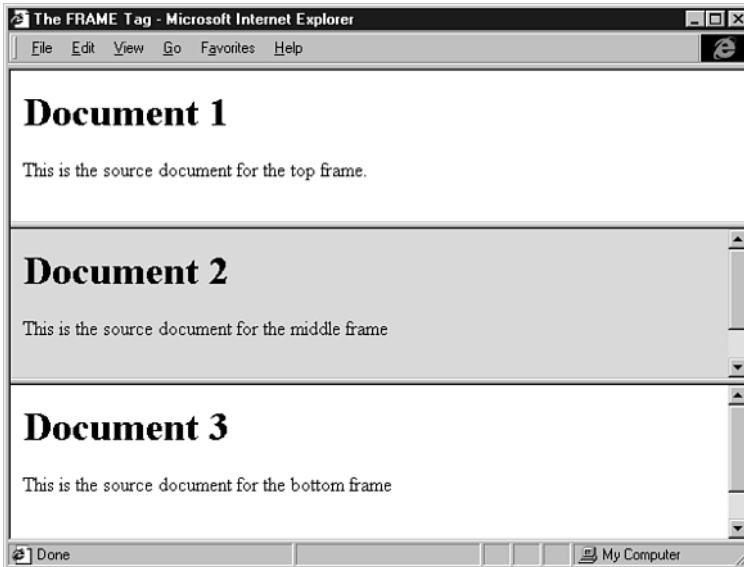


Abbildung 12.10:
Das `<frame>`-Tag
wird benutzt, um
den Inhalt jedes
Frame zu definie-
ren.

Die `<noframes>`-Kennung

Was geschieht, wenn ein Browser, der keine Frames unterstützt, zu einem Frameset-Dokument navigiert? Gar nichts; sie werden nur eine leere Seite zu sehen bekommen. Aber auch hierfür gibt es eine Lösung.

Ein spezieller Tag-Block namens `<noframes>` ermöglicht Ihnen, zusätzliche HTML-Codes als Teil des Frameset-Dokuments miteinzuschließen. Der innerhalb des `<noframes>`-Tag einbezogene Code wird nicht in Frame-kompatiblen Browsern angezeigt. Allerdings wird er in Browsern angezeigt, die keine Frames unterstützen. Das `<noframes>`-Tag sieht wie folgt aus:

```
<html>
<head>
<title>Frameset with No Frames Content</title>
</head>
<frameset>
  Hier fügen Sie Ihr Frameset ein.
</noframes>
```

```
Fügen Sie an dieser Stelle Text, Hyperlinks, Tags nach Ihren Vorstellungen ein.
</noframes>
</frameset>
</html>
```

Browser, die Frames unterstützen, werden den Text innerhalb des `<noframes>`-Blocks nicht anzeigen. Wenn die Seite aber in einem Web-Browser geladen wird, der keine Frames unterstützt, wird der Text angezeigt. Mit dem Frame-Inhalt und den Tags innerhalb von `<noframes>`, können Sie Seiten erstellen, die in beiden Browser-Typen angezeigt werden können. Etwas später in diesem Kapitel, werden sie einige `<noframes>`-Inhalte einem Frameset zufügen.

Frame-Ränder verändern

Beachten Sie bitte, dass alle Frames in diesem Kapitel dicke Ränder haben, die sie von einander trennen. In der ursprünglichen Implementation der Frames in Netscape 2.0 konnten Sie nur wenig gegen diese Situation tun. Mit der Einführung von Netscape 3.0 und Internet Explorer 3.0 wurden neue Attribute des `<frame>`- und `<frameset>`-Tag verfügbar, die eine gewisse Kontrolle über die Farbe und die Breite der Frame-Ränder bieten.

Beginnen wir mit dem `<frame>`-Tag. Durch die Verwendung von zwei Attributen, `bordercolor` und `frameborder`, können Sie die Ränder an- und abschalten und deren Farbe festlegen. Dem `bordercolor`-Attribut kann jeder gültige Farbwert entweder als Name oder als Hexadezimal-Wert zugewiesen werden. Für `frameborder` gibt es zwei zulässige Werte: 1 (zeigt Rahmen an) oder 0 (schaltet Rahmenanzeige aus).



Wenn Sie die Anzeige der Ränder abschalten, zeigt Netscape zwar die Standard-3-D-Ränder an, aber es lässt für die Ränder entsprechend Freiraum.



In HTML 4.0 ist bisher nur das `frameborder`-Attribut aufgeführt. Das `bordercolor`-Attribut stellt eine Erweiterung dar.

Die folgende Zeile erzeugt zum Beispiel einen Frame mit einem dunkelroten Rand um den mittleren Frame im Frameset:

```
<html>
<head>
<title>The frame Tag</title>
</head>
```

```
<frameset rows="*,*,*">
  <frame src="document1.html">
  <frame frameborder="1" bordercolor="#cc3333" src="document2.html">
  <frame src="document3.html">
</frameset>
</html>
```

Beide Attribute können Sie auch im `<frameset>`-Tag verwenden. Hier definieren sie dann die Standardeinstellung für das gesamte Frameset in Netscape und dem Microsoft Internet Explorer. HTML 4.0 bietet allerdings keines der beiden Attribute in Verbindung mit dem `<frameset>`-Tag.

Es kann natürlich zu Verwirrungen kommen, wenn farbige Ränder definiert werden. In der folgenden Frameset-Definition zum Beispiel ergibt sich ein Konflikt, da die beiden Frames über einen gemeinsamen Rand verfügen. Diesem Rand wird aber in den einzelnen Frames jeweils eine andere Farbe zugewiesen. Hierfür muss das `bordercolor`-Attribut angewandt werden:

```
<html>
<head>
<title>Conflicting Borders</title>
</head>
<frameset frameborder="0" rows="*,*,*">
  <frame frameborder="1" bordercolor="yellow" src="document1.html">
  <frame bordercolor="#cc3333" src="document2.html">
  <frame src="document3.html">
</frameset>
</html>
```

Zusätzlich sind in der Frameset-Definition die Ränder abgeschaltet. Der erste Frame soll allerdings über Ränder verfügen. Wie wird dieses Problem gelöst? Netscape hat drei einfache Regeln festgelegt, die in solchen Situationen angewendet werden:

- ▶ Die Attribute in der Definition des äußersten Frameset haben die geringste Priorität.
- ▶ Attribute werden in verschachtelten Frameset-Strukturen durch Attribute in den tieferen Frameset-Definitionen überschrieben.
- ▶ Das `bordercolor`-Attribut im aktuellen Frame überschreibt vorherige innerhalb übergeordneter `<frameset>`-Tags.

Zusätzliche Attribute

Sie können den `<frame>`-Kennungen noch ein paar extra Attribute hinzufügen, durch die Sie mehr Kontrolle darüber erhalten, wie der Benutzer mit Ihren Frames umgeht. Tabelle 12.1 gibt Einzelheiten darüber an. Andere Attribute kontrollieren den Seitenrand oder den Abstand zwischen den Frames und ob, bei Bedarf, eine Bildlaufleiste erscheinen soll.

Attribut	Wert	Beschreibung
<code>frameborder</code>	1	Zeigt Randlinien der Frames (default)
<code>frameborder</code>	0	Erstellt Frames ohne Randlinien
<code>longdesc</code>	URL	Spezifiziert eine URL, die eine längere Inhaltsbeschreibung des Framesets bereitstellt. Wird hauptsächlich in nicht visuellen Browsern verwendet.
<code>marginheight</code>	Pixel	<code>marginheight</code> gibt die Höhe der Umrandung an, die einen Frame oben und unten abschließt. Sie können <code>marginheight</code> durch eine Zahl als Pixel-Wert festlegen.
<code>marginwidth</code>	Pixel	Das <code>marginwidth</code> -(Randbreite)-Attribut erlaubt es Ihnen, den Rand links und rechts eines Frame mit einer Zahl festzulegen, die in Pixeln angegeben wird.
<code>name</code>	String	Weist dem Frame einen Namen zu, damit sich das <code>target</code> -Attribut auch auf den Frame beziehen kann.
<code>noresize</code>		Benutzer können die Position und Größe der Umrandung jedes Frame auf dem Bildschirm standardgemäß ändern, indem Sie die Umrandung mit der Maus festhalten und herumziehen. Um die Umrandungen des Frame zu sperren, sodass sie nicht mehr bewegt werden können, wenden Sie das <code>noresize</code> -Attribut an.
<code>scrolling</code>	yes	Wenn Sie <code>scrolling</code> auf <code>yes</code> stellen, wird der Frame mit Rollbalken ausgestattet, auch wenn diese unnötig sind (d. h. wenn der Text ganz in den Frame passt).
<code>scrolling</code>	no	Wenn Sie <code>scrolling</code> auf <code>no</code> stellen, setzen Sie damit die Verwendung von Rollbalken für den aktuellen Frame außer Kraft. (Beachten Sie, dass der Benutzer, wenn Sie <code>scrolling</code> auf <code>no</code> einstellen, nicht mehr zu dem Text blättern kann, der nicht in den Rahmen passt.)

Tabelle 12.1: Kontroll-Attribute für die `<frame>`-Kennung

Attribut	Wert	Beschreibung
scrolling	auto	(Default). Wenn der Frame-Inhalt mehr Platz in Anspruch nimmt als der im Frame zur Verfügung stehende Bereich, können Frame-kompatible Browser automatisch auf Scroll-Leisten (seitlich oder unterhalb des Frames) einfügen. Somit kann der Anwender durch das Dokument scrollen.
src	URL	Legt die URL des Dokument fest, das in einem Frame angezeigt wird, wenn das Dokument zum ersten Mal geladen wird.

Tabelle 12.1: Kontroll-Attribute für die `<frame>`-Kennung

Komplexe Framesets gestalten

Die Framesets, die Sie bisher kennen gelernt haben, stellen die grundlegendsten Typen von Frames dar, die dargestellt werden können. Im alltäglichen Gebrauch werden Sie solche einfachen Frame-Designs jedoch nur selten anwenden. Mal abgesehen von den einfachsten Sites, werden Sie wahrscheinlich komplexere Framesets aufbauen.

Übung 12.2: Erstellen Sie den Seiteninhalt für Ihr Frameset

In den meisten Fällen haben Framesets Navigationsleisten, die dem Leser Ihrer Site das Navigieren erheblich erleichtern. Die Navigationsleiste befindet sich fast ausschließlich auf der linken Seite Ihres Browser-Fensters. Jedesmal wenn der Leser ein Link im linken Navigations-Frame anklickt, wird die Seite, die der Leser /-in gewählt hat, im Hauptrahmen angezeigt. Das (wirklich einfache) Frameset, das Sie in dieser Übung erstellen werden, demonstriert diese Technik. Obwohl es sich hier nicht gerade um ein praktisches Beispiel handelt, ist es doch einfach und amüsant, und demonstriert exakt die gleiche Technik, die Sie auch für eine Navigationsleiste verwenden würden.

Wenn Sie normalerweise eine Webseite mit Frames gestalten, müssen Sie sich zuerst daran machen, das Frameset zu gestalten, bevor Sie mit dem doch so schwierigem Gestalten des Inhalts anfangen. Hierfür gibt es einen einfachen Grund: Natürlich wollen Sie zuerst wissen, wie groß Ihre Frames sein werden, bevor Sie mit dem Design von Grafiken und Seiteninhalt, die in den Frames platziert werden sollen, beginnen.

In dieser Übung wird es aber etwas anders ablaufen. Damit Sie den Zusammenhang auch wirklich verstehen, werden wir zuerst den Inhalt gestalten.

Folgende Inhaltsseiten enthalten keine der Tags, die Sie bis jetzt kennen gelernt haben. Insgesamt werden es acht Seiten werden. Ich versprechen Ihnen, die Codes hierfür kurz zu halten, damit Sie diese Seiten auch möglichst schnell erstellen können. Wollen wir anfangen?



Wenn Sie die Basisstrukturen eines Framesets gestalten, möchten Sie natürlich nicht mit Details, wie zum Beispiel dem aktuellen Inhalt der Frames, belästigt werden. Ihr Frameset wird aber nicht richtig angezeigt werden können, wenn es in einem Frame-kompatiblen Browser zum Testen geladen wird. Es sei denn, Sie definieren ein `<frame>`-Tag, das gültige Dokumente beinhaltet. Falls Sie aber ein Frameset vor dem Inhalt erstellen wollen, gibt es die Möglichkeit, ein kleines HTML-Dokument namens `dmmy.html` zu erstellen, welches Sie zum Testen Ihrer Frames benutzen können.

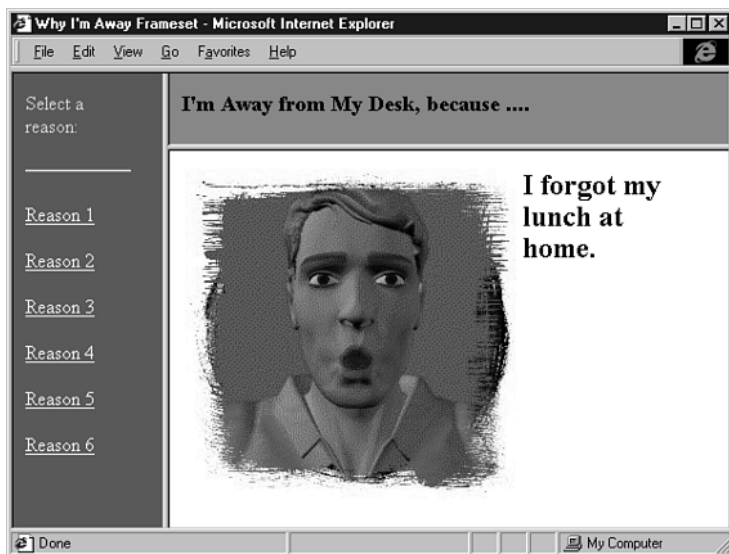


Abbildung 12.11:
Ihr Frameset wird
aus drei Frames
bestehen: top, left
und main (oben,
links, Hauptframe)

Das Frameset, das Sie in den Übungen 12.3 bis 12.7 erstellen werden, besteht aus drei Frames. Das Layout des Framesets wird wie in Bild 12.11 aussehen. Die Frameset-Seite wird als erste geladen und gibt dem Browser Anweisungen, das Browser-Fenster in drei Frames zu unterteilen. Als Nächstes werden die drei Seiten geladen, die sich oberhalb und links der Seite befinden, sowie das Hauptframe. Schließlich wird eine Alternativ-Seite angezeigt, sobald der Anwender das Frameset ohne Frame-kompatiblen Browser durchsucht.

Der sich oberhalb der Seite befindende Frame zeigt immer die gleiche Webseite namens `away.html` an. Die `choices.html`-Seite, die im linken Frame angezeigt ist, enthält eine Liste mit Links zu sechs verschiedenen Seiten namens `reason1.html` bis `reason6.html`. Jede dieser sechs Seiten wird in den Hauptframe im unteren rechten Teil des Framesets geladen.

Beginnen Sie mit dem Code, der die Seite für den oberen Frame erstellt. Diese Seite wird immer im Frameset angezeigt (allerdings nur so lange, bis ein Link angeklickt wird). Informationen die, sobald ein Browser Ihre Site durchsucht, ständig angezeigt werden sollen, können leicht in die Frames eingefügt werden. Im tatsächlichen Gebrauch, und nicht nur in diesem Beispiel, könnte der Inhalt dieses Frame Ihr Webseiten-Name sein, oder aber auch ein Site-Logo, ein Link zu Ihrer E-Mail-Adresse usw. Geben Sie den folgenden Code ein und speichern Sie ihn auf Ihrem Laufwerk als `away.html`. Bild 12.12 zeigt ein Beispiel dieser Seite.

```
<html>
<head>
<title>I'm Away from My Desk Because</title>
</head>
<body bgcolor="#cc6600" text="#ffcc33">
<h3>I'm Away from My Desk, because .... </h3>
</body>
</html>
```

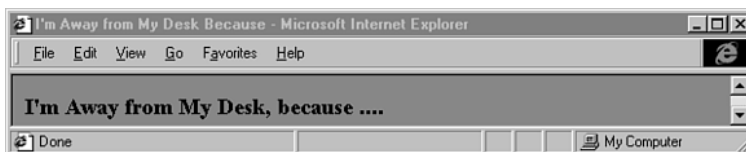


Abbildung 12.12:
Der TOP-Frame
im Frameset

Als Nächstes werden Sie den linken Frame des Framesets erstellen. Dieser Frame wird im tatsächlichen Gebrauch für die Navigationsleiste verwendet, mit der Ihre Leser auf verschiedene Schlüsselseiten Ihrer Site gelangen können. Eine individuelle Site könnte zum Beispiel eine Navigationsleiste haben, mit der die Leser auf eine Homepage, ein Gästebuch, eine Linkseite, usw. gelangen können. Eine Business-Site könnte unter anderem Links zu einer Produkt- oder Kundendienst-Sektion, einer Sektion, die die meist gestellten Leserfragen enthält oder aber eine Stellenangebot-Sektion haben.

Die Inhaltsseite des folgenden Beispiels funktioniert genauso, wie es eine Navigationsleiste im tatsächlichen Gebrauch auch tut. Wenn das entsprechende Link angeklickt wird, zeigt sie eine von sechs Seiten im Hauptframe des Framesets an. Die Inhaltsseite enthält Links zu sechs Seiten, `reason1.html` bis `reason6.html`. Diese Seiten werden Sie als Nächstes erstellen.

Nachdem Sie den folgenden Code in einer neuen Seite eingegeben haben, speichern Sie ihn auf Ihrem Laufwerk als `choice.html`. Ihre Seite sollte, sobald sie in einem Browser geöffnet wird, wie die auf Bild 12.13 aussehen.

```
<html>
<head>
<title>Reason I'm Out</title>
</head>
<body bgcolor="#006699" text="#ffcc66" link="#ffffff" vlink="#66ccff"
alink="#ff6666">
<p>Select a reason:</p>
<hr />
<p><a href="reason1.html">Reason 1</a></p>
<p><a href="reason2.html">Reason 2</a></p>
<p><a href="reason3.html">Reason 3</a></p>
<p><a href="reason4.html">Reason 4</a></p>
<p><a href="reason5.html">Reason 5</a></p>
<p><a href="reason6.html">Reason 6</a></p>
</body>
</html>
```

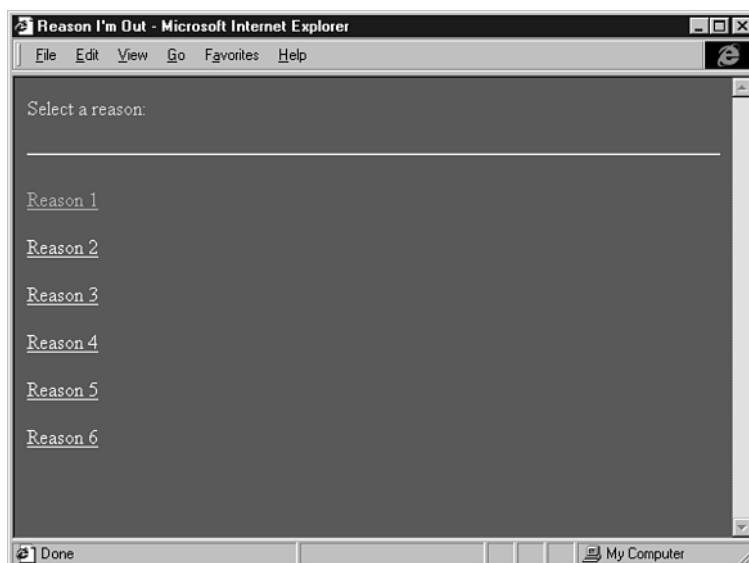


Abbildung 12.13:
Der linke Frame
im Frameset

Jetzt müssen noch die sechs Seiten erstellt werden, die im Hauptframe erscheinen, sobald ein Leser eines der Links im Inhalts-Frame wählt. Der Hauptframe dient dazu, Seiten anzuzeigen, die normalerweise in einem vollen Browser-Fenster angezeigt werden. Eines dürfen Sie aber nicht vergessen: Da Sie Ihre Seiten in einem Frameset mit

einer Navigationsleiste auf der linken Seite anzeigen, muss die Größe Ihrer Grafiken und Ihres Seiteninhalts kleiner sein. Vergessen Sie nicht, dass die Seiten nicht in einem ganzen Browser-Fenster angezeigt werden.

Damit die Seiten-Beispiele auch weiterhin einfach bleiben, habe ich die Seiten so erstellt, dass sie im Ansatz alle gleich sind, was bedeutet, dass der Code dieser Seiten so ziemlich der gleiche sein wird. Von Seite zu Seite ändern sich nur folgende Einzelheiten:

- ▶ Der Titel der Seite
- ▶ Das Bild, das auf jeder Seite erscheint. Die Bilder sind `uhoh.jpg`, `flirty.jpg`, `grumpy.jpg`, `happy.jpg`, `scared.jpg`, und `duh.jpg`. Diese Bilder sind auf der Web-Site für dieses Buch unter <http://www.tywebpub.com/> erhältlich.
- ▶ Der Text, der jedes Bild beschreibt.

Für diejenigen der sechs Seiten, die im Hauptframe erscheinen sollen, geben Sie folgenden Code auf einer neuen Seite ein und speichern Sie die Seite als `reason1.html`. Bild 12.14 zeigt Ihnen, wie die Seiten im Internet Explorer aussehen sollten.

```
<html>
<head>
<title>Reason 1 - Forgot My Lunch</title>
</head>
<body bgcolor="#ffffff">
<h2>I forgot my lunch
  at home.</h2>
</body>
</html>
```

Das Kodieren der restlichen fünf Seiten für den Hauptframe ist ähnlich. Ändern Sie den von Ihnen gerade erstellten Code, um die zweite der sechs Seiten zu erstellen. Die Unterschiede zum vorherigen Code (`reason1.html`) sind grau markiert. Speichern Sie die neue Seite als `reason2.html`. Der komplette Code lautet wie folgt:

```
<html>
<head>
<title>Reason 2 - By the Water Cooler</title>
</head>
<body bgcolor="#ffffff">
<h2>I'm flirting by
the water cooler.</h2>
</body>
</html>
```



Abbildung 12.14:
Die erste der
sechs Seiten, die
im Hauptframe
angezeigt wird.

Für die dritte Seite müssen Sie den Code wieder ändern und die neue Seite als `reason3.html` speichern:

```
<html>
<head>
<title>Reason 3 - Don't Ask!</title>
</head>
<body bgcolor="#ffffff">
<h2>None of
your business!</h2>
</body>
</html>
```

Hier ist die vierte Seite (`reason4.html`)

```
<head>
<title>Reason 4 - Out to Lunch</title>
</head>
<body bgcolor="#ffffff">
<h2>I'm out
to lunch.</h2>
</body>
</html>
```

Für die fünfte Seite (`reason5.html`) sieht der Code wie folgt aus:

```
<head>
<title>Reason 5 - Boss es Office</title>
</head>
<body bgcolor="#ffffff">
<h2>The boss
```

```
called me into his office.</h2>
</body>
</html>
Die sechste Seite (reason6.html)
<head>
<title>Reason 6 - I Don't Work Here Anymore</title>
</head>
<body bgcolor="#ffffff">
<h2>I just
got fired.</h2>
</body>
</html>
```

Die sechs Seiten für den Hauptframe des Framesets sind nun erstellt. Fangen wir also mit dem Frameset an.

Übung 12.3: Reihen und Spalten kombinieren

Damit Sie sich noch einmal vor Augen führen können, wie das Layout Ihres Frameset aussehen soll, haben wir es noch einmal in Bild 12.15 aufgeführt.



Für das Design eines kompletten Frame-Layouts, brauchen Sie unbedingt ein Storyboard. Das Storyboard hilft Ihnen, die Strukturen eines Framesets festzulegen. Wie Sie in Übung 12.5 »Verwenden Sie Frames mit Namen und Hyperlinks« sehen werden, ist ein Storyboard unbezahlbar, wenn es darum geht, Hyperlinks in das Frameset einzugeben.

In Abbildung 12.15 wird der rechte Abschnitt des Bildschirms in zwei horizontale Frames eingeteilt, und der dritte Frame, der sich unten auf der Seite befindet, überspannt die gesamte Breite des Bildschirms. Um ein Frame-Definitionsdocument zu erstellen, das dieses Layout beschreibt, öffnen Sie Ihren Text-Editor, und geben Sie die folgenden grundlegenden strukturellen HTML-Definitionen ein:

```
<html>
<head>
<title>Why I'm Away Frameset</title>
</head>
<frameset>
</frameset>
</html>
```

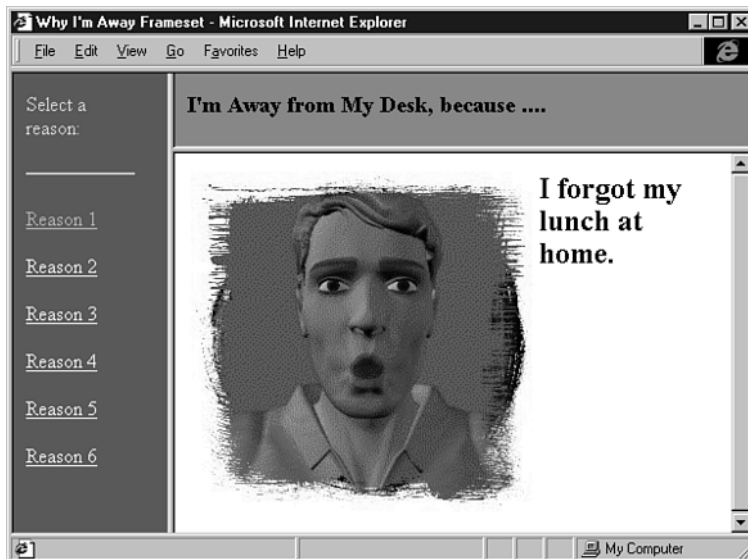


Abbildung 12.15:
Das Frameset mit
drei Frames: Fra-
me oben, unten
und Hauptframe

Als Nächstes müssen Sie entscheiden, ob Sie `rows`- und `cols`-Attribute in ihrem Basis-`<frameset>` verwenden wollen. Dazu sehen Sie sich Ihr Storyboard an – in diesem Fall Abbildung 14.2 – und versuchen Sie herauszubekommen, ob es Frame-Bereiche gibt, die sich quer über den Bildschirm erstrecken oder von oben nach unten. Wenn es Frames gibt, die sich von oben nach unten erstrecken, müssen Sie mit dem `cols`-Frameset beginnen; andernfalls fangen Sie mit dem `rows`-Frameset an. Wenn sich jedoch keine Frames in irgendeiner Richtung über den gesamten Bildschirm erstrecken, sollten Sie mit dem `COLS`-Frameset anfangen.

Um es einfacher auszudrücken, folgen hier noch mal drei leicht zu merkende Regeln:

- ▶ Von links nach rechts benutzen Sie `rows`.
- ▶ Von oben nach unten benutzen Sie `cols`.
- ▶ Unentschlossen? Benutzen Sie `cols`.



Die Grund hinter der Regel »von links nach rechts `rows` benutzen« kommt von der Art und Weise, wie Netscape Frames aufbaut. Jede getrennte `<frameset>`-Definition kann den Bildschirm (oder den Frame) entweder vertikal oder horizontal einteilen, nicht jedoch in beide Richtungen gleichzeitig. Deshalb müssen Sie beim Definieren von Framesets in einer logischen Reihenfolge vorgehen, wenn Sie das gewünschte Layout erreichen wollen.

Der rechte Frame in Abbildung 12.15 erstreckt sich von einer Seite des Bildschirms zur anderen. Wenn Sie die eben erwähnten Regeln anwenden, bedeutet das, dass Sie mit dem `rows`-Frameset beginnen müssen. Diesen definieren Sie folgendermaßen:

```
<frameset cols="125,*">
  <frame src="choice.html"> <!-- this loads the choices page into the left frame --
>
  <frame src="dummy.html"> <!-- this line is only temporary -->
</frameset>
```

Bisher haben Sie erreicht, dass Ihr Bildschirm in zwei Abschnitte aufgeteilt ist: ein kleiner Frame unten, der 125 Pixel hoch ist, und ein großer Frame rechts, der den verbleibenden Platz einnimmt.

Ich habe schon etwas früher in diesem Kapitel erwähnt, dass das Frameset-Dokument allein nicht den Frame-Inhalt beschreibt. Die Dokumente, auf die das SRC-Attribut hinweist, enthalten den Text, die Bilder und die Tags, die vom Frameset angezeigt werden. In der dritten Zeile des vorherigen Codes sehen Sie ein Beispiel eines solchen Tag. Die zweite Zeile spezifiziert die URL der Webseite, die im linken Frame (die von Ihnen erstellte `choice.html`-Seite) angezeigt wird. Die dritte Zeile würde eine Webseite namens `dummy.html` anzeigen (sofern Sie eine solche Seite erstellt haben). Diese Seite heben wir uns aber noch für die nächste Übung auf.

Übung 12.5: <frameset>-Kennungen ineinander verschachteln

Als nächsten Schritt in diesem Vorgang spalten Sie den oberen Frame in zwei vertikale Frames. Sie erreichen dies, indem Sie einen zweiten <frameset>-Block in den ursprünglichen <frameset>-Block einbauen. Wenn ein <frameset>-Block in einen anderen verschachtelt ist, muss der innere Block eines der <frame>-Tag des äußeren Frameset ersetzen. In diesem Fall müssen Sie die Codezeile der `dummy.html`-Seite ersetzen.

Um deshalb den ursprünglichen rechten Frame in die beiden vertikalen Frame-Bereiche aufzuteilen, ersetzen Sie die <frame>-Kennung für den ersten Frame mit einem eingebetteten <frameset>-Block. Auf diese Weise wird der Frameset dort eingebettet, wo sich vorher das <frame>-Tag befand, das nun ersetzt wird. Innerhalb des <frameset>-Tag für diesen neuen Block müssen Sie dann noch ein `cols`-Attribut definieren:

```
<html>
<head>
<title>Why I'm Away Frameset</title>
</head>
```

```
<frameset cols="125,*">
  <frame src="choice.html"> <!--hierdurch wird choice.html in den linken Frame
geladen -->
  <frameset rows="60,*">    <!--Der Frame für die zweite Spalte -->
    <frame src="away.html"> <!--wurde durch einen eingebetten -->
    <frame src="reason1.html"> <!--Frameset-Block -->
  </frameset>                <!-- ersetzt -->
</frameset>
```

Der eingebettete `rows`-Frameset definiert zwei Spalten, von denen die erste 60% der Höhe des eingebetteten Frame-Bereichs einnimmt und die zweite den verbleibenden Platz im verschachtelten Frame-Bereich ausfüllt. Außerdem sind innerhalb des `<frameset>`-Blocks zwei `<frame>`-Kennungen eingebettet, die den Inhalt jeder Spalte definieren. Der obere Frame lädt `away.html`, der untere Frame lädt `reason1.html`.



Wenn Prozentsätze innerhalb eines eingebetteten Frameset angewendet werden, dann beziehen sich diese auf den gesamten Bereich des eingebetteten Frame und nicht auf die gesamte Bildschirmgröße.

Schließlich speichern sie das fertige HTML-Dokument auf Ihrer Festplatte und testen es mit einem Frame-fähigen Browser. Wenn Sie eine Version eines anderen, nicht Frame-fähigen Web-Browsers zur Verfügung haben, versuchen Sie, das Dokument dort ebenfalls zu laden. (Sie sollten mit einem anderen Browser allerdings nichts vom Frameset zu sehen bekommen.)

Übung 12.5: Benannte Frames und Hyperlinks anwenden

Wenn Sie Ihre `frameset.html`-Seite in einen Frame-kompatiblen Browser laden, wird das, was auf Ihrem Bildschirm angezeigt wird, so ähnlich aussehen wie im Bild 12.15. Textgröße und -abstände werden vielleicht nicht ganz übereinstimmen, doch die Seite sieht zwar korrekt aus, was aber noch lange nicht heißt, dass sie auch gut funktioniert. Sollten Sie in diesem Stadium eines der Hyperlinks anklicken, würden Sie wahrscheinlich vom Ergebnis überrascht sein. Genauer gesagt, die Frame-kompatiblen Browser würden versuchen, den von Ihnen gewählten Dateiinhalt in den linken Frame zu laden, obwohl Sie in Wirklichkeit die Dokumente in den größeren rechten Frame laden wollten.

Etwas früher in diesem Kapitel haben Sie einiges über das `target`-Attribut gelernt. Das `target`-Attribut lädt verschiedene Seiten in unterschiedliche Browser-Fenster. Damit das Frameset auch so arbeitet wie Sie wollen, müssen Sie das `target`-Attribut etwas variieren. Sie wollen nicht, dass das `target`-Attribut auf ein neues Fenster hinweist,

sondern auf einen der Frames im aktuellen Frameset. Wenn Sie jedem Frame in Ihrem Frameset einen Frame-Namen (oder einen Fenster-Namen) geben, kann dies erreicht werden. Hierfür müssen Sie ein `name`-Attribut innerhalb des `<frame>`-Tag eingeben:

```
<frame src="document url" name="frame name">
```

Um jedem Frame des `frameset.html`-Dokuments einen Namen zu geben, müssen Sie jedem `<frame>`-Tag ein `name`-Attribut zuweisen. Im folgenden Beispiel sind diese zusätzlichen Angaben fett gedruckt. Ihre Frameset-Seite sieht nun wie folgt aus:

```
<html>
<head>
<title>Why I'm Away Frameset</title>
</head>
<frameset cols="125,*">
  <frame src="choice.html" name="left"> <!-- this loads the choices page into the
left frame -->
  <frameset rows="60,*">    <!-- the frame for column 2 -->
    <frame src="away.html" name="top"> <!-- has been replaced -->
    <frame src="reason1.html" name="main"> <!-- with an embedded -->
  </frameset>
  <!-- frameset block -->
</frameset>
```

Dieser Quellen-Code gibt dem linken Frame den Namen »left«, dem sich oberhalb rechts befindenden Frame den Namen »top« und dem sich unterhalb links befindenden Frame den Namen »main«. Speichern Sie als Nächstes diese `frameset.html`-Datei wieder ab. Wir sind schon fast fertig.

Übung 12.6: Dokumente mit individuellen Frames verknüpfen

Die Schlacht ist noch nicht geschlagen. Nach dem Benennen der Frames, müssen Sie nun die Links in der `choice.html`-Seite fixieren, damit die Zielseiten im `main`-Frame und nicht im `left`-Frame geladen werden.

Vielleicht erinnern Sie sich ja noch daran, dass das `target`-Attribut, zusammen mit dem `<a>`-Tag dafür verwendet wurden, ein Dokument in einem bestimmten Fenster zu laden. Verwenden Sie das gleiche Attribut, um zu kontrollieren, in welchen Frame ein Dokument geladen werden soll.

Sehen Sie nun, was mit dieser Übung erreicht werden soll: Jedesmal, wenn Sie ein Hyperlink im `left`-Frame anklicken, soll eine Seite im `main`-Frame geladen werden. Da Sie dem Frame, der sich unten rechts befindet bereits den Namen »main« zugewiesen

haben, müssen Sie nur noch `target="main"` jedem Tag des `choice.html`-Dokuments hinzufügen. Der folgende Ausschnitt der HTML-Quelle zeigt Ihnen, wie dies aussieht:

```
<p><a href="reason1.html" target="main">Reason 1</a></p>
<p><a href="reason2.html" target="main">Reason 2</a></p>
<p><a href="reason3.html" target="main">Reason 3</a></p>
<p><a href="reason4.html" target="main">Reason 4</a></p>
<p><a href="reason5.html" target="main">Reason 5</a></p>
<p><a href="reason6.html" target="main">Reason 6</a></p>
```

Da jedes Tag im `choice.html`-Dokument auf den gleichen Frame hinweist, können sie als Alternative das `<base target="value">`-Tag verwenden. Hier muss `target="main"` nicht in jedem `<a>`-Tag enthalten sein. Platzieren Sie stattdessen Folgendes in den `<head>...</head>`-Block des Dokuments.

```
<base target="main">
```

Nach all diesen Änderungen und neu erstellten Dokumenten, sollten Sie jetzt in der Lage sein, `frameset.html` in Ihrem Frame-kompatiblen Browser zu laden und alle HTML-Dokumente zu laden, die Sie in `choice.html` als Link definiert haben.



Wenn Sie das Layout richtig hinkriegen wollen, nachdem alle Ihre Links gut funktionieren, können Sie noch einmal zurückgehen und mit der Größe der Reihen und Spalten experimentieren, die vom `<frameset>`-Tag definiert werden. Bedenken Sie jedoch, dass die endgültige Darstellung eines Frameset immer noch von der Bildschirmgröße und dem Betriebssystem der Betrachter Ihres Dokuments bestimmt wird.

Übung 12.7: Fügen Sie den NOFRAMES-Inhalt hinzu

Obwohl Sie jetzt ein perfekt funktionierendes Frameset haben, müssen Sie ihm noch ein Feature hinzufügen. Denken Sie daran, dass es vielleicht einige Personen gibt, die durch Ihre Site navigieren, aber keinen Frame-kompatiblen Browser haben. Mit folgenden zusätzlichen Angaben auf Ihrer Frameset-Seite werden Inhalte erstellt, die dem Besucher Ihrer Site angezeigt werden, sobald er das Frameset öffnet.

Öffnen Sie also noch einmal Ihre `frameset.html`-Seite. Ihr Code hierfür sieht wie folgt aus:

```
<html>
<head>
<title>Why I'm Away Frameset</title>
</head>
<frameset cols="125,*">
  <frame src="choice.html" name="left"> <!--hierdurch wird choice.html in den linken
```

Frame geladen -->

```
<frameset rows="60,*">  <!--Der Frame für die zweite Spalte -->
  <frame src="away.html" name="top">  <!--wurde durch einen eingebetten -->
  <frame src="reason1.html" name="main">  <!--Frameset-Block -->
</frameset>                <!-- ersetzt -->
</frameset>
</html>
```

Setzen Sie gleich nach dem `</frameset>`-Tag und vor dem schließenden `</html>`-Tag das `<noframes>...</noframes>`-Element und den Inhalt ein.

```
<noframes>
  <body bgcolor="#ffffff">
<h1>I'm Away from My Desk, because ...</h1>
<ul>
  <li>Reason 1 - <a href="reason1.html">I forgot my lunch at home.</a></li>
  <li>Reason 2 - <a href="reason2.html">I'm flirting by the water cooler.</a></li>
  <li>Reason 3 - <a href="reason3.html">None of your business.</a></li>
  <li>Reason 4 - <a href="reason4.html">I'm out to lunch.</a></li>
  <li>Reason 5 - <a href="reason5.html">The boss just called me in his office.</
a></li>
  <li>Reason 6 - <a href="reason6.html">I just got fired.</a></li>
</ul>
</body>
</noframes>
```



Abbildung 12.16:
Diese Seite wird
angezeigt, wenn
ein Anwender das
Frameset mit
einem Frame-
kompatiblen
Browser ansieht.

Wenn nun ein Anwender, der einen Frame-kompatiblen Browser benutzt, zum Frameset navigiert, wird er oder sie eine Seite angezeigt bekommen, die ungefähr wie die in Abbildung 12.16 aussieht.

Magische TARGET-Namen

Da Sie jetzt wissen, was das `target`-Attribut in einem Frameset bewirkt, müssen Sie auch noch ein paar spezielle `target`-Namen, die man in einem Frameset anwenden kann, kennen. Sie können vier spezielle Werte für das `target`-Attribut bestimmen, von denen Sie bereits zwei (`_blank` und `_self`) kennen gelernt haben. Netscape nennt diese Werte magische `target`-Namen (Zielnamen). Tabelle 12.2 führt diese magischen `target`-Namen auf und beschreibt ihre Anwendung.

TARGET-Name	Beschreibung
TARGET="_blank"	(leeres Ziel) Zwingt das Dokument, das durch das <code><a></code> -Tag festgelegt ist, in einem neuen, »namenlosen« Fenster geladen zu werden.
TARGET="_self"	Lädt das Dokument, das durch das <code><a></code> -Tag bestimmt ist, in das Fenster oder den Rahmen, der das <code><a></code> -Tag beinhaltet. Dies kann nützlich sein, wenn das <code><base></code> -Tag einen anderen Frame als Ziel definiert, ein bestimmter Link allerdings in den aktuellen Frame geladen werden muss.
TARGET="_parent"	Zwingt das Link dazu, im <code><frameset></code> -Ursprungsdocument des gegenwärtigen Dokuments geladen zu werden. Wenn das aktuelle Dokument jedoch keinen Ursprung hat, wird TARGET="_self" angewandt.
TARGET="_top"	Zwingt das Link dazu, im vollen Web-Browser-Fenster zu laden, wobei der aktuelle <code><frameset></code> ersetzt wird. Wenn das gegenwärtige Dokument sich jedoch bereits oben (= <code>_top</code>) befindet, wird TARGET="_self" angewandt.

Tabelle 12.2: Magische TARGET-Namen

Floating Frames

Mit dem Internet Explorer 3.0 stellte Microsoft eine neue Variante der Frames vor: *Floating Frames* (Auch als *Inline Frame* bezeichnet). Dieses Konzept, das sich auch in der HTML 4.0-Spezifikation wiederfindet, unterscheidet sich von der ursprünglichen Idee hinter den Frames, die von Netscape vorgestellt wurde.

Schwebende Frames haben ihre Vor- und Nachteile. Der Vorteil ist, dass Sie einen schwebenden Frame, wie ein Bild, eine Tabelle oder irgendein anderes Webseiten-Element, an eine beliebige Stelle in Ihrer Webseite platzieren können. Hiermit werden Ih-

nen viele Layout Möglichkeiten geboten, die Sie mit den heute erlernten Standard-Framesets nicht hätten.



Die Autoren der HTML 4.0 Frames-Spezifikation haben im Bereich der Frame-Definition auch Floating Frames mit aufgenommen – allerdings mit Bedenken. Nach dem Entwurf kann nämlich derselbe Effekt über das `<object>`-Tag erreicht werden, was die Aufnahme dieses Frame-Typs in Frage stellt. Im Entwurf sind sie weiterhin enthalten und der Internet Explorer 3.0 unterstützt diese Technologie, was das Erlernen dieser Technologie sinnvoll macht.

Dennoch müssen Sie bei schwebenden Frames einige Vorsichtsmaßnahmen treffen. Zunächst einmal wäre da, dass der Internet Explorer 3.0 und spätere Versionen die einzigen Browser sind, die schwebende Frames derzeit unterstützen. Bevor Sie schwebende Frames für Ihr Webseiten-Design verwenden, müssen Sie wissen, dass es da noch einen Vorbehalt gibt.

Standard-Framesets ermöglichen es Ihnen, einen sich ändernden Inhalt zu spezifizieren, der auch denjenigen angezeigt werden kann, die ohne einen Frame-kompatiblen Browser zu einem Frameset navigieren. Diese Option haben Sie aber mit dem `<iframe>`-Element / Tag leider nicht. Wenn Sie nun einen schwebenden Frame in Ihrer Webseite haben und ein Anwender navigiert mit einem Browser, der diese nicht unterstützt, zu Ihrer Seite, wird er oder sie absolut nichts in dem Bereich, in dem eigentlich der Frame sein sollte, erkennen können. Es kann natürlich gut möglich sein, dass Sie überhaupt keine schwebenden Frames verwenden wollen, zumindest so lange nicht, bis Sie sichergehen können, dass alle Ihre den Besucher den Internet Explorer 3.0 oder spätere Versionen benutzen.

So, die Warnungen sind ausgesprochen. Sprechen wir noch kurz an, wie man schwebende Frames erstellt. Wie Bilder auch, erscheinen diese Frames inline, in der Rumpfmittle eines HTML-Dokuments. Ihre schwebenden Frames werden mit dem `<iframe>`-Tag definiert (das »I« in `<iframe>` bezieht sich auf inline). Das `<iframe>`-Tag ermöglicht es Ihnen, ein HTML-Dokument in irgendeinen Frame eines anderen HTML-Dokuments einzugeben.

Attribut	Beschreibung
width	Legt die Breite des Floating Frame, der das HTML-Dokument aufnimmt, in Pixeln fest.
height	Legt die Höhe des Floating Frame, der das HTML-Dokument aufnimmt, in Pixeln fest.
src	Gibt die URL des HTML-Dokuments an, das in dem Frame angezeigt werden soll.

Tabelle 12.3: Schlüsselattribute

Attribut	Beschreibung
name	Legt den Namen des Frame fest, der für Verknüpfungen bzw. Zielangaben verwendet werden kann.
frameborder	Gibt an, ob der Frame ein Rahmen anzeigen soll. Nach dem Entwurf von Cougar legt der Wert 1 fest, dass ein Rahmen angezeigt werden soll, wohingegen der Wert 0 die Anzeige eines Rahmens verhindert.
marginwidth	Legt die Breite des Randes in Pixeln fest.
marginheight	Legt die Höhe des Randes in Pixeln fest.
noresize	Gibt an, dass die Größe des Frame nicht vom Benutzer geändert werden kann. (Internet-Explorer-Erweiterung)
scrolling	Wie beim <frame>-Tag. Gibt an, ob der Frame über Bildlaufleisten verfügen soll. (Dieses Attribut kann die Werte yes, no oder auto annehmen; der Standardwert ist auto.)
vspace	Gibt die Höhe des Randes an. (Internet-Explorer-Erweiterung)
hspace	Gibt die Breite des Randes an. (Internet-Explorer-Erweiterung)
align	Wie beim -Tag. Legt die Positionierung des Frame in Bezug auf die Textzeile fest, in der sich dieser befindet. Mögliche Werte sind left, middle, right, top und bottom, wobei der letzte der Standardwert ist. Die Werte absbottom, absmiddle, baseline und texttop stehen als Internet-Explorer-Erweiterungen zur Verfügung.

Tabelle 12.3: Schlüsselattribute (Forts.)

Da Sie mit der Anwendung von normalen Frames und Bildern innerhalb von Dokumenten vertraut sind, ist es ziemlich einfach das <frame>-Tag anzuwenden. Der folgende Code zeigt Ihnen eine Methode, wie Sie die Away from My Desk-Seite mit einem Floating Frame verbinden können. Erstellen Sie in diesem Beispiel zuerst eine Seite mit rotem Hintergrund. Die Links, die der Anwender anklickt, erscheinen in einer einzigen, oberhalb des Floating Frame zentrierten Zeile. Jeder dieser Links steht im folgendem Code separat in einer eigenen Zeile.

Gleich nach den Links (die auf den schwebenden Rahmen namens »reason« abzielen), erscheint der Code für die schwebenden Rahmen innerhalb eines zentrierten <div>-Elements. Im folgenden Code werden Sie sehen, dass der schwebende Rahmen der Seite zentriert wird, 450 Pixel breit und 315 Pixel hoch ist. Bild 12.17 zeigt das Ergebnis.

```
<html>
<head>
<title>I'm Away From My Desk</title>
</head>
<body bgcolor="#ffcc99">
```

```

<h2>I'm away from my desk because ...</h2>
<p align="center">
  <a href="reason1.html" target="reason">Reason 1</a> |
  <a href="reason2.html" target="reason">Reason 2</a> |
  <a href="reason3.html" target="reason">Reason 3</a> |
  <a href="reason4.html" target="reason">Reason 4</a> |
  <a href="reason5.html" target="reason">Reason 5</a> |
  <a href="reason6.html" target="reason">Reason 6</a> </p>
<div align="center">
<iframe name="reason"
  src="reason1.html"
  width="450"
  height="315">
</div>
</body>
</html>

```



Abbildung 12.17:
Ein Inline (oder
floating) Frame

Zusammenfassung

Wenn Sie von diesem Kapitel Kopfschmerzen bekommen haben, sind Sie wahrscheinlich nicht allein. Obwohl die grundlegenden Konzepte, die hinter der Verwendung von Frames stehen, ziemlich einfach sind, ist ihre Anwendung doch ein wenig schwieriger zu verwirklichen. Deshalb ist die beste Methode mehr über Frames zu lernen die, damit zu experimentieren.

In diesem Kapitel haben Sie gelernt, wie Sie ein Dokument mit einem neuen oder bereits existierenden Fenster verknüpfen. Außerdem lernten Sie, wie Sie Framesets aufbauen und sie miteinander verknüpfen können, indem Sie die in der Tabelle 12.4 aufgezählten Kennungen anwenden.

Kennung	Attribut	Beschreibung
<base tar-get="fenster">		Bestimmt das globale Link-Fenster für ein Dokument.
<frameset>		Definiert die grundlegende Struktur eines Frameset.
	cols	Definiert die Anzahl der Frame-Spalten und deren Breite innerhalb eines Frameset.
	rows	Bestimmt die Anzahl der Frame-Zeilen und deren Höhe innerhalb eines Frameset.
	frameborder	Legt fest, ob im Frameset-Rahmenlinien zwischen den Frames angezeigt werden sollen.
<frame>	BORDERCOLOR	Gibt die Farbe der Rahmen in einem Frameset an.
		Legt den Inhalt eines Frame innerhalb eines Frameset fest.
	src	Die URL des Dokuments, das innerhalb des Frame dargestellt werden soll.
	marginwidth	Die Breite des Randes links und rechts vom Frame, angegeben als Anzahl der Pixel.
	marginheight	Die Breite des Randes ober- und unterhalb des Frame-Inhalts, gemessen in Pixeln.
	scrolling	Schaltet das Display von Rollbalken für einen Frame ein oder aus. Erlaubte Werte sind yes, no oder auto.
	noresize	Verhindert, dass der Anwender die Frame-Größe verändern kann.
	frameborder	Legt fest, ob im Frame Rahmenlinien zwischen den Frames angezeigt werden sollen.

Tabelle 12.4: Die neuen Tags von Tag 12 im Überblick

Kennung	Attribut	Beschreibung
<iframe>	bordercolor	Gibt die Farbe der Rahmen in einem Frameset an.
	longdesc	Spezifiziert die URL, die eine längere Beschreibung des Frameset-Inhalts liefert. Wird von nicht visuellen Browsern verwendet.
	name	Weist dem Frame einen Namen zu, um bestimmte Ziele verfolgen zu können.
		Definiert einen Inline Frame.
	src	Gibt die URL des HTML-Dokuments an, das in dem Frame angezeigt werden soll.
	name	Legt den Namen des Frame fest, der für Verknüpfungen bzw. Zielangaben verwendet werden kann.
	width	Legt die Breite des Frame in Pixeln fest.
	height	Legt die Höhe des Frame in Pixeln fest.
	marginwidth	Legt die Breite des Randes in Pixeln fest.
	marginheight	Legt die Höhe des Randes in Pixeln fest.
	scrolling	Gibt an, ob der Frame über Bildlaufleisten verfügen soll. Mögliche Werte sind yes, no oder auto.
	frameborder	Gibt an, ob der Frame einen Rahmen anzeigen soll. Mögliche Werte sind 1 und 0.
	bordercolor	Legt die Farbe des Rahmens fest. (Internet-Explorer-Erweiterung)
<noframe>		Erlaubt es, einen anderen Inhalt für Web-Browser zu definieren, die keine Frames unterstützen.

Tabelle 12.4: Die neuen Tags von Tag 12 im Überblick (Forts.)

Workshop

Wenn Sie es bis hierher im Buch geschafft haben, können Sie sich jetzt selber gratulieren. Mit dem Wissen, das Sie innerhalb der letzten Woche erworben haben, haben Sie so ziemlich alles getan, was mit einem einzelnen Computer noch möglich ist. Jetzt sind Sie bereit dazu, Ihre Webseiten aufs Internet zu laden und ihnen interaktivere Merkmale hinzuzufügen, wie zum Beispiel Formulare, Imagemaps und eingebettete Animationen. Morgen, am achten Tag, werden Sie genau damit beginnen.

Ein kleiner Kurs zum auffrischen gefällig? Als ob Sie für heute nicht schon genug hätten... Sie wissen ja schon wie unser Workshop abläuft: Fragen, Quiz und Übungen werden Ihnen helfen, die wichtigsten Punkte dieses Kapitels noch einmal in Erinnerung zu rufen.

Fragen und Antworten

F Gibt es eine Begrenzung dafür, wie viele Ebenen von `<frameset>`-Kennungen ich innerhalb eines einzigen Bildschirms verschachteln kann?

A *Nein, da gibt es keine Einschränkung. Praktisch gesehen wird jedoch die verfügbare Fenstergröße zu klein, wenn Sie mehr als vier Ebenen benutzen.*

F Was würde passieren, wenn ich innerhalb eines `<frame>`-Tag einen Bezug auf ein Frame-Definitionsdocument einfügte?

A *Netscape wird damit fertig, indem es das verschachtelte Frame-Definitionsdocument als verschachtelten Frameset behandelt. Diese Technik wird oft angewandt, um die Komplexität verschachtelter Frames zu reduzieren. Es gibt da jedoch eine Einschränkung. Sie können keinen Bezug auf das gegenwärtige Frame-Definitionsdocument von einem seiner eigenen Frames aus herstellen. Diese Situation wird Wiederholung genannt und verursacht eine Endlosschleife. Netscape Communications hat eine Sicherung gegen diese Art von Bezug eingebaut.*

Quiz

1. Was ist der Unterschied zwischen einem Frameset-Dokument, einem Frame und einer Seite?
2. Sie erstellen Links zu Seiten, die dann in ein Frameset geladen werden sollen. Welches Attribut lässt die Seiten im richtigen Frame erscheinen? (Kleiner Tipp: Das Attribut bezieht sich auf das `<a>`-Tag)
3. Wenn eine Webseite ein `<frameset>`-Element enthält, welches Element kann dann nicht am Anfang des HTML-Dokuments verwendet werden?
4. Welche zwei Attribute des `<frameset>`-Tag teilen das Browser-Fenster in viele Sektionen?
5. Welches Attribut des `<frame>`-Tag definiert das HTML-Dokument, das als Erstes in ein Frameset geladen wird?

Antworten

1. Ein Frameset-Dokument ist das HTML-Dokument, das die Definition des Framesets enthält. Ein Frameset ist der Teil des Frameset-Dokuments, der vom `<frameset>`-Tag definiert wird, welches dem Browser die Anweisung gibt, das Fenster in eine Vielzahl von Sektionen zu unterteilen. Ein Frame ist eine dieser Sektionen, oder Fenster, innerhalb eines Framesets. Die Seite ist das Web-Dokument, das in einen Frame geladen wird.
2. Das `target`-Attribut des `<a>`-Tags dirigiert die verwiesenen Seiten in dem für sie vorgesehenen Frame, in dem sie geladen werden sollen.
3. Wenn eine Webseite ein `<frameset>`-Element enthält, kann es das `<body>`-Element nicht am Anfang der Seite enthalten. Sie schließen sich gegenseitig aus.
4. Die `cols`- und `rows`-Attribute des `<frameset>`-Tag unterteilen das Browser-Fenster in eine Vielzahl von Frames.
5. Das `src`-Attribut des `<frame>`-Tag definiert das HTML-Dokument, das als Erstes in das Frameset geladen wird.

Übungen

1. Erstellen Sie wie folgt ein Frameset, das Browser-Fenster in drei Sektionen unterteilt:

Die linke Sektion des Framesets soll eine Spalte sein, die die gesamte Höhe und 1/3 der Breite des Browser-Fensters einnehmen wird. Nennen Sie diesen Frame `contents`.

Teilen Sie die rechte Sektion des Framesets in zwei Reihen, wobei jede Reihe die Hälfte der Höhe des Browser-Fensters einnehmen soll. Nennen Sie die obere Sektion `top` und die untere Sektion `bottom`.
2. Erstellen Sie für das in 1. erstellte Frameset eine Seite, die Sie für eine Inhaltstabelle im linken Frame verwenden werden. Geben Sie zwei Links in diese Seite, wobei eines die Seite in den oberen Frame und das andere die Seite in den unteren Frame laden soll.



**Multimedia:
Einbinden von
Sounds, Videos
und mehr**

**Woche
2**

Das Einbinden von Multimedia-Dateien in eine Webseite ist genauso einfach wie das Erstellen von Hyperlinks für Sound- oder Video-Dateien. Wie sagt man doch: Presto! Und schon haben Sie Multimedia in Ihre Web-Site integriert. Natürlich ist das noch nicht alles! Sie können nicht nur Multimedia-Dateien verknüpfen, sondern diese auch in Ihre Web-Sites einbetten. Leider könnte dieses Einbetten von Dateien etwas knifflig werden! Während Sie für das Verstehen von HTML nur einige Elemente kennen müssen, sind die Multimedia HTML-Elemente von einer Art Schizophrenie befallen. Sie werden beide jeweils verschieden im Microsoft Internet Explorer und im Netscape Navigator implementiert. Oft werden sie nur von einem oder sogar von keinem der beiden Browser unterstützt, oder sie sind Teil von HTML – Standards, die von diesen nicht als solche erkannt werden. Zusätzlich ist heute schon eine gewisse Auswahl an Audio- und Videoformaten erhältlich. Es ist fast unmöglich, alle Vor- und Nachteile eines jeden Formats zu lernen, bevor nicht schon neue Formate auf dem Markt erscheinen, die dann die perfekten Multimediaprodukte sein sollen.

Obwohl die Übertragungsgeschwindigkeiten heutzutage immer höher werden (fast alle Modems haben für gewöhnlich eine Durchschnittsgeschwindigkeit von 56 Kb pro Sekunde), die Sound- und Videokompression bzw. Dekompressionstechnologie ständig verbessert wird (denken wir nur an MP3 Audiodateien) und Audio- und Videoadapterkarten immer leistungstärker werden, zeichnet sich das Web – entgegen den Wunschvorstellungen der Multimedia-Befürworter – wohl nicht durch Sound- und Videoqualität aus – zumindest jetzt noch nicht.

Ein Grund, warum dieses Problem so schlecht in den Griff zu bekommen ist, ist die große Kluft zwischen dem, was wir von unseren Computern gewohnt sind und was wir vom Web erwarten. Legt man eine CD oder eine DVD in ein Laufwerk ein, dann springen einem regelrecht 3D-Graphiken, Stereo-surround Sounds und Full-screen Digitalvideos mit 30 Frames pro Sekunde entgegen, überfallen Hör- und Sehsinn und nehmen diese so lange ein, bis diese packenden Eindrücke schließlich nur noch als monotone Überhäufung von Multimedia-Eindrücken aufgenommen werden können. Vergleichen Sie das nun mit den meisten Multimediaanwendungen im Web, und Sie werden sehr enttäuscht sein. Sound von schlechter Qualität, kleine Videoformate und lange Downloadzeiten gehören im Web dazu.

Aber es gibt bereits Verbesserungen; Beispiel dafür sind das MP3 Audio-Format, Macromedia Flash Animationen und ihre wachsende Beliebtheit. Beide bieten Multimedia bei geringer Bandbreite mit höchster Qualität; dennoch muss für die erbrachten Fortschritte bezahlt werden. Als Web-Benutzer werden wir nämlich von Audio- und Videoformaten überrollt. Für jedes einzelne Format werden spezielle Plugins oder Hilfsprogramme benötigt. Als Web-Entwickler müssen Sie sich mit teurerem Audio- und Videoequipment und Software ausstatten, um Ihren eigenen Multimedia-Bestand aufzubauen.

Nachdem ich Ihnen dies nun alles erklärt habe, möchte ich versuchen, Ihnen in dieser Lektion sowohl die Techniken zu zeigen, die Sie sofort anwenden können als auch die Verfahrensweisen, für deren Verständnis Sie eine bedeutenden Zeit- und Energieaufwand in Kauf nehmen müssen. Sie werden Folgendes lernen:

- ▶ Die Erstellung von Links zu Audio- und Videodateien, damit diese von Besuchern heruntergeladen oder abgespielt werden können
- ▶ Die Verwendung vom `<embed>`-Element und vom `<object>`-Element zur Integration von Sound- und Video-Dateien in Webseiten
- ▶ Das Einbetten von QuickTime, Shockwave, Flash und RealAudio oder RealVideo-Dateien in Ihre Webseiten
- ▶ Das Verwenden von einigen speziellen Multimedia-Fähigkeiten vom Microsoft Internet Explorer
- ▶ Das Erkennen der verbreitetsten Multimedia Dateitypen und der notwendigen Plugins oder Hilfsprogramme

Präsentieren von Sound und Video

Die Komplexität von erhältlichen Multimedia-Dateien und Formaten ist sehr groß. Letztendlich läuft es darauf hinaus, eine Methode finden zu müssen, diese Multimedia-Dateien in Ihre Webseite zu integrieren. Sie können entweder Hyperlinks zu diesen Dateien erstellen oder sie direkt in Ihre Webseite integrieren. Die Dateienverknüpfung ist kinderleicht, aber das Einbetten von Dateien könnte problematisch werden. Eine verknüpfte Sound- oder Videodatei – es spielt übrigens keine Rolle, um was für eine Sound- oder Videodatei es sich dabei handelt – hat innerhalb der Webseite einen Hyperlink zur Quelldatei. Wenn man eine verknüpfte Sound- oder Videodatei anklickt, erscheint eine von drei Optionen. Als Erstes können Sie die Datei downloaden und auf Ihrem Computer speichern. Diese Methode ermöglicht es Ihnen, Ihre Datei zu einem späteren Zeitpunkt anzuhören oder anzusehen, ganz gleich in welcher Anwendung Sie sich gerade *befinden*. Zweitens kann die Datei herunter geladen werden während automatisch ein Hilfsprogramm oder ein Plugin zum Abspielen der Datei gestartet wird. Diese Vorgänge werden dann nicht mehr angezeigt, wenn die Datei identifiziert ist und ein entsprechender Player oder Plugin konfiguriert wurde, und so die Datei zum Laufen gebracht wird. Drittens: wenn die Audio- oder Video-Dateien als Streaming-Dateien identifiziert werden, wird zusätzlich ein Player gestartet, der die Datei während des Herunterladens startet.

Eingebettete Sound- und Video-Dateien werden im Web-Browser integriert und werden mit Hilfe von Plugins oder Hilfsprogrammen abgespielt. Eingebettete Sound- und Videodateien bieten dem Anwender weniger Optionen, da die Dateien direkt in der Webseite eingebettet sind. Sehr oft werden ein Hilfsprogramm oder Plugins im Browser des Web-Fensters angewendet, um die Datei abzuspielen.

Stürzen wir uns jetzt voller Elan in die Arbeit und bringen etwas Dynamik in Form von Sound und Video auf unsere Webseite.

Die altbewährte Methode: Linking

Die sicherste und einfachste Methode, Multimedia-Dateien auf ihre Webseiten zu integrieren ist das Erstellen eines Hyperlinks zu diesen Dateien, der von allen Browserversionen unterstützt wird. Man kann dann entscheiden, ob man diese Datei downloaden möchte, oder sie je nach Interesse anhören oder ansehen möchte.

Es ist heute üblich, einen Link zu einer Datei zu erstellen, dem Besucher eine kurze Vorschau auf das Minibild eines Medienclips zu ermöglichen, eine Beschreibung und die Dateigröße anzugeben, sozusagen als nette Geste, damit der Besucher die Ladezeit abschätzen kann. Sie sollten auch Links für alle Player erstellen, die für den Besucher interessant sein könnten. So kann sich der Besucher bei Bedarf den gewünschten Player downloaden. Wenn ich zum Beispiel der Öffentlichkeit ein QuickTime Video zur Verfügung stellen wollte, so würde ich folgenden Code erstellen:

```
<div align="center">
<h1>Apollo 17 Videos</h1>
<p><a href="apollo_17_flag.qt">Astronauts placing the flag on the Moon</a><br>
[2.75Mb]</p>

<p>Apple <a href="http://www.apple.com/quicktime">QuickTime</a> is required
to view this movie.
<a href="http://www.apple.com/quicktime"></a></p>
</div>
```

In Abbildung 13.1 sehen Sie die fertiggestellte Webseite.

Ihre Webseite wird attraktiver, wenn Sie viele verschiedene Mediendateien zum Herunterladen anbieten, und die Besucher Ihrer Webseite nach ihren Präferenzen auswählen können.

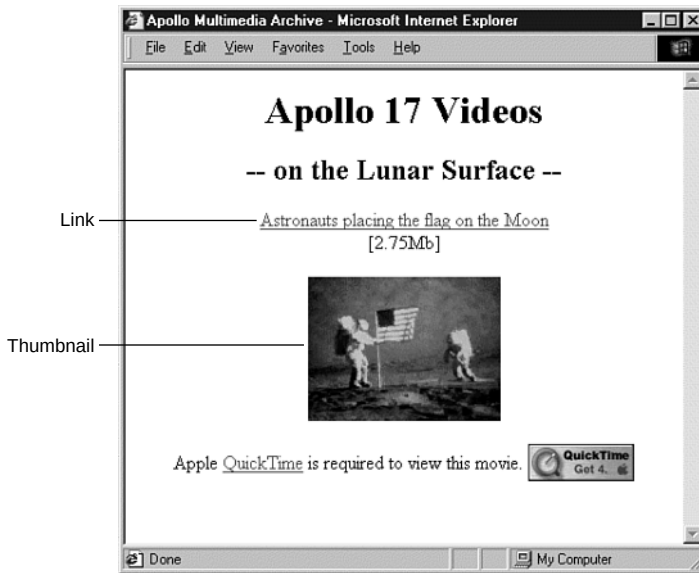


Abbildung 13.1:
Dateiverknüpfungen sind einfach und effektiv.

```
<html>
<head><title>Apollo Multimedia Archive</title></head>
<body>
<div align="center">
<h1>Apollo 17 Videos</h1>
<p>Astronauts placing the flag on the Moon</p>
<table border="0">
  <tr>
    <td rowspan="3"></td>
    <td><a href="apollo_17_flag.qt">QuickTime</a> [2.75Mb]</td>
  </tr>
  <tr>
    <td><a href="apollo_17_flag.mpg">MPEG</a> [2.45Mb]</td>
  </tr>
  <tr>
    <td><a href="apollo_17_flag.avi">AVI</a> [3.11Mb]</td>
  </tr>
</table>
<br />
<a href="http://www.apple.com/quicktime">
</a>
<br>
<a href="http://microsoft.com/windows/mediaplayer/download/default.asp">
</a>
<br />
</div>
</body>
</html>
```

In Abbildung 13.2 sehen Sie die fertiggestellte Webseite.

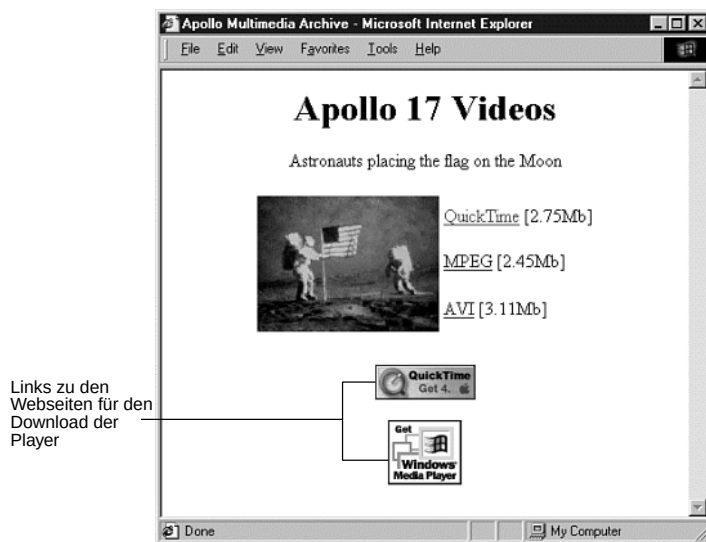


Abbildung 13.2:
Bieten Sie bei
verknüpften Sound-
und Video-Dateien
möglichst mehrere
Formate an.

Übung 13.1: Erstellen eines Multimedia-Familienalbums

Eine sehr häufig anzutreffende Art von Webseiten ist ein *Medienarchiv*. Ein Medienarchiv ist eine Webseite, die einfach nur einen schnellen Zugang zu Bildern oder anderen Mediendateien verschafft, die man ansehen oder downloaden kann.

Bevor das Web populär wurde, wurden Mediendateien wie Bilder, Sounds und Video in FTP (File Transfer Protocol) oder Gopher -Archiven gespeichert. Da diese Archivdateien aus textorientierten Oberflächen bestehen, hat man oft Probleme, das zu finden, was man sucht, denn der Dateiname ist die einzige Information über den Dateinhalt. Sogar sinnvolle, aussagekräftige Dateinamen wie *the-trees-last-fall.gif* oder *ave-maria.waw.* sind von keiner großen Hilfe, wenn es um Bilder oder Sounds geht. Obwohl diese Dateinamen den Dateinhalt beschreiben, kann man sie nur dann sampeln, wenn man die ganze Datei herunterlädt und abspielt.

Wenn Sie Inline Bilder als Minibilder anwenden und Sound- oder Videodateien in kleine »Sample«-Clips mit größeren Dateien aufsplitten, können Sie im Web ein Medienarchiv erstellen, das um einiges nützlicher ist als jede Art von Volltextarchiv.



Denken Sie daran, dass diese Art von Archivdatei, die voller Inline-Graphiken und großen Mediendateien ist, optimal für Graphikbrowser, die an schnelle Netzwerke angeknüpft sind, ist.

In dieser Übung werden Sie ein einfaches Beispiel für ein Medienarchiv mit verschiedenen GIF (Graphic Interchange Format) – und JPEG (Joint Photographic Experts Group) – Bildern, WAV Sounds und einer Mischung von MPEG (Moving Pictures Expert Group) und AVI (Audio Video Interleave) Videos erstellen.

Verwenden Sie Ihr bevorzugtes Bildbearbeitungsprogramm, um Minibilder von jedem einzelnen Ihrer Bilder erstellen zu können. Diese Minibilder dienen als Inline Icons. Sie können dann diese Bilddateien an der gewünschten Stelle in die Archivdatei über das `<img>`-Tag einfügen.

Beginnen Sie zuerst mit der Framework für die Archivdatei und fügen Sie dann eine Tabelle für die Minibilder wie folgt hinzu:



```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>
<head>
<title>My Family History</title>
</head>
<body>
<h1>My Family Media Archive</h1>
<div align="center">
<table border="0">
<tr>
<td width="80"><h2>Images</h2></td>
<td><p>Select an image to view it in a larger size</p></td>
</tr>
<tr>
<td width="80">A bunch of family members in the early 1950s.</td>
<td></td>
</tr>
<tr>
<td width="80">Aunts Betsy and Phyllis sitting on the porch.</td>
```

```
<td></td>
</tr>
<tr>
  <td width="80">Don when he was a child.</td>
  <td></td>
</tr>
</table>
</div>
</body>
</html>
```

Bitte beachten Sie, dass ich dem `<img>`-Tag Werte für das `alt`-Attribut hinzufüge, die zur Ersetzung der Bilder dienen, die vom Browser nicht erkannt werden können. Obwohl Sie natürlich nicht davon ausgehen sollten, dass Ihr Browser sich als ungeeignet entpuppt, was die Erstellung der Graphik auf Ihrer Webseite betrifft, sollte man aber schließlich, sozusagen vorbeugend denjenigen einen Tipp geben, die trotzdem über dieses Problem stolpern.

In Abbildung 13.3 sehen Sie, wie die Seite bis jetzt aussieht.



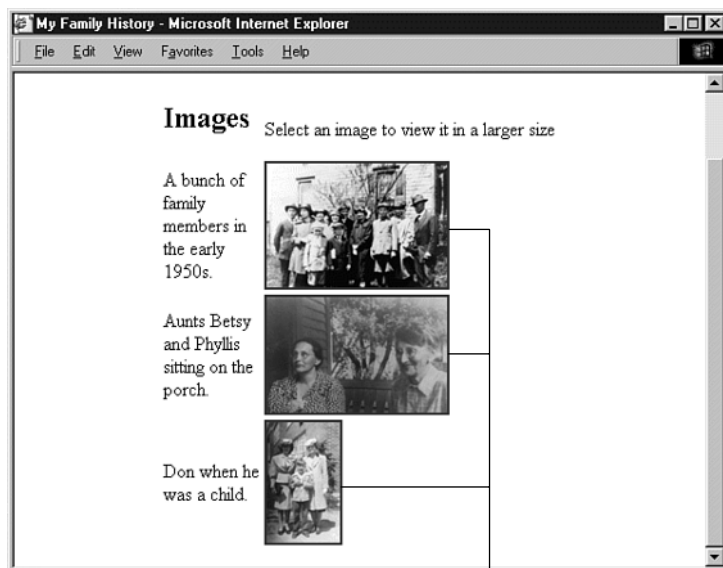
Abbildung 13.3:
Die fast fertig
gestellte Web-
seite mit dem
Bildarchiv

Der nächste Schritt besteht darin, Hyperlinks zu jedem Bild zu erstellen, das auf die größere Datei verweist. Wenn man diese Hyperlinks anklickt, kann man sich aussuchen, ob man die Datei nur ansehen möchte oder sie lieber downloaden möchte.



```
...
<tr>
  <td width="80">A bunch of family members in the early 1950s.</td>
  <td><a href="groupoldlarge.jpg">
    </a></td>
</tr>
<tr>
  <td width="80">Aunts Betsy and Phyllis sitting on the porch.</td>
  <td><a href="auntslarge.jpg"></a></td>
</tr>
<tr>
  <td width="80">Don when he was a child.</td>
  <td><a href="donoldlarge.jpg"></a></td>
</tr>
```

In Abbildung 13.4 sehen Sie das Ergebnis.



Bilder werden als Links verwendet

Abbildung 13.4:
Das Bild ist jetzt
mit einem größe-
ren Bild ver-
knüpft. Minibil-
der stellen die
Links dar.

Wenn ich das fertiggestellte Archiv jetzt betrachte, kann ich feststellen, dass es gut aussieht, aber ich breche dabei eine meiner eigenen Regeln: ich habe noch nicht auf die Größe der Datei verwiesen. Ihnen stehen hier mehrere Formatierungsmöglichkeiten zur Verfügung, aber das Problemloseste ist, auf die Dateigröße ganz einfach wie folgt nach jeder Bilderklärung hinzuweisen:



```
<tr>
  <td width="80">A bunch of family members in the early 1950s. [103k]</td>
  <td><a href="groupoldlarge.jpg"></a></td>
</tr>
<tr>
  <td width="80">Aunts Betsy and Phyllis sitting on the porch. [83k]</td>
```

```
<td><a href="auntslarge.jpg"></a></td>
</tr>
<tr>
<td width="80">Don when he was a child. [284k]</td>
<td><a href="donoldlarge.jpg"></a></td>
</tr>
```

In Abbildung 13.5 sehen sie das Ergebnis

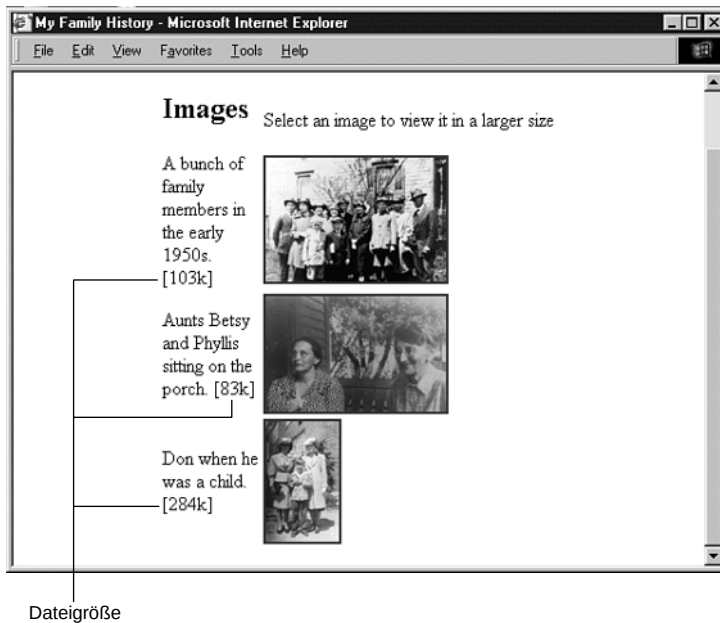


Abbildung 13.5:
Dateigrößen-
Angaben zu
jeder Bildbe-
schreibung
ermöglichen
Besuchern Ihrer
Webseite das
Einschätzen der
Ladezeit.

Kommen wir nun zu den Bereichen Sound und Video. Es gibt zwei Formatierungsmöglichkeiten für diese beiden Bereiche. Sie können das Material derselben Tabelle hinzufügen, die auch die Bilder enthält. Sie können aber auch zwei einzelne Tabellen erstellen- beide Möglichkeiten sind gleich gut. In folgender Übung werden Sie nun einzelne Tabellen erstellen, die virtuell identisch zu der Tabelle sind, in der die Bilder enthalten sind.

Beginnen Sie nun Ihre Arbeit, indem Sie drei Sound- und zwei Videodateien hinzufügen. Da Sound-Dateien nicht auf einfache Minibilder reduziert werden können, müssen Sie diese Dateien im Archivtext ausführlich beschreiben. Bei Video-Dateien können Sie im Normalfall mit Hilfe Ihres Video Players ein Frame des Clips kopieren und es als Minibild darstellen. Der Code für das Einfügen einer Sound-Datei in Ihrem Archiv ist wie folgt:



```
<div align="center">
<table border="0">
<tr>
  <td width="150"><h2>Sound Bites</h2></td>
  <td><p>Select a sound bite to download or listen to it.</p></td>
</tr>
<tr>
  <td width="150">An oral family history describing how we survived the tornado of
1903. [1192k]</td>
  <td><a href="tornado.wav"></a></
td>
</tr>
<tr>
  <td width="150">Don describing his first job. [1004k]</td>
  <td><a href="donjob.wav"></a></
a></td>
</tr>
<tr>
  <td width="150">Grandma Jo telling how she came to America. [2459k]</td>
  <td><a href="jo.wav"></a></a></
td>
</tr>
</table>
</div>
```

In Abbildung 13.6 sehen Sie, wie die verknüpften Sound-Dateien auf Ihrer Webseite aussehen.

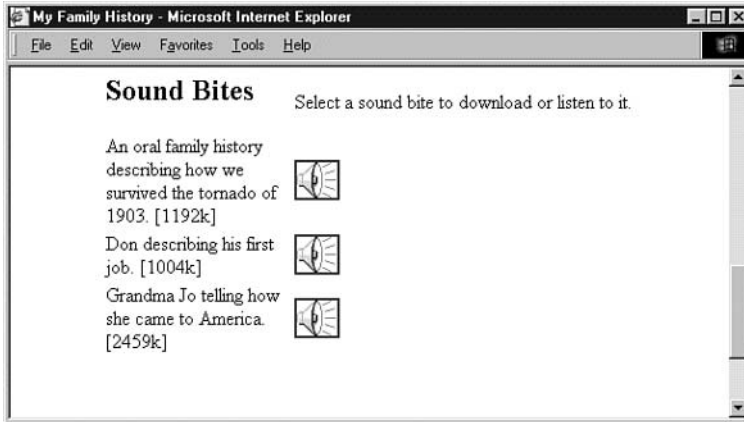


Abbildung 13.6:
Verknüpfte
Sound-Dateien
in einem Multi-
media-Familien-
album.

Fügen Sie zu guter Letzt nach demselben Schema die Videoclip-Dateien hinzu. Sie sehen, es geht schon viel einfacher!



```
<div align="center">
<table border="0">
<tr>
  <td width="100"><h2>Video Clips</h2></td>
  <td><p>Select a video clip to download it.</p></td>
</tr>
<tr>
  <td width="100">A video of a family wedding. [2492k]</td>
  <td><a href="wedding.mpeg"></a></td>
</tr>
<tr>
  <td width="100">Don and Mary talking. [3614k]</td>
  <td><a href="donandmary.mpeg"></a></td>
</tr>
</table>
</div>
```

In Abbildung 13.7 sehen Sie, wie die verknüpften Video-Dateien und die Minibilder auf der Webseite aussehen.



Abbildung 13.7:
» Die Videoseite
in unserem
Multimedia-
archiv«

Et voilà, Ihr Medienarchiv ist fertig. Mit Inline-Bildern und externen Dateien können Sie ein Medienarchiv ganz einfach selbst erstellen. Mit Hilfe der `alt`-Dateiattribute kann man ein Medienarchiv auch ganz problemlos erstellen, wenn nur ein Textbrowser vorhanden ist.

Einbetten von Sound-und Video-Dateien

Sie können Sound-und Video-Dateien mit Hilfe der `<embed>`- oder `<object>`-Elemente einbetten. Denken sie daran; das Prinzip hinter dem Einbetten von Sound-oder Video-Dateien steckt in ihrer Integration auf einer Webseite, sodass diese Dateien Teil der Webseite sind.

Das `<embed>`-Element war vor einiger Zeit aktuell und man findet es auch heute immer wieder. Es wird sowohl vom Internet Explorer als auch vom Netscape Navigator unterstützt. Es wurde so entwickelt, dass Dateitypen, die ohne Plugins (vor allem bei Multimedia) nicht funktionieren, auf Webseiten hinzugefügt werden konnten. Trotz dieses Vorteils wird `<embed>` vom World Wide Web Consortium (W3C) nicht unterstützt und man findet es auch nicht im offiziellen HTML-Standard. Da beide Browser das Element unterstützen, können Sie natürlich in Zukunft einfach außer Acht lassen, dass W3C dieses Tag nicht unterstützt.

Das »Konkurrenzelement« `<object>` wird offiziell vom W3C unterstützt, obwohl Browserunterstützung in unserer heutigen Zeit (vor allem beim Navigator) eine heikle Angelegenheit ist. Das `<object>`-Element ist wohl eine gute Möglichkeit, jegliche Art von Dateitypen in Webseiten einzubetten, von Bildern zu Java Applets bis hin zu Sound- und Video-Dateien. Man hofft, dass durch die Einführung eines allgemeinen Tags das Einbetten einer großen Menge verschiedener Dateitypen ermöglicht werden kann, und dass man somit für die Multimediateypen der Zukunft gerüstet ist.

Verwendung des `<embed>`-Elements

Obwohl `<embed>` nicht im HTML-Standard enthalten ist, unterstützen Microsoft oder Netscape weiterhin `<embed>` in ihren neuesten Web-Browsern. `<embed>` wird auf vielen Webseiten benutzt.

Die Syntax für die Verwendung von `<embed>` ist einfach:

```
<embed attributes />
```

Bitte beachten Sie die das abschließende Tag, das fakultativ ist.

Obwohl Microsoft und Netscape beide `<embed>` unterstützen, teilen sie leider nur eine Handvoll gemeinsamer Attribute. Der Vorteil dieser ganzen Sache aber ist, dass jeder Web-Browser die Attribute, die er nicht versteht, ignoriert, und es Ihnen dann offen steht, so viele Attribute wie Sie möchten, hinzuzufügen. Deshalb sollten Sie sich am besten eine ganze Reihe von Attributen beschaffen, die in jedem Fall funktionieren. Sie sollten diese Attribute auch gewissenhaft anwenden, während Sie die anderen Attribute zu »added value« hinzufügen.

Sehen wir uns einmal die Attribute an, die Sie unbedingt für die Verwendung des `<embed>`-Elements benötigen.

```
<embed src="a01607av.avi" height="120" width="160" />
```

Das `<src>`-Attribut gibt den Pfad und den Namen der Mediendatei an, die Sie in Ihre Webseite einbetten möchten, während die `height`- und `width`-Attribute die Größe im Browser-Fenster festlegen, in der die Mediendatei erscheinen soll.

Im Internet Explorer und im Netscape Navigator werden `hight` und `width` beide jeweils unterschiedlich angezeigt. Im Internet Explorer 5 werden die Media-Controls der entsprechenden Plugins immer angezeigt, während sie im Netscape Navigator nie angezeigt werden. Wenn im Internet Explorer die `hight`- und `width`-Attribute festgelegt werden, so erhält man ein Video, das sich auch über den Bereich erstreckt, der für die Controls bestimmt ist. Das Video verdeckt also die Controls, und das ist äußerst problematisch. Das Festlegen von `hight`- und `width` auf die exakte Größe des Video-Anzeigebereichs lässt das Video gestaucht aussehen, da die Controls Teile des Videos einfach einnehmen. Manch einer möchte nun glauben, die Vergrößerung von `hight`- und `width` würde das Problem lösen; es würde mehr Platz geben, und so könnten die Controls und das Video in ihrer eigenen, richtigen Größe angezeigt werden. Netscape Navigator lässt dies aber nicht zu. Im Netscape Navigator werden die Videos nach den Werten der `hight`- und `width`-Attribute angezeigt, die Controls werden vollkommen ignoriert und das Ergebnis ist ein gedehntes, verzerrtes Bild.

In Abbildung 13.8 und 13.9 sehen Sie das Problem mit folgendem Code dargestellt:

```
<embed src="a01607av.avi" type="video/x-msvideo" height="120" width="180" />
```



Abbildung 13.8:
Der Internet Explorer zeigt die Controls und das Video im richtigen Größenverhältnis zur Box an.

Es funktioniert, wenn Sie die `hight`- und `width`-Attribute nicht verwenden und den Browsern die Arbeit überlassen. Allerdings tut er dies nur im Internet Explorer. Der Netscape Navigator zeigt ein kleines Fenster für das Plugin an, das viel vom Video abschneidet.

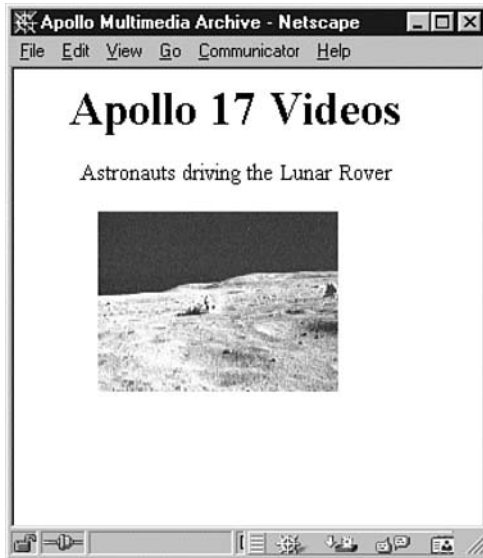


Abbildung 13.9:
 Der Netscape Navigator ignoriert die Controls und dehnt die Videos, damit sie in die Box passen.

Was kann man also tun? Außer Ihre Hände verzweifelt über den Kopf zusammenzuschlagen, genervte Briefe an sämtliche Firmen zu schreiben oder Ihren Computer aus dem Fenster zu werfen, sollten Sie lieber noch folgende zwei Dinge tun:

- ▶ Seien Sie einfach eiskalt und ignorieren Sie, wie Ihr Video in dem anderen Browser aussieht
- ▶ Verwenden Sie ein Script, um den Browsertyp des Besuchers zu ermitteln und verwenden Sie `<embed>`, dem Browser des Besuchers entsprechend.
- ▶ Benutzen Sie die `<object>`- und `<embed>`-Elemente für bestimmte Dateitypen, um browserübergreifende Unterstützung zu erhalten.

Tabelle 13.1 fasst die `<embed>`-Attribute zusammen, die vom Internet Explorer unterstützt werden.

Attribut	Beschreibung
align	Sie können gewünschte Elemente für Textabsätze auf Ihrer Webseite ausrichten. Folgende Angaben sind möglich: <code>absbottom</code> , <code>absmiddle</code> , <code>baseline</code> , <code>bottom</code> , <code>left</code> , <code>middle</code> , <code>right</code> , <code>texttop</code> und <code>top</code> .
alt	Der alternativ zur Grafik anzuzeigende Text.
class	Legt fest zu welcher Klasse in Bezug auf Cascading Style Sheets das Element gehört.

Tabelle 13.1: `<embed>`-Attribute, die im Internet Explorer verwendet werden

Attribut	Beschreibung
height	bestimmt die Höhe des Elements.
hspace	bestimmt den horizontalen Abstand zum Element.
id	Legt die ID (Identification) eines Elements fest.
name	Legt den Namen eines Elements fest.
src	Legt die Quelle der Mediendatei fest.
style	Legt die Style-Sheet-Eigenschaften fest.
title	Legt den Titel des Elements fest.
units	Definiert die Maßeinheit für die Attribute height oder width. Pixel sind die Standardeinheit für Maßeinheiten.
vspace	Bestimmt den vertikalen Abstand zum Element.
width	Bestimmt die Anzeigebreite des Elements.

Tabelle 13.1: *<embed>-Attribute, die im Internet Explorer verwendet werden (Forts.)*

In Tabelle 13.2 sind alle Attribute zusammengefasst, die im Netscape Navigator verwendet werden.

Attribut	Beschreibung
src	Bezeichnet den Datei-Speicherort
type	Bezeichnet den Mime-Type der eingebundenen Datei.
pluginspace	Bestimmt eine URL im Internet, die Installationshinweise für ein Plugin enthält, das die eingebundene Multimedia-Datei anzeigen bzw. abspielen kann.
pluginurl	Die URL der Java-Archivdatei (JAR).
align	Richtet auf der Webseite aus. Folgende Angaben sind möglich: left, right, top und bottom.
border	Bestimmt die Breite des Rahmens eines Elements.
frameborder	Falls der Wert dieses Attributs no ist, wird kein Rahmen um das Element herum angezeigt.
height	Bestimmt die Höhe des Elements.
width	Bestimmt die Breite des Elements.
units	Definiert die Maßeinheit für die Attribute height oder width. Pixel sind die Standardeinheit für Maßeinheiten.
hidden	Bei true wird ein Element nicht angezeigt, bei false wird es angezeigt. false ist die Standardeinstellung.
hspace	Bestimmt den horizontalen Abstand zum Element.

Tabelle 13.2: *<embed>-Attribute, die im Netscape Navigator verwendet werden*

Attribut	Beschreibung
<code>vspace</code>	Bestimmt den vertikalen Abstand zum Element.
<code>name</code>	Bezeichnet das für die Datei benötigte Plugin.
<code>palette</code>	Ist nur unter Windows relevant. Mit <code>foreground</code> benutzt das Plugin die Vordergrundpalette, während <code>background</code> das Plugin dazu zwingt, die eingestellten Hintergrundfarben zu benutzen.

Tabelle 13.2: `<embed>`-Attribute, die im Netscape Navigator verwendet werden (Forts.)

Zu diesen Attributen gibt es zusätzliche Attribute für spezifische Plugins, wie den Macromedia Flash-Player.

Schließlich können Sie das `<noembed>`-Tag hinzufügen, um die Besucher zu unterstützen, die keinen Web-Browser besitzen, der Plugins anzeigen kann.

```
<noembed>This Web page requires a web browser that can display objects.</noembed>
<embed src="a01607av.avi" height="120" width="160" />
```

Verwendung des `<object>`-Elements

Dem World Wide Web Consortium zufolge sollten Sie das `<object>`-Element benutzen, wenn Sie Sound und Videos (und anderes) in Ihre Webseite einbetten möchten. Das könnte allerdings etwas problematisch werden, denn der Netscape Navigator unterstützt `<object>` nicht voll und der Internet Explorer ist wohl etwas launisch, wenn es darum geht, `<object>` zu unterstützen.

Um das `<object>`-Element zu verwenden, versehen Sie `<object>` zuerst mit dem einleitenden Tag und fügen Sie folgende Attribute hinzu:

```
<object data="movie.mpeg" type="application/mpeg">
```

Das `data`-Attribut zeigt die Quelldatei für Ihren Sound oder Video an, und `type` gibt den MIME-Typ der Datei an.

Als Nächstes fügen Sie den Inhalt ein, den Sie gerne anzeigen möchten, wie zum Beispiel eine Überschrift, und fügen Sie ein abschließendes `<object>`-Element hinzu.

```
<object data="movie.mpeg" type="video/mpeg">
My homemade movie.

</object>
```

Sie können auch Objekte überlappend anordnen, sodass, wenn eines der Objekte nicht angezeigt werden kann, der Browser die ganze Liste durchsucht.

```
<object data="movie.mpeg" type="video/mpeg">
  <object data="moviesplash.gif" type="image/gif">
    </object>
  My homemade movie.
</object>
```

`<object>` kann auch mit Hilfe des `<param>`-Tag jegliche Art von Parameter initialisieren, das für die eingebettete Datei benötigt werden könnte. Das `<param>`-Tag wird im Rumpf des `<object>`-Elements integriert und hat, wie man im Folgenden sieht, kein abschließendes Tag:

```
<object data="movie.mpeg" type="video/mpeg">
  <param name="height" value="120" valuetype="data">
  <param name="width" value="160" valuetype="data">
  My homemade movie.
</object>
```

Dieser Code legt Höhe und Breite des eingebetteten Objekts in 120×160 Pixel fest.

Parameter, die vom `<param>`-Tag initialisiert werden, sind vom Objekttyp abhängig, den Sie einbetten möchten.

Kombinieren von `<embed>` und `<object>`

Wie Sie in den nächsten Schritten sehen werden, können Sie `<embed>` und `<object>` oft gleichzeitig anwenden, damit eine Unterstützung vom Netscape Navigator und Internet Explorer gewährleistet werden kann. Dafür müssen Sie das `<object>`-Element mit allen erforderlichen Parametern erstellen (`<param>`), und dann das `<embed>`-Element hinzufügen. Dies müssen Sie jedoch tun, bevor Sie das abschließende `<object>`-Element bestätigen. Ein Ausschnitt aus dem entsprechenden Code zeigt, wie es funktioniert:

```
<object classid="value" codebase="value" height="480" width="512" name="myname">
  <param name="src" value="source location" />
  <embed src="filename" height="480" width="512" name="myname" />
</object>
```

Wenn der Internet Explorer die Webseite lädt, wird er das `<object>`-Element lesen und es zur Einbettung der Multimedia-Datei in die Seite interpretieren, während er das `<embed>`-Element völlig ignorieren wird. Der Netscape Navigator wird das `<embed>`-Element aber erkennen und verwenden.

Einbetten von Flash Animationen

Macromedia Flash lässt sich ziemlich einfach in Webseiten einbetten, da die Dateien innerhalb des Flash-PlugIns angezeigt werden. In Abbildung 13.10 sehen Sie die Macromedia Flash Oberfläche, die zur Erstellung von Flash-Dateien dient.

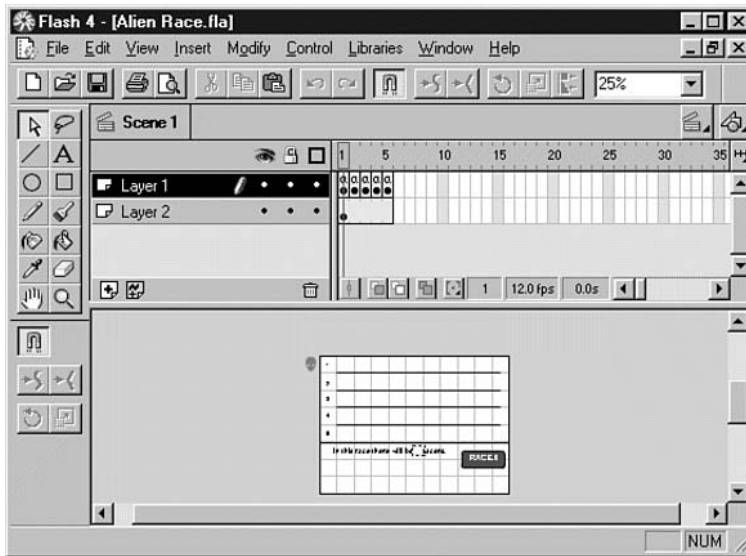


Abbildung 13.10:
Benutzen Sie
Macromedia
Flash, um Flash-
Dateien zu erstellen
und um diese
in Ihrer Webseite
zu speichern.

Flash verwendet HTML-Vorlagen, die entsprechend Ihren Vorgaben modifiziert werden. Sie sehen im folgenden Beispiel die Default-Schablone:

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://active.macromedia.com/flash2/cabs/swflash.cab#version=4,0,0,0"
id=$ti width=$wi height=$he>
$PO
<embed $pe width=$wi height=$he
type="application/x-shockwave-flash" pluginspage="http://www.macromedia.com/
shockwave/download/index.cgi?pl_prod_version=shockwaveflash">
</embed>
</object>
```

Bitte beachten Sie, dass Flash sowohl `<object>` als auch `<embed>` verwendet, um eine Animation in eine Webseite einzubetten. Die Dollarzeichen sind Variablen, die Flash dann durch Ihre Vorgaben ersetzt, wenn Sie die Datei veröffentlichen. Sie können die HTML-Einstellungen durch eine entsprechende Dialogbox modifizieren, wie Sie in Abbildung 13.11 sehen können.

Einbetten von Shockwave Animationen

Auch wenn Sie Macromedia Dreamweaver verwenden, der automatisch den entsprechenden HTML-Code in Ihre Webseite integriert, müssen Sie, wenn Sie die Shockwave-Datei einbetten möchten, den Code manuell eingeben.

Um eine Shockwave -Datei einzubetten, müssen Sie eine Kombination aus `<object>` (für Microsoft-Unterstützung) und `<embed>` (für Netscape-Unterstützung) verwenden. Um eine maximale Kompatibilität zu erhalten, sollten Sie beides verwenden.

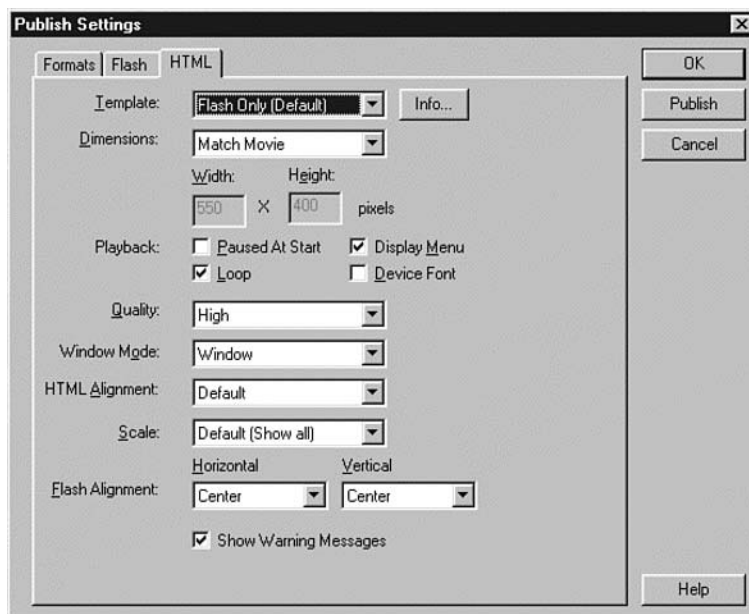


Abbildung 13.11: Flash ermöglicht es Ihnen, mittels HTML-Schablonen Animationen zu veröffentlichen. Diese Schablonen findet man in den Dialogboxen.

Für die meisten Shockwave-Anwendungen, können Sie den Code wie folgt eingeben und einfach Ihre eigenen Angaben zur *Location* und Anzeigengröße des Movies einsetzen. Für andere eingebettete Objekte sind womöglich andere Informationen erforderlich. Sie sollten sich beim Produkthändler oder -hersteller über die exakten Parameter informieren.

Der Quellcode für das Einbetten von Shockwave-Dateien sieht folgendermaßen aus:

```
<object classid="clsid:166b1bca-3f9c-11cf-8075-44553540000" codebase="http://
download.macromedia.com/pub/shockwave/cabs/director/sw.cab#version=7,0,0,0"
width="512" height="480" name="movienamen">
<param name="src" value="mymovie.dcr">
<embed src="mymovie.dcr" height=480 width=512 name="movienamen">
</object>
```

In Tabelle 13.3 sehen Sie die Attribute, die Sie im `<object>`-Element verwenden können:

Attribute	Beschreibung
classid	Ist der universelle Klassenbezeichner für das Shockwave Active X-Control. Die Angabe besteht aus folgender Zeichenfolge: <code>clsid:166b1bca-3f9c-11cf-8075-44553540000</code>
codebase	Gibt die URL an, von der das Shockwave Active X-Control installiert werden kann, wenn es nicht installiert ist. Die Angabe besteht aus folgender Zeichenfolge: <code>http://download.macromedia.com/pub/shockwave/cabs/director/sw.cab#version=7,0,0,0</code>
width	Bestimmt die Anzeigenbreite der Shockwave-Datei (in Pixeln)
height	Bestimmt die Anzeighöhe der Shockwave-Datei (in Pixeln)
name	Bestimmt den Namen des Shockwave Movies (Text)
src	Diese Attribut wird in einem <code><param></code> -Tag innerhalb eines <code><object></code> -Elements verwendet. <code>name</code> sollte <code>src</code> sein und das <code>value</code> sollte auf die URL des Movies verweisen. Der Code ist wie folgt: <code><param name="src" value="URL"></code>
bgcolor	Bezeichnet die Farbe für den Bildschirmhintergrund des Anzeigenfensters, wenn kein Movie dargestellt wird (der Wert wird mit Hexadezimal-Ziffern definiert).
swmodifyreport	Ist dieser Wert <code>true</code> , dann wird die <code>src</code> -URL aus der Statistik von Shockwave entfernt. Standardwert ist <code>true</code> .

Tabelle 13.3: `<object>`-Attribute

In Tabelle 13.4 sind die `<embed>`-Attribute aufgelistet.

Attribut	Beschreibung
width	Bezeichnet die Anzeigenbreite der Shockwave-Datei (in Pixeln)
hight	Bezeichnet die Anzeighöhe der Shockwave-Datei (in Pixeln)
name	Bestimmt den Namen des Shockwave Movies (Text)
src	Bezeichnet die URL des Movies.
pluginspage	Kann nur beim Netscape Navigator angewendet werden. Url des Shockwave Plugins. Der Wert sollte wie folgt sein: <code>http://www.macromedia.com/shockwave</code>

Tabelle 13.4: `<embed>`-Attribute

Attribut	Beschreibung
bgcolor	Bezeichnet die Farbe für den Bildschirmhintergrund des Anzeigenfensters, wenn kein Movie dargestellt wird (der Wert wird mit Hexadezimal-Ziffern definiert).
swmodifyre- port	Ist dieser Wert true, dann wird die src-URL aus der Statistik von Shockwave entfernt. Standardwert ist true.

Tabelle 13.4: <embed>-Attribute (Forts.)

Shockwave-Dateien werden vom Macromedia Director erstellt (siehe Abbildung 13.12).

Einbetten von RealAudio und RealVideo

RealNetwork, RealAudio und RealVideo-Dateien werden auch über <object>- und <embed>-Elemente eingebunden. Über die folgende Syntax können Sie diese Dateien einbetten.

```
<object id="rvocx" classid="clsid:cfcdaa03-8be4-11cf-b84b-0020afb3ccfa"
width="300" height="134">
Optional parameters
<embed src="source" width="value" height="value" />
<noembed><a href="download page">Play with RealPlayer.</a></noembed>
</object>
```

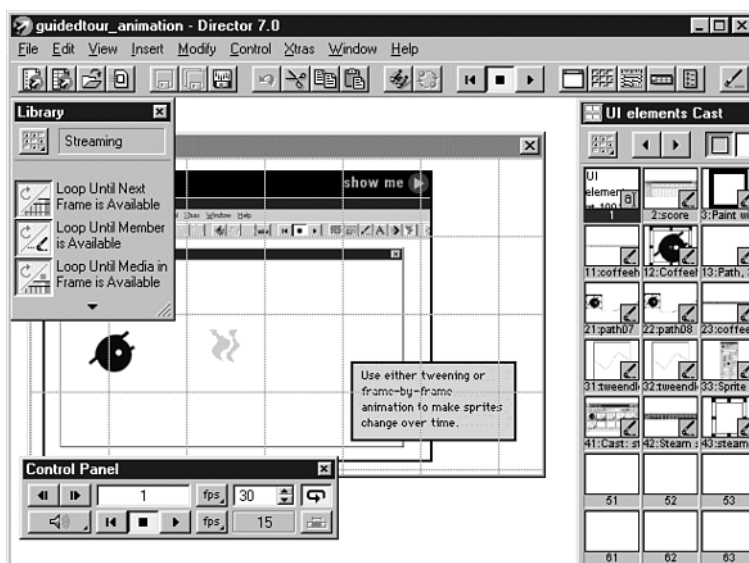


Abbildung 13.12: Macromedia Director 7 kann Movies als Shockwave-Dateien speichern.

In Tabelle 13.5 sind die verfügbaren Attribute für `<embed>` und die Parameter (`<param name="name" value="value" />`) für das `<object>`-Element aufgelistet.

Attribut/ Parameter	Beschreibung
autostart	Erzwingt das sofortige Abspielen einer Datei
backgroundcolor	Bezeichnet die Hintergrundfarbe (Hexadezimalfarbe in Werten oder Namen)
center	Zentriert einen Clip im Fenster (true oder false)
console	Verknüpft mehrere Controls (yes, name, _master oder _unique).
controls	Fügt Realplayer-Bedienelemente hinzu (Name des Bedienelements).
height	Legt die Fenster bzw. Controlhöhe fest (in Pixel oder Prozent).
loop	Erzwingt die endlose Wiederholung einer abspielbaren Datei (true oder false).
maintainaspect	Erhält das Seitenverhältnis der Datei
nojava	Verhindert das Starten der Java Virtual Machine (true oder false)
nolabels	Unterdrückt Präsentationsinformation (true oder false)
nologo	Unterdrückt die Anzeige des Reallogo (true oder false)
numloop	Legt fest wie oft eine Datei abgespielt wird (ein Zahlenwert).
region	Bindet einen Clip an eine SMIL-Region (Synchronized Multimedia Integration Language).
shuffle	Ermöglicht eine zufallsgesteuerte Wiedergabe (true oder false).
src	Legt den Quellclip fest (URL).
width	Legt die Breite des Fenster bzw. des Controls (in Pixel oder Prozent) fest.

Tabelle 13.5: `<embed>`-Attribute und `<object>`-Parameter

Sie können auch den Webseitenassistenten Wizzard in RealProducerPlus benutzen, um automatisch Webseiten mit eingebettetem RealMedia zu erstellen.

Die Multimedia-Techniken des Microsoft Internet Explorer

Der Microsoft Internet Explorer bietet einige besondere Fähigkeiten, die hier erwähnenswert sind: Background Sounds und Inline Video. Bitte beachten Sie trotzdem, dass, auch wenn der Netscape Navigator keine dieser Techniken unterstützt, ein Einbetten dieser Techniken in Ihre Webseite möglich ist. Der Netscape Navigator ignoriert Background Sounds und verschlüsselt Inline Videos so, dass der Navigator anstelle eines Videos ein statisches Bild anzeigt.

Hinzufügen von Background Sounds

Der Internet Explorer besitzt ein Element, das Audio-Dateien im Hintergrund lädt und abspielt. Diese Sound-Dateien werden heruntergeladen, wenn die Seite vom Web-Browser des Besuchers heruntergeladen wird. Sie werden abgespielt, ohne dass der Benutzer etwas Spezielles tun muss, wie zum Beispiel einen Audio-Player selbst zu starten. Da kein visueller Effekt entsteht, gibt es auch keinen Hinweis auf den gerade abspielenden Sound, es sei denn, ein Benutzer verwendet eine Soundkarte und hat die Lautstärke entsprechend hoch eingestellt. Um einen eingebetteten Background-Sound einer Seite hinzuzufügen, benutzen Sie das `<bgsound>`-Tag:

```
<bgsound src="elevatormusic.wav" />
```

Benutzen Sie das `LOOP`-Attribut, um den Sound endlos oft zu wiederholen. Wenn der Wert von `LOOP` in einer bestimmten Zahl ausgedrückt ist, richtet sich die Anzahl der Wiederholungen nach dieser Zahl. Wenn `LOOP=-1` oder `infinite` ist, wird der Sound sooft wiederholt, bis der Betrachter die Seite verlässt.

```
<bgsound src="elevatormusic.wav" loop="-1" />
```

Der Explorer unterstützt drei verschiedene Inline-Sound-Formate: das bekannte AU-Format von Sun, Windows WAV-Dateien, und MIDI-Dateien mit MID -Erweiterung.

Wie beim `dynsrc`-Attribut, über das wir übrigens gleich sprechen werden, wird das `<bgsound>`-Tag von Netscape-Browsern nicht unterstützt.

Inline-Video mit `<dynsrc>`

Sie können Videoclips (AVI oder MPEG) auf eine Webseite, die im Microsoft Internet Explorer 4 oder neueren Versionen angezeigt werden, integrieren, indem Sie das `dynsrc`-Attribut im `<img>`-Element wie im folgenden Beispiel benutzen:

```

```

Diesem Code zufolge spielt der Internet Explorer nach Beendigung des Ladevorgangs auf der Webseite den durch das `dynsrc`-Attribut angegebenen Videoclip zweimal ab. Das `loop`-Attribut gibt die Anzahl der Abspiel-Wiederholungen des Videoclips an, wobei das einmalige Abspielen des Videoclips ein Default-Wert ist. Wenn Sie den Clip unendlich oft wiederholen möchten, geben Sie stattdessen `-1` ein. Das `start`-Attribut gibt den Abspielzeitpunkt des Videoclips an. Sie können zwischen `fileopen`, dem Default-Wert oder `mouseover` wählen; letzteres spielt das Video dann ab, wenn sich die Maus im Anzeigebereich des Videos befindet.

Da Sie das `<img>`-Element verwenden, können sie zur Formatierung Ihres Videoclips auch andere `<img>`-Attribute benutzen, wie zum Beispiel `alt`, `align`, `border`, `height`, `wight`....

Damit eine Kompatibilität zum Netscape Navigator erstellt werden kann, sollten Sie das `src`-Attribut verwenden, um ein statisches GIF oder JPG-Bild zu erhalten, das anstelle eines Videos angezeigt wird. Der Code würde wie folgt aussehen:

```

```

Der Internet Explorer ignoriert den Wert des `src`-Attribut solange das Video von `dynsrc` unterstützt wird.

Übung 13.2: Einbetten eines QuickTime-Movies

In der zweiten Übung werden sie nun versuchen, QuickTime- Movies in Ihre Webseite einzubetten. QuickTime ist ein von Apple entwickeltes Format. Videos dieses Formats sind vor allem bei Macintosh-Benutzern relativ bekannt.

Apple zufolge sollten Sie das `<embed>`-Element benutzen, um QuickTime-Movies in Ihre Webseite einzubetten. Apple ist ein gutes Beispiel für ein Unternehmen, das einen Multimedia-Dateityp veröffentlicht, der Ihnen zusätzliche Attribute zur Verfügung stellt.

Der Code für eine einfache Webseiten-Schablone könnte wie folgt aussehen:

```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>
<head>
<title></title>
</head>
<body>
</body>
</html>
```

Geben Sie Ihrer Seite nun einen Titel und fügen Sie ein `<div>`-Tag hinzu, um die Elemente auf Ihrer Seite zentrieren zu können.

Fügen Sie als Nächstes eine Überschrift hinzu, die den Titel des Videos und das Video selbst ungefähr beschreiben.

```
<div align="center">
<h1>Apollo 17 Videos</h1>
<p>Astronauts placing the flag on the Moon</p>
</div>
```

Jetzt ist es an der Zeit, das Video selbst hinzuzufügen. Beginnen Sie mit dem `<embed>`-Element (stellen Sie es unter den Videotitel) und geben Sie die Quelle und die Anzeigengröße Ihres Videos an. Denken sie daran, diese Attribute sind notwendig.

```
<embed src="Apollo_17_Flag.qt"
width="160"
height="136"
```

In dieser Übung werden Sie eine ganze Reihe von Attributen hinzufügen (nähere Beschreibung der Attribute erfolgt später). Fügen Sie diese unter `height`, also der Höhe des Videos ein. Vergessen Sie nicht das abschließende `<embed>`-Element, nachdem Sie das letzte Attribut hinzugefügt haben.

```
autoplay="false"
controller="true"
kioskmode="true"
dontflattenwhensaving
pluginspage="http://www.apple.com/quicktime/download/" />
```

OK. Und um zu sehen, ob dies nun funktioniert, testen Sie dies mit Hilfe Ihres Web-Browsers. Wenn Sie mit dem Ergebnis zufrieden sind, fügen Sie ein letztes Element in Ihre Webseite ein, das anzeigt, wo das – eventuell benötigte – QuickTime-Plugin erhältlich ist. Da es sich um das letzte Element auf Ihrer Webseite handelt, vergessen Sie nicht das abschließende `<DIV>`-Tag.

```
<p>Apple <a href="http://www.apple.com/quicktime">QuickTime</a> is required to
view this movie. <a href="http://www.apple.com/quicktime"></
A></P>
```

Wenn nun alles kombiniert wird, sieht die Quelldatei für Ihre Webseite folgendermaßen aus:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">
<html>
<head>
<title>Apollo Multimedia Archive</title>
</head>
<body>
<div align="center">
<h1>Apollo 17 Videos</h1>
<p>Astronauts placing the flag on the Moon</p>
<embed src="Apollo_17_Flag.qt"
width="160"
height="136"
autoplay="false"
controller="true"
kioskmode="true"
dontflattenwhensaving
pluginspage="http://www.apple.com/quicktime/download/" />
```



```
<p>Apple <a href="http://www.apple.com/quicktime">QuickTime</a> is required to
view this movie. <a href="http://www.apple.com/quicktime"></
A></P>
</div>
</body>
</html>
```

In Abbildung 13.13 sehen Sie das Ergebnis im Internet Explorer

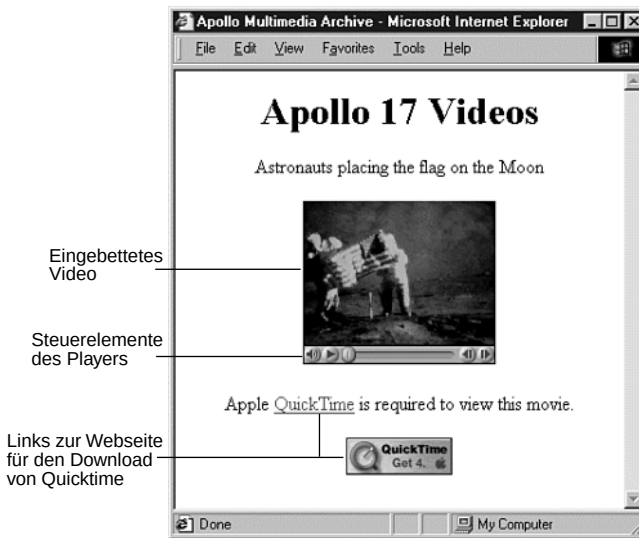


Abbildung 13.13:
Eingebettete QuickTime-Movies verwenden spezielle Attribute, die von Apple entwickelt wurden.

Tabelle 13.6 fasst die Attribute zusammen, die QuickTime4 unterstützt.

Attribut	Beschreibung
autoplay	Bei true wird das Movie abgespielt, wenn das Plugin den Clip so einschätzt, dass er ohne auf weitere Daten zu warten abgespielt werden kann (true oder false).
bgcolor	Bestimmt die Hintergrundfarbe der Bereiche, die nicht vom Movie abgedeckt sind. QuickTime4 akzeptiert die 16 HTML Farbbezeichnungen, die von W3C definiert werden (eine Hexadezimalzahl).
cache	Der Browser speichert das Movie im Cache (=Pufferspeicher). So kann der Browser ein Movie aus dem Cache abspielen, ohne es noch einmal downloaden zu müssen.

Tabelle 13.6: Attribute, die von QuickTime4 unterstützt werden

Attribut	Beschreibung
controller	Bei <code>true</code> werden die Bedienelemente für das QuickTime-PlugIn angezeigt. In diesem Fall sollten Sie, sofern Sie das <code>height</code> -Attribut einsetzen, zur Moviehöhe 16 Pixel hinzufügen. (<code>true</code> oder <code>false</code>).
correction	Ist nur bei QuickTime VR anwendbar (<code>none</code> oder <code>full</code>).
dontflattenwhensaving	Speichert das Movie ohne flimmern (keine Werte).
endtime	Definiert den letzten Frame des Videos (Zeit in Stunden:Minuten:Sekunden:Frames).
fov	Bestimmt den Initialwinkel für QuickTime VR -Movies (setzen Sie einen Wert zwischen 8 und 64 ein).
height	Obligatorisch. Bestimmt die Anzeighöhe des Bereichs, in dem das Movie angezeigt werden soll. Wenn die QuickTime-Bedienelemente sichtbar sind, fügen Sie 16 Pixel zur Moviehöhe hinzu, um die erforderliche Höhe zu erreichen. (In Pixeln)
hidden	Versteckt das Movie und ist nur bei Background-Sound wirklich nützlich.
hotspotn	Ermöglicht Hotspots in einem VR-Panorama zu setzen, in dem <code>n</code> die Hotspot ID ist (URL).
href	Stellt eine Verknüpfung zu einer anderen Webseite oder einem anderen Movie (URL) her.
kioskmode	Bei <code>true</code> steht kein Pop-up-Menü für das Video zur Verfügung, und Sie können es auch nicht mit Hilfe der drag and drop-Funktion speichern (<code>true</code> oder <code>false</code>).
loop	Bei <code>true</code> wird das Video unendlich oft wiederholt. Bei <code>palindrome</code> wird das Movie abwechselnd vorwärts und rückwärts abgespielt (<code>true</code> , <code>false</code> oder <code>palindrome</code>).
movieeid	Eine Zahlen-ID (Integer)
moviename	Der Moviename (Text)
node	Erstellt den Initialnode für Multinode QuickTime VR Videos (Integer).
pan	Erstellt die Initialwinkel für Quicktime VR Movies (Integer von 0–360 Grad).
playeveryframe	Bei <code>true</code> werden Audiotracks ausgeschaltet, und jedes Movie-frame muss abgespielt werden, selbst wenn dadurch die Frame-rate niedriger wird (<code>true</code> oder <code>false</code>).

Tabelle 13.6: Attribute, die von QuickTime4 unterstützt werden (Forts.)

Attribut	Beschreibung
pluginspage	Die URL zur QuickTime-Download-Seite. Sie sollten dies unter folgender Adresse einrichten: http://www.apple.com/quicktime/download/ .
qtextn	Identifiziert die URL für ein Movie, das nach Ablauf des Movies geladen und abgespielt werden soll. n kann ein Integer von 1 bis 255 sein und gibt den Index der URL in der Abspiel-liste anzeigen. nn ist der Index der nächsten qtext URL, der ge-laden werden soll (URL oder gotonn).
qtsrc	Zwingt einen Web-Browser, das QuickTime -Plugin zu verwen-den. URL überschreibt den Wert des src-Attributs (URL).
qtsrccokespeed	Spezifiziert die Datensequenz des Movies und lässt dabei die aktuelle Verbindungsgeschwindigkeit außer Acht (Zahl).
scale	tofit legt die Größe des Movies nach den Werten der height und widht-Attribute fest, aspect passt das Movie auf die Größe der Box an, wobei das Originalgröße des Movies beibehalten wird. Die Größe des Movies wird durch eine Zahl bestimmt (tofit, aspect oder eine Zahl).
src	Legt die URL des Movies fest (URL).
starttime	Legt den ersten Frame des Movies fest (Zeit in Stunden:Minu-ten:Sekunden:Frames).
target	Startet den QuickTime Player, der wiederrum das Movie ab-spielt. Zu diesem Zweck muss das href-Attribut erstellt werden (QUICKTIMEPLAYER).
targetn	Wird mit hotspot und href verwendet. Definiert das Ziel für Links, in denen das hotspot- und href-Attribut verwendet wird. Die Zahl n ist mit der hotspot-ID gleichzusetzen (Name eines gültigen HTML-Frame).
targetcache	Legt fest, dass das Movie, das von einem anderen Movie als Ziel definiert ist, gepuffert wird (true oder false).
tilt	Erstellt den Initialwinkel für QuickTime VR-Movies (Integer).
type	Definiert den MIME-Type des Movies. Wenn das Movie sichtbar ist und height und width-Werte besitzt, muss type hinzugefügt werden. Dieses Attribut wird vom Netscape Navigator 2 oder neueren Versionen unterstützt (MIME-Type).
volume	Erstellt das Initialaudiovolumen. Defaulteinstellung ist 100 (Inte-ger von 0 bis 100).
width	Erstellt die Breite des Anzeigenbereichs für den des Movies (in Pixeln).

Tabelle 13.6: Attribute, die von QuickTime4 unterstützt werden (Forts.)

Sound- und Videodateitypen

Erstellen Sie eine Liste von all den Audio- und Videoformaten, die es heutzutage im Web gibt! Sie werden sehen, es gibt so viele davon, dass eine genauere, leicht verständliche Betrachtung der verschiedenen Multimedia-Dateien, ihrer Erweiterungen und ihrer MIME-Typen erforderlich ist. Ich werde jetzt also versuchen, nur die bekanntesten dieser Dateien zu erwähnen, oder zumindest die, die Sie höchstwahrscheinlich in Ihre Webseite einbetten werden.

Bevor ich diese Dateien aufliste, wäre es wohl sinnvoll, dass sie zuerst einmal wissen, welche Faktoren man beim Auswählen eines Medientyps beachten sollte. Ich meine damit vor allem Sound- und Videoqualität, Größe der Zieldatei, die Anzahl der Plugins oder Player, die mit dem Dateityp kompatibel sind, und die Empfehlbarkeit einer solchen Datei.

Die Qualität von Sound- oder Videodateien hängt vor allem von der Originalsampling-Frequenz, der benötigten Bits pro Sample und der Anzahl des Channels ab.

Die Sampling-Frequenz gibt an, wie oft pro Sekunde das Signal abgetastet wird. Dieser Wert wird in Tausenden von Signalen pro Sekunde oder in Kilohertz (KHz) ausgedrückt. Stellen Sie sich vor, Sie gehen durch einen Raum und können Ihre Augen nur alle fünf Sekunden einmal öffnen. Können Sie sich vorstellen, wie gefährlich das werden könnte? Nun stellen Sie sich vor, Sie öffnen und schließen Ihre Augen jede Sekunde einmal. Sie bekommen ein viel besseres Bild von dem, was sich um Sie herum befindet, und Sie sehen die Dinge viel realistischer und klarer. Das Gleiche geschieht bei Sampling-Frequenzen: je schneller das Sample ist, desto näher ist der Sound oder das Video am Original. Das einzige Problem ist, dass die Erhöhung der Sampling-Frequenz eine schnell zunehmende Anhäufung von Daten zur Folge hat.

Die von Ihnen ausgewählte Bit-Anzahl bestimmt die Qualität des Sounds. Ein 8-bit-Sample, zum Beispiel, kann 256 diskrete Werte messen, während ein 16-bit Sample mehr als 65.000 Werte messen kann. Je mehr Bits Sie verwenden, desto exakter wird die Auflösung des Sounds oder Videos in einem Videoclip.

Die Anzahl der Kanäle spielt vor allem bei Audio-Dateien eine Rolle. Hier unterscheidet man Mono (ein Kanal), Stereo (zwei Kanäle) und Aufnahmen mit mehr als zwei Kanälen. Je mehr Kanäle Sie besitzen, desto besser ist die Soundqualität und desto mehr Freude werden Sie haben; aber wieder einmal auf Kosten der Dateigröße.

Es gibt fünf Dateitypen, die heutzutage im Web Verwendung finden, und jede einzelne hat besondere Vor- und Nachteile.

Audio Interchange File Format (AIFF) sind unkomprimierte Audio-Dateien und werden meistens bei Macintosh verwendet. Die Dateien können sehr groß sein.

Musical Instrument Digital Interface (MIDI) werden beim Synthetisieren der aufgenommenen Klänge vorgezogen. Die Datei ist sehr klein, aber nachdem keine aufgenommenen Klänge abgespielt werden können, ist MIDI hier eine Ersatzlösung.

l-law (normalerweise wird es mü-law ausgesprochen, wird auch Basic Sound genannt) ist die älteste Form von Audio im Internet; aber seine eher magere Soundqualität macht es heute nicht besonders attraktiv. Der Vorteil ist, dass die Dateien klein sind.

Motion Picture Expert Group (MPEG) Audio bietet drei verschiedene Sound-Dateitypen und ist sehr bekannt und beliebt, da MPEG fast überall als Audio- und Videoformat akzeptiert wird. Der neueste, erfolgreiche Dateityp ist MP3 Audio, die einen Sound liefert, der in seiner Qualität vergleichbar mit dem Sound einer CD ist. MP3 Audio ist sehr klein.

Waveform (WAVE)-Dateien wurden von Microsoft und IBM entwickelt und werden meistens auf Windows-Computern verwendet.

In Tabelle 13.7 sehen Sie die bekanntesten Audioformate

Name	Erweiterung(en)	MIME-Typ
AIFF	AIFF, AIF, AIFC	audio/aiff
µ-law	AU, SND	audio/basic
MIDI	MID, RMI	audio/mid
Waveform (WAVE)	WAV	audio/wav
MPEG Audio	MP2, MP3	audio/x-mpeg

Tabelle 13.7: Common Audio File Formats

Die gängigen Videotypen sind AVI, MPEG und QuickTime und werden in der folgenden Liste beschrieben:

- ▶ AVI, was für *Audio/Video Interleaved* steht, ist ein beliebtes Microsoft Windows Format. Es ist mit den meisten Windows Computern kompatibel; In den letzten Jahren hat es aber auch in anderen Computer-Kreisen an Akzeptanz gewonnen.
- ▶ MPEG Video liegt ungefähr zwischen AVI und QuickTime und ist deshalb sehr beliebt.
- ▶ QuickTime Video wird von Macintosh entwickelt und findet, wie AVI, in den Kreisen die meiste Unterstützung, in denen das System des Herstellers verwendet wird. QuickTime wird nun auch außerhalb des Macintosh-Kreises akzeptiert und deshalb sollten Sie sich nicht scheuen, es zu benutzen.

In Tabelle 13.8 sehen Sie eine Zusammenfassung dieser Videoformate

Name	Erweiterung(en)	MIME-Typ
Audio/Video Interleaved	AVI	video/x-msvideo
MPEG	MPEG, MPG	video/mpeg
QuickTime	MOV, QT	video/quicktime

Tabelle 13.8: Gängige Video-Dateiformate

Plugins und Player

Man kann heutzutage große Fortschritte hinsichtlich der Unterstützung von Inline Multimedia und Animationen im Web feststellen, aber längerfristig gesehen werden Plugins sicherlich den besten Effekt erzielen.

Plugins sind eine Art eingebaute Unterstützung für den Browser. Sie sind also nicht völlig unabhängig vom Browser, sondern sie dienen als Plugin, also als Erweiterungskomponente im Browser, arbeiten in ständiger Verbindung zum Browser und statten den Browser mit neuen Fähigkeiten aus. Ein Video-Plugin erlaubt das direkte Inline-Abspielen der Video-Dateien vom Browser. Genauso erlaubt zum Beispiel ein Spreadsheet Plugin einem editierbaren Spreadsheet, als Element auf einer Webseite integriert zu werden. Plugins können auch Links zurück zum Browser erstellen. So könnte ein Spreadsheet theoretisch Links enthalten, die vom Plugin aus aktiviert und nachverfolgt werden könnten.

Netscape führte das Plugin-Konzept mit der Version 2.0 seines Browsers ein und stellt bereits eine Liste dieser Plugins auf seiner Web-Site zur Verfügung. Bestimmte Plugins sind für viele Sound- und Videoformen erhältlich; tatsächlich enthält Netscape 4.7 schon installierte Sound- und Video-Plugins, die Formate wie AU, AIFF, WAV, MIDI, AVI und Quicktime unterstützen.

Wie Sie heute schon gelernt haben, ist das Problem bei Plugins die Tatsache, dass wenn diese auf Ihrer Webseite benutzt werden sollen, jeder Besucher Ihrer Webseite einen Browser benötigt, der Plugins unterstützt (wie zum Beispiel Netscape oder der Internet Explorer). Die Besucher Ihrer Webseite müssen das entsprechende Plugin besitzen. Dieser muss installiert und der Start muss möglich sein. (Besucher, ohne dem entsprechenden Plugin werden anstelle der eigentlichen Datei, eine leere Fläche oder ein Icon auf Ihrer Seite sehen, das auf diesen Umstand hinweist.) Ein weiteres Problem ist, dass viele Plugins nur für einige wenige Plattformen erhältlich sind. Bei manchen Mediendateien könnte es sein, dass Sie Ihren Server konfigurieren müssen, damit die neuen Mediendateien mit dem richtigen Inhalt geliefert werden können.

Windows Media Player

Der Windows Media Player, der unter www.microsoft.com/windows/mediaplayer erhältlich ist, ist Teil des Windows Betriebssystems und kann verschiedene Multimedia-Dateien abspielen. Version 6.4 (siehe Abbildung 13.4) ist zur Zeit in Windows 98 und bald in Windows 2000 integriert. Benutzer von älteren Windows-Versionen (95 und NT4) können aber die neueste Version gratis herunterladen.



Abbildung 13.14:
Der Windows Media Player kann AVI und andere verbreitete Mediaformate abspielen.

Obwohl Version 6.4 für den Microsoft Internet Explorer 4 und neuere Versionen optimiert wurde, sind auch Versionen für ältere Versionen des Internet Explorer und andere nicht-Microsoft-Browser erhältlich. Eine Macintosh Version ist übrigens auch erhältlich.

Version 6.4 des Mediaplayers kann folgende Dateitypen abspielen: ASF, WAV, AVI, MOV, MPEG, MIDI, IVF, AIF, VOD, AU, MP3, ID3.

Macromedia Flash

Der Flash Player, der von Macromedia entwickelt wurde, ist ein bekanntes Plugin, das das Streamen von Animationen niedriger Bandbreite (von Macromedia Flash erzeugt) in Ihre Webseite ermöglicht. Verglichen mit »traditionellen« Bitmap Animationen sind Flash Animationen extrem kompakt. Der Flash Player hat den Vorteil, dass Animationen gestreamt werden können, sobald der Browser sie erhält, und man so nicht mehr warten muss, bis die gesamte Animation heruntergeladen ist. Für zusätzliche Informationen, besuchen Sie die Macromedia Flash Web Site unter:

<http://macromedia.com/flash>

Macromedia Shockwave

Shockwave ist ein Plugin, das das Abspielen von Macromedia Director Movies als Inline Media auf einer Webseite ermöglicht. Macromedia Director ist als Hilfsprogramm unter professionellen Multimedia-Entwicklern sehr verbreitet, es dient vor allem der Erstellung von Multimedia-Präsentationen, die synchronisierten Sound und Videos wie auch interaktive Elemente enthalten (Tatsächlich wurden viele CD-ROMs, die man heute kaufen kann mit Hilfe von Macromedia Director erstellt). Wenn Sie schon länger mit Director arbeiten, können Sie auch die von Shockwave angebotene Möglichkeit nutzen, mit Director Präsentationen ins Web zu bringen. Wenn Sie ernsthafte Multimediaarbeit im Web oder anderswo leisten wollen, ist Director auch ein geeignetes Tool, dass man sich unbedingt ansehen sollte. Zusätzliche Informationen zu Macromedia Shockwave finden Sie unter: <http://www.macromedia.com/shockwave>.

QuickTime 4 von Apple

Apple QuickTime 4 ist sowohl ein Dateiformat als auch ein Player. Der Player, der von Apple unter www.apple.com/quicktime erhältlich ist, spielt Apple QuickTime-Movies ab (QT, MOV) und es gibt ihn sowohl für Macintosh als auch für die Windows Plattform (siehe 13.15).



Abbildung 13.15:
Benutzen Sie den QuickTime 4 Player,
um gespeicherte Dateien oder Streaming-Dateien abzuspielen.

Wenn Sie den Player installieren, sind Plugins sowohl für den Internet Explorer als auch für den Netscape Navigator installiert.

Es können QuickTime -Movies abgespielt werden, wobei auch QuickTime VR (virtual reality) zusätzlich unterstützt wird. Das sind natürlich dann keine Movies, aber interaktive 3D Videos, die es Ihnen ermöglichen, bestimmte, den Besucher interessierende Ausschnitte der Präsentation zu bewegen, zu schwenken oder zu zoomen.

RealNetworks

RealNetworks hat zwei bekannte Player für Macintosh und für Windows, die es heutzutage gibt: RealPlayer G2 und RealJukebox (siehe Abbildung 13.16).

RealPlayer G2 spielt folgende Dateiformate ab:

- ▶ RM, RA, RAM-RealAudio/ RealVideo Streaming-Text
- ▶ RT-RealText Streaming-Textformat
- ▶ RP-RealPix Streaming – GIF und JPG -Bilder
- ▶ GIF, JPG-Standalone JPG und GIF-Bilder
- ▶ MP3 – MPEG Layer 3 Audioformat
- ▶ SWF – RealFlash und Shockwave Animation
- ▶ SMIL, SMI – SMIL-Dateien
- ▶ VIV, VIVO – VIVO Video-Dateien
- ▶ .MPG, .MPEG – Standard MPEG Layer 1 Video und Layer 2 Video Audioformate
- ▶ WAV, AU, AIFF – Legacy Sound-Dateien
- ▶ QT, MOV – QuickTime Video (Unkomprimiert)
- ▶ AVI – Audio/Video Interleave Video (Microsoft)
- ▶ ASF – Active Streaming Format (Microsoft)
- ▶ MID – MIDI Sound-Dateien

RealNetworks stellt auch ein Produkt her, das RealJukebox genannt wird. Es handelt sich um einen Player, der das Abspielen und Aufnehmen von CD's, das Downloaden und das Abspielen von Musik aus dem Internet und das Managen Ihrer Musikkollektion ermöglicht. (Siehe Abbildung 13.17). RealJukebox kann integrierte RealAudio-Dateien, wie auch MP3 und WAV kodieren. RealJukebox kann alle Dateien der vorhergehenden Liste abspielen, sowie Liquid Audio (LQT), A2B, und EMMS-Dateien. Das Erscheinen neuer Formate ist bereits angekündigt.



Abbildung 13.16:
Benutzen Sie RealPlayer G2, um
Streaming Audio- oder Video-
Dateien zu überprüfen.

Weitere Plugins

Da hier aus Platzgründen nicht alle erhältlichen Audio- und Video-Plugins aufgeführt werden können, werden hier nur einige bekannte Plugins in einer kurzen Liste beschrieben.

1. MBed, von mBed Software (<http://www.mbed.com>), ist sowohl als Netscape-Plugin als auch als OCX Version für den Internet Explorer erhältlich. Dieses Plugin ermöglicht das Anzeigen von Animationen, Sounds, interaktive Schaltflächen und synchronisiertes RealAudio.
2. Sizzler, von Totally Hip Software (<http://www.totallyhip.com/Products.html>) ermöglicht das Betrachten von Webseiten und Arbeiten mit Webseiten und das gleichzeitige Streamen von Animationen zum Browser. Sie können ganz einfach gängige Animationen in Sizzler Format konvertieren.
3. Crescendo, von Live Update (<http://www.liveupdate.com/crescendo.html>), ermöglicht das Streamen von MIDI-Dateien über das Web. Das Plugin verfügt über CD-Player ähnliche Bedienelemente für die Steuerung der Wiedergabe und einen Digitalzähler.
4. Beatnik, von Headspace (<http://www.headspace.com>), spielt RichMusicFormat (RMF) und andere Sound-Dateiformate (MIDI, MOD, WAV, AIFF und AU) im Web-Browser ab. Der Sound wird in HiFi-Qualität abgespielt, die mit High-End Soundcards vergleichbar ist. Dieser Sound hört sich auf vielen Plattformen gleich an.

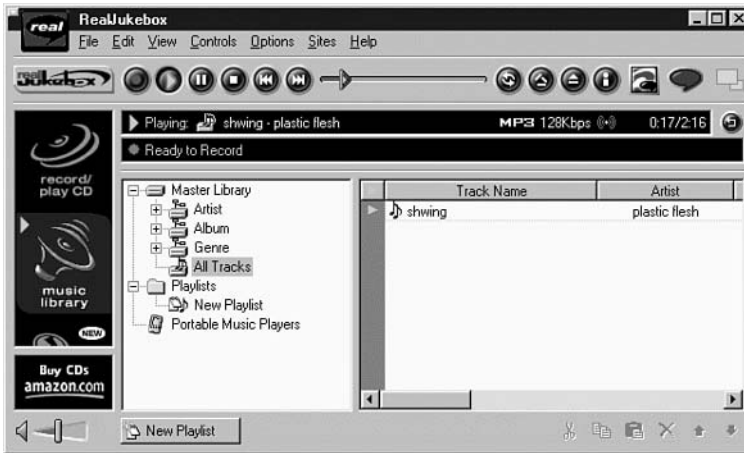


Abbildung 13.17:
RealJukebox
beim Abspielen
einer herunterge-
ladenen MP3
Audio-Datei.

Zusammenfassung

Die heutige Lektion war sicherlich sehens- und hörens- wert! Sie haben gelernt, dass es nur zwei Möglichkeiten gibt, Sound- und Video-Dateien in Ihre Webseite einzubinden: entweder durch Verknüpfung oder durch Einbetten.

Externe Mediendateien können von Ihrem Web-Browser nicht direkt integriert werden. Wenn Sie dann aber eine Verknüpfung zu einer externen Datei erstellen, startet ihr Browser eine »Hilfsanwendung«, damit Sie diese Dateien sehen oder abspielen können. Sie haben auch gelernt, wie sich externe Medien verhalten, wie man Sound- und Video-Dateien als externe Medien verwendet, und Sie haben einige Tipps kennen gelernt, wie Sie mit externen Mediendateien arbeiten können.

Diese Lektion brachte viele Beispiele, in denen gezeigt wurde, wie man Mediendateien direkt in den Web-Browser einbettet. Sie können das `<embed>`-Element verwenden oder eine Kombination aus `<embed>` und `<object>`.

In Tabelle 13.9 sehen Sie eine Zusammenfassung der Tags, die Sie heute gelernt haben.

Tag	Attribut	Verwendung
<code><a></code>	<code>href</code>	Erzeugt eine Verknüpfung zu einer Audio- oder Video-Datei. Dies funktioniert genauso wie Sie eine Verknüpfung auf einen beliebigen anderen Dateityp erstellen.
<code><embed></code>		Bettet Objekte in Webseiten ein.

Tabelle 13.9: Tags für Inline Media

Tag	Attribut	Verwendung
<object>...</object>		Bettet Objekte in Webseiten ein.
<param>...</param>		Legt Parameterwerte fest, die an eingebettete Objekte übergeben werden sollen. Wird innerhalb des <object>-Elements verwendet.
	dynsrc	Fügt eine Sound- oder Video-Datei anstelle eines Bildes ein. Falls die Datei nicht gefunden wird oder nicht abgespielt werden kann, wird das normale Bild (festgelegt im src-Attribut) angezeigt. Wird nur vom Internet Explorer unterstützt.
<bgsound>		Spielt einen Hintergrundsound ab. Wird nur vom Internet Explorer unterstützt

Tabelle 13.9: Tags für Inline Media (Forts.)

Workshop

Im folgenden Workshop werden eventuell aufkommende Fragen über das Einbinden von Sound- oder Video-Dateien in Webseiten behandelt, außerdem werden Sie ein Wissensquiz und zwei kurze Übungen finden.

Fragen und Antworten

F Hilfe! Ich bin verwirrt. Was muss ich hier tun?

A Denken Sie daran, dass es eine bombensichere Methode gibt, Sound- und Video-Dateien in Webseiten einzubinden: Erstellen Sie Links zu diesen Dateien. Obwohl diese Methode nicht so schick wie das Einbetten von Dateien im Web-Browser-Fenster ist, wissen Sie trotzdem: Ihre Webseite funktioniert. Natürlich muss der Besucher Ihrer Webseite über die entsprechenden Anwendungen verfügen, um die Datei abspielen zu können, aber Sie können ihm helfen, indem sie zu allen in Frage kommenden Players oder Plugins Links erstellen.

F Muss man sich über die Kompatibilität von Web-Browser und HTML bei Audio und Video Sorgen machen?

A Leider ja. Fast alle anderen HTML-Elemente und Techniken sind so standardisiert, dass Sie sich auf das richtige Funktionieren Ihres Codes bei den bekanntesten Web-Browsern verlassen können. Das Einbetten von Sound- und Video-Dateien ist jedoch etwas völlig anderes. Mein Tipp für Sie ist: Sie sollten die verschiedenen Möglichkeiten, wie man Sound und Video einbindet

und die unterschiedlichen Wege, wie Web-Browser diese Dateien einbetten, studieren. Schließlich sollten Sie dann Ihre Webseite hartnäckig im Microsoft Internet Explorer und im Netscape Navigator testen.

F Was sind die Unterschiede zwischen AVI, MPEG und QuickTime-Movies?

A Die feinen Unterschiede würden den Rahmen dieser Lektion sprengen. Eine Rolle spielen hier die Art der Kodierung und der Komprimierung sowie die Anzeigenbereiche in der Zielformat von Audio- und Videodateien. Bei jedem Dateityp werden unterschiedliche Methoden angewendet, die aber im Endeffekt alle gleich sind. Der praktische Unterschied ist der, dass jeder Dateityp einen anderen Player benötigt, damit man sie richtig hören bzw. sehen kann.

F Soll/kann ich Techniken benutzen, die nur mit dem Internet Explorer kompatibel sind, so wie zum Beispiel dynsrc?

A Ich rate Ihnen, diese Techniken nicht anzuwenden, da ein Großteil Ihres Publikums sich bald nicht mehr für Ihre Webseite interessieren wird (ja, vielleicht wird Ihre Webseite sogar gemieden). Wenn Sie sich nur in einem »Internet Explorer-Kreis« befinden, können sie diese Techniken getrost anwenden. Im Endeffekt liegt die Wahl bei Ihnen.

Quiz

1. Was sind die Unterschiede zwischen einem Hilfsprogramm (auch Player genannt) und einem Plugin?
2. Welche Möglichkeiten gibt es, Multimedia in Webseiten zu integrieren?
3. Was sind die Vor- und Nachteile von Plugins?
4. Was ist Streaming Multimedia?

Antworten

1. Hilfsprogramme funktionieren ohne Web-Browser und öffnen Dateien, die Ihr Browser nicht unterstützt. Der Browser lädt eine Datei herunter und gibt sie dann an das Hilfsprogramm weiter, das die Datei liest und abspielt.
2. Sie können Links zu diesen Dateien erstellen oder sie einbetten.
3. Der Vorteil von Plugins ist, dass Sie die Möglichkeit haben, viele verschiedene Inhaltstypen in Ihre Webseite zu integrieren. Der Nachteil von Plugins ist, dass es keine Garantie gibt, dass jeder Ihrer Besucher im Besitz des entsprechenden Plugins ist und jeder bereit ist, sich die Zeit zu nehmen, Ihr Plugin herunterzuladen, um

Ihre Webseite sehen zu können. Manche benutzen Browser, die Ihre Plugins nicht unterstützen, und nicht alle Plugins werden immer von Web-Browsern und Betriebssystemen unterstützt.

4. Streaming Multimedia kann schon während dem Herunterladens abgespielt werden.

Übungen

1. Surfen Sie im Web und besuchen Sie Sites, auf denen Multimedia zu sehen ist. Sehen Sie sich an, wie andere Multimedia in Ihre Webseite integrieren. Versuchen Sie die gleiche Site sowohl mit Internet Explorer als auch mit Netscape Navigator zu besuchen. Gibt es einen Unterschied? Wie kann man ein Plugin downloaden?
2. Sehen Sie sich die Plugins Seite auf der Netscape Site an (<http://home.netscape.com/plugins/index.html>), um mehr über die zahlreichen Plugins für die entsprechenden Plattformen zu erfahren.



Definieren von Formularen

**Woche
2**

Bis jetzt haben Sie fast alles gelernt, was Sie wissen müssen, um funktionierende, attraktive und interaktive Webseiten zu erstellen. Wenn Sie genauer hinsehen, werden Sie jedoch erkennen, dass Ihre Webseiten, so wie sie jetzt sind, nur eine Informationsfließrichtung haben. Ihre HTML-Dokumente, Bilder, Sounds und Videos wurden ohne Rückfahrchein in die Web-Browser Ihrer Besucher heruntergeladen.

Die heutige Lektion behandelt das Erstellen von HTML-Formularen, die den Erhalt von Rückinformationen Ihrer Webseiten-Besucher ermöglichen. Formulare geben Ihnen die Möglichkeit, jegliche Art von Information zur sofortigen Bearbeitung durch Server- Scripts oder für spätere Analysen durch andere Anwendungen zu sammeln. Wenn Sie mit Surfen im Web viel Zeit verbracht haben, sind Sie sicherlich schon auf die unterschiedlichsten Formulare gestoßen. Es gibt viele Formulare: einfache Formulare, die für Suchvorgänge bestimmt sind, Formulare, die Sie in Web-Sites einloggen, Formulare, die Ihnen die Möglichkeit geben, Produkte Online zu bestellen usw. Alle diese Formulare haben eines gemeinsam: sie akzeptieren Inputs von einem Webseiten-Besucher.

Sollten Sie sich Sorgen um die Kompatibilität machen, kann ich Sie an dieser Stelle beruhigen. HTML-Formulare gibt es nun schon seit Beginn der HTML-Sprache und sie werden von fast allen führenden Web-Browsern unterstützt. Ich werde Sie aber natürlich in unserer heutigen Lektion auf eventuelle Kompatibilitätsprobleme jeder Art aufmerksam machen.

Lassen Sie sich von Formularen nicht einschüchtern! Obwohl sie sehr komplex und kompliziert wirken, sind sie eigentlich sehr einfach zu verschlüsseln. Das Schwierigste ist das Formatieren von Formularen. Nach der heutigen Lektion werden Sie jede Art von Formular mit HTML erstellen können. Folgende Themen werden wir behandeln:

- ▶ Interaktion von HTML-Formularen und Server-Scripts, um Interaktivität zu kreieren
- ▶ Erstellung einfacher Formulare zum Einstieg
- ▶ Kennen lernen aller verfügbaren Formularfelder, die zur Erstellung von Radio-Buttons (Optionsfelder), Checkboxes (Kontrollkästchen) und mehr dienen
- ▶ Verwendung komplizierterer Formularfelder
- ▶ Das gut durchdachte und geplante Erstellen von Formularen, damit Daten jeder Art dem von Ihnen verwendeten Server- Script gewachsen sind

Verstehen von Formularen und ihrer Funktionen

Ja, Sie müssen natürlich einige Dinge über Formulare verstehen. Erstens, ist ein Formular Teil einer Webseite, die Sie mittels HTML-Elementen erstellen. Jedes Formular enthält ein `<form>`-Element, zu dem spezielle Formularfelder wie Schaltflächen, Textfelder, Checkboxes, Submit-Schaltflächen und Menüs gehören. Diese Formularfelder dienen als Benutzer – Interface für das Formular (es handelt sich hierbei um die Elemente, die Formularbenutzer auf der Webseite sehen können). Wenn Formulare ausgefüllt werden, entsteht eine Interaktion zwischen dem Benutzer und den Formularfeldern des Formulars. Sie können auch viele andere HTML-Elemente innerhalb von Formularen zur Erstellung von Legenden, zusätzlicher Informationen oder Strukturen usw. verwenden. Diese Elemente sind nicht Teil des Formulars, aber sie können das Aussehen Ihres Formulars verbessern oder seine Anwendungsmöglichkeiten erhöhen.



Ein *HTML-Formular* ermöglicht das Sammeln von Informationen, die Ihnen der Besucher Ihrer Webseite zukommen lässt. Sie können die aufgenommene Information an ein Script in Ihrem Web-Server senden, damit sie dort entweder bearbeitet oder gespeichert werden können.

Beim Ausfüllen eines HTML-Formulars werden Informationen eingegeben oder eine der Formularoptionen mittels der Formularfelder ausgewählt. Schließlich erfolgt die Datenübertragung, die sich in mehrere Schritte unterteilen lässt. Zunächst ermittelt das Formular die Elemente, die Daten enthalten und bildet daraus einen Datensatz. Dieser wird im Anschluss codiert und an den Webserver gesendet, um dort verarbeitet zu werden.

Es ist sehr wichtig, diese Vorgänge der Datenübertragung gut zu verstehen. Die Daten sind immerhin das Wichtigste an der ganzen Sache: letztendlich haben Sie sich für die Erstellung eines Formulars entschlossen, weil Sie an diesen Besucher-Daten interessiert sind. Nachdem der Benutzer die Submit-Schaltfläche angeklickt hat, handelt es sich nicht mehr um einen reinen *HTML-Vorgang*. Die Verarbeitung wird meistens serverseitig durch Programme oder Scripten wie zum Beispiel CGI-Scripten (Common Gateway Interface) erfolgen. Mit anderen Worten, damit Ihr Formular erfolgreich verarbeitet werden kann, muss bereits ein Script auf Ihrem Server liegen, das die Daten entgegen nehmen und sie speichern oder bearbeiten kann.

Es gibt einige wichtige Ausnahmen von dieser Regel, die man nicht außer Acht lassen sollte. Zum einen kann ein Formular einen Besucher abhängig von seinen Eingaben zu einer anderen Webseite weiterleiten. Die zweite Besonderheit ist sehr nützlich und ist gleichzeitig eine einfache Möglichkeit für Sie, Formulare zu testen. Anstatt ein Script zu benutzen, kann der Inhalt des Formulars auch an Sie gemailt werden. Eine letzte Besonderheit ist das gelegentliche Verwenden von Formularen in Dynamic

HTML, da sie Benutzeraktionen, wie zum Beispiel das Mausklicken, abfangen. Ein Formular in Dynamic HTML dient nicht der Datensammlung, sondern der Erstellung von Schaltflächen, die es dem Benutzer ermöglichen, bestimmte Aktionen auszulösen.

Übung 14.1: Erstellen eines einfachen Formulars, das einen Namen und ein Passwort akzeptiert

O.K., gehen wir es an und erstellen ein einfaches Formular, das die eben erklärten Konzepte illustriert.

Ich habe eine Webseite erstellt, die den Benutzer zwingt, einen Namen und ein Passwort einzugeben, um mit der Anwendung fortzufahren. Werfen Sie einen Blick auf Abbildung 14.1 und sehen Sie sich die Webseite an.

Beginnen Sie mit dem Öffnen Ihres bevorzugten HTML-Editors (ich bevorzuge Notepad) und erstellen Sie eine Webseiten-Schablone. Geben Sie als Erstes die Standard-HTML Kopfzeileninformation ein, fügen Sie das `<body>`-Tag hinzu, und setzen Sie die `<body>`- und HTML-Elemente in abschließende Tags, um eine Schablone zu erhalten, die als Grundlage zur folgenden Arbeit dient. Wenn Sie bereits eine ähnliche Schablone besitzen, laden Sie sie einfach in Ihren HTML-Editor herunter.



Ich tendiere dazu, Transitional HTML zu benutzen und dies in `<!doctype>` zu vermerken. Das gibt mir die Flexibilität, verworfene HTML-Elemente hinzuzufügen, ohne dass ich mir Sorgen um Validierungsfehler machen müsste.

```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>
<head>
<title>myTitle</title>
</head>
<body>
</body>
</html>
```

Fügen Sie als Nächstes Ihren Titel hinzu, um den Inhalt Ihrer Webseite anzukündigen.

```
<title>Enter the Wabbit Hole</title>
```

Fügen Sie innerhalb des Bodys der Webseite ein `<form>`-Element hinzu. Ich habe sowohl das einleitende als auch das abschließende Tag eingefügt, und dazwischen eine Freizeile gelassen, damit ich nicht vergesse, `<form>` am Ende zu schließen.

```
<form action="http://www.fakeurl.com/cgi-bin/entrance.cgi" method="post">
</form>
```

Bevor wir fortfahren, müssen Sie mehr über das `<form>`-Element und die Attribute, die Sie im einleitenden Tag sehen können, wissen. `<form>` ist hier also der Anfang des Elements und gibt an, dass Sie ein HTML-Formular erstellen. Das obligatorische `action`-Attribut spezifiziert die URL zum Server-Script (Sie fügt den Dateinamen hinzu), das das Formular nach seiner Übertragung bearbeiten wird. Es ist sehr wichtig, dass das mit dem Namen eingegebene Script auf dem Web-Server an der von der URL definierten Stelle auch vorhanden ist.



Bevor Sie Ihr Formular ins Web bringen, sollten Sie Ihren ISP kontaktieren und sich erkundigen, ob Sie dessen Scripts verwenden können oder Ihre eigenen Scripts hinzufügen können. Sie müssen auch die URL festlegen, die auf die Verzeichnisse verweisen, in denen die Scripts auf dem Server vorhanden sind. Einige ISPs führen aus Sicherheitsgründen sehr strenge Kontrollen durch und werden Ihnen das Erstellen oder Hinzufügen von Scripts auf Ihrem Server nicht gestatten. Wenn dies der Fall ist, Sie aber unbedingt Formulare auf Ihrer Webseite benötigen, rate ich Ihnen, sich an einen anderen ISP zu wenden.

Das nächste Attribut ist `method`, das eines von zwei möglichen Werten akzeptiert: `post` oder `get`. Diese Werte bestimmen, wie Formulardaten zu Ihrem Web-Server übertragen werden. Wenn Sie `post` wählen, werden die Formulardaten im Rumpf des Formulars eingefügt und zum Web-Server gesendet. Wenn Sie `get` verwenden, dann werden die Daten zur im `action`-Attribut spezifizierten URL als URL-Parameter gesendet. `get` wird oft bei Suchformulare verwendet.

Fügen Sie nun Formularfelder und Informationen hinzu, um Ihrem Besucher das Ausfüllen der Formulare zu erleichtern. Beginnen Sie mit dem Hinzufügen – innerhalb des `<form>`-Elements – einer informativen Beschreibung der vom Besucher erwarteten Daten, und fügen Sie anschließend ein `text`-Formularelement hinzu. Der Benutzer muss dann nämlich seinen Namen im Texteingabefeld eintippen. Machen Sie sich jetzt über das Positionieren noch keine Gedanken, denn später werden Sie alle Formularelemente in eine Tabelle eingeben.

```
<form action="http://www.fakeurl.com/cgi-bin/entrance.cgi" method="post">
Your wascawy name:<input type="text" name="username" />
</form>
```

Fügen Sie einen Informationstext und ein Passwort-Element hinzu.

```
<form action="http://www.fakeurl.com/cgi-bin/entrance.cgi" method="post">
Your wascawy name:<input type="text" name="username" />
Your just as wascawy password:<input type="password" name="userpassword" />
</form>
```

Beachten Sie, dass beide Formular-Elemente Input-Elemente sind. Das `type`-Attribut definiert, welcher Elementtyp erstellt wird. In diesem Fall haben Sie ein Passwort- und ein Text-Element. Jedes Formularfeld sieht unterschiedlich aus, wenn es im Web-Browser angezeigt wird (wie zum Beispiel ein Texteingabefeld, Schaltfläche oder Checkbox), akzeptiert unterschiedliche Benutzereingaben (wie Texteingabe oder Mausklicken) und stellt gleichzeitig mehrere Optionen zur Verfügung. Jedes Formularfeld erhält außerdem einen Namen, der das Formularfeld und seine Daten von anderen Formularfeldern unterscheidet. Sie können den Namen frei wählen, solange dieser Name nur einmal vorkommt (Normalerweise müssen Formularfeld-Namen alle unterschiedlich sein; es gibt jedoch Ausnahmen, die etwas später erklärt werden).

Fügen Sie schließlich eine `submit`-Schaltfläche hinzu, damit der Benutzer die im Formular eingegebene Information senden kann. Fügen Sie eine `reset`-Schaltfläche ein, um dem Benutzer die Möglichkeit zu geben, die Eingaben zu löschen, falls er einen Fehler gemacht hat oder neu anfangen will.

```
<form action="http://www.fakeurl.com/cgi-bin/entrance.cgi" method="post">
Your wascawy name:<input type="text" name="username" />
Your just as wascawy password:<input type="password" name="userpassword" />
<input type="submit" value="enter" />
<input type="reset" value="oops!" />
</form>
```

Die Submit-und Reset-Schaltflächen sind ein anderer Type des `<input>`-Elements. Beachten Sie, dass ich mit jedem der beiden ein `value`-Attribut verbunden habe. In diesem Fall ändert das `value`-Attribut den in der Schaltfläche und im Web-Browser angezeigten Text. Sie müssen das `value`-Attribut nicht unbedingt mit diesen Schaltflächen verwenden. Wenn dies der Fall ist, so zeigt der Web-Browser einen Standardtext für diese beiden Schaltflächen an.



Wenn Sie Formularfeldern einen Namen geben und Schaltflächen beschriften, sollten Sie vor allem auf Klarheit und Sinn achten. Wenn die Arbeit mit einem Formular frustrierend und schwierig ist, wird derjenige, von dem sie ja eine Information erwarten, schnell die Flucht ergreifen und Ihre Web-Site für immer und ewig verlassen!

Sie haben jetzt also das Erstellen eines Formulars gelernt. Es könnte also losgehen! Wenn Sie das Formular jetzt in Ihren Web-Browser laden, werden Sie merken, dass es alles andere als repräsentativ ist. Es ist an der Zeit, ein paar Grafik-Elemente hinzuzufügen um Ihrer Seite ein flotteres Aussehen zu verleihen, und die Elemente des Formulars in eine schöne, saubere Tabelle zu ordnen.

Ich habe drei Grafiken für den Seitenkopf erstellt und eine Tabelle verwendet, um diese zu ordnen.

```
<table border="0">
  <tr>
    <td></td>
    <td align="center"><br>
    </td>
  </tr>
</table>
```

Erstellen Sie als Nächstes eine weitere Tabelle, die aus vier Querspalten und zwei Längsspalten besteht. Geben Sie nun `<form>` in die Tabelle ein, um alle Elemente zu platzieren.

```
<table border="0">
<tr>
  <td></td>
  <td></td>
</tr>
<tr>
  <td></td>
  <td></td>
</tr>
<tr>
  <td></td>
  <td></td>
</tr>
<tr>
  <td></td>
  <td></td>
</tr>
<tr>
  <td></td>
  <td></td>
</tr>
</table>
```

Fügen Sie nun die Formularelemente hinzu: Geben Sie das einleitende `<form>`-Element nach dem einleitenden `<table>`-Tag ein. Fügen Sie die Formularfelder in die Zeilen und Spalten Ihrer Tabelle, wie ich es bereits gezeigt habe, ein, und vergessen Sie nicht das abschließende `<form>`-Tag.

Beachten Sie, dass ich die dritte Querspalte der Tabelle freigelassen habe – Sie sehen, ich schummele ein bisschen! Nachdem ich den Quellcode für dieses Beispiel fertiggestellt hatte, wollte ich etwas Platz zwischen dem Password-Feld und den Schaltflächen schaffen. Nach einigen Versuchen richtete ich die Texteingabenaufforderungen ebenfalls nach rechts aus, erstellte Tabellenspalten, die 50% der Gesamtbreite der Tabelle einnahmen und zentrierte die Schaltflächen in den entsprechenden Spalten.

```
<table border="0">
  <form action="http://www.fakeurl.com/cgi-bin/entrance.cgi" method="post">
  <tr>
    <td align="right" width="50%">Your wascawy name:</td>
    <td width="50%"><input type="text" name="username" /></td>
  </tr>
```

```
<tr>
  <td align="right">Your just as wascaw password:</td>
  <td><input type="password" name="userpassword" /></td>
</tr>
<tr>
  <td></td>
  <td></td>
</tr>
<tr>
  <td align="center"><input type="submit" value="enter" /></td>
  <td align="center"><input type="reset" value="oops!" /></td>
</tr>
</form>
</table>
```

Dies sieht so aus, als könnte es funktionieren; Stellen Sie nun die einzelnen Code-Teile zusammen. Hierzu noch einige wichtige abschließende Anmerkungen: Um die Tabelle auf der Seite zu zentrieren, schloss ich diese in ein `<div>`-Tag ein und wies dem `align`-Attribut den Wert `center` zu. Achten Sie darauf, wo die einleitenden und abschließenden `<div>`-Tags auf der Seite platziert sind. Um zusätzlichen Freiraum zwischen den Grafiken und dem Formular zu schaffen, fügte ich mit dem `<br>`-Tag einen Zeilenumbruch ein.



```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>
<head>
<title>Enter the Wabbit Hole</title>
</head>
<body>
<div align="center">
<table border="0">
  <tr>
    <td></td>
    <td align="center"><br>
    </td>
  </tr>
</table>
<br>
<table border="0">
  <form action="http://www.fakeurl.com/cgi-bin/entrance.cgi" method="post">
    <tr>
      <td align="right" width="50%">Your wascaw name:</td>
```

```

        <td width="50%"><input type="text" name="username" /></td>
    </tr>
    <tr>
        <td align="right">Your just as wascawy password:</td>
        <td><input type="password" name="userpassword" /></td>
    </tr>
    <tr>
        <td></td> <!-- comment: left blank to add space -->
        <td></td> <!-- comment: left blank to add space -->
    </tr>
    <tr>
        <td align="center"><input type="submit" value="enter" /></td>
        <td align="center"><input type="reset" value="oops!" /></td>
    </tr>
</form>
</table>
</div>
</body>
</html>

```

Dies bedeutet zwar einen größeren Arbeitsaufwand, aber ich denke, das Ergebnis, das in Abbildung 14.1 gezeigt wird, lässt sich sehen.

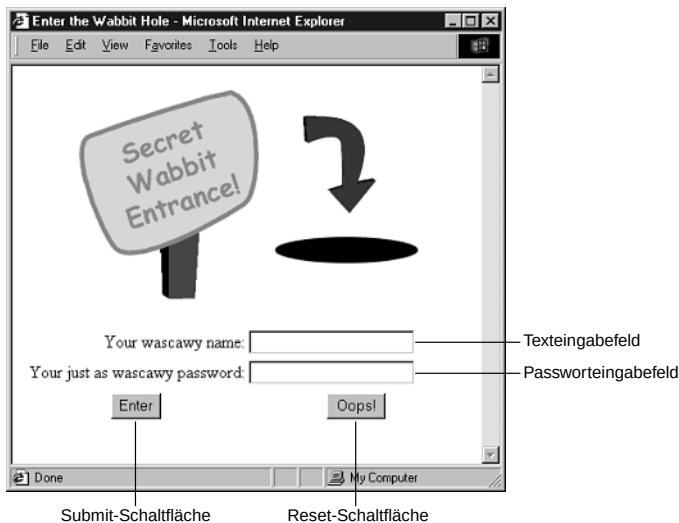


Abbildung 14.1:
Ein einfaches HTML-
Formular mit vier For-
mularfeldern ermög-
licht Zutritt zum Secret
Wabbit hole

Um die Übung zu vervollständigen, habe ich das Formular daraufhin getestet, ob es die von mir verlangten Daten wieder zurückgibt. Die übertragenen Daten sehen folgendermaßen aus:

```
username=Elmer&userpassword=hunter
```

Es ist zwar etwas mickrig, aber Sie können erkennen, dass jedes Formularelement die Daten aufweist, die ich im Formular eingegeben hatte. Mit Hilfe von Scripts können Sie diese Daten auf einfache Weise sowohl vor dem Versenden als auch danach prüfen und weiterverarbeiten.

Die wichtigsten Elemente eines Formulars

Nach dieser ersten Übung sollten Sie schon einige wichtige Dinge mit auf den Weg nehmen: wie das `<form>`-Element verwendet wird, wie einfache Formularfelder hinzugefügt werden, und wie die Submit – und Reset-Schaltflächen verwendet werden. In diesem Teil der Lektion wird auf diesen Kenntnissen aufgebaut, um sie dann zu vertiefen. Wir werden über zwei Hauptbereiche sprechen: das `<form>`-Element und die Formularfelder, die über das `<input>`-Element erstellt werden. Danach werden Sie eine etwas kompliziertere Übung meistern.

Verwendung des `<form>`-Elements zur Erstellung von Formularen

Da das `<form>`-Element die Basis für alle Formulare, die auf Webseiten erstellt werden, ist, ist es wichtig, dass Sie die Verwendung dieses Elements verstehen.

Als Erstes müssen Sie wissen, dass das `<form>`-Tag als Block-Level-Element betrachtet wird. Um das zu verstehen, müssen Sie auch das Gegenstück – die Inline-Elemente – verstehen. Sie verwenden Inline-Elemente wie `<b>` oder `<i>` bei Text oder einzelnen Wörtern, um deren Aussehen zu verändern. Inline-Elemente enthalten den zu verändernden Text, wie `<b>LOVE THIS GAME!</b>`, und können noch verschachtelte Elemente besitzen. `<b>I <i>LOVE</i> THIS GAME!</b>` ist gültiges HTML.

Block-Level-Elemente sind größere Organisationsstrukturen (wie zum Beispiel das `<form>`-und `<table>`-Element). Sie können Web-Inhalte, Inline und andere Block-level HTML-Elemente enthalten. Wenn Sie folgende HTML-Anweisung in einem Web-Browser anzeigen, wird bei verschachtelten Inline-Elementen keine neue Zeile begonnen.

```
<b>Nesting <i>inline</i> elements does not cause a line-break.</b>
```


Bei Block-Level-Elementen wie `<form>` jedoch wird ein Zeilenumbruch im Web-Browser bewirkt. Das Unangenehme an der ganzen Sache aber ist, dass bei der Erstellung des Seitenlayouts das einleitende Tag des `<form>`-Elements einen Zeilenumbruch bewirkt. Verwenden Sie folgenden Code-Ausschnitt:



Inline-Elemente wie `<b>...</b>` können Text oder andere Inline-Elemente enthalten. Block-Level-Elemente wie `<div>...</div>` können aus Inhalt, anderen Block-Level-Elementen und Inline-Elementen bestehen. Die meisten Block-Level-Elemente werden in einer neuen Zeile angezeigt, während bei Inline-Elementen kein Zeilenumbruch bewirkt wird.

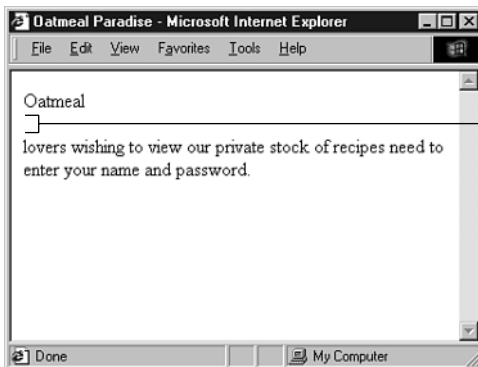
```
<p>Oatmeal <form action="http://www.fakeurl.com/cgi-bin/register.cgi"
method="post">
lovers wishing to view our private stock of recipes need to enter your name
and password.</p>
```

Das Setzen des `<form>`-Elements an seine übliche Stelle bewirkt einen Zeilenumbruch nach `Oatmeal`. Das Ergebnis ist ein ungleichmäßiges Bild (siehe Abbildung 14.2).

Um bis hierher kurz zusammenzufassen: Das `<form>`-Element bewirkt einen Zeilenumbruch. `<form>` kann aus Inhalt, Inline oder anderen Block-Level-Tags bestehen. Sie können jedoch keine verschachtelten Formulare erstellen – bitte nur ein Formular nach dem anderen!

Die Syntax, die zum Erstellen des `<form>`-Elements notwendig ist, ist klar und einfach. Sie müssen meistens nur zwei Attribute verwenden `action` und `method`. `<action>` ist obligatorisch, wie Sie im folgenden Code-Ausschnitt erkennen können.

```
<form action="someaction" method="get or post">
content, form controls, and other HTML elements
</form>
```



Die Zeile bricht
am `<form>`-Tag um

*Abbildung 14.2:
Das Platzieren eines
`<form>`-Elements
innerhalb eines Inline-
Elements, wie zum
Beispiel des Paragra-
fen `<p>`, bewirkt
einen Zeilenumbruch.*

`action` definiert den Übertragungsablauf eines Formulars und enthält normalerweise eine URL in einem CGI Script auf dem Web-Server. Denken Sie auch hier daran, dass sich das Script exakt an der von Ihnen spezifizierten Stelle befinden muss und dass es für das Formular richtig funktionieren muss, um eine erfolgreiche Übertragung versprechen zu können. Obwohl die meisten Formulare ihre Daten zu Scripts senden, können Sie das `<action>` -Link auch zu einer anderen Webseite oder eine Mail zu Link erstellen. Der Code sieht folgendermaßen aus:

```
<form action="mailto:somebody@isp.com" method="post">
```

Dieser Code verknüpft die Formulardaten mit einer E-Mail, die dann an die im `action`-Attribut beschriebene E-Mail-Adresse gesendet wird.



Um Ihre Formulare zu testen, empfehle ich Ihnen, `<mailto>` zu verwenden, um sich die Ergebnisse Ihres Testformulars selbst zuzusenden. Dadurch können Sie Fehler in Ihrem Formular korrigieren und sich vergewissern, dass Sie die korrekten Daten erhalten und Sie so Links zum Script erstellen können.

Das `method`-Attribut kann einer von zwei möglichen Werten sein: `get` oder `post`.

Diese Werte bestimmen, wie die Formulardaten des `action`-Attributs zum Script gesendet werden. Formulare, die `get` verwenden, fügen den Formulardaten der `action` URL ein Fragezeichen hinzu und senden dann das Ganze zur modifizierten URL. Sites mit Suchmaschinen verwenden normalerweise diese Methode. Wenn ich ein Text - Formularfeld namens `searchstring` hätte und im Feld `orangutans` eingegeben hätte, würde die URL am Ende so aussehen:

```
http://www.fakeurl.com/cgi-bin/search?searchstring=Orangutans
```

Wird das `method`-Attribut nicht verwendet, so gilt der Standardwert `get`.

Der andere mögliche Wert ist `post`. Anstatt die Formulardaten mit einer URL zu verbinden und die kombinierte URL-Daten-Zeichenfolge zum Server zu senden, sendet `post` die Formulardaten direkt an den im `action`-Attribut festgelegten Empfänger. Dasselbe geschieht in etwa beim `<form>`-Element. Aber das ist leider nur der Anfang. Haben Sie bemerkt, dass das `<form>`-Element außer die Aktion zur Verarbeitung des Formulars festzulegen nicht viel macht? Um Informationen zu sammeln, benötigen Sie Elemente, die man als Formularfelder bezeichnet.

Erstellen von Formularfeldern mit dem Input-Element

Es ist jetzt an der Zeit zu lernen, wie man die Elemente erstellt, die innerhalb eines Formulars Informationen sammeln. Das Input-Element ermöglicht Ihnen das Erstellen vieler verschiedener Formularfeld-Typen.



Formularfelder sind spezielle HTML-Elemente, die in einem Formular verwendet werden. Sie ermöglichen es einem Besucher Ihrer Web-Site Informationen einzugeben und diese an Sie zu senden. Die Informationen werden in einem Datensatz zusammengefasst, der dann an die im `action`-Attribut festgelegte Adresse gesendet wird, wenn das Formular übertragen wird.

Das `<input>`-Element besteht aus einem einleitendem Tag und Attributen. Es verfügt weder über Inhalt noch über ein abschließendes Tag.

```
<input attributes />
```

Der Schlüsselpunkt hier ist das Auswählen der richtigen Attribute, die das Formularfeld erstellen können, das Sie benötigen. Das wichtigste Attribut ist hier `type`, das den Formularfeldtyp definiert. Für alle Formularfelder außer Submit- und Reset-Schaltflächen, ist das `name`-Attribut erforderlich. Das `name`-Attribut bezeichnet das Formularfeld, sodass Daten dem Formularfeld zugewiesen werden können, wenn sie zum Server gesendet werden. Im Folgenden werden die verschiedenen Formularfeldtypen beschrieben, die Sie bei Verwendung des Input-Elements erstellen können.

Erstellen von Text-Formularfeldern

Text-Formularfelder ermöglichen Ihnen das Sammeln von Benutzerinformationen in geringer Menge. Es wird so ein einzeliges Input-Feld erstellt, in das Benutzerinformationen eingeben werden können, wie zum Beispiel ihren Namen oder ein Suchbegriff. Im Gegensatz zu anderen Formularfeldern wie Optionsfelder (die nur zwei Zustände besitzen: on und off) oder Auswahllisten (in diesen können nur vorgegebene Werte ausgewählt werden), bieten Text-Formularfelder dem Besucher die Freiheit, alle gewünschten Texte einzugeben.

Formularfelder sind immer Teil eines Formulars. Entsprechend befinden sich `<input>`-Elemente immer zwischen dem öffnenden und dem schließenden `<form>`-Element. Beginnen Sie mit dem Erstellen eines Input-Elements und wählen Sie `<text>` als Wert für das `type`-Attribut. Gehen Sie sicher, dass Sie die Formularfelder mit einem Namen bezeichnen, damit das Server-Script den Wert verarbeiten kann. In Abbildung 13.4 sehen Sie ein Text-Formularfeld, in dem ein hilfreicher Text angeboten wird. Dieser Text enthält eine Beschreibung eines Inhalts, den der Benutzer eingeben soll.

```
<p>Enter the name of your pet: <input type="text" name="petname" /></p>
```

Sie können das Aussehen der Text-Formularfelder verändern, indem Sie das `size`-Attribut verwenden. Die eingegebene Zahl bestimmt die Breite des Text-Formularfelds, wie Sie im folgenden Beispiel erkennen können:

```
<input type="text" name="petname" size="15" />
```

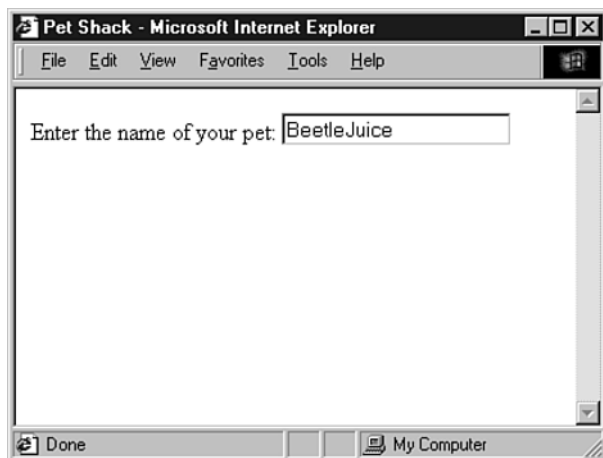


Abbildung 14.3:
Text-Formularfelder erstellen
ein einzeliges Texteingabefeld

Um die Buchstabenanzahl, die ein Benutzer eingeben kann, zu reduzieren, fügen Sie das `maxlength`-Attribut dem Texteingabefeld hinzu. So wird die maximale Buchstabenanzahl bestimmt, die der Besucher eingeben kann. Wenn ein Benutzer versucht, mehr Text einzugeben, wird ihr Web-Browser Inputs ab einer bestimmten Länge für dieses spezielle Formularfeld nicht mehr akzeptieren.

```
<input type="text" name="petname" size="15" maxlength="15" />
```

Um ein Textformularfeld mit Text vorzubelegen, der nach dem Laden des Dokuments angezeigt wird, verwenden Sie das `value`-Attribut. Das `value`-Attribut kann als Eingabeaufforderung oder als Hinweis dienen, und den Benutzer auf die Dateneingabe aufmerksam machen. Bitte sehen Sie sich dazu folgendes Beispiel an:

```
<input type="text" name="petname" size="15" maxlength="15" value="enter pet name" />
```

In diesem Fall erscheint *Enter Pet Name* auf dem Formularfeld, wenn das Formular zum Web-Browser übertragen wird. Diese Beschriftung ist für das Formularfeld festgelegt, solange der Benutzer sie nicht ändert.



Wenn Sie ein `value`-Attribut benutzen und einen Wert verwenden, der größer als das Text-Formularfeld ist, kann es zu Verwirrung beim Benutzer kommen, da der Text abgeschnitten erscheint. Versuchen Sie, wirklich nur mit den wichtigsten Informationen zu arbeiten. Vergewissern Sie sich, dass die Länge des Textes in `value` kleiner oder gleich der Buchstabenanzahl ist, die Sie in `maxlength` angegeben haben.

Um Nur-Lese-Formularfelder zu erstellen, fügen Sie das `readonly`-Attribut, wie im folgenden Beispiel hinzu:

```
<input type="text" name="year" value="1999" readonly />
```

So können Benutzer keinen Text in die Text-Formularfelder eingeben. Es ist auch sehr nützlich, einen konstanten Standardwert für ein Text-Formularfeld zu erstellen, wenn das Formular übertragen wird.

Erstellen von Passwort-Formularfeldern

Das Passwort-Formularfeld gleicht dem Text-Formularfeld, in das Benutzer nur wenig Text für das Formular eingeben können, sehr stark. Es handelt sich hier jedoch um einen Sonderfall, da das Passwort-Formularfeld Sternchen verwendet, um Benutzereingaben zu maskieren.



Das `<password>`-Formularfeld dient nicht nur ausschließlich der Erstellung von Passwörtern. Sie können es für alles verwenden, was Ihrer Meinung nach versteckt werden soll, wenn der Benutzer die Information in das Formular eingibt.

Zur Erstellung eines `password`-Formularfelds, erstellen Sie ein `<input>`-Element und weisen Sie `password` dem `type`-Attribut als Wert zu. Um die Größe des Passwort-Formularfelds und die maximale Buchstabenanzahl, die ein Benutzer eingeben kann, zu reduzieren, können Sie die `length`- und `maxlength`-Attribute, wie im folgenden Code verwenden:

```
<p>Enter your password: <input type="password" name="userpassword" size="8"
maxlength="8" /></p>
```

Abbildung 14.4 zeigt ein Passwort-Formularfeld mit hilfreichen Text, der dem Benutzer den Zweck des Formularfeldes erklärt.

Beachten Sie, dass Sie das `value`-Attribut nicht integrieren müssen, da der von Ihnen eingegebene Wert maskiert ist, wenn er zum Web-Browser übermittelt wird.



Jegliche Informationen, die von einem `password`-Formularfeld gesendet werden, sind nicht maskiert oder verschlüsselt; es handelt sich also um keine sichere Informationsübertragung. Der Benutzer kann zwar nicht lesen, was er eintippt, aber das war auch schon die einzige Sicherheitsmaßnahme des `password`-Formularfelds.



Abbildung 14.4:
Erstellen Sie maskierte Texteingabefelder mit Passwort-Formularfeldern.

Erstellen von Submit-Schaltflächen

Submit-Schaltflächen dienen der Übertragung von Formulardaten, damit diese von einem Server-Script oder einem anderen Programm bearbeitet werden können. Das Eingeben von Submit als Formularfeldtyp kreiert eine Submit-Schaltfläche, die die Beschriftung SUBMIT trägt. Um den in der Schaltfläche angezeigten Text zu ändern, benutzen Sie das `value`-Attribut und geben Sie Ihren eigenen Text, wie im folgenden Beispiel, ein:

```
<input type="submit" value="send form data" />
```



Sie können mehr als eine Submit-Schaltfläche je Formular erstellen.

Erstellen von Reset-Schaltflächen

Reset Schaltflächen dienen der inhaltlichen Überarbeitung Ihres Formulars sowie der Einstellung aller Formularfelder nach dem Wert des Standard-Anfangswerts. In den meisten Fällen wird dieser Wert mittels des `value`-Attributs bei der Erstellung eines jeden Formularfeldes spezifiziert. Mit der Submit-Schaltfläche können Sie die Default-Überschrift von Reset nach Ihren Wünschen ändern, indem Sie das `value`-Attribut verwenden, wie es das folgende Beispiel zeigt:

```
<input type="reset" value="clear form" />
```

Erstellen von Checkboxes – Kontrollkästchen

Kontrollkästchen (Checkboxes) ermöglichen es dem Benutzer, einen Wert zu wählen bzw. die Auswahl zurückzunehmen (siehe Abbildung 14.5). Das Checkbox-Formularfeld kennt entsprechend zwei Zustände: ausgewählt und nicht ausgewählt. Diese werden durch die Werte `on` und `off` repräsentiert. Um eine Checkbox zu erstellen, verwenden Sie das `Input`-Element und geben Sie `Checkbox` als Formularfeldtyp ein. Vergessen Sie nicht, das `name`-Attribut einzugeben, damit das Script erkennen kann, mit welcher Checkbox die Dateien verbunden sind. Folgendes Beispiel zeigt dies:

```
<p>Check to receive SPAM e-mail <input type="checkbox" name="spam" /></p>
```

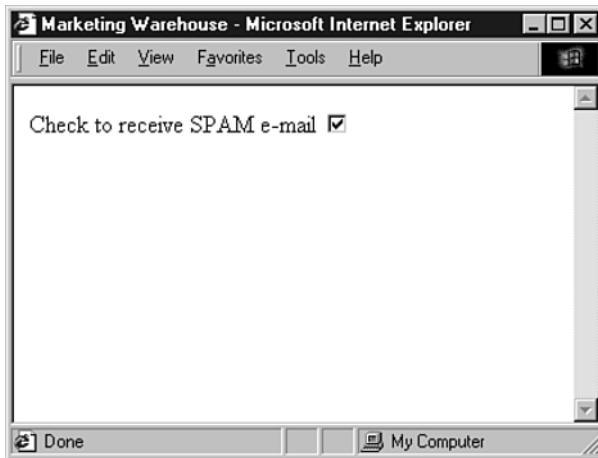


Abbildung 14.5:
»Checkbox – Formularfelder –
ein Beispiel«

Um Checkboxes von vornherein als markiert anzuzeigen, fügen Sie das `checked`-Attribut wie im folgenden Beispiel hinzu:

```
<input type="checkbox" name="year" checked />
```

Sie können Checkboxes auch als Gruppe verwenden, indem Sie allen Checkboxes der Gruppe denselben Namen geben. Dadurch können viele verschiedene Werte ausgewählt und eine gemeinsame Eigenschaft angewendet werden.

```
<p>Check all symptoms that you are experiencing:<br />
Nausea <input type="checkbox" name="symptoms" value="nausea" /><br />
Light-headedness <input type="checkbox" name="symptoms" value="lightheadedness" /><br />
Fever <input type="checkbox" name="symptoms" value="fever" /><br />
Headache <input type="checkbox" name="symptoms" value="headache" /><br />
</p>
```

Wenn dieses Formular zur Bearbeitung auf ein Script übertragen wird, gibt es für jede bearbeitete Checkbox einen Wert an, der mit dem Namen der Checkbox in Verbindung steht.

Erstellen von Optionsfeldern

Optionsfelder sind mit Checkboxes fast identisch, aber sie erscheinen unterschiedlich im Web-Browser. Anstelle einer Box, die markiert oder unmarkiert sein kann, sieht man eine kleine runde Schaltfläche (auf manchen Plattformen gleicht diese Form auch einem Diamanten), auf der ein schwarzer Punkt erscheint, wenn der Benutzer sie anklickt (siehe Abbildung 14.6). Sie erstellen Optionsfelder in einem Input-Element, wenn Sie `radio` als Formularfeldtyp eingeben. Das obligatorische `value`-Attribut verbindet Daten mit dem Optionsfeld. Wenn das Formular übertragen wird, wird `value` zum Server gesendet. Dafür muss das Optionsfeld angeklickt sein.

```
<p>Do you agree? <input type="radio" name="agree" value="yes" /></p>
```

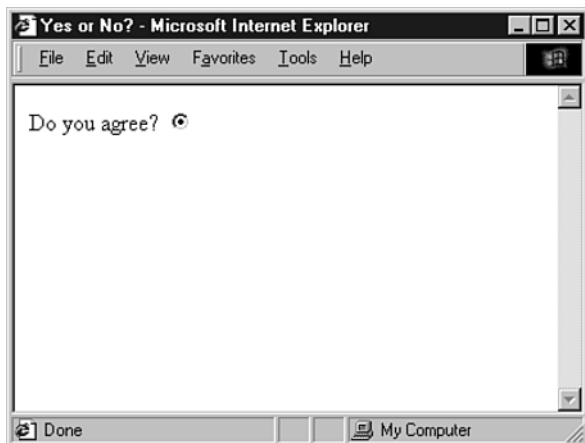


Abbildung 14.6:
Radio-Schaltflächen sind bei Ja- oder Nein-Fragen oder beim Auswählen eines Punktes aus einer Liste nützlich

Wenn zwei oder mehr Optionsfelder denselben Formularfeld-Namen teilen, kann man hier von einer Gruppe sprechen, in der die verschiedenen Möglichkeiten einander ausschließen. Anders gesagt, es kann nur eine Radio-Schaltfläche markiert sein. Im folgenden Codeausschnitt haben beide Radio-Schaltflächen den Namen `sex`. Das bedeutet, dass der Benutzer nur eines der Felder auswählen kann. Wenn ein Benutzer `male` auswählen würde, dann aber erkennen würde, dass er sich geirrt hat und schließlich `female` auswählen würde, würde das Optionsfeld `male` automatisch deselektiert werden.

```
<input type="radio" name="sex" value="male" />
<input type="radio" name="sex" value="female" />
```


Erstellen von Grafik – Schaltflächen

Wenn Sie `<image>` als Eingabeformularfeldtype verwenden, erhalten sie eine Submit-Schaltfläche, die mit dem Bild verziert ist, das sie mittels des `src`-Attributs, wie im folgenden Beispiel, auswählen:

```
<input type="image" src="submit.gif" name="submitformbtn" />
```

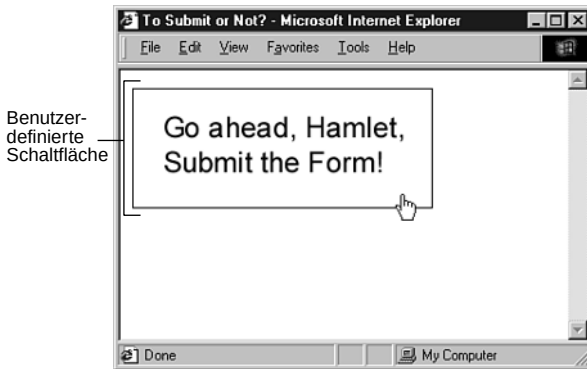


Abbildung 14.7:
Erstellen Sie eine Schaltfläche
für Ihre Besucher mit `type=`
`IMAGE`

Zusätzlich zu den Formulardaten werden die `x` und `y`-Koordinaten des angeklickten Bereichs zum Server übermittelt. Die Daten werden in Form von `name.x = coord` und `name.y = coord` übertragen, wobei `name` der Formularfeld-Name ist. Wenn man nun den vorhergehenden Code verwendet, sieht das Ergebnis folgendermaßen aus:

```
submitformbtn.x=150&submitformbtn.y=200
```

Wenn Sie möchten, können Sie den Namen auch weglassen. Wenn Sie das tun, sehen die zurückgesendeten Koordinaten folgendermaßen aus: `x =` und `y =`

Erstellen von benutzerdefinierten Schaltflächen

Außer Submit-, Reset- und Image-Schaltflächen können Sie auch benutzerdefinierte Schaltflächen erstellen. Diese Schaltflächen haben standardmäßig keine Funktion. Sie können Sie aber mit Scripts verbinden. Ein solches Script wird dann ausgeführt, wenn der Benutzer auf die Schaltfläche klickt. In Abbildung 14.8 sehen Sie eine Schaltfläche, die das Besucher-Script aktiviert, wenn dies die Fähigkeit hat, aktiv zu werden.

```
<input type="button" name="verify" value="verify" onclick="verifydata()" />
```

In diesem Beispiel wird eine Schaltfläche erstellt, die bei Anklicken die Script-Funktion `verifydata()` startet. Geben Sie den Namen, der auf der Schaltfläche erscheint, im `value`-Attribut an. Für dieses Beispiel wäre das *Verify Data*.

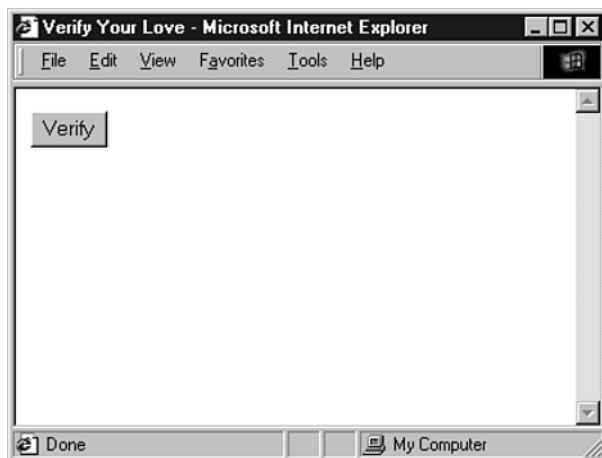


Abbildung 14.8:
Push-Schaltflächen können mit
Server-Scripts verbunden sein.

Schaltflächen, die mit dieser Methode erzeugt werden, unterliegen größeren Einschränkungen, als solche, die über das `<button>`-Element erstellt wurden.

Versteckte Formularfelder

Versteckte Formularfelder sind spezielle Formularfelder, die vor dem Benutzer versteckt sind und von diesem auch nicht verändert werden können. Wenn der Benutzer das Formular übermittelt, wird der dem versteckten Formularfelder zugeteilte Wert mit den anderen Daten gesendet.

```
<input type="hidden" name="surveynumber" value="1402" />
```

Meiner Erfahrung nach wird diese Technik meistens bei Umfragen verwendet, um die Daten zuzuordnen. Für das folgende Beispiel habe ich ein verstecktes Formularfeld erstellt. Es hat den Namen `surveynumber` und den Wert `1402`. Wenn das Formular zur Verarbeitung übermittelt wird, kann ich es mit Hilfe dieser Information identifizieren. Wenn ich morgen die Umfrage ändern wollte, könnte ich für `value` eine neue Zahl eingeben.

Laden von Dateien mit dem Dateiauswahldialog

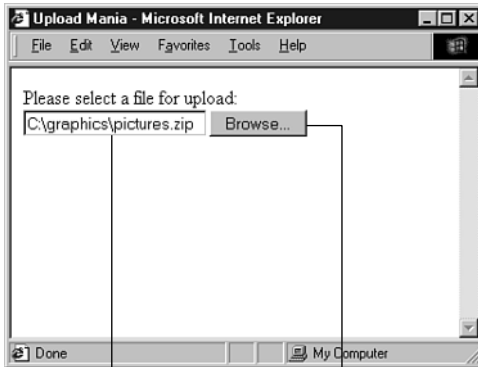
Ein Dateiauswahldialog erlaubt es dem Anwender, eine Datei von seinem lokalen Rechner zusammen mit dem Formular hochzuladen.

Abbildung 14.9 zeigt ein Dateiauswahldialog.

```
<p>Please select a file for upload: <input type="file" name="fileupload" /></p>
```

Benutzer können einen Dateinamen eingeben (mit dem Pfad) oder sie können auch den Inhalt ihres Festplattenlaufwerks browsen, um die betreffende Datei zu finden. Wenn Sie das `value`-Attribut verwenden, können Sie theoretisch einen Ausgangsdateinamen spezifizieren, sodass dieser angezeigt wird; diese Technik wird jedoch nicht vom Microsoft Internet Explorer 5 oder Netscape Navigator 4.5 unterstützt.

```
<p>Please select a file for upload: <input type="file" name="fileupload" /></p>
```



Geben Sie den Pfad und den Dateinamen ein

Wählen Sie Browse zur Auswahl von Dateien auf Ihrem System

Abbildung 14.9:
Das Hochladen von Dateien mit den Dateiauswahldialogen ist einfach.

Übung 14.2: Erstellen eines Formulars mit verschiedenen Eingabefeldern

Es ist Zeit für eine neue Übung. In dieser Übung werden Sie mit allen Formularfeldern arbeiten, die in dieser Lektion vorgestellt wurden.

Zu Beginn, öffnen Sie die HTML-Schablone, die Sie in Übung 14.1 erstellt haben. Fügen Sie drei Formularrümpfe ein. Das erste Formular wird als Umfrage dienen, das zweite wird Ihnen Zutritt zu einem Mitgliedsbereich einer virtuellen Mime-Akademie verschaffen und das dritte Formular dient der Site-Suche. Benutzen Sie folgenden Code:

```
<form action="http://www.fakeurl.com/cgi-bin/survey.cgi method="post">
</form>
<form action="http://www.fakeurl.com/cgi-bin/members.cgi method="post">
</form>
<form action="http://www.fakeurl.com/cgi-bin/search.cgi method="get">
</form>
```

Vergessen Sie nicht, dass Sie für jedes Formular zur Verarbeitung der Daten verschiedene Scripts verwenden, und dass das dritte Formular die `get`-Methode zur Datenübertragung verwendet.

Nun ist es an der Zeit, jedem Formular Formularfelder hinzuzufügen. Für die Umfrage werden Sie einige einfache Multiple-Choice-Fragen stellen. Für die erste Frage benutzen Sie Checkboxes und geben Sie diesen Checkboxes allen den gleichen Namen. Platzieren Sie die Text-Eingabeaufforderung vor das `<input>`-Element, sodass der Text im Browser-Fenster links von der Checkbox erscheint. Im folgenden Code sehen Sie wie es gemacht wird:

```
<form action="http://www.fakeurl.com/cgi-bin/survey.cgi" method="post">
Take today es survey
I like miming because: (check all that apply)
It es fun<input type="checkbox" name="whyilikeit" value="fun" />
I meet interesting people<input type="checkbox" name="whyilikeit" value="people" />
>
It es who I am<input type="checkbox" name="whyilikeit" value="personality" />
The money<input type="checkbox" name="whyilikeit" value="money" />
```

Die zweite Frage in diesem Formular ist eine weitere Multiple-Choice-Frage. Diesmal jedoch werden Sie nur eine Antwort akzeptieren. Benutzen Sie hierfür Optionsfelder, die alle den gleichen Namen haben. Anstatt die Text-Eingabeaufforderungen vor die Schaltfläche zu platzieren, stellen Sie diese lieber hinter die Optionsfelder, sodass sie rechts von den Optionsfeldern erscheinen. So gehen Sie sicher, dass die Schaltflächen in einer Reihe angeordnet sind.

```
The most difficult thing about being a mime is:
<input type="radio" name="difficult" value="makeup" />Putting on the makeup
<input type="radio" name="difficult" value="unitard" />Wearing a spandex unitard
<input type="radio" name="difficult" value="nottalking" />Not talking
```

Fügen Sie als Nächstes einen Dateiauswahldialog ein, sodass den Benutzern das Auswählen und Uploaden einer Datei möglich ist. Fügen Sie Submit und Reset-Schaltflächen hinzu. Um die Umfrage schließlich zu tracken, fügen Sie ein verstecktes Formularfeld hinzu, das festkodiert ist und den Wert `061300` hat. Wenn Sie das versteckte Formularfeld verwenden, können Sie alle Überblick-Antworten überwachen, wann immer Sie möchten, und die Kontrollfläche jeden Tag ändern. Nachdem Sie das versteckte Formularfeld hinzugefügt haben, geben Sie das abschließende `<form>`-Element ein.

```
Upload a picture of yourself miming!
<input type="file" name="picture" />
<input type="submit" value="submit" />
<input type="reset" value="reset" />
<input type="hidden" name="surveynumber" value="091899" />
</form>
```

Das zweite und dritte Formular ist viel einfacher. Im zweiten Formular müssen Text- und Passwort-Formularfelder für Benutzernamen und Passwort hinzugefügt werden. Integrieren Sie zum Schluss noch eine Submit-Schaltfläche in Ihr Formular. Beschränken Sie die Maximalzahl der Zeichen (normalerweise können 20 oder 10 Zeichen eingegeben werden). Begrenzen Sie dann das Passwort-Feld, wie im folgenden Beispiel, auf 10 Zeichen:

```
<form action="http://www.fakeurl.com/cgi-bin/members.cgi method="post">
Enter our Member es Only area!
Username:<input type="text" name="username" maxlength="20" />
Password:<input type="password" name="userpassword" size="10" maxlength="10" />
<input type="submit" value="submit" />
</form>
```

Das fertige Formular besteht letztendlich nur aus einem Textfeld und einer Image-Schaltfläche. Die Image-Schaltfläche dient hier als Submit-Schaltfläche. Auf dieser Schaltfläche ist eine Grafik zu sehen, die hier eine gewöhnliche Schaltflächenoberfläche ersetzt.

```
<form action="http://www.fakeurl.com/cgi-bin/search.cgi method="get">
Search our database<input type="text" name="searchstring" />
<input type="image" src="searchbtn.gif" name="searchbtn" />
</form>
```

Die Formulare, Formularfelder und andere Informationen sind nun fertig; Damit diese in einem HTML-Dokument gut aussehen, müssen sie jedoch noch formatiert werden.



Ich habe festgestellt, dass mir das Formatieren meiner Formulare nach ihrer Erstellung die Möglichkeit gibt, jeden Aspekt einzeln zu betrachten. So erhalte ich noch viel bessere Ergebnisse.

Wenn man die Seite so betrachtet, kann man erkennen, dass sie schön in fünf Elemente unterteilt ist: drei Formulare, eine Titelgrafik, die ganz oben auf der Webseite erscheinen wird und eine andere Grafik, die das Interesse der Besucher weckt.

Um die Elemente auf Ihrer Seite zu organisieren, zentrieren Sie als Erstes die Titelgrafik ganz oben auf der Seite, indem Sie folgenden Code verwenden:

```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>
<head>
<title>Anne es Mime Academy</title>
</head>
<body>
<div align="center">

</div>
```

Es geht weiter: Bitte beachten Sie, dass zwei Formulare eher kurz, und eines lang ist. Denken sie an den Rest der Seite: Es sind zwei Spalten vorhanden, in die Sie die zwei kleineren Formulare eingeben können. Auf der einen Seite der Spalte ist die »Interesse«-Grafik integriert, auf der anderen das längere Formular. Wie sollen Sie das nur machen? Ganz einfach: mit Tabellen.

Beginnen Sie mit dem Erstellen einer Haupttabelle, die aus zwei Spalten und drei Zeilen besteht. Im folgenden Code habe ich die Rolle einer jeden Zeile der Tabelle aufgezeigt:

```
<table border="0">
<tr>
  <td></td> <!-- Zeile 1, Spalte 1: Die Grafik, um Interesse zu wecken -->
  <td></td> <!-- Zeile 1, Spalte 2: Fügen Sie hier die verschachtelte Tabelle für
das erste Formular ein -->
</tr>
<tr>
  <td></td> <!-- Zeile 2, Spalte 1: Hier fügen Sie eine weitere verschachtelte
Tabelle für das zweite Formular ein -->
  <td></td> <!--Zeile 2, Spalte 2: Fortsetzung des ersten Formulars. -->
</tr>
<tr>
  <td></td> <!-- Zeile 3, Spalte 1: Auch hier eine verschachtelte Tabelle. Diese
für das dritte Formular. -->
  <td></td> <!-- Zeile 3, Spalte 2: Fortsetzung des ersten Formulars. -->
</tr>
</table>
```

Verwenden Sie nun folgenden Code, um die verschachtelten Tabellen hinzuzufügen:

```
<table border="0">
<tr>
  <td></td> <!-- Zeile 1, Spalte 1: Die Grafik, um Interesse zu wecken -->
  <td><!-- Zeile 1, Spalte 2: Fügen Sie hier die verschachtelte Tabelle für das
erste Formular ein -->
    <table border="0">
      <tr>
        <td></td>
        <td></td>
      </tr>
      <!-- Bitte fügen Sie hier 12 weitere Tabellenzeilen ein wie die vorherige -->
    </table>
  </td>
</tr>
<tr>
  <td><!-- Zeile 2, Spalte 1: Hier fügen Sie eine weitere verschachtelte Tabelle
für das zweite Formular ein -->
```

```

<table border="0">
  <tr>
    <td></td>
    <td></td>
  </tr>
  <tr>
    <td></td>
    <td></td>
  </tr>
  <tr>
    <td></td>
    <td></td>
  </tr>
  <tr>
    <td></td>
    <td></td>
  </tr>
</table>
</td>
<td></td> <!-- Zeile 2, Spalte 2: Fortsetzung des ersten Formulars. -->
</tr>
<tr>
  <td><!-- Zeile 3, Spalte 1: Auch hier eine verschachtelte Tabelle. Diese für das
dritte Formular. -->
  <table border="0">
    <tr>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <td></td>
      <td></td>
    </tr>
  </form>
</table>
</tr>
</table>

```

Fügen Sie die Grafik- und Formular-Tags an ihre entsprechenden Stellen in den Tabellen hinzu und fügen Sie nun alles zusammen.



```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>
<head>
<title>Anne es Mime Academy</title>
</head>
<body>
<div align="center">
  
</div>
<table border="0">
<tr>
  <td align="center"></td>
  <td rowspan="3">
    <table border="0" bgcolor="#ffcc33">
      <form action="http://www.fakeurl.com/cgi-bin/survey.cgi" method="post">
        <tr>
          <td align="center" colspan="2">Take today es survey</td>
        <td></td>
        </tr>
        <tr>
          <td colspan="2">I like miming because: (check all that apply)</td>
          <td></td>
        </tr>
        <tr>
          <td>&nbsp;</td>
          <td>It es fun<input type="checkbox" name="whyilikeit" value="fun" /></td>
        </tr>
        <tr>
          <td>&nbsp;</td>
          <td>I meet interesting people<input type="checkbox"
name="whyilikeit" value="people" /></td>
        </tr>
        <tr>
          <td>&nbsp;</td>
          <td>It es who I am<input type="checkbox" name="whyilikeit"
value="personality" /></td>
        </tr>
        <tr>
          <td>&nbsp;</td>
          <td>The money<input type="checkbox" name="whyilikeit" value="money" /></td>
        </tr>
```



```
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| The most difficult thing about being a mime is:</td> |  |  | | --- | --- | | <td>&nbsp;</td>  <td><input type="radio" name="difficult" value="makeup" /> Putting on the makeup</td> | | | <td>&nbsp;</td>  <td><input type="radio" name="difficult" value="unitard" /> Wearing a spandex unitard</td> | | | <td>&nbsp;</td>  <td><input type="radio" name="difficult" value="nottalking" />Not talking</td> | | | Upload a picture of yourself miming!</td> |  |  |  | | --- | --- | --- | | <td></td>  <td><input type="file" name="picture" /></td> | | | | <td colspan="2" align="center"><input type="submit" value="submit" /> | <input type="hidden" name="surveynumber" value="091899" /> </form> | | | | | | | |

```

```

        <td>Username:</td>
        <td><input type="text" name="username" maxlength="20" /></td>
    </tr>
    <tr>
        <td>Password:</td>
        <td><input type="password" name="userpassword" size="10" maxlength="10" /></td>
    </tr>
    <tr>
        <td align="center" colspan="2"><input type="submit" value="submit" /></td>
    </tr>
</table>
</td>
</tr>
<tr>
    <td>
        <table border="0" width="100%" bgcolor="#ffcccc">
            <form action="http://www.fakeurl.com/cgi-bin/search.cgi" method="get">
                <tr>
                    <td>Search our database</td>
                    <td><input type="text" name="searchstring" /></td>
                </tr>
                <tr>
                    <td align="center" colspan="2"><input type="image"
src="searchbtn.gif" name="searchbtn" /></td>
                </tr>
            </form>
        </table>
    </td>
</tr>
</table>
</body>
</html>

```

In Abbildung 14.10 sehen Sie die fertig erstellte Webseite mit verschiedenen Formularen und Formularfeldern ergänzt.



The screenshot shows a web browser window titled 'Anne's Mime Academy - Microsoft Internet Explorer'. The page has a header 'Welcome to Anne's Mime Academy' and a profile picture of a woman. Below the header, there are several form sections:

- Passworteingabefeld**: A section for entering a password, including a 'Username:' label, a password input field, and a 'Submit' button.
- Texteingabefeld**: A section for entering text, including a 'Search our database' label, a search input field, and a 'Search' button.
- Formular 1**: A survey section titled 'Take today's survey' with the instruction 'I like miming because: (check all that apply)'. It includes several checkboxes: 'It's fun', 'I meet interesting people', 'It's who I am', and 'The money'. Below these is a question 'The most difficult thing about being a mime is:' followed by three radio button options: 'Putting on the makeup', 'Wearing a spandex unitard', and 'Not talking'. At the bottom of this section is an 'Upload a picture of yourself miming!' label, a file input field, and a 'Browse...' button.
- Kontrollkästchen**: A label pointing to the 'Submit' and 'Reset' buttons at the bottom of the survey section.
- Formular 3**: A label pointing to the 'Submit' button in the password section.
- Formular 2**: A label pointing to the 'Search' button in the search section.
- Benutzerdefinierte Schaltfläche**: A label pointing to the 'Submit' button in the password section.
- Optionskästchen**: A label pointing to the radio button options in the survey section.
- Submit-Schaltfläche**: A label pointing to the 'Submit' button in the survey section.
- Reset-Schaltfläche**: A label pointing to the 'Reset' button in the survey section.
- Datenauswahl-Dialog**: A label pointing to the 'Browse...' button in the survey section.

Abbildung 14.10:
Annes Mime
Akademie hat
drei Formulare
auf einer Seite.

Verwenden weiterer Formularfeld-Elemente

Zusätzlich zu Formularfeldern, die Sie mit Hilfe des `<input>`-Elements erstellen können, gibt es drei Elemente, die selbst noch Inhalte aufweisen.

Verwenden des Button-Elements

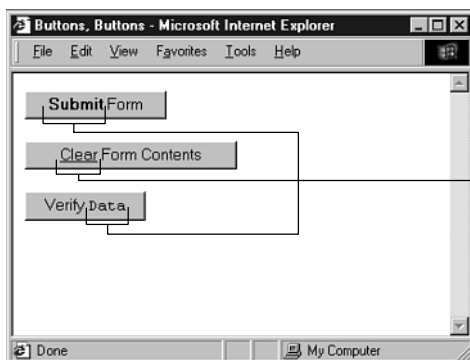
Eine mit dem `<button>`-Element erstellte Schaltfläche gleicht den Schaltflächen, die man mit dem `<input>`-Element erstellen kann. Diese ersteren Schaltflächen können aber innerhalb der einleitenden und abschließenden Tags einen Inhalt aufweisen, der dann auf den Schaltflächen erscheint.



Das `<button>`-Element (im Gegensatz zu dem `<input>`-Element von `type="BUTTON"`) wird vom Netscape Navigator nicht unterstützt.

Sie können drei verschiedene Schaltflächen definieren: Submit, Reset- und Benutzer-definierte-Schaltflächen. Um diese zu erstellen, öffnen Sie als Erstes das einleitende `<button>`-Tag und geben Sie anschließend das `name`-Attribut ein. Das `name`-Attribut legt den Namen für die Schaltfläche fest. Geben Sie als Nächstes das `type`-Attribut ein und wählen Sie den Schaltflächentyp, den Sie verwenden möchten. Schließen Sie danach das Tag und geben Sie den Text ein, der auf der Schaltfläche erscheinen soll. Fügen Sie schließlich ein abschließendes `<button>`-Tag hinzu. Der folgende Code-Ausschnitt dient zur Erstellung der drei verschiedenen Schaltflächen. In Abbildung 14.11 sehen Sie drei `<button>`-Tag. Jedes dieser Tags dient einem anderen Zweck. Beachten Sie, dass ich zwei Zeilenumbruch-Tags verwendet habe, `<br>`, um genügend Abstand zwischen den Schaltflächen zu schaffen.

```
<button name="mysubmitbutton" type="submit"><b>Submit</b> Form</button><br />
>
<button name="myresetbutton" type="reset"><u>Clear</u> Form Contents</button><br />
><br />
<button name="mycustombutton" type="button">Verify <code>Data</code></button>
```



HTML-Formatierung

*Abbildung 14.11:
Das `<button>`-Tag ermöglicht Ihnen das Integrieren von Inhalt und HTML-Elementen in die Schaltflächen.*

Da Sie verschiedene HTML-Elemente in ein `<button>`-Tag integrieren können, ist das folgende Beispiel durchaus möglich:

```
<button name="myresetbutton" type="reset"><b>Clear</b> Form Contents</button>
```

Wenn Sie das `<b>`-tag innerhalb einer Schaltfläche verwenden, erscheint das Wort »Clear« fettgedruckt.

Sie können auch Bilder einfügen, die auf den Schaltflächen erscheinen sollen. Geben Sie einfach das `<img>`-Tag zum Inhalt der Schaltflächen, wie im folgenden Beispiel, ein:

```
<button name="mysubmitbutton" type="submit">
</button>
```

Erstellung von umfassenden Texterfassungsfeldern mit `<textarea>`

Mit dem `<textarea>`-Tag erstellen Sie ein umfassendes Texterfassungsfeld, in dem Besucher so viel Text eingeben können, wie Sie es wünschen. Um ein solches Textfeld zu erstellen, beginnen Sie mit dem einleitenden `<textarea>`-Tag des Elementnamens und geben Sie dann die `rows`- und `cols`-Attribute ein, bevor Sie das Tag schließen. Diese obligatorischen Attribute definieren die Größe des Texterfassungsfelds in Zeilen und Spalten. Sie können auch eine hilfreiche Information, die im Texterfassungsfeld als Anfangswert erscheint, eingeben. Vergessen Sie am Ende nicht das abschließende `</textarea>`-Tag. In Abbildung 14.12 sehen Sie, das Ergebnis.

```
<p>Please comment on our customer service.
<textarea name="question4" rows="10" cols="60">
Enter your answer here
</textarea>
</p>
```

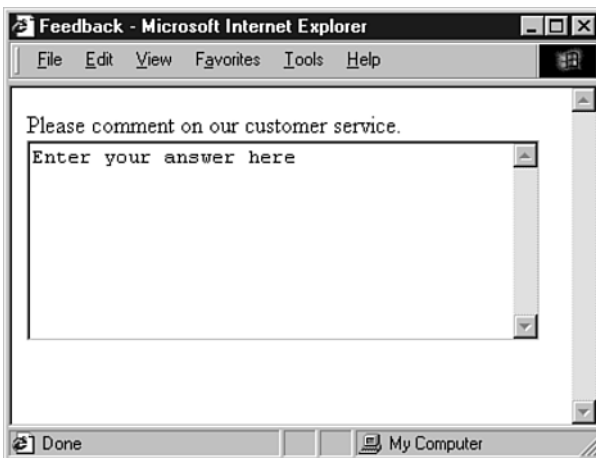


Abbildung 14.12:
Verwenden Sie `<textarea>`, um
umfassende Texterfassungsfel-
der zu erstellen.



Sie mögen sich fragen: »Aber wo ist denn das `<isindex>`-Tag?« Über das `<isindex>`-Tag erzeugen Sie, ähnlich wie die mit dem Typ `text` des `<input>`-Elements, ein einzeliges Textfeld. Allerdings wurde es vom W3C verworfen. Die Syntax ist `<isindex prompt="prompt for user input">`. Als ich dieses

Kapitel geschrieben habe, habe ich `<isindex>` in den aktuellen Versionen vom Internet Explorer und Navigator getestet und konnte feststellen, dass beide `<isindex>` nicht mehr unterstützen. Eine detaillierte Besprechung ist deshalb nicht notwendig.

Erstellen von Menüs mit `<select>` und `<option>`

Das `<select>`-Element erstellt ein Auswahlmenü. In diesem Auswahlmenü kann der Benutzer, wie in den Menüs, die man in den meisten Software-Anwendungen findet, auswählen. Im Grunde dient `<select>` nur zur Definition des Gerüsts einer Auswahlliste. Zur Erstellung der Einträge müssen Sie zumindest ein `<option>`- oder `<optgroup>`-Tag einfügen.

Um ein Auswahlmenü zu erstellen, fügen Sie zunächst das öffnende `<select>`-Tag ein und weisen dessen `name`-Attribut einen Wert – den Namen des Elements – zu:

```
<select name="location">
```

Geben Sie als Nächstes Ihren Menüeintrag unter Verwendung des `<option>`-Tags ein.

```
<select name="location">
  <option>Indiana</option>
  <option>Fuji</option>
  <option>Timbuktu</option>
  <option>Alaska</option>
```

Geben Sie schließlich das abschließende Tag nach dem `</select>`-Element ein.

```
<p>Please pick a travel destination:
```

```
<select name="location">
  <option>Indiana</option>
  <option>Fuji</option>
  <option>Timbuktu</option>
  <option>Alaska</option>
</select>
</p>
```

In Abbildung 14.13 sehen sie eine Auswahlliste mit vier Einträgen.

Zur Erstellung einer Bildlaufleiste am Rand der Liste fügen Sie einfach nur das `size`-Attribut in das einleitende `<select>`-Element, wie im folgenden Beispiel, hinzu.

```
<select name="location" size="2">
```

In Abbildung 14.14 sehen Sie das gleiche `<select>`-Element wie in Abbildung 14.13. Allerdings ist das `size`-Attribut auf 2 eingestellt. Beachten Sie, dass die Auswahlliste nun eine Bildlaufleiste aufweist.



Abbildung 14.13:
Sie können `<select>`-Formularfelder zur Erstellung von Auswahllisten verwenden.

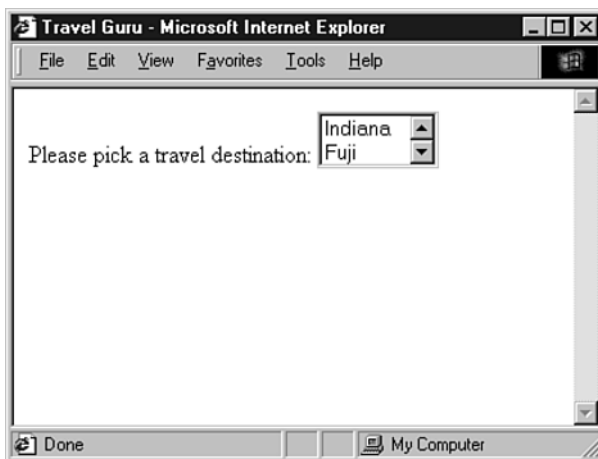


Abbildung 14.14:
Sie können auch Bildlauf Listen mit Hilfe des `<select>`-Elements erstellen.

Das `<select>`-Element reduziert die Höhe der sichtbaren Fläche auf zwei Einträge der Auswahlliste. Wenn Benutzer weitere Optionen sehen möchten, so müssen sie das Bild scrollen.

Um Daten zum Server zu senden, die sich von den angezeigten Einträgen der Liste unterscheiden, können Sie das `value`-Attribut der `<option>`-Tags verwenden. Das ist vor allem dann sinnvoll, wenn Sie die Daten verarbeiten und einen Zahlenwert oder andere codierte Werte benötigen. Der folgende Code zum Beispiel erzwingt das Zurücksenden von `bw499` für `courses` anstatt `Basket Weaving 499`.

```
<select name="courses">
  <option value="p101">Programming 101</option>
```

```
<option value="e312">Ecomomics 312</option>
<option value="pe221">Physical Education 221</option>
<option value="bw499">Basket Weaving 499</option>
```

```
</select>
```

Um Optionen vorauszuwählen, fügen Sie das `selected`-Attribut in einem `<option>`-Tag, wie im folgenden Beispiel, hinzu:

```
<select name="courses">
  <option value="p101">Programming 101</option>
  <option value="e312">Ecomomics 312</option>
  <option value="pe221" selected>Physical Education 221</option>
  <option value="bw499">Basket Weaving 499</option>
</select>
```

Dieser Code zwingt die Auswahlliste zum Anzeigen von `Physical Education 221` als Anfangswert.

Bis jetzt haben Sie Auswahllisten erstellt, in denen ein Benutzer nur eine Wahl treffen kann. Sie können das aber ganz einfach ändern: Fügen Sie das `multiple`-Attribut in das `<select>`-Element, wie im folgenden Beispiel, hinzu, und schon ist die Auswahl mehrerer Einträge möglich:

```
<select name="courses" multiple>
```



Ein Benutzer kann mehrere Optionen mit  +Klicken (bei Windows), mit  +Klicken oder mit  +Klicken (bei Macintosh) auswählen.

Um nun langsam das Thema Auswahllisten zu beenden, möchte ich kurz auf die `<optgroup>`-Tags eingehen. Theoretisch teilt dieses Tag `<option>`-Elemente in Gruppen ein und erstellt Untermenüs.

Sie sollten `<optgroup>`-Tags in `<select>`-Elemente platzieren. Das `<optgroup>`-Tag sollte, wie im folgenden Beispiel auch `<option>`-Tags enthalten.

```
<select name="futility">
  <optgroup label="mild">
    <option>Swimming upstream</option>
    <option>Shaving with no shaving cream</option>
  </optgroup>
  <optgroup label="extreme">
    <option>Squeezing blood from turnips</option>
    <option>Crying over spilled milk</option>
  </optgroup>
</select>
```




Beachten Sie, dass das `<optgroup>`-Tag aktuell vom Internet Explorer und Netscape Navigator nicht unterstützt wird.

Hinzufügen von Extras

Sie haben all die Formularfelder erstellt, die Benutzereingaben akzeptieren. Es ist jetzt an der Zeit, Funktionsvielfalt ins Spiel zu bringen und die Formularfelder ein bisschen freundlicher zu gestalten.

Feldbeschriftungen

Das `<label>`-Tag zeigt nützliche Informationen für Formularfelder im Formular an. Sie sollten das `for`-Attribut des Tag mit dem Formularfeld verknüpfen, das beschriftet werden soll. Um eine Beschriftung zu erstellen, beginnen Sie mit dem einleitenden `<label>`-Tag und fügen dann das `for`-Attribut hinzu. Der Wert dieses Attributs muss, falls vorhanden, an das `id`-Attribut für das zu beschriftende Formularfeld angepasst werden. Geben Sie als Nächstes den Text ein, der als Beschriftung dienen soll, und schließen Sie das Ganze mit dem `</label>`-Tag wie im folgenden Beispiel:

```
<label for="control4">Who is your favorite NFL Quarterback?</label>  
<input type="text" name="favqb" id="control4" />
```

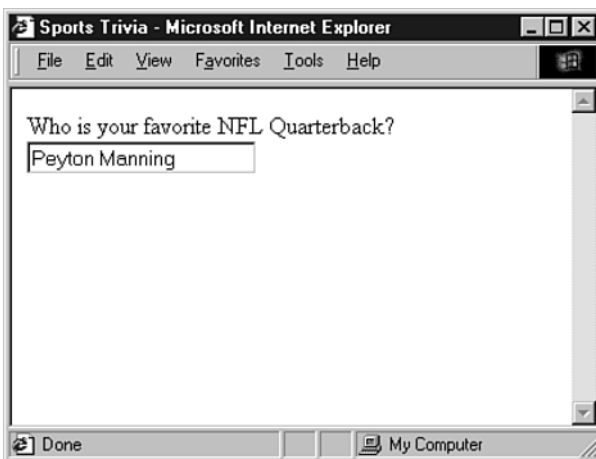


Abbildung 14.15:
Sie können `<label>`-Tags mit jeder Art von Formularfeld verbinden. Beachten Sie, dass die Beschriftungen mit der Kontrollfläche angezeigt werden.



Obwohl das `<label>`-Tag nicht vom Netscape Navigator unterstützt wird, wird der Inhalt, der zwischen den einleitenden und abschließenden `<label>`-Tags steht, angezeigt.

In Abbildung 14.15 sehen Sie diese Text-Kontrollfläche mit Beschriftung.

Wenn Sie ein Formularfeld, wie im folgenden Beispiel gezeigt wird, innerhalb des `<label>`-Tags definieren, können Sie das `for`-Attribut auslassen.

```
<label>User name: <input type="text" name="username" /></label>
```

Gruppieren von Formularfeldern mit `<fieldset>` und `<legend>`

Das `<fieldset>`-Tag organisiert Formularfelder in Gruppen, die im Web-Browser erscheinen. Das `<legend>`-Tag zeigt eine Kopfzeile für das `<fieldset>` an. Um ein `<fieldset>`-Tag zu erstellen, beginnen Sie mit dem einleitenden `<fieldset>`-Tag und fügen Sie dann das `<legend>`-Tag, wie im folgenden Code, hinzu:

```
<fieldset>
  <legend>Oatmeal Varieties</legend>
```



Der Netscape Navigator unterstützt weder `<fieldset>` noch `<legend>`.

Beachten Sie, dass `<legend>` Text enthält; es könnte jedoch ebenso Inline-HTML-Elemente für die Formatierung des Textes enthalten. Geben Sie als Nächstes Ihre Formularfelder ein und beenden Sie das Ganze mit dem abschließenden `<fieldset>`-Tag. In Abbildung 14.16 sehen Sie das Ergebnis.

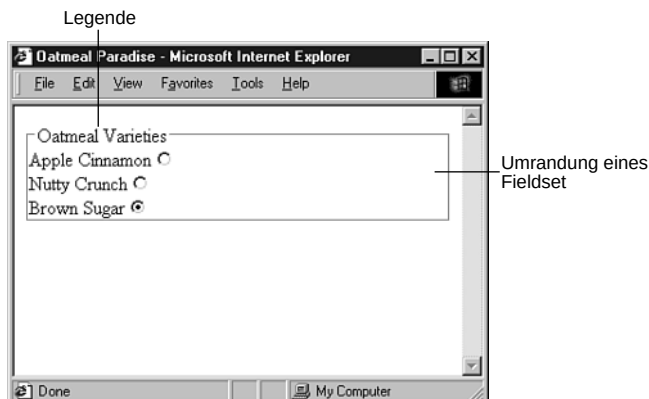


Abbildung 14.16:
Das `<fieldset>` und das
`<legend>`-Tag ermöglichen das Organisieren
Ihrer Formulare.

```
<fieldset>
  <legend>Oatmeal Varieties</legend>
  <label>Apple Cinnamon<input type="radio" name="applecinnamon" /></label><br />
  <label>Nutty Crunch<input type="radio" name="nuttycrunch" /></label><br />
  <label>Brown Sugar<input type="radio" name="brownsugar" /></label>
</fieldset>
```

Ändern der Standardtabulator – Navigation

Bei einigen Computeranwendungen erleichtert die Arbeit mit der Tastatur das Ausfüllen von Formularen. Mit Hilfe der Tabulator-Taste können Benutzer ein Formularfeld nach dem anderen in der von Ihnen angegebenen Reihenfolge anspringen.



Obwohl der Netscape Navigator es Ihnen ermöglicht, mit Hilfe des Tabulators ein Formularfeld nach dem anderen anzuspringen, können Sie die Tabulator-Reihenfolge nicht mit dem `tabindex`-Attribut ändern.

Um die Tabulator-Navigation zu ändern, fügen Sie das `tabindex`-Attribut allen von Ihnen gewünschten Formularfelder hinzu. Beginnen Sie mit 1 und ordnen Sie Ihre Formularfelder vom Anfang bis zum Ende. Formularfelder, die duplizierte `tabindex`-Werte besitzen, erscheinen in der gleichen Reihenfolge, in der sie in HTML-Dokumenten erscheinen. Folgendes Beispiel zeigt, wie man das Attribut einem Formularfeld hinzufügt:

```
<p>Enter your name: <input type="text" name="username" tabindex="1" /></p>
```

Verwenden von Zugriffstasten

Zugriffstasten erleichtern die Navigation in Ihren Formularen. Sie verbinden einen Buchstaben mit einem Element. Ein Feld erhält den Fokus, sobald der Anwender die dafür definierte Tastenkombination drückt. Verwenden Sie folgenden Code, um einer Checkbox eine Zugriffstaste hinzuzufügen:

```
<p>What are your interests?</p>
<p>Sports<input type="checkbox" name="sports" accesskey="S"></p>
<p>Music<input type="checkbox" name="music" accesskey="M"></p>
<p>Television<input type="checkbox" name="tv" accesskey="T"></p>
```



Leider unterstützt der Netscape Navigator den Gebrauch von `accesskey` nicht. `accesskey` wird jedoch vom Internet Explorer unterstützt. Der Benutzer muss in diesem Fall **Alt** zusammen mit der Taste drücken, damit es funktioniert. Andere Betriebssysteme und Web-Browser erfordern eine andere Methode.

Erstellen von <disabled> und <readonly> Formularfeldern

Vielleicht möchten Sie einmal ein Formularfeld anzeigen, ohne dass Ihre Besucher das Feld verwenden oder neue Informationen eingeben können. Um ein Formularfeld auszuschalten, fügen Sie der Definition des Feldes das `disabled`-Attribut hinzu, wie es im folgenden Beispiel gezeigt wird:

```
<p>What is the meaning of life?
<textarea name="question42" disabled>
Enter your answer here.
</textarea>
</p>
```

Wenn die Kontrollfläche nun im Web-Browser angezeigt wird, erscheint es in hellgrauer Farbe oder matt, damit man erkennen kann, dass es ausgeschaltet ist. Zur Erstellung eines Nur-Lese-Formularfeldes verwenden Sie das `readonly`-Attribut, wie es im folgenden Beispiel gezeigt wird:

```
<p>This month: <input type="text" name="month" value="september" readonly /></p>
```

Ein Nur-Lese-Formularfeld wird normal angezeigt; wenn Besucher jedoch versuchen, neue Informationen einzugeben (oder im Falle von Schaltflächen oder Checkboxen eine Wahl treffen), werden sie bemerken, dass sie den Wert nicht verändern können. In Abbildung 14.17 sehen Sie sowohl ein ausgeschaltetes Formularfeld, als auch ein Nur-Lese-Formularfeld.

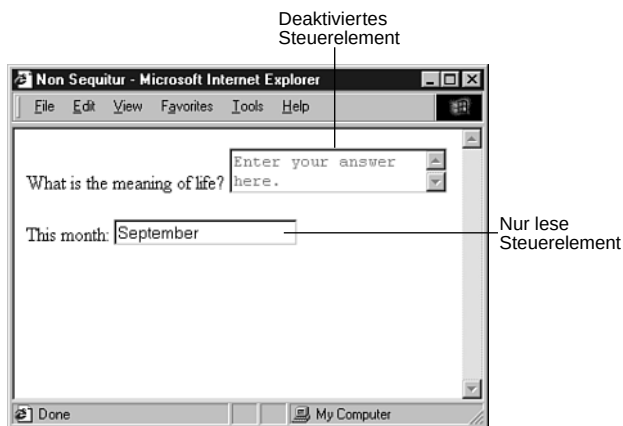


Abbildung 14.17:
Deaktivierte Formularfelder
erscheinen grau untermalt,
während Nur-Lese-Formular-
felder normal erscheinen –
sie können nur nicht verän-
dert werden.



Der Netscape Navigator unterstützt die `disabled` oder `readonly`-Attribute nicht.

Übung 14.3: Erstellen eines Formulars mit erweiterten Formularfeldern

In Ihrer letzten Übung für dieses Kapitel werden Sie die Formularfelder der letzten beiden Unterkapitel in ein großes Formular in einer Matchmaker Webseite integrieren.

Sie beginnen mit einem großen Formular. Öffnen Sie Ihr HTML-Schablonendokument und erstellen Sie ein Formular im Rumpf des Dokuments.

Auf dieser Seite werden Sie zwei Teile erstellen: einen Teil, in dem Fragen über den Besucher gestellt werden, und ein Teil, in dem Fragen über den Grund seines Besuchs auf Ihrer Webseite gestellt werden. Das ist eine Aufgabe für `<fieldset>`-Tags.

Erstellen Sie jetzt das erste `<fieldset>`-Tag. Beginnen Sie mit dem `<form>`-Element und fügen Sie ein `<fieldset>`-Tag und `<legend>` wie im folgenden Beispiel hinzu:

```
<form action="http://www.fakeurl.com/cgi-bin/match.cgi" method="post">
<fieldset>
  <legend>Tell us about yourself.</legend>
```

Fügen Sie eine Beschriftung für das `<select>`-Element und anschließend das `<select>`-Element hinzu, und geben Sie schließlich alle Optionen ein. Wenn Sie schon dabei sind, definieren Sie die `tabindex`- und `accesskey`-Attribute.

```
<label for="control1">I have a funny bone that...</label>
  <select name="humor" id="control1" tabindex="1" accesskey="h">
    <option value="-1">Please pick one</option>
    <option value="-1">-----</option>
    <option value="0">has been removed.</option>
    <option value="1">is my skull.</option>
    <option value="2">is ticklish.</option>
  </select>
```

Das war doch nicht allzu schwierig. Geben Sie nun das zweite `<label>` und `<select>`-Element wie im folgenden Beispiel ein:

```
<label for="control2">I like to take long walks...</label>
  <select name="walks" id="control2" tabindex="2" accesskey="w">
    <option value="-1">Please pick one</option>
    <option value="-1">-----</option>
    <option value="fields">in the corn fields at our farm.</option>
    <option value="streets">on the streets of my city.</option>
    <option value="beach">down the hall in my dorm.</option>
    <option value="apartment">to the refrigerator.</option>
  </select>
```

Vervollständigen Sie das `<fieldset>`, indem Sie ein letztes `<label>` und `<textarea>`-Formularfeld hinzufügen. Vergessen Sie nicht das abschließende `<fieldset>`-Tag.

```
<label for="control3">Why should we care?</label>
  <textarea name="mycomments" id="control3" rows="5" cols="30" tabindex="3"></
textarea>
</fieldset>
```

Sie haben schon fast die Hälfte geschafft. Verwenden Sie folgenden Code, um ein zweites `<fieldset>` zu erstellen, das Fragen zum Grund des Besuchs auf Ihrer Webseite enthält.

```
<fieldset>
  <legend>What are you looking for?</legend>
  <label for="control4">I want...</label>
  <select name="romance" id="control4" size="4" tabindex="4" accesskey="r">
    <option>somebody to watch movies with.</option>
    <option>a serious relationship.</option>
    <option>a Hot-N-Steamy romance.</option>
    <option>someone to clean the catbox.</option>
  </select>
  <label for="control5">Describe your perfect partner.</label>
  <textarea name="lookingfor" id="control5" rows="5" cols="30" tabindex="5"></
textarea>
</fieldset>
```

Nachdem die beiden `<fieldsets>` nun vervollständigt sind, fügen Sie ein deaktiviertes und ein Nur-Lese-Formularfeld hinzu, um sicher zu gehen, dass Sie alle Grundlagen erstellt haben.

```
Check if you want to join our club.<input type="checkbox"
name="single" checked disabled />
Do you agree to pay us for this service?<input type="text"
name="pay" value="yes" readonly />
```

Fügen Sie nun die Submit und Reset-Schaltflächen hinzu, und schließen Sie das `<form>`-Element. Beachten Sie, dass ich mit Hilfe des `<button>`-Tag spezielle Grafiken für die Schaltflächen einfüge.

```
<button name="submit" type="submit" tabindex="6">

</button>
<button name="reset" type="reset" tabindex="7">

</button>
</form>
```

Da Ihr Formular nun fertig ist, können Sie am Seitenanfang eine Titelgrafik hinzufügen und die meisten Ihrer Formularfelder in eine Tabelle einbinden. Das vollständige Listing zeigt das Endergebnis:



```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>
<head>
<title>Matchmaker Matchmaker</title>
</head>
<body>
<div align="center">

</div>
<form action="http://www.fakeurl.com/cgi-bin/match.cgi" method="post">
<table border="0" width="100%">
  <tr valign="top">
    <td width="50%">
      <table border="0">
        <tr>
          <td align="center">
            <fieldset>
              <legend>Tell us about yourself.</legend>
              <label for="control1">I have a funny bone that...</label><br />
              <select name="humor" id="control1" tabindex="1" accesskey="h">
                <option value="-1">Please pick one</option>
                <option value="-1">-----</option>
                <option value="0">has been removed.</option>
                <option value="1">is my skull.</option>
                <option value="2">is ticklish.</option>
              </select>
            <br />
            <label for="control2">I like to take long walks...</label>
            <select name="walks" id="control2" tabindex="2" accesskey="w">
              <option value="-1">Please pick one</option>
              <option value="-1">-----</option>
              <option value="fields">in the corn fields at our farm.</option>
              <option value="streets">on the streets of my city.</option>
              <option value="beach">down the hall in my dorm.</option>
              <option value="apartment">to the refrigerator.</option>
            </select>
            <br />
          </td>
        </tr>
      </table>
    </td>
  </tr>
</table>
</form>
```

```

        <label for="control3">Why should we care?</label>
        <br />
        <textarea name="mycomments" id="control3" rows="5" cols="30"
tabindex="3"></textarea>
    </fieldset>
</td>
</tr>
</table>
</td>
<td width="50%">
    <table border="0">
        <tr>
            <td align="center">
                <fieldset>
                    <legend>What are you looking for?</legend>
                    <label for="control4">I want...</label>
                    <br />
                    <select name="romance" id="control4" size="4" tabindex="4" accesskey="r">
                        <option>somebody to watch movies with.</option>
                        <option>a serious relationship.</option>
                        <option>a Hot-N-Steamy romance.</option>
                        <option>someone to clean the catbox.</option>
                    </select>
                    <br />
                    <label for="control5">Describe your perfect partner.</label>
                    <textarea name="lookingfor" id="control5" rows="5" cols="30"
tabindex="5"></textarea>
                </fieldset>
            </td>
        </tr>
        </table>
    </td>
</tr>
<tr>
    <td colspan="2">Check if you want to join our club.<input type="checkbox"
name="single" checked disabled /></td>
    <td></td>
</tr>
<tr>
    <td colspan="2">Do you agree to pay us for this service?<input
type="text" name="pay" value="yes" readonly /></td>
    <td></td>
</tr>
</table>
<div align="center">
<table border="0">

```



```
<tr>
  <td><button name="submit" type="submit" tabindex="6"></button></td>
  <td><button name="reset" type="reset" tabindex="7"></button></td>
</tr>
</table>
</div>
</form>
</body>
</html>
```

In Abbildung 14.18 sehen Sie das Endergebnis dieser Übung.



Textbereich

<select> als Kombinationslistenfeld

<select> als Listenfeld

Schaltflächen mit Grafiken

Nur-Lese

Deaktiviert

Abbildung 14.18:
Hinzufügen von
Extras in einem
Matchmaker-Formular

Planen Sie Ihre Formulare

Bevor Sie anfangen, komplexe Formulare für Ihre Webseite zu erstellen, sollten Sie das Ganze planen, um sich auf Dauer gesehen Zeit und Ärger zu ersparen.

Entscheiden Sie zuerst, welche Informationen Sie benötigen. Das mag zwar selbstverständlich klingen, aber Sie müssen sich mit den Methoden zur Formularerstellung dann auseinandersetzen, wenn Sie noch keine Probleme damit haben.

Gehen Sie als Nächstes diese Information noch einmal durch und wählen Sie zu jedem Element ein entsprechendes Formularfeld. Fragen Sie sich selbst, welches Formularfeld am besten zu welcher Frage passt. Wenn Sie eine ja/nein Antwort möchten, sind Optionsfelder oder Checkboxes ideal, während das `<textarea>`-Tag zuviel des Guten ist. Versuchen Sie dem Besucher, der Ihr Formular ja ausfüllen soll, das Leben leichter zu machen, indem Sie ein Formularfeld erstellen, das auch zu der Frage passt. So wird das Analysieren einer Information mittels eines Scripts (wenn nötig) auch viel einfacher sein.

Sie sollten sich außerdem mit der Person, die das CGI-Script schreibt, die Feldnamen besprechen und die übrige Struktur mit dieser besprechen.

Es ist keine große Arbeit, jedes Formularfeld vor der Zusammenarbeit mit dem Script-Autor mit einem Namen zu versehen. Sie können auch Tabellen erstellen, die ausführliche Beschreibungen sowie die verfügbaren Werte von jedem Formularfeld enthalten.

Schließlich werden Sie vielleicht noch das Validieren von Formular-Inputs durch Scripting in Betracht ziehen. Mit einem clientseitigen Script – die Verarbeitung erfolgt also im Browser des Benutzers – können Sie die Benutzereingaben noch vor der Übermittlung zum Server prüfen. Auf diesem Weg stellen Sie sicher, dass das Formular erst dann an den Server übermittelt wird, wenn es vollständig ist. Sie können Script-Funktionen in jedes Formular-Element integrieren, wenn Sie das `<submit>`-Button integrieren und Immanent-Tags wie `<onsubmit>` verwenden.

Zusammenfassung

Wie Sie sehen, ist die wundervolle Welt der Formulare voller verschiedener Formularfelder, mit denen der Besucher Ihrer Webseite in Interaktion treten kann. Das ist doch wirklich DIE Möglichkeit, Ihre Webseite interaktiv zu machen.

Seien Sie trotzdem vorsichtig. Web-Surfer, die ununterbrochen mit den verschiedensten Formulartypen bombardiert werden, werden vom ständigen Seitenwechsel schnell müde. Sie müssen Ihren Besuchern Gründe geben, warum Sie ausgerechnet Ihre Webseite so interessant finden!

In Tabelle 14.1 sehen Sie eine Zusammenfassung der HTML-Tags, die in diesem Kapitel verwendet wurden. Denken Sie immer an folgende Punkte, und Ihnen kann nichts passieren:

- ▶ Verwenden Sie das `<form>`-Element zur Erstellung von Formularen
- ▶ Fügen Sie immer das `action`-Attribut in einem Formular hinzu
- ▶ Erstellen Sie Formularfelder mit dem `<input>`-Element oder anderen Formularfeld-Tags
- ▶ Testen Sie Ihre Formulare ausführlich

Tag	Verwendung
<code><form></code>	Erstellt ein HTML-Formular. Sie können innerhalb eines Dokuments mehrere Formulare erstellen, aber Sie können Formulare nicht verschachteln.
<code>action</code>	Ein <code>form</code> -Attribut, das das Server-Script angibt (mit einem URL-Pfad). Dieses Server-Script verarbeitet die Formulardaten.
<code>method</code>	Ein <code>form</code> -Attribut, das definiert, wie die Formulardaten zum Server gesendet werden. Erlaubte Werte sind <code>get</code> und <code>post</code> .
<code><input></code>	Ein <code><form></code> -Element, das Formularfelder für Benutzer-Inputs erstellt.
<code>type</code>	Gibt den Typ des Formularfeldes an. Erlaubte Werte sind <code>text</code> , <code>password</code> , <code>submit</code> , <code>reset</code> , <code>checkbox</code> , <code>radio</code> , <code>image</code> , <code>button</code> , <code>hidden</code> und <code>file</code> und werden in folgender Liste erklärt: <code>text</code>: erstellt ein einzeliges Texteingabefeld <code>password</code>: erstellt ein einzeliges Textfeld, in dem die Benutzereingaben maskiert dargestellt werden. <code>submit</code>: erstellt eine Submit-Schaltfläche, die die Formulardaten zum Server sendet. <code>reset</code>: erstellt eine Reset-Schaltfläche, die alle Formularfelder auf ihre Anfangswerte zurücksetzt. <code>checkbox</code>: erstellt eine Checkbox <code>radio</code>: erstellt ein Optionsfeld <code>image</code>: macht aus einem Bild eine Submit-Schaltfläche <code>button</code>: erstellt eine Schaltfläche. Diese hat für sich noch keine Funktion. <code>hidden</code>: erstellt ein verstecktes Formularfeld, das vom Verwender nicht gesehen werden kann. <code>file</code>: erstellt ein Datei-Upload-Formularfeld, dass Benutzern die Möglichkeit gibt, eine Datei über das Formular auszuwählen. Diese Datei wird dann mit dem Formular zum Server geschickt.
<code><button></code>	Erstellt eine Schaltfläche, die HTML-Inhalt enthalten kann.

Tag	Verwendung
<textarea>	Ein mehrzeiliges Texteingabefeld
<select>	Eine Auswahlliste. Die einzelnen Elemente werden vom <option>-Tag bestimmt.
<option>	Einzelne Elemente innerhalb eines <select>-Tag.
<label>	Erstellt eine Beschriftung, die mit einem Formularfeld verknüpft ist.
<fieldset>	Organisiert Formularfelder in Gruppen.
<legend>	Zeigt eine Überschrift für ein <fieldset>-Tag an.

Workshop

Ich denke, Sie werden bis hierher immer noch einige Fragen haben. Ich habe deshalb die Fragen aufgelistet, die Sie meiner Meinung nach am meisten interessieren könnten. Überprüfen Sie zuerst einmal in einem Quiz das, was Sie behalten haben, und erweitern Sie Ihr Wissen mit einigen Übungen.

Fragen&Antworten

- F Ich möchte ein Formular definieren, dass später ganz einfach in einer Tabellenkalkulation oder einer Datenbank analysiert werden kann. Wie soll ich mein Formular definieren, um die ganze Sache etwas einfacher zu machen?**
- A** Stellen Sie zuerst einmal sicher, dass Sie einen Standardwert für jedes Formularfeld in Ihrem Formular haben. Das legt die Anzahl der Dateneingaben fest und bewirkt, dass diese Dateneingaben viel einfacher in eine Tabellenkalkulation übernommen werden können. Scheuen Sie nicht die Verwendung von Werten, die beim Programmieren manipuliert werden können. Ich könnte zum Beispiel mit Hilfe des *value*-Attributs jede Ja-Antwort in meinem Formular durch 1 und jede Nein-Antwort durch 0 ersetzen. Wenn ich die Daten dann in meine Tabellenkalkulation einbinden würde, könnte ich die Antworten in Tabellen und Schaubildern mit Einsern und Nullern anstatt Text ausdrücken. Sie können auch noch viele andere Tricks anwenden, wie zum Beispiel das Verwenden von Tabellen in Ihren Scripts zur Dekodierung von Formular-Antworten. Experimentieren Sie ein wenig, um herauszufinden, was für Sie und Ihre Daten am besten funktioniert.

F Ich möchte ein Formular erstellen und es testen, aber ich besitze nicht die entsprechenden Scripts. Gibt es die Möglichkeit, mit einem Arbeits-script feststellen zu können, ob das Formular die richtige Information sendet?

A *Ich gerate andauernd in diese Situation! Gott sei Dank ist das Umgehen des Problems sehr einfach.*

Ändern Sie das *action*-Attribut innerhalb des einleitenden *<form>*-Elements und erstellen Sie ein *<mailto>*-Link zu Ihrer E-Mail-Adresse, wie es im folgenden Beispiel gezeigt wird:

```
<form action="mailto:ihreemailadresse@rsp.com"method="post">
```

Jetzt können Sie Ihr Testformular vervollständigen und es auch ohne dem entsprechenden Script übermitteln. Wenn Sie Ihr Formular übermitteln, wird es Ihnen als Anhang gemailt. Öffnen Sie einfach den Anhang in einem Text-Editor, et voilà! Ihre Formulardaten sind da.

F Ich habe gehört, dass über HTML 4.01 diskutiert wird. Wird es Änderungen für Formulare geben?

A *Eine Änderung von HTML 4.01 wurde momentan vom World Wide Web Consortium vorgeschlagen. Formularen stehen keine größeren Veränderungen bevor. Folgende kleine Veränderungen werden in Betracht gezogen:*

- ▶ *Hinzufügen des *name*-Attributs zum *<form>*-Element. Diese Veränderung würde die Abwärtskompatibilität herstellen. Über diesen Namen könnten Sie in Ihren Scripts auf Ihre Formulare zugreifen.*
- ▶ *Entfernen der Referenz und des Beispiels des *mailto*-Werts des *action*-Attributs für das *<form>*-Element. Es ist unklar, ob es dazu führen würde, dass *mailto* als Aktion nicht mehr unterstützt würde. Der Text des Änderungsvorschlags gibt an, dass alle Maßnahmen, dieses Feature zu implementieren, beim Browser liegen.*
- ▶ *Hinzufügen des fehlenden *ismap*-Attributs in das *<input>*-Element, um Unterstützung für Server-Bildermappen zu erhalten.*
- ▶ *Hinzufügen einer Randnotiz für das *<select>*-Element, die Folgendes erklärt: Wenn keine Option für das *<select>*-Element ausgewählt wurde, werden keine Daten für dieses Element zum Server gesendet, wenn das Formular übertragen wird.*

Quiz

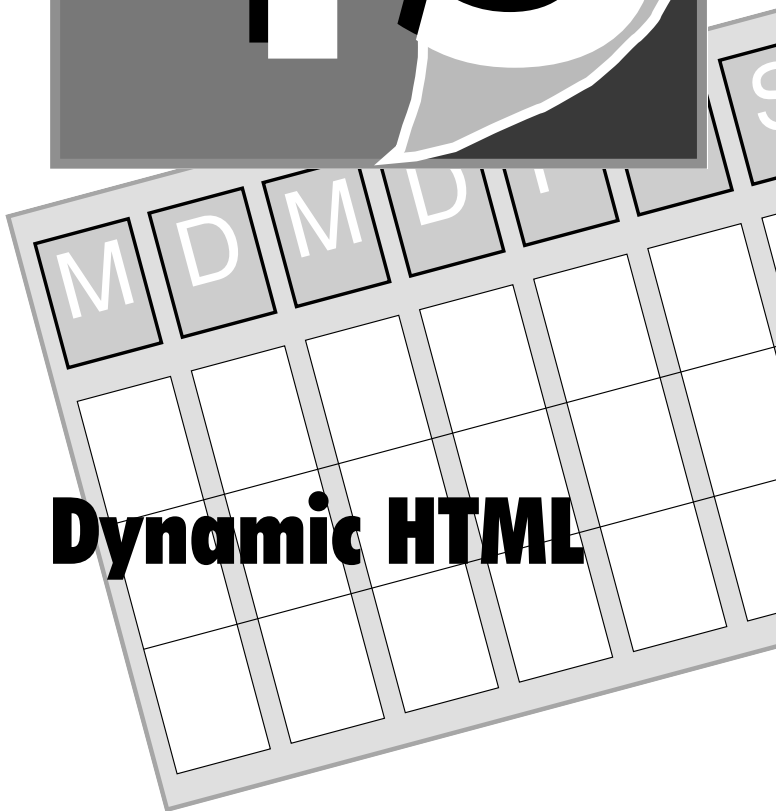
1. Wie viele Formulare kann es auf einer Webseite geben?
2. Wie erstellt man Formularfelder, Radio-Schaltflächen und Checkboxes?
3. Sind gesendete Passwörter, für die es ein `<password>`-Formularfeld gibt, sicher?
4. Erklären Sie den Vorteil des Verwendens von versteckten Formularfeldern.
5. Auf welchen anderen Technologien beruhen Formulare auch noch?

Antworten

1. Theoretisch sind Sie nur in Bezug auf den Speicherplatz des Client-Computers eingeschränkt. Dennoch können Sie Formulare nicht verschachteln. Ein anderer interessanter Aspekt ist, dass jedes Formularfeld Speicherplatz benötigt. Ich bin nur allzu oft in diese heikle Situation geraten, in der ich den ganzen freien Speicherplatz eines Computers verbraucht habe, weil ich Formularfelder eingefügt habe.
2. Diese Formularfelder werden mit dem `<input>`-Element erstellt und können einzeln gar nicht existieren.
3. Nein!
4. Versteckte Formularfelder sind eigentlich mehr etwas für Sie, als für denjenigen, der das Formular ausfüllt. Indem Sie einzelne `value`-Attribute verwenden, können Sie zwischen verschiedenen Formularen, die zum gleichen Script oder zu verschiedenen Zeiten gesendet werden sollen, unterscheiden.
5. Um die Formulardaten bearbeiten zu können, müssen diese mittels des `action`-Attributs mit einem Server-Script in Verbindung stehen. Diese Scripts werden in Perl oder anderen Formatiersprachen geschrieben.

Übungen

1. Klären Sie mit Ihrem Internet Service Provider, ob er Scripts zur Verfügung stellt, die Sie für die Verarbeitung Ihrer Formulare verwenden können. Wenn Sie die Scripts verwenden können, fragen Sie wie die Daten verarbeitet werden und welche Namen in Ihren Formularfeldern verwendet werden können. Wenn Sie Formulare verwenden wollen und Ihr ISP Ihnen das Benutzen von Scripts nicht erlaubt, sollten Sie sich woanders umsehen, um Ihre Web-Site unterzubringen.
2. Besuchen Sie verschieden Sites, die Formulare verwenden, und sehen Sie sich diese Formulare genau an, wie zum Beispiel www.fedex.com. Achten Sie darauf, welche Formularfelder dort verwendet werden, wie sie angeordnet sind und werfen Sie einen Blick auf den HTML-Code.



**Woche
3**

Gestern haben Sie gelernt wie man umfangreiche HTML-Formulare erstellt, die Ihnen helfen Informationen von Besuchern Ihrer Web-Site zu erhalten. In der ein oder anderen »Form« gibt es diese Formulare schon seit der virtuellen Geburt von HTML und sind mehr oder weniger mit den Browsern kompatibel. Kümmern Sie sich also nicht zu sehr darum, wer Ihre Site nun richtig angezeigt bekommt oder nicht.

Die heutige Lektion beginnen wir in den relativ ruhigen Gewässern der Formulare. Dann heißt es aber Segel hissen auf der rauen See der Dynamic HTML!

Dynamic HTML bietet Ihnen etwas Einzigartiges, nämlich die Möglichkeit beliebige Elemente der WWW-Seite während der Anzeige dynamisch zu verändern, sei es nun automatisch oder durch Einwirken des Anwenders. Bei konventionellem HTML, werden die Webseiten von einem Browser geladen und bleiben dort, bis Sie ein Link anklicken oder mit Formularen interaktiv einwirken. Das ist zwar alles sehr einfach, kann allerdings auch sehr schnell langweilig werden.

Also hat man sich daran gemacht, die statischen Webseiten zu verbessern, und zwar mit Technologien, die nicht aus dem »Königreich« des HTML stammen. CGI zum Beispiel war lange Zeit die bevorzugte Methode für die Erzeugung dynamischer Web-Sites. Die Einbettung interaktiver Animationen ließen und lassen sich mit Shockwave und Flash aufbauen. Außerdem konnte man Java Applets (Softwarekomponente, die in der Regel nicht eigenständig benutzt wird, sondern das Leistungsspektrum eines Programms ergänzt) oder ActiveX-Controls benutzen, um programmtechnische Funktionen zu integrieren. Diese Methoden sind aber von Browser Plugins und virtuellen Maschinen abhängig. Was sich wiederum auf die Ladezeit auswirken kann.

Mit Dynamic HTML, bekannt als DHTML, ist das ganz anders. Über diese Technik können Sie den Besuchern Ihrer Web-Site ein Look & Feel bieten, wie sie das von anderen Programmen auf dem Computer gewohnt sind. Der Web-Browser wird als Gerüst für die Benutzerschnittstelle verwendet, die Funktionalität stützt sich auf die extern programmierte Lösung. Können Sie schon ahnen, welche Vorteile DHTML bietet?

Hoffentlich, denn es bedarf schon einer großen Portion Begeisterung, um die »dunkle Seite« des DHTML zu meistern. Ehrlich gesagt, kann es bei dieser »dunklen Seite« manchmal ganz schön »drunter und drüber« gehen.

Um ein möglichst breitgefächertes Publikum anzusprechen (innerhalb praxisorientierter Grenzen), müssen Sie die verschiedenen Web-Browser und die Art und Weise, wie sie verschiedene Web-Technologien einbinden (wie etwa HTML, Cascading Style Sheets und Scripting) kennen.

Machen Sie sich also darauf gefasst, dass Sie heute mit Informationen bombardiert werden. Fangen wir gleich damit an! Heute werden folgende Themen behandelt:

- ▶ Definiton von Dynamic HTML und verschiedene DHTML-Technologien.
- ▶ Grundzüge von JavaScript

- ▶ Was sind Document Object Models?
- ▶ Erstellen von Browser-übergreifenden (cross-Browser) Routinen mit DHTML, wie etwa Browsererkennung und DOM Referenzen (Document Object Model; Definition einer Gruppe von Objekten, die ein Dokument im Browser repräsentiert).
- ▶ Erstellen eines erweiterten Menüs und eines netten Tic-Tac-Toe-Spiels mit DHTML

Was genau ist denn Dynamic HTML?

Einfach ausgedrückt, benutzt Dynamic HTML normale HTML-Elemente zur Erstellung von Webseiten, die sich beim Formatieren, Positionieren und Scripting auf Style-Sheets stützen. Dabei muss die Seite vom Server nicht jedesmal vollständig geladen werden. Dynamic HTML darf man nicht als eigenständiges »Ding« verstehen, sondern als eine Sammlung von Technologien, die zusammenarbeiten, um interaktive Effekte zu erzielen. Wozu wird DHTML nun verwendet? Es gibt schier unerschöpfliche Möglichkeiten bei der Kontrolle über das Geschehen am Bildschirm. Hier sind einige davon, die Sie hoffentlich auf den Geschmack bringen werden.

- ▶ Objekte können beliebig auf Ihrer Webseite platziert werden.
- ▶ Objekte können angezeigt oder verdeckt werden.
- ▶ Erstellen von Listen, die durch die Auswahl von Elementen erweitert, bzw. verkürzt werden können.
- ▶ Farbe und Größe des Inhalts Ihrer Webseite kann dynamisch verändert werden.
- ▶ Drag-and-Drop Funktionen, die denen der modernen Betriebssystemen mit grafischer Oberfläche ähnlich sind (Windwos, Macintosh).

Die Geburt von DHTML erfolgte zur gleichen Zeit, wie der Beginn der Ära der großen »Vier«. Das World Wide Web Konsortium entwickelte HTML 4 und veröffentlichte deren offizielle Empfehlung ungefähr zur gleichen Zeit als Netscape und Microsoft die konkurrierenden Web-Browser, Internet Explorer 4 und Navigator 4 auf den Markt brachten. Leider ist das Thema »Dynamic HTML« jedoch zu einem Kriegsschauplatz im Duell der beiden führenden WWW-Browser geworden. Große Werbekampagnen wurden für HTML 4 gestartet, wobei vorallem betont wurde, dass die neuen Features nicht nur den Erwartungen der Anwender gerecht werden, sondern sich auch besser in CSS integrieren werden. Für die Marktanteile lieferten sich Microsoft und Netscape einen unerbittlichen Kampf im fürchterlichen Browser-Krieg. Das Resultat war aber ein ganz anderes: Anwender, die ihre Webseiten interaktiv und dynamisch gestalteten. Bis dahin war das ohne Rückgriff auf externe Programme überhaupt nicht möglich. DHTML wurde geboren.



Dynamic HTML ist eine Kombination von Web-Technologien zur dynamischen und interaktiven Gestaltung von Webseiten, ohne dabei auf externe Objekte oder Programme zurückgreifen zu müssen.

Für das Arbeiten mit DHTML bedarf es drei von den Web-Browsern unterstützten Techniken:

- ▶ HTML
- ▶ Style-Sheets
- ▶ Scripting

Natürlich müssen Sie auch weiterhin HTML benutzen. Vergessen Sie aber nicht, dass HTML 4 neue Elemente, wie etwa `<div>` und `<span>`, und neue Attribute, wie `id` und `style`, hat. Diese tragen zur höheren Integration der Style Sheets und Manipulation mit Scripting beim Strukturieren und Formatieren Ihres Inhalts bei. Der Nachteil aber ist, dass Web-Browser heutzutage immer noch nicht systematisch mit HTML 4 ausgestattet werden. Das wird uns auch leider das ganze Kapitel hindurch beschäftigen.



Die größte Herausforderung für DHTML sind die Widersprüche zu überwinden, – nicht nur in HTML selbst, sondern in allen DHTML-Technologien – und dynamische Browser-übergreifende Webseiten zu erstellen. Mit viel Mühe ist dies tatsächlich zu schaffen.

Wenn man erst einmal weiß, wie man Style Sheets anwenden muss, kann man sich ein »Leben« ohne sie gar nicht mehr vorstellen. Sie sind eine entscheidende Komponente des DHTML. Obwohl DHTML technisch gesehen von keiner bestimmten Style-Sheet-Technologie abhängig ist, ist die offizielle W3C Style-Sheet-Technologie, die Cascading Style Sheets, dennoch der Standard. Mit CSS Level 2 (CSS2) können Sie nicht nur Elemente Ihrer Webseite formatieren und Eigenschaften wie Schriftart, Farbe, Abstand verwenden, sondern auch Elemente auf Ihrer Seite positionieren. Wenn Sie etwas über CSS nachschlagen wollen, schlagen Sie im Kapitel des 10. Tages nach.

Mit DHTML rücken Sie in die nächste Ebene des Web-Designs vor. Anstatt eine Style-Regel zu erstellen oder ein Element auf Ihre Seite zu platzieren und dann wieder *alles zu vergessen*, können Sie DHTML zur dynamischen Änderung der visuellen Styles oder zum Positionieren Ihrer Elemente anwenden. Wie mit HTML auch, gibt es mit DHTML Probleme beim Einbinden der CSS in den Internet Explorer und Netscape Navigator. Ihre Angaben werden nicht nur unvollständig eingebunden, sondern jeder Browser interpretiert unterschiedliche Teile auch noch anders. Das kann ja heiter werden.....

Kommen wir nun zum Scripting. Scripting muss man sich wie einen Kleister vorstellen, der alles in DHTML zusammenhält und für das »Dynamische« in DHTML sorgt.

Scripts laufen nämlich selbst dann noch weiter, wenn zum Beispiel eine Seite zur gleichen Zeit heruntergeladen wird. Selbst wenn auf eine Aktion des Anwenders reagiert wird oder wenn der Anwender die Site verlässt, laufen die Scripts noch weiter.



Web-Browser sind bei der Implementierung inkonsequent. Sie sollten wissen, dass es immer gefährlich ist, sich auf »offizielle« Angaben zu beziehen, bzw. ihnen zu vertrauen. Lesen Sie immer die Browser-spezifischen Dokumentationen und versichern Sie sich, dass der Web-Browser, den Sie verwenden wollen, auch alles unterstützt, was Sie zu tun gedenken.

Die Script-Sprache, die am meisten verwendet wird, ist JavaScript. JavaScript war die erste Script-Sprache, die im Web angewandt wurde und wird auch heute noch von den meisten Web-Browsern unterstützt. Obwohl Sie bei DHTML auch andere Script-Sprachen verwenden können (zum Beispiel VBScript), empfehle ich Ihnen immer JavaScript zu benutzen, es sei denn Sie arbeiten ausschließlich mit dem Microsoft Internet Explorer.

Wenn wir erst einmal bei den DHTML Beispielen angelangt sind, werden Sie sehen, dass DHTML mit Scripten arbeitet, um unter anderem Style und Position der Elemente zu verändern und Berechnungen auszuführen. Diese Scripten machen DHTML erst »dynamisch«. Deswegen wurde diesem Kapitel eine kurze Einführung in JavaScript beigelegt. Darin werden Sie zwar nicht alles über JavaScript lernen, doch wenigstens werden Sie die aufgeführten Beispiele verstehen und nachvollziehen können.

Noch zwei abschließende Bemerkungen über das, was DHTML eigentlich ist. Der Scripting-Teil von DHTML ist auf so genannte DOM (Document Object Model) angewiesen. Damit können Objekte einer Webseite identifiziert, erstellt und manipuliert werden. Sie müssen zum Beispiel in der Lage sein, ein Element zu identifizieren (zum Beispiel ein Bild), um es dann mit einem Script manipulieren zu können. Genau dies tut DOM, indem es eine Brücke zwischen dem Inhalt der Webseite und dem Script schlägt. Obwohl das W3C momentan mit der offiziellen DOM-Spezifikation arbeitet, verfügt jeder Web-Browser über ein eigenes DOM. Deshalb werden wir die DOMs etwas später in diesem Kapitel separat behandeln.

DHTML beruht auf Event-Handling (Ereignis-Behandlung). Hierbei handelt es sich um HTML-Attribute, die eine Verknüpfung zu einer Scriptsprache herstellen. Ein Ereignis ist zum Beispiel ein Mausklick oder ein Tastendruck des Anwenders. Ein Element, das einen Event-Handler enthält, kann auf ein solches Ereignis reagieren, wenn das Ereignis im Anzeigebereich des Elements stattfindet. Wie mit allem anderen auch in DHTML, handhaben der Microsoft Internet Explorer und der Netscape Navigator dieses Attribut auf unterschiedliche Art und Weise. Event-Handling wird aber später noch eingehender behandelt.

JavaScript verstehen

JavaScript ist eine Scriptsprache, die ursprünglich von Netscape und Sun Microsystems (den Gründern von Java) entwickelt wurde und dazu dient, die Möglichkeiten von HTML zu erweitern. HTML ist nur eine Dokumentauszeichnungssprache, die unfähig ist, Programmieraufgaben zu erfüllen. Nehmen Sie folgenden HTML-Code:

```

```

In diesem Fall gibt der HTML-Code dem Browser nur den Befehl, das Bild myImage.gif auf der Seite anzuzeigen und zwar in einem 100 Pixel hohem und 300 Pixel breitem Bereich. Eigentlich handelt es sich hierbei um eine ziemlich strenge Eingrenzung, denn außer Hyperlinks und Formkontrolle können sie weder Information, Daten oder Objekte manipulieren, geschweige denn auf Anwender-Ereignisse reagieren.



JavaScript ist eine Scriptsprache, die es Ihnen erlaubt Scripts oder kleinere nicht compilierte Programme zu schreiben, die von einem Web-Browser innerhalb der Webseite integriert werden.

Scriptsprachen, insbesondere JavaScript, sind reine Programmiersprachen und erfüllen eine komplett andere Rolle als HTML. Das bedeutet, dass Scripts nur Berechnungen ausführen, Objekte manipulieren und auf eine Vielzahl von Anwender-Ereignisse reagieren können. Scripts erfüllen Aufgaben wie zum Beispiel die Validierung in einem HTML-Formular, auf Mausklicks und auf das Drücken von Tasten der Tastatur reagieren, sowie Style und Position der HTML-Elemente dynamisch verändern.

Anders als Programme in Programmiersprachen, wie C/C++ oder Java, werden Programme in Scriptsprachen nicht vor der Ausführung in einen von der Maschine lesbaren Code compiliert. Sie sind als normaler Text geschrieben, der von einem Interpreter erst bei der Ausführung des Programms in die entsprechenden Maschinen-sprachenbefehle übersetzt wird. Bei Webseiten übernimmt der Web-Browser die Rolle des Interpreters und interpretiert Scripts innerhalb der Webseiten bzw. in externen Script-Dateien, während diese geladen werden.



Ich möchte noch einmal auf einen wichtigen Punkt hinweisen. So wie ältere Browser viele HTML 4 Elemente nicht anzeigen können, können nicht alle Browser die neueste JavaScript Version interpretieren. Denken Sie daran: Der Browser übernimmt die Rolle des Scriptinterpreters. Wenn ein Browser ein Script nicht versteht, kann man nichts machen.

Erläuterungen zu JavaScript

Leider kann man in einem so begrenzten Rahmen, wie der unseres Buches, JavaScript nicht ausführlich behandeln. Ich werde jedoch versuchen Ihnen, zumindest ansatzweise, JavaScript und seine Funktionsweise verständlich zu machen. Danach sollten Ihnen die Codes in dieser Lektion und auch das, was gemacht wird, keine Probleme mehr bereiten.

Grundsätzlich sind Scripts nichts anderes als eine Reihe von Anweisungen, die dem Scriptinterpreter (dem Web-Browser) mitteilen, was zu tun ist. Diese Anweisungen können einfache Anweisungen sein, die Variablen Werte zuweisen. Beispiel:

```
var x = 14;
```

Oder auch – wie im folgenden Beispiel – eine Reihe von Anweisungen, die Funktionen definieren sollen:

```
function init() {  
    if (x == 14) {  
        alert(x);  
    }  
    else {  
        alert("X doesn't equal 14");  
    }  
}
```

Jede Anweisung endet mit einem Strichpunkt. Anweisungen können auch in einem Block innerhalb von geschweiften Klammern gruppiert werden. Der Code oben verwendet Klammern, um Anweisungen in einer Funktion namens `init` zu gruppieren. Die `if-else`-Anweisungen haben beide Klammern, welche die Anweisungen darunter zu einem Block zusammenfassen.

Sie können Informationen in JavaScript-Script, die nicht interpretiert werden, einfügen. Solche Informationen nennt man Kommentare (Comments), die dazu benutzt werden, Stellen des Scripts zu erklären. Einzeilige Kommentare werden mit zwei Schrägstrichen (`//`), die am Anfang einer Zeile stehen, eingegeben.

Beispiel:

```
// Diese Variable wird verwendet, um temporäre Informationen zu speichern  
var x;
```

Sie können sie aber auch am Ende einer Anweisung platzieren

Beispiel:

```
var x; // Temporäre Variable
```

Der Script-Interpreter wird alles, was nach dem doppelten Schrägstrich kommt, ignorieren. Seien Sie also beim Eingeben Ihrer Codes vorsichtig!

Einzeilige Kommentare können Sie verwenden, um einen größeren Kommentarblock zu erstellen.

Beispiel:

```
// Script created on 9 Oct 99
// Copyright Sams Publishing
// All rights reserved
```

Oder aber sie verwenden einen speziellen mehrzeiligen Kommentar. Einen Kommentar leiten Sie mit `/*`, also einem Schrägstrich, gefolgt von einem Sternzeichen, ein. Mit der umgekehrten Folge `*/`, also einem Sternchen, gefolgt von einem Schrägstrich, beenden Sie den Kommentar. Folgendes Beispiel wird Ihnen diese Art mehrzeiliger Kommentare verdeutlichen.

```
/* Script created on 9 Oct 99
   Copyright Sams Publishing
   All rights reserved */
```

Variablen (Speicherbereiche, in denen Sie Daten, die Sie im Laufe Ihrer Programmprozeduren benötigen, speichern können) sind mit die meistgenutzten Elemente eines Programmes. Der Inhalt, der in einer Variablen gespeichert ist, wird als Wert bezeichnet. Mit Variablen können Sie Werte Platzhaltern zuweisen, um später dann Daten dieser Platzhalter zu manipulieren. Um eine Variable zu erstellen, müssen Sie nur `var`, gefolgt vom Namen des Platzhalters (der Name, an dessen Stelle später Daten eingefügt werden) angeben.

Beispiel:

```
var myVariable;
```



JavaScript ist case-sensitive, dass bedeutet, dass `myVariable` sich von `myvariable` und `Myvariable` unterscheidet.

Es ist ganz einfach, eine Variable anzugeben und sofort einen Wert in ihr zu speichern.

```
var myVariable = 1999;
```

JavaScript verfügt ebenfalls über Operatoren für mathematische Berechnungen und Vergleiche. Tabelle 15.1 listet die gebräuchlichsten Operatoren und ihre jeweilige Funktion auf.

Kalkulation		Komparativ	
Symbol	Beschreibung	Symbol	Beschreibung
+	Addition	=	Bestimmung der Werte
-	Subtraktion	==	gleich
*	Multiplikation	<	weniger als
/	division	>	größer als
++	Inkrement (+1)	<=	weniger oder gleich
--	dekrement (-1)	>=	größer oder gleich
		!=	ungleich
		&&	logisches AND
			logisches OR

Tabelle 15.1: Die gebräuchlichsten Operatoren

Der folgende Code deklariert drei Variablen, weist zwei davon einen Wert zu und weist dann die Summe der beiden ersten Variable der dritten Variablen zu.

```
var xPos = 142;
var yPos = 15;
var Pos;
Pos = xPos + yPos;
```

Um einen Wert zu inkrementieren oder zu dekrementieren, verwenden Sie den entsprechenden Operator. Dieser kann entweder vor dem jeweiligen Wert oder der Variablen bzw. dahinter stehen, wie das im folgenden Beispiel gezeigt wird:

```
Pos++;
```

Diese Anweisung fügt dem gegenwärtigen Wert `Pos`, eins hinzu.

Um Werte zu vergleichen, benutzen Sie den logischen Operator.

```
if (Pos == 150) {
    anotherFunction();
}
```

Das führt uns direkt zum nächsten Punkt: `if-else`-Anweisungen. Mit `if` leiten Sie eine `if-else`-Anweisung ein. Verwenden Sie diese Anweisung, um einen Wert mit einem anderen zu vergleichen (oder um eine ganze Gruppe von Vergleichen durchzuführen). Über das Ergebnis dieses Vergleichs können Sie dann zwischen verschiedenen Alternativen entscheiden. Der vorherige Code verglich die Variable `Pos` mit 150. Wenn `Pos` gleich 150 ist, wird der Code innerhalb der Klammern ausgeführt. Sollte `Pos` ungleich

150 sein, fährt das Script fort als wäre nichts geschehen. Wenn Sie nun auf alle Ergebnisse reagieren wollen, verwenden Sie mehrere `if`-Anweisungen oder Sie binden die `else`-Anweisung ein.

```
if ((Pos >= 150) && (Pos <=200)) {
    functionOne();
}
else {
    functionTwo();
}
```

Die Schleife untersucht die Variable `Pos` auf zwei Werte: 150 und 200. Wenn `Pos` größer oder gleich 150 und kleiner oder gleich 200 ist (also wenn `Pos` zwischen 150 und 200 liegt), dann wird der Code im Rumpf der `if`-Anweisung ausgeführt. In diesem Fall ist das der Aufruf der Funktion `functionOne`. Sollte der Vergleich `false` (falsch) als Ergebnis liefern, wird der `else`-Block ausgeführt und `function2` aufgerufen.



if-else-Anweisungen bieten zwei oder mehrere Möglichkeiten, den der Pfad der Ausführung eines Scripts nehmen kann. Bedingte Anweisungen wie zum Beispiel (`x>y`) werden benutzt, um Werte und Variablen zu testen und den gewünschten Pfad festzulegen.

Wenn Sie Berechnungen und Vergleiche durchführen, werden Wertgruppen oder Variablen in Klammern gesetzt. Dadurch wird die Reihenfolge, in der sie interpretiert werden, festgelegt. Die folgende Code-Zeile legt eine Anweisung fest, in der `Pos >= 150` und `Pos <=200` jeweils zuerst interpretiert werden und dann das jeweilige Ergebnis vom logischen AND-Operator verglichen wird.

```
((Pos >= 150) && (Pos <= 200))
```

`((156 * 4 / 24) + 99) * 24` wird mit Hilfe der Klammern interpretiert. In diesem Fall wird 156 mit 4 multipliziert und das Ergebnis durch 24 geteilt. Diesem Ergebnis wiederum wird 99 hinzugefügt und schließlich mit 24 multipliziert. Das endgültige Ergebnis ist 3000 (rein zufällig eine schöne runde Zahl).



Wenn Sie eine Vielzahl von Werten oder Variablen in Klammern gesetzt haben, müssen Sie sich von innen nach außen durcharbeiten. Sie sollten ebenfalls sicherstellen, dass Sie die gleiche Anzahl schließender Klammern wie öffnender Klammern haben.

Verwenden Sie für diese Aufgabe Klammern, wenn Sie eine Funktion erstellen oder sie innerhalb eines Scripts aufrufen. Die Klammern umfassen Parameter oder Argumente, die von der Funktion zur Erfüllung der Aufgaben benutzt werden. Folgender Code illustriert eine simple Funktion, die für die Ausgabe einer Meldung geschaffen wurde, und die Werte der `user`-Variablen aufzeigt (in diesem Fall `Laura`).

Um die Funktion aufzurufen, müssen Sie, wie folgt, nur den Funktionsnamen mit dem Parameter `user` innerhalb der Klammern eingeben:

```
var user = "Laura";  
function alertUser(user) {  
    alert(user);  
}
```

Viele Funktionen haben keine Parameter, müssen aber alle Klammern haben. Eine einfache INIT-Funktion, wie im folgenden Beispiel aufgelistet, hat keine Parameter, wenn sie wie folgt aufgerufen wird: (*18)

Das führt mich gleich zum nächsten Punkt: Funktionen.

Normalerweise führen Scripts die Anweisungen von oben nach unten und in der Reihenfolge aus, in der sie eingegeben werden. Funktionen aber unterbrechen diese Reihenfolge, denn sie sind nichts anderes als Codeblöcke, die nur ausgeführt werden, wenn Sie von einem anderen Teil des Scripts aufgerufen werden. Die `init`-Funktion befindet sich getrennt vom übrigen Script und wird nur ausgeführt, wenn man sie aufruft.

Deshalb sollten Sie beim Schreiben von Scripts Funktionen verwenden, die einer kontrollierten Anordnung von Anweisungen und einmaligen Ereignissen entsprechen.



Funktionen sind Codeblöcke, die getrennt vom übrigen Fluss der Ausführung des Scripts sind. Funktionen werden nur ausgeführt, wenn man sie von anderen Teilen des Scripts oder des HTML-Dokuments aus aufruft.

Mit dem Schlüsselwort `function` leiten Sie die Definition einer Funktion ein. Dahinter folgt, durch ein Leerzeichen getrennt, ein frei wählbarer Funktionsname. Unmittelbar hinter dem Funktionsnamen folgt eine öffnende Klammer. Wenn die Funktion keine Parameter enthalten soll, notieren Sie dahinter sofort wieder eine schließende Klammer. Der gesamte Rumpf der Funktion wird in geschweifte Klammern `{ }` eingeschlossen. Diese Klammern dürfen niemals fehlen! Bedingte Anweisungen, wie zum Beispiel `if`, und Schleifen innerhalb der Funktionen können wie folgt mit in den Code einbezogen werden:

```
if ((Pos >= 150) && (Pos <=200)) {  
    functionOne();  
}  
else {  
    functionTwo();  
}  
}
```

Überprüfen Sie, dass die Anzahl der öffnenden und schließenden Klammern übereinstimmen. In dem vorherigen Listing hat die `determinePosition` – Funktion oben und unten Klammern. Innerhalb der Funktion hat die `if-else`-Anweisung ebenfalls jeweils Klammern. Wenn Sie diese öffnenden und schließenden Klammern nicht eingeben, wird eine Fehlfunktion bei der Ausführung des Scripts angezeigt.

So weit so gut. Sie haben nun einen Einblick in die Basis-Syntax von JavaScript und in einige Details dieser Sprache gewonnen. Allerdings haben Sie, außer einige Werte zu addieren, nicht viel gelernt. Das fehlende Glied, und jetzt aufgepasst, ist nämlich das Objekt. Objekte sind eine Kombination von Eigenschaften und Methoden. Stellen Sie sich Objekte als einen Container vor, der sowohl Informationen, als auch die Mittel zu deren Manipulation beinhaltet.



Objekte sind eine Kombination von Eigenschaften und Methoden und beinhalten sowohl Informationen, als auch Möglichkeiten zur Manipulation der Information darin.

Eigenschaften beschreiben das Objekt und Methoden liefern, unter Zuhilfenahme des Scripts, die Mittel zu deren Manipulation. Nehmen wir an, das Objekt ist ein Auto. Die Eigenschaften dieses Autos sind Farbe, Baujahr und Wert. Sie haben eine Methode mit der Sie jährlich den Wert des Autos, auf jährlicher Wertminderung basierend, ausrechnen. Um diese Methode des Objekts aufzurufen, geben Sie zuerst den Objektnamen, dann einen Punkt und anschließend den Namen der Methode ein. Eine Eigenschaft des Objekts anzusprechen, funktioniert auf dieselbe Weise: Geben Sie den Namen des Objekts ein, gefolgt von einem Punkt und anschließend den Namen der Eigenschaft. Mit einer solchen Eigenschaft können Sie wie mit einer Variablen arbeiten. Sie können den Wert auslesen, ihr einen Wert zuweisen bzw. den Wert darin verändern. In diesem Beispiel hat das Objekt Auto die Farbe blau.

```
Car.color = "Blue";
```

Die Methode wird aufgerufen, wenn Sie nach dem Objekt und nach der Eigenschaft, die Methode, wie im folgenden Beispiel, eingeben.

```
Car.value.calculate();
```

Im Web stehen die meisten der Objekte und deren Eigenschaften und Methoden in Bezug zum Web-Browser und HTML-Dokument. Das wichtigste Objekt ist hier das `document`-Objekt. Dieses Objekt enthält Informationen über das im Web-Browser geladene HTML-Dokument, verfügt über Methoden, die den Anwender-Ereignissen entsprechen und kann viele andere Aufgaben verrichten.

Und genau hier wird es ein bisschen schwieriger. Jeder Web-Browser hat unterschiedliche Objekte definiert, die verschiedene Eigenschaften und Methoden haben können.

Hinzu kommt, dass jeder Browser über eine Reihe von Event-Handler verfügt, die verwendet werden, um auf Anwenderereignisse zu reagieren (wie etwa Anklicken der Maus oder Drücken einer Taste).



Im Netscape Navigator stehen nicht alle Event-Handler für sämtliche Objekte zur Verfügung. Deshalb sollten Sie sich in der Netscape JavaScript-Dokumentation darüber informieren, welche Ereignisse von welchen Objekten verwendet werden können.

Objekte und Ereignisse stellen eine gewisse Problematik dar, deshalb werde ich Sie später in diesem Kapitel noch eingehender behandeln. Zuerst jedoch sollten Sie mehr über das Integrieren Ihrer Scripts in HTML-Dokumente erfahren.

Scripts mit HTML integrieren

JavaScript ist natürlich nicht gleich HTML. Beides sind vollkommen unterschiedliche Sprachen und führen grundverschiedene Funktionen aus. Sie können nicht einfach so Script -Anweisungen in ein HTML-Dokument »werfen« und dann hoffen, dass alles gut geht und der Browser schon wissen wird, was er zu tun hat. Das Erste, was Sie tun sollten ist, eine Standard-Scriptsprache im Dokumentkopf zu bestimmen und dabei Meta-Angaben zu verwenden.

Beispiel:

```
<meta http-equiv="Content-Script-Type" content="text/javascript" />
```

Meta-Angaben sind für die Ausführung dieser Funktion optimal. Mit dieser Anweisung wird sichergestellt, dass der Web-Browser weiß, dass JavaScript als Scriptsprache benutzt wird. Sämtliche Scripts werden als solche interpretiert, es sei denn Sie machen andere Angaben. Diese Angaben sind wirklich wichtig, denn für den Web-Browser sollten keine Zweifel bestehen, welche Art von Script angewandt wird. Doch nun weiter im Einbinden der Scripts.

JavaScript Scripts können auf dreierlei Art und Weise in ein HTML-Dokument eingebunden werden:

- ▶ Einbinden von Inline-Scripts als Ereignisbehandlung
- ▶ Eingeben von Scripts im Dokumentkopf oder -rumpf
- ▶ Link auf externe Script-Dateien

Eine Ereignisbehandlung (Event-Handler) wird aufgerufen, wenn ein Ereignis auf der Webseite auftritt (Mausklick, Tastendruck). Der Rumpf des HTML-Dokuments hat einen Event-Handler namens `onload`, der aufgerufen wird, sobald das Dokument im Web-Browser geladen ist. Diese Event-Handler stellen Attribute der HTML-Elemente

dar. Der Wert eines solchen Attributs kann eine Scriptanweisung oder eine ganze Reihe von Scriptanweisungen sein. Folgendes Beispiel weist das `onload`-Event dem `<body>`-Element zu und führt eine einzige Anweisung aus, die eine Meldung mit dem eingebundenen Text erstellen soll.

```
<body onload="alert('The page has loaded.')">
```

Hier ist allerdings zu beachten, dass sich in Inline-Scriptanweisungen nur einfache Anführungszeichen befinden dürfen. Mit dem ersten doppelten Anführungszeichen wird der Attributwert eingeleitet. Beim nächsten Anführungszeichen würde der Wert aber beendet werden. Falls Sie den Code falsch eingegeben haben, erscheint am Scriptende eine Meldung (und das wollen wir doch vermeiden, oder?)

```
<body onload="alert("The page has loaded.")">
```



Sie dürfen sich nicht von den offiziellen HTML-Angaben in Bezug auf Ereignisse in die Irre leiten lassen. Obwohl das W3C eine Vielzahl von Ereignissen auflistet, die auf bestimmte Elemente angewendet werden können, liegt es letztendlich am Web-Browser das jeweilige Ereignis zu implementieren. Im Gegensatz zum Internet Explorer, werden im Netscape Navigator nicht so viele Ereignisse unterstützt, wie das beim Internet Explorer der Fall ist. In Bezug hierauf ist der Navigator schwieriger zu programmieren.

Das Problem hierbei ist jedoch, dass die Praxis derzeit noch stark von der Theorie abweicht – zumindest bei Netscape. Der MS Internet Explorer 4.x dagegen interpretiert Event-Handler weitgehend so, wie vom W3-Konsortium vorgesehen.

Wenn Sie auf ein Ereignis reagieren, können Sie ebenfalls vielfältige Scriptanweisungen anwenden, die in ein HTML-Dokument integriert sind (inline). Hierzu folgendes Beispiel:

```
<body onload="var message = 'The page has loaded'; alert(message)">
```

Dieses kleine Script enthält zwei Anweisungen, die auf das `onload`-Ereignis (beim Laden einer Datei) reagieren. Die erste Anweisung deklariert die Variable und weist ihr ein String zu, das zweite eine Meldung mit einer Nachricht.

Die zweite Methode ein Script zu integrieren ist, dieses in den Kopf oder Rumpf des HTML-Dokuments einzugeben. Hierfür muss ein `<script>`-Element verwendet und die Art des Scripts mit den `type`- oder `language`-Attributen im einleitenden Script-Tag festgelegt werden. Beide Attribute haben die gleiche Funktion und teilen dem Browser mit, welche Scriptsprache Sie benutzen. Das `type`-Attribut entspricht hierbei zwar dem offiziellen Standard, dennoch wird das `language`-Attribut öfters benutzt. Man kann aber auch beide Attribute anwenden:

```
<script type="text/javascript" language="javascript">
```

Geben Sie nach dem einleitenden Befehl Ihre erstellten Scriptanweisungen ein. Beenden Sie das Scriptelement anschließend mit einem abschließendem Tag.

```
<head>
<title>Sample Script</head>
  <meta http-equiv="Content-Script-Type" content="text/javascript" />
  <script type="text/javascript" language="javascript">
    var message="Hello World";
    alert(message);
  </script>
</head>
```

Das eben aufgeführte Beispiel wird problemlos von Script-kompatiblen Browsern unterstützt, dennoch ist es üblich die Scriptanweisungen in einen HTML-Kommentar einzuschließen.

```
out the script statements with HTML comment tags, as follows:
<script type="text/javascript" language="javascript">
<!-- Hide JavaScript from older browsers
  var message="Hello World";
  alert(message);
// stop hiding the script -->
</script>
```

Achten Sie auf die letzte Zeile des Scripts. Ich habe einen JavaScript-Kommentar verwendet, um sicherzugehen, dass der JavaScript-Interpreter nicht versucht den schließenden HTML-Kommentar als Script-Code zu interpretieren. Hätte ich das Script lediglich mit dem HTML-Kommentar abgeschlossen, würde der Browser versuchen diesen als JavaScript zu interpretieren.

```
stop hiding -->
```

Eine weitere Technik bei der Herstellung der Kompatibilität zu möglichst vielen Browser-Versionen ist die Verwendung des `<noscript>`-Elements. Mit diesem Element wird es Ihnen ermöglicht, alternative Inhalte für Browser, die keine Scriptsprachen unterstützen, anzubieten. Beginnen Sie mit dem einleitenden `<noscript>`-Befehl; geben Sie anschließend den alternativen Inhalt ein. Schließen Sie das `</noscript>`-Element mit dem abschließendem Tag.

```
<script type="text/javascript" language="javascript">
<!-- Hide JavaScript from older browsers
  var message="Hello World";
  alert(message);
// stop hiding the script -->
</script>
<noscript>
  <p>You have a browser that is not compatible with scripting.</p>
</noscript>
```

Wie schon erwähnt, können Sie Scripts in den Kopf oder den Rumpf eines Dokuments eingeben. Die Reihenfolge in der die Scripts erscheinen, ist gleichzeitig die Reihenfolge in der sie ausgeführt werden. Ein Script, das sich im Kopf eines Dokuments befindet, wird deswegen vor einem Script, das sich im Rumpf eines Dokuments findet, ausgeführt. Verwenden Sie nun Funktionen, die den Ablauf Ihres Scripts kontrollieren, können Sie diese Regel übergehen. Die Interpretation Ihres Scripts hängt also nur von Ihrer Vorgehensweise ab.

Wenn Sie das `src`-Attribut innerhalb des Scriptelements verwenden, können Sie auf ein oder mehrere externe Scripts verweisen (die externen Scriptdateien werden in der Regel mit der Endung `.js` abgespeichert). Folgendes Beispiel zeigt Ihnen wie man auf zwei externe Scripts verweist.

```
<script type="text/javascript" language="javascript" src="detect.js"></script>
<script type="text/javascript" language="javascript" src="animate.js"></script>
```

Indem Sie mit Links auf externe Scripts verweisen, können Sie häufig verwendete Scriptroutinen, die Sie bei vielen Webseiten benutzen, einfacher verwalten und wiederverwenden.

Sie wissen jetzt also, wie man Scripts in Webseiten integriert. Sie müssen nur noch wissen, dass die aktuelle JavaScript-Version 1.3 ist, die 1.4-Version aber schon bald erscheinen wird. Tabelle 15.2 listet die JavaScript Versionen auf, die vom Netscape Navigator und MS Internet Explorer unterstützt werden.

JavaScript	Navigator	Internet Explorer
1.0	2.0x	3.0x
1.1	3.0x	4.0x
1.2	4.0-4.05	4.0x
1.3	4.06-4.61	5.0x
1.4	TBD	TBD

Tabelle 15.2: Vom Netscape Navigator und MS Internet Explorer unterstützte JavaScript-Versionen



Technisch gesehen, interpretiert MS Internet Explorer JavaScript nicht. Microsoft verwendet eine Sprache, die auf JavaScript basiert und Jscript genannt wird. Lassen Sie sich davon nicht verwirren. Schreiben Sie weiterhin Ihre Scripts und beziehen Sie sich dabei auf die JavaScript-Dokumentation.

Anwendung des Document Object Models (DOM)

Ich habe ja schon erwähnt, dass der Microsoft Internet Explorer und der Netscape Navigator jeweils verschiedene Document Object Models haben. Jedes Objekt einer Webseite (und vieler Elemente des Web-Browsers) haben verschiedene Eigenschaften (zu deren Beschreibung), Methoden (zur Manipulation der Eigenschaften) und Event-handler (um auf Anwenderereignisse zu reagieren). Als Nächstes werden wir noch einmal kurz jedes Browsermodell durchgehen und dabei versuchen, einige Gemeinsamkeiten festzustellen.



Die Fähigkeiten eines jeden DOM durchzugehen, würde den Rahmen dieses Buches bei weitem überschreiten. Im Web gibt es viele Beispiele für den Einsatz von DHTML und DOM. Nehmen Sie sich etwas Zeit und gehen Sie den Code dieser Beispiele durch – hiervon können Sie unheimlich viel lernen.

Das Navigator DOM

Das Navigator DOM entspricht dem der offiziellen JavaScript-Dokumentationen. Wenn man Browser-übergreifendes DHTML erstellen will, kann dies aber durchaus verwirrend sein. Wenn der Internet Explorer JavaScript interpretiert, geschieht das nämlich in einem anderen DOM-Kontext: dem eigenen. Mit anderen Worten: Geben Sie acht, wenn Sie offizielle JavaScript-Dokumentationen lesen. Die darin enthaltenen Informationen lassen sich nur im Netscape Navigator anwenden.

Außer einigen vordefinierten JavaScript-Objekten, wie zum Beispiel `screen`, entspricht die Struktur der Navigator DOMs, der eines Baumes, wobei `window` das an der Spitze stehende Objekt ist. Das `window`-Objekt repräsentiert das Web-Browser-Fenster oder einen Frame innerhalb des Browsers. Die Eigenschaften und Ereignisse des `window`-Objekts betreffen also den Web-Browser.

Dem `window`-Objekt sind andere Objekte untergeordnet, wovon sich einige wiederum in mehrere Objekte verzweigen. Das Objekt, das uns in der heutigen Lektion am meisten interessieren wird, ist das direkte Subobjekt von `window`, das `document`-Objekt. Das `document`-Objekt wird mit dem HTML `<body>`-Tag im Navigator erstellt. Sie müssen sich also nicht selbst darum kümmern.

Unterhalb des `document`-Objekts kann sich eine Fülle von verschiedenen Objekten befinden, jedes mit eigenen Eigenschaften und Methoden. Die meisten der Objekte, die Sie verwenden werden, sind HTML-Elemente, wie etwa Bilder, Style-Sheet-Objekte, Formulare und deren Felder.

Sie werden sich immer wieder auf das `document`-Objekt und seine »Abkömmlinge« beziehen müssen, deshalb ist es wichtig zu wissen, wie man die Referenz zu ihnen erstel-

len muss. Wenn Sie ein Objekt in JavaScript aufrufen, verwenden Sie Punktnotationen, um auf jedes Objekt in der DOM Hierarchie zu verweisen, und dies so lange, bis Sie zu Ihrem Ziel gelangen. Folgendes Beispiel zeigt wie die Referenz zum `myimage`-Bild aussieht:

```
document.myimage
```

Um auf die Eigenschaften des Bildes zu verweisen, hängen Sie einfach das Ende dieser Anweisungen wie folgt an:

```
document.myimage.src
```

Führen Sie doch einmal folgenden einfachen Versuch aus. Erstellen Sie eine Webseite, die ein Bild hat. (Dabei können Sie denselben Namen wie ich nehmen. Das bleibt Ihnen überlassen. Versichern Sie sich nur, dass Sie im `src`-Attribut ein Bild einsetzen, das sich auch tatsächlich auf Ihrer Platte befindet.) Das Bild ist im Rumpf Ihres HTML-Dokuments lokalisiert. Gleich unterhalb des Bildes befindet sich ein Script, das eine Meldung ausgibt, welche die Breite und die Höhe des Bildes, wie sie im `width` bzw `height`-Attribut festgelegt sind, ausgibt. In diesem Beispiel ist es also wichtig, dass Sie das Script nach dem Bild platzieren, da das Bild erst geladen werden muss, um das Objekt erstellen zu können.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">
<html>
<head>
<title>Netscape Navigator DOM Test</title>
<meta http-equiv="Content-Script-Type" content="text/javascript" />
</head>
<body>



<script language="javascript" type="text/javascript">
<!-- Hide JavaScript
alert(document.barnimage.width);
// end hide JavaScript -->
</script>

</body>
</html>
```


Bild 15.1 zeigt die Meldung, die vom Netscape Navigator generiert wurde. Das Bild erschien, bevor die Meldung angezeigt wurde.



Abbildung 15.1:
Zugriff auf die Eigen-
schaften eines Bildes
mittels JavaScript
und Navigator-DOMs

Zugriff auf andere Eigenschaften des Bildes zu haben, ist genauso leicht, wie die `width`-Eigenschaft in eine andere Eigenschaft abzuändern (zum Beispiel `src`).

Um die Methode eines Objekts zu verwenden, müssen Sie nur die Eigenschaft durch die Methode ersetzen. Sämtliche Werte, die von der Methode verlangt werden, sind dabei inbegriffen. Folgender Code verwendet die `write`-Methode des `document`-Objekts, um in das Browser-Fenster zu schreiben:

```
document.write("Hello world!")
```

Einige Eigenschaften, wie etwa Bildhöhe, werden nur gelesen.

Das bedeutet, dass der Wert der Eigenschaft nur angezeigt wird, jedoch nicht verändert werden kann. Andere Eigenschaften wiederum, insbesondere CSS-Eigenschaften (Abkömmlinge des Objekts selbst) können modifiziert werden. Für die Modifizierung einer Objekt-Eigenschaft geben Sie wie gewohnt das Objekt und den Eigenschaftsnamen ein. Fügen Sie ein Gleichheitszeichen (=) und den neuen Wert ein. Folgende Anweisung verändert die Position des Bildobjekts `barnimage` um 150 Pixel vom oberen Teil des Fensters oder vorherigen Elements, abhängig davon, ob das Bild absolut oder relativ positioniert war.

```
document.barnimage.top = "150"
```

Zu guter Letzt noch eine Information über das Navigator-DOM. Die Objekte »unterhalb« eines Objekts werden in einem Array abgelegt (Reservierter Speicherbereich, der auf eine besondere Art angesprochen wird). Beispiel: Alle Bild-Objekte einer Webseite werden in der `document.images`-Eigenschaft gespeichert (`document.images` ist das Array). Das erste Bild wird an der Position 0 gespeichert. Jedes weitere Bild wird nacheinander durchnummeriert und abgespeichert. Um auf das erste Bild im Array zuzugreifen zu können, müssen Sie wie folgt vorgehen:

```
document.images[0].width
```

Wenn Sie diese Anweisung im vorhergehenden Bildbeispiel (Meldung) einsetzen, würde der gleiche Wert erhalten werden: 150

Das Internet Explorer DOM

Das DOM des Internet Explorers (in den Versionen 4 und 5) ist dem Navigator-DOM ähnlich: Jeder Browser erstellt Objekte mit Eigenschaften, Methoden und Ereignissen. Der Unterschied liegt in der Art und Weise wie sie organisiert sind und wie Sie in Ihren Scripts die Referenz zu diesen herstellen.

Der Internet Explorer verfügt über ein `document`-Objekt, das dem des Navigators sehr ähnlich ist. Unterhalb des `document`-Objekts ist jedoch alles in einer `all`-Sammlung enthalten, das heißt, jedes HTML-Element der Webseite ist dort enthalten. Wenn Sie auf ein Bild im Internet Explorer Bezug nehmen müssen, würde die Anweisung wie folgt aussehen:

```
document.all.myimage
```

Um Zugriff auf eine Eigenschaft der Methode zu haben, hängen Sie diese, wie im Navigator auch, an das Ende der Anweisung.

```
document.all.myimage.src
```

Style-Objekte werden im Internet Explorer anders gehandhabt. Alles was die Styles betrifft, wird mit der Angabe `.style` nach dem Objekt-Namen versehen. Nach `.style` müssen Sie noch die Style-Eigenschaft eingeben.

```
document.all.barnimage.style.top
```

Die Erläuterungen der DOMs im Internet Explorer enden hier mit demselben Bildbeispiel wie dem, das für den Netscape Navigator verwendet wurde.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">
<html>
<head>
<title>Internet Explorer DOM Test</title>
<meta http-equiv="Content-Script-Type" content="text/javascript" />
</head>
<body>



<script language="javascript" type="text/javascript">
<!-- Hide JavaScript
alert(document.all.barnimage.width);
// end hide JavaScript -->
</script>

</body>
</html>
```

Bild 15.2 zeigt die Meldung. Im Internet Explorer wird das Bild aufgezeigt, bevor die Meldung generiert wird.



Abbildung 15.2:
Zugriff auf die Bildeigenschaften
mittels JavaScript und
DOM des Internet Explorers.

Mit Ereignissen arbeiten

Bei Anwendererscheinungen spricht man von Ereignissen. Die Bedeutung von Ereignissen ist, dass man Scripts erstellen kann, die auf diese ansprechen. Leider gibt es große Unterschiede in der Art und Weise wie der Navigator und der Internet Explorer die Ereignisse handhaben. Allgemein gesehen, tendiert der Navigator dazu, strenger in Bezug auf HTML-Elemente, die automatisch Ereignisse auslösen, zu sein. Eine Vielzahl von Ereignissen können fast allen Elementen im Internet Explorer zugewiesen werden.



Wegen der großen Anzahl an Elementen und Unterschieden in der Implementierung, ist es schier unmöglich sämtliche Eventualitäten in diesem Buch abzudecken. Verwenden Sie deshalb für beide Browser die spezifischen Dokumentationen für ausführlichere Details.

Ein weiterer Unterschied im Arbeiten mit Ereignissen im Navigator und Internet Explorer besteht darin, an welcher Stelle das Ereignis zuerst aufschlägt und in welcher Richtung es in der Dokument-Objekt Hierarchie folgt (nach oben oder unten). Der Navigator benutzt ein absteigendes Ereignismodell, wobei Ereignisse im höchsten Level des DOM (dem `window`-Objekt) aufschlagen und den Objekt-Baum so lange hinuntersteigen, bis ein Event-Handler auf das Ereignis reagiert. Es ist wohl keine sehr große Überraschung, wenn ich Ihnen jetzt sage, dass der Internet Explorer eine ganz andere Vorgehensweise hat. Anstatt ein absteigendes Ereignismodell zu verwenden, verwendet der Internet Explorer ein aufsteigendes Ereignismodell. Das Ereignis wird zuerst von dem Element, das das Ereignis hervorruft (zum Beispiel ein `<img>`- oder `<div>`-Element) aufgespürt. Das Ereignis steigt dann die Ereigniskette so lange hoch, bis eine Reaktion erfolgt oder es verworfen wird.

Das Praktische hierbei ist, dass das Behandeln von Ereignissen im Internet Explorer ziemlich einfach ist.

Willkommen in der Wirklichkeit: Browser-DHTML-Techniken im tatsächlichen Gebrauch

Der größte Teil des Kapitels wurde dem Verstehen der Technologien hinter DHTML und wie man sie einzeln anwendet gewidmet. Nun ist es an der Zeit DHTML Webseiten zu erstellen. Die wahre Herausforderung besteht nämlich im Erstellen von DHTML, um innerhalb praxisorientierter Grenzen das größtmögliche Publikum ansprechen zu können. »Praxisorientiert« deswegen, weil DHTML nicht in Web-Browsern möglich ist, die keine Scriptsprachen oder Style-Sheets implementieren, und in solchen Web-Browsern unbrauchbar sind, die nur teilweise oder schlecht implementie-

ren. Mit anderen Worten, mit DHTML werden Sie nie alle von Ihnen anvisierten Zielgruppen erreichen können.

Die gute Nachricht ist, dass Sie mit guten Kenntnissen der herrschenden Unterschiede bei den führenden Web-Browsern und mit kreativer Programmierung Browser-übergreifende DHTML-Webseiten erstellen können, die einige oder die meisten der Dinge, die Sie zu tun gedenken, auf einer Webseite vollziehen.

Eine Alternative wäre, mehrere Versionen Ihrer Webseite für die verschiedenen Web-Browser zu erstellen. Meiner Meinung nach ist das keine sehr praktische Lösung, denn Ihre Arbeitslast würden um Faktor zwei, drei, vier oder x größer sein. Deswegen werde ich mich in der folgenden Lektion auf Browser-übergreifendes DHTML konzentrieren.

Browser erkennen

Die schwierigste Aufgabe beim Erstellen von Browser-übergreifendem DHTML ist, den Web-Browser ausfindig zu machen, der Ihre Seite lädt (den Web-Browser erkennen). Haben Sie diese Aufgabe erst einmal ausgeführt, können Sie Ihre Scripts festlegen, deren Funktionen oder Anweisungen dann auf einen bestimmten Browser zugeschnitten werden.

Aber keine Sorge, Sie müssen dies nicht alles selbst erstellen. Netscape hat diese Aufgabe schon übernommen und einen leicht verständlichen »Schnüffler« (Sniffer) auf seiner DevEdge-Web-Site veröffentlicht: <http://devedge.netscape.com/tech/dynhtml/index.html>. Das gesamte Listing können Sie herunterladen und kostenlos für Ihre Webseiten benutzen.

Ich werde in diesem Kapitel nur eine verkürzte Version des Sniffer anwenden, der nur den Netscape Navigator 4 und den MS Internet Explorer 4 ausfindig macht. Im folgenden Listing werden Variablen erzeugt. Diesen werden bestimmte Informationen der Browser-Objekte zugewiesen. Durch Verknüpfung der Werte wird den endgültigen Variablen `is_nav4up` und `is_ie4up` der Wert `true` bzw. `false` zugewiesen. Je nachdem, um welchen Browser es sich handelt.

```
// Trimmed down browser sniffer that
// detects Navigator 4+ and IE 4+.
// Reference is_nav4up and is_ie4up in other
// portions of the script
var agt = navigator.userAgent.toLowerCase();
var is_major = parseInt(navigator.appVersion);
var is_minor = parseFloat(navigator.appVersion);
var is_nav = ((agt.indexOf('mozilla')!=-1) && (agt.indexOf('es poofers')==-1)
    && (agt.indexOf('compatible') == -1) && (agt.indexOf('opera')==-1)
    && (agt.indexOf('webtv')==-1));
```

```
var is_nav4up = (is_nav && (is_major >= 4));
var is_ie = (agt.indexOf("msie") != -1);
var is_ie4up = (is_ie && (is_major >= 4));
```

Verwenden Sie das folgende Script, um zu überprüfen, ob es vom Netscape Navigator und MS Internet Explorer in den Versionen 4 oder höher ausgeführt wird:

```
if (is_nav4up) {
    do something }
if (is_ie4up) {
    do something else }
```

Die Funktion des Sniffer existiert einzig und allein aus dem Grunde, damit Ihre eigenen Variablen zu Ihnen zurückgesendet werden können. Somit können Sie Teile Ihres Scripts noch einmal überprüfen. Alleinstehend würde diese Funktion überhaupt nichts bewirken.

Folgendes Beispiel zeigt den Browser-Sniffer in Aktion. Das <body>-Element der Webseite ruft die doSniff-Funktion auf. Diese wiederum kreiert eine Variable namens sniff und weist sie einem string-Wert zu, der von der aufgefundenen Browser-Version abhängig ist.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">

<html>
<head>
<title>Browser Sniffing</title>
<meta http-equiv="Content-Script-Type" content="text/javascript" />

<script language="javascript" type="text/javascript">
<!-- Hide JavaScript

// Trimmed down browser sniffer that
// detects Navigator 4+ and IE 4+.
// Reference is_nav4up and is_ie4up in other
// portions of the script
var agt = navigator.userAgent.toLowerCase();
var is_major = parseInt(navigator.appVersion);
var is_minor = parseFloat(navigator.appVersion);
var is_nav = ((agt.indexOf('mozilla')!=-1) && (agt.indexOf('es poofers')==-1)
    && (agt.indexOf('compatible') == -1) && (agt.indexOf('opera')==-1)
    && (agt.indexOf('webtv')==-1));
```

```
var is_nav4up = (is_nav && (is_major >= 4));
var is_ie = (agt.indexOf("msie") != -1);
var is_ie4up = (is_ie && (is_major >= 4));

// Function to display the browser version
function doSniff() {
    var sniff
    if (is_nav4up == true) {
        sniff = "Netscape Navigator 4+" }
    if (is_ie4up == true) {
        sniff = "Microsoft Internet Explorer 4+" }
    alert(sniff);
}
// end hide -->
</script>
</head>
<body onload="doSniff()">
</body>
</html>
```

Bild 15.3 zeigt das Ergebnis im Internet Explorer 5.



Abbildung 15.3:
Der Sniffer für Web-
Browser hat die
Browser-Version aus-
findig gemacht.

Entwickeln einer Browser-übergreifenden DOM-Referenz

Weitere nützliche Werkzeuge sind die Mittel, die sich automatisch, je nach Browser, auf das korrekte DOM beziehen. Anstatt sich immer auf jedes Document Object Model einzeln beziehen zu müssen, erstellen Sie doch eine Browser-übergreifende Terminologie. Folgend aufgeführter Code wird Ihnen dies ermöglichen. Dabei werden die `objref`- und `styleref`-Variablen verwendet, um entweder den NULL-Wert (für Navigator) oder jeweils `.all` und `.style` (für Internet Explorer) festzulegen. Nachdem der Browser erkannt wurde, werden die Variablen dementsprechend eingegeben.

```
// Object and Style Reference conventions
// for Navigator and IE. Use objRef and
// when referring to objects and both when
// accessing style properties.
var objRef, styleRef;
if (is_nav4up == true) {
    objRef = ""
    styleRef = ""
}
if (is_ie4up == true) {
    objRef = ".all"
    styleRef = ".style"
}
```

Folgendes Beispiel erstellt eine Webseite unter Anwendung dieser Logik. Nachdem der Browser-Sniffer eingesetzt wurde, habe ich den DOM-Referenz- Code eingegeben. Zwei Funktionen für die Erstellung von Meldungen folgen. Hiermit werden die adäquaten DOM-Informationen für den Navigator oder den Internet Explorer aufgezeigt. Innerhalb des Rumpfs der Webseite befinden sich zwei `div`-Elemente zur Positionierung Ihrer Text-Links. Diese Links reagieren auf das `onmouseover`-Ereignis und lösen die `warn`-Funktion aus. Das `href`-Attribut der Links ist auf `javascript:void(0)` gesetzt. Das ist eine einfache Art und Weise, um Links als Auslöser für Funktionen zu verwenden, ohne dabei auf irgendetwas zu verweisen.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">

<html>
<head>
<title>Document Object Model References</title>
<meta http-equiv="Content-Script-Type" content="text/javascript" />
```



```
<script language="javascript" type="text/javascript">
<!-- Hide JavaScript

// Trimmed down browser sniffer that
// detects Navigator 4+ and IE 4+.
// Reference is_nav4up and is_ie4up in other
// portions of the script
var agt = navigator.userAgent.toLowerCase();
var is_major = parseInt(navigator.appVersion);
var is_minor = parseFloat(navigator.appVersion);
var is_nav = ((agt.indexOf('mozilla')!=-1) && (agt.indexOf('es poofers')!=-1)
    && (agt.indexOf('compatible') == -1) && (agt.indexOf('opera')!=-1)
    && (agt.indexOf('webtv')!=-1));
var is_nav4up = (is_nav && (is_major >= 4));
var is_ie = (agt.indexOf("msie") != -1);
var is_ie4up = (is_ie && (is_major >= 4));

// Object and Style Reference conventions
// for Navigator and IE. Use objRef and
// when referring to objects and both when
// accessing style properties.
var objRef, styleRef;
if (is_nav4up == true) {
    objRef = ""
    styleRef = ""
}
if (is_ie4up == true) {
    objRef = ".all"
    styleRef = ".style"
}

function alertstyle() {
    alert(eval("document" + objRef + ".div01" + styleRef + ".top"));
}

function alertobject() {
    alert(eval("document" + objRef + ".div02.id"));
}

// end hide JavaScript -->
</script>

<style type="text/css">
<!--
#div01 { position: absolute; }
```

```
#div02 { position: absolute; }
-->
</style>

</head>

<body>

<div id="div01" style="top: 50; left: 50;">
<a href="javascript:void(0)" onmouseover="alertstyle()">Style Alert</a>
</div>

<div id="div02" style="top: 100; left: 100;">
<a href="javascript:void(0)" onmouseover="alertobject()">Object Alert</a>
</div>

</body>
</html>
```

Bild 15.4 zeigt das Ergebnis Ihres Codes. Beachten Sie den Cursor über dem Element, das den Style anzeigt.

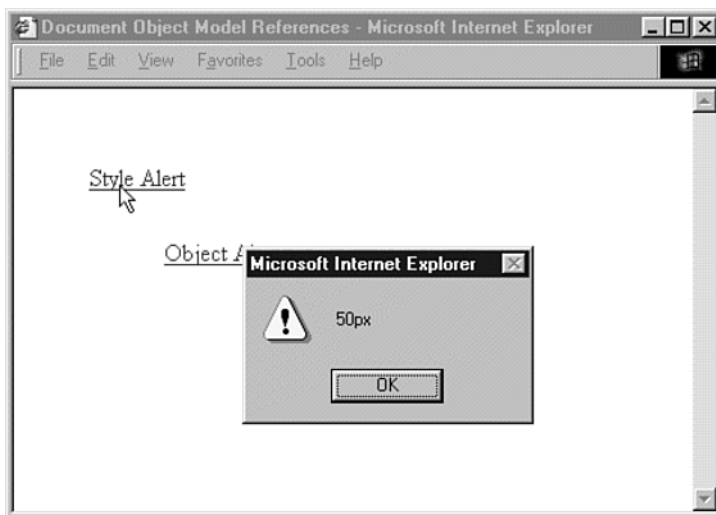


Abbildung 15.4:
Verwenden einer
Browser-übergreifen-
den Objekt-DOM-
Referenz-Routine.

Die heutigen zwei Übungen werden Ihnen ohne großes Trara zeigen, wie man Browser erkennt und diese Information in Ihrem Script verwendet, wie man Rollover-Effekte für Bilder erzeugt, das Verdecken und Anzeigen von Elementen steuert, Menüs erweitert, und noch einige andere Browser-übergreifende DHTML-Techniken.

Das hauptsächliche Ziel dieser Übungen ist es, Ihnen zu zeigen, wie man folgende Aufgaben erfüllt:

- ▶ Kombination von HTML, CSS und JavaScript, um nützliche DHTML-Effekte zu bekommen.
- ▶ Ihre Browser-Version zu überprüfen, um sicherzugehen, dass Ihre DHTML-Seiten Browser-übergreifend verwendet werden können. Die erhaltenen Informationen können Sie dann anschließend in Ihren Scripts verwenden.
- ▶ Ihnen ein Gefühl für eine Methode zu vermitteln, wie Sie an DHTML herangehen.
- ▶ Praxisnahe Beispiele von DHTML aufzuzeigen, damit Sie die Konzepte auch in Aktion sehen und Sie als Basis für zusätzlich zu Lernendes weiterverwenden können.

Übung 15.1: Erstellen aufklappbarer Menüs

In Ihrer ersten Übung werden Sie ein aufklappbares Menü für eine Webseite, das auf DHTML basiert, erstellen. Wenn ein Anwender ein Bild neben einem Menüeintrag anklickt, erweitert sich das Menü und zeigt Untermenüs mit dem jeweiligen Inhalt an. Wenn Anwender ein Menüeintrag anklicken, erscheint der Inhalt in einem `<div>`-Element im Hauptbereich des Browser-Fensters. Sie werden keine `<table>`- oder `<frame>`-Elemente für die Strukturierung verwenden. Anstelle dieser Elemente werden Sie absolut positionierte `<div>`-Elemente benutzen und ihre Sichtbarkeit modifizieren. Diese Übung enthält außerdem `<style>`-Deklarationen im Dokumentkopf.

Erstellen Sie zuerst eine »Schablone« der Webseite mit den wichtigsten Struktur-Elementen und Kommentaren und den Stellen, wo Sie Scripting, CSS und HTML-Elemente verwenden wollen.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">
```

```
<html>
```

```
<head>
<title>Ye old Brick House</title>
<meta http-equiv="Content-Script-Type" content="text/javascript" />

<!-- Most JavaScript will go here -->

<!-- CSS will go here -->

</head>
<body>

<!-- HTML will go here -->

</body>
</html>
```

Als Nächstes fügen Sie den HTML-Code hinzu, um das Inhaltsformular der Webseite zu strukturieren. Der Struktur zuliebe, habe ich alle TOP-LEVEL Menüs `toc100` und `toc200` genannt. Jedes Untermenü erhält einen dementsprechenden Namen (je nachdem unter welches Menü es fällt). Damit wird die Organisation erheblich erleichtert. Ich verwende ebenfalls Inline-Scripting, um herauszufinden, ob das Element gerade sichtbar ist, und tausche das Pfeilbild nach der Menüerweiterung um.

Schließlich habe ich noch den Inhalt einbezogen, der in jedem Menü erscheinen wird. Sie müssen sich nicht darum kümmern, dass alles sichtbar ist. Setzen sie die `visibility`-Eigenschaft in Ihrer Style-Sheet-Sektion auf `hidden` (verdeckt).



```
<div id="title">

</div>

<div id="toc0100">
<a href="javascript:void(0)" onclick="menuaction();
  if (show == 1) { document.menuimg.src = ea.src; return true }
  if (show == 0) { document.menuimg.src = ca.src; return true }">
</a>
History
</div>

<div id="toc0101">
```

```
<a href="javascript:void(0)" onclick="contentID='0101'; showcontent(0101)">The
Early Days</a>
</div>
```

```
<div id="toc0102">
<a href="javascript:void(0)" onclick="contentID='0102'; showcontent(0101)">The
Modern Era</a>
</div>
```

```
<div id="toc0200">
<a href="javascript:void(0)" onclick="contentID='0200';
showcontent(0101)">Pictures</a>
</div>
```

```
<div id="splash" align="center">

</div>
```

```
<div id="content0101">
<p>This brick house was built in the 1800s. Originally part of a large farm that
consisted of several hundred acres, the house and barn now sit on a 4 acre plot of
land.</p>
</div>
```

```
<div id="content0102">
<p>There have been several modifications to the house, such as a refurbished
kitchen, but the exterior is largely as it was in the 1800s. The barn, on the
other hand, has been left unchanged.</p>
</div>
```

```
<div id="content0200">



</div>
```

Als Nächstes geben Sie den CSS-Code im Dokumentkopf ein. Verwenden Sie absolute Positionierung, um sowohl jeden Menüeintrag auf Ihrer Seite zu platzieren, als auch die Titel-Grafik und den Splash Screen. Es muss etwas »experimentiert« werden, um herauszufinden, wo man alle Elemente platzieren soll, aber keine Sorge, das wird nicht allzuviel Zeit beanspruchen. Untermenüs und sämtliche Inhalte sind vorerst verdeckt. Scripting wird zur Menüerweiterung und zum Platzieren aller Elemente angewandt, um eine Überlappung zu vermeiden (siehe Bild 15.5).



```
<style type="text/css">
<!--
#title { position: absolute; top: 20; left: 50; }
#splash { position: absolute; top: 150; left: 200; visibility: visible; }
#toc0100 { position: absolute; top: 130; left: 20; visibility: visible; }
#toc0101 { position: absolute; top: 150; left: 50; visibility: hidden; }
#toc0102 { position: absolute; top: 170; left: 50; visibility: hidden; }
#toc0200 { position: absolute; top: 150; left: 20; visibility: visible; }
#content0101 { position: absolute; top: 150; left: 200; visibility: hidden; }
#content0102 { position: absolute; top: 150; left: 200; visibility: hidden; }
#content0200 { position: absolute; top: 150; left: 200; visibility: hidden; }
-->
</style>
```

Nun können wir mit dem Scripting beginnen. Ihre erste Aufgabe besteht darin, globale Variablen, die im Script verwendet werden, zu erstellen.

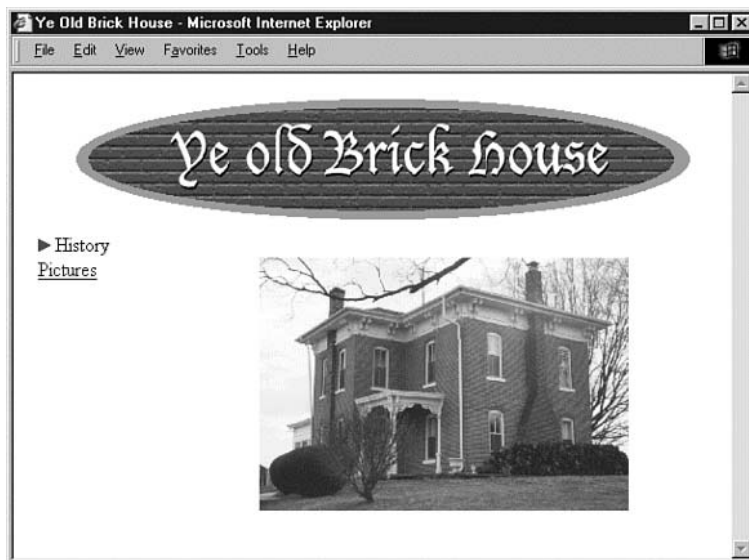


Abbildung 15.5:
Erweitertes Menü
vor dem Hinzufügen des Haupt-
Scripts.



```
<script language="javascript" type="text/javascript">
<!-- Hide JavaScript

// Global Variable declarations
// and image creation.
var ea = new Image(10,10); //ea for expand arrow
var ca = new Image(10,10); //ca for contract arrow
ea.src = "expand.gif";
ca.src = "contract.gif";
var show = 1;
var contentID;
```

Fügen Sie als Nächstes die Sniffer-Funktion hinzu.



```
// Trimmed down browser sniffer that
// detects Navigator 4+ and IE 4+.
// Reference is_nav4up and is_ie4up in other
// portions of the script
var agt = navigator.userAgent.toLowerCase();
var is_major = parseInt(navigator.appVersion);
var is_minor = parseFloat(navigator.appVersion);
var is_nav = ((agt.indexOf('mozilla')!=-1) && (agt.indexOf('es poofer')==-1)
    && (agt.indexOf('compatible') == -1) && (agt.indexOf('opera')==-1)
    && (agt.indexOf('webtv')==-1));
var is_nav4up = (is_nav && (is_major >= 4));
var is_ie = (agt.indexOf("msie") != -1);
var is_ie4up
```

Nun müssen Sie den Browser-übergreifenden DOM-Referenz-Code eingeben.



```
// Object and Style Reference conventions
// for Navigator and IE. Use objRef and
// when referring to objects and both when
// accessing style properties.
var objRef, styleRef;
```

```
if (is_nav4up == true) { //for Nav these are blank
    objRef = ""
    styleRef = ""
}
if (is_ie4up == true) { //for IE the are set
    objRef = ".all"
    styleRef = ".style"
}
```

Jetzt kommen wir zu den Funktionen, die die Arbeit machen werden. Die erste Funktion zeigt das Menü auf, verdeckt es und positioniert es wieder. Beim Erstellen dieser Funktion wende ich die DOM-Referenz-Logik an, ohne mich um den angewandten Browser zu kümmern.



```
// Function to reveal or hide
// sub-menus.
function menuaction() {
    if (show == 1) {
        eval("document" + objRef + ".toc0101" + styleRef + ".visibility = 'visible'");
        eval("document" + objRef + ".toc0102" + styleRef + ".visibility = 'visible'");
        eval("document" + objRef + ".toc0200" + styleRef + ".top = '190'");
        show = 0;
        return true;
    }
    if (show == 0) {
        eval("document" + objRef + ".toc0101" + styleRef + ".visibility = 'hidden'");
        eval("document" + objRef + ".toc0102" + styleRef + ".visibility = 'hidden'");
        eval("document" + objRef + ".toc0200" + styleRef + ".top = '150'");
        show = 1;
        return true;
    }
}
```

Die nächste Funktion zeigt den Inhalt. Auch hier müssen Sie erkennen, welche Elemente zuerst verdeckt werden müssen, um dann den Platz für den Inhalt, den Sie anzeigen wollen, freizumachen.



```
// Function to hide other content layers
// and reveal the chosen layer.
```



```
function showcontent() {
  if (contentID == "0101") {
    eval("document" + objRef + ".splash" + styleRef + ".visibility = 'hidden'");
    eval("document" + objRef + ".content0102" + styleRef + ".visibility = 'hidden'");
    eval("document" + objRef + ".content0200" + styleRef + ".visibility = 'hidden'");
    eval("document" + objRef + ".content0101" + styleRef + ".visibility = 'visible'");
  }
  if (contentID == "0102") {
    eval("document" + objRef + ".splash" + styleRef + ".visibility = 'hidden'");
    eval("document" + objRef + ".content0101" + styleRef + ".visibility = 'hidden'");
    eval("document" + objRef + ".content0200" + styleRef + ".visibility = 'hidden'");
    eval("document" + objRef + ".content0102" + styleRef + ".visibility = 'visible'");
  }
  if (contentID == "0200") {
    eval("document" + objRef + ".splash" + styleRef + ".visibility = 'hidden'");
    eval("document" + objRef + ".content0101" + styleRef + ".visibility = 'hidden'");
    eval("document" + objRef + ".content0102" + styleRef + ".visibility = 'hidden'");
    eval("document" + objRef + ".content0200" + styleRef + ".visibility = 'visible'");
  }
}
// end hiding -->
</script>
```

Das abschließende Listing sehen Sie im Anschluss:



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">

<html>
<head>
<title>Ye old Brick House</title>
<meta http-equiv="Content-Script-Type" content="text/javascript" />

<script language="javascript" type="text/javascript">
```

```
<!-- Hide JavaScript

// Global Variable declarations
// and image creation.
var ea = new Image(10,10); //ea for expand arrow
var ca = new Image(10,10); //ca for contract arrow
ea.src = "expand.gif";
ca.src = "contract.gif";
var show = 1;
var contentID;

// Trimmed down browser sniffer that
// detects Navigator 4+ and IE 4+.
// Reference is_nav4up and is_ie4up in other
// portions of the script
var agt = navigator.userAgent.toLowerCase();
var is_major = parseInt(navigator.appVersion);
var is_minor = parseFloat(navigator.appVersion);
var is_nav = ((agt.indexOf('mozilla')!=-1) && (agt.indexOf('es poofer')==-1)
    && (agt.indexOf('compatible') == -1) && (agt.indexOf('opera')==-1)
    && (agt.indexOf('webtv')==-1));
var is_nav4up = (is_nav && (is_major >= 4));
var is_ie = (agt.indexOf("msie") != -1);
var is_ie4up = (is_ie && (is_major >= 4));

// Object and Style Reference conventions
// for Navigator and IE. Use objRef and
// when referring to objects and both when
// accessing style properties.
var objRef, styleRef;
if (is_nav4up == true) { //for Nav these are blank
    objRef = ""
    styleRef = ""
}
if (is_ie4up == true) { //for IE the are set
    objRef = ".all"
    styleRef = ".style"
}

// Function to reveal or hide
// sub-menus.
function menuaction() {
    if (show == 1) {
        eval("document" + objRef + ".toc0101" + styleRef + ".visibility = 'visible'");
        eval("document" + objRef + ".toc0102" + styleRef + ".visibility = 'visible'");
```

```

    eval("document" + objRef + ".toc0200" + styleRef + ".top = '190'");
    show = 0;
    return true;
}
if (show == 0) {
    eval("document" + objRef + ".toc0101" + styleRef + ".visibility = 'hidden'");
    eval("document" + objRef + ".toc0102" + styleRef + ".visibility = 'hidden'");
    eval("document" + objRef + ".toc0200" + styleRef + ".top = '150'");
    show = 1;
    return true;
}
}

// function to hide other content layers
// and reveal the chosen layer.
function showcontent() {
    if (contentID == "0101") {
        eval("document" + objRef + ".splash" + styleRef + ".visibility = 'hidden'");
        eval("document" + objRef + ".content0102" + styleRef + ".visibility = 'hidden'");
        eval("document" + objRef + ".content0200" + styleRef + ".visibility = 'hidden'");
        eval("document" + objRef + ".content0101" + styleRef + ".visibility = 'visible'");
    }
    if (contentID == "0102") {
        eval("document" + objRef + ".splash" + styleRef + ".visibility = 'hidden'");
        eval("document" + objRef + ".content0101" + styleRef + ".visibility = 'hidden'");
        eval("document" + objRef + ".content0200" + styleRef + ".visibility = 'hidden'");
        eval("document" + objRef + ".content0102" + styleRef + ".visibility = 'visible'");
    }
    if (contentID == "0200") {
        eval("document" + objRef + ".splash" + styleRef + ".visibility = 'hidden'");
        eval("document" + objRef + ".content0101" + styleRef + ".visibility = 'hidden'");
        eval("document" + objRef + ".content0102" + styleRef + ".visibility = 'hidden'");
        eval("document" + objRef + ".content0200" + styleRef + ".visibility = 'visible'");
    }
}
// end hiding -->
</script>

```

```
<style type="text/css">
<!--
#title { position: absolute; top: 20; left: 50; }
#splash { position: absolute; top: 150; left: 200; visibility: visible; }
#toc0100 { position: absolute; top: 130; left: 20; visibility: visible; }
#toc0101 { position: absolute; top: 150; left: 50; visibility: hidden; }
#toc0102 { position: absolute; top: 170; left: 50; visibility: hidden; }
#toc0200 { position: absolute; top: 150; left: 20; visibility: visible; }
#content0101 { position: absolute; top: 150; left: 200; visibility: hidden; }
#content0102 { position: absolute; top: 150; left: 200; visibility: hidden; }
#content0200 { position: absolute; top: 150; left: 200; visibility: hidden; }
-->
</style>
</head>

<body>

<div id="title">

</div>

<div id="toc0100">
<a href="javascript:void(0)" onclick="menuaction();
  if (show == 1) { document.menuimg.src = ea.src; return true }
  if (show == 0) { document.menuimg.src = ca.src; return true }">
</a>
History
</div>

<div id="toc0101">
<a href="javascript:void(0)" onclick="contentID='0101'; showcontent(0101)">The
Early Days</a>
</div>

<div id="toc0102">
<a href="javascript:void(0)" onclick="contentID='0102'; showcontent(0101)">The
Modern Era</a>
</div>

<div id="toc0200">
<a href="javascript:void(0)" onclick="contentID='0200';
showcontent(0101)">Pictures</a>
</div>

<div id="splash" align="center">
```

```

</div>

<div id="content0101">
<p>This brick house was built in the 1800s. Originally part of a large farm that
consisted of several hundred acres, the house and barn now sit on a 4 acre plot of
land.</p>
</div>

<div id="content0102">
<p>There have been several modifications to the house, such as a refurbished
kitchen, but the exterior is largely as it was in the 1800s. The barn, on the
other hand, has been left unchanged.</p>
</div>

<div id="content0200">



</div>

</body>
</html>
```

Dieser Code führt zu dem in Bild 15.6 gezeigten Ergebnis. Ich habe das Menü zwar erweitert, aber noch keinen Inhalt dafür gewählt, was angezeigt werden soll.



Abbildung 15.6:
Erweitertes Menü im MS Inter-
net Explorer 5

Bild 15.7 zeigt, dass ich ein Menüeintrag aufgerufen habe. Der zuerst verdeckte Inhalt wird nun angezeigt und ist sichtbar.

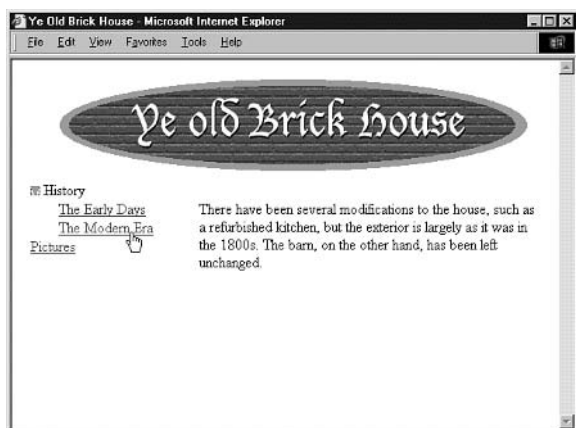


Abbildung 15.7:
Verdeckter Inhalt wird mit
DHTML aufgedeckt.

Bild 15.8 zeigt dieselbe Seite im Netscape Navigator und zeigt, dass es sich hier wirklich um Browser-übergreifende DHTML handelt. Hier habe ich das Menü erweitert und für das Aufzeigen der Fotos `Pictures` angeklickt.



Abbildung 15.8:
Bestätigung der Browser-übergrei-
fenden Kompatibilität im Net-
scape Navigator 4.61.

Übung 15.2: Das Tic-Tac-Toe-Spiel

Die letzte Übung in diesem Kapitel ist, eine Webseite für ein Tic-Tac-Toe Spiel zu erstellen. Hierfür soll einzig und allein DHTML benutzt werden. Das Spiel und die Webseite auf dem es abgebildet ist, benutzt einige praktische DHTML-Techniken, wie etwa Browser-Sniffer, Beeinflussen der z-Ordnung, Element-Sichtbarkeit und Animation. Bild 15.9 zeigt, wie das Ergebnis dieser Übung aussehen soll.

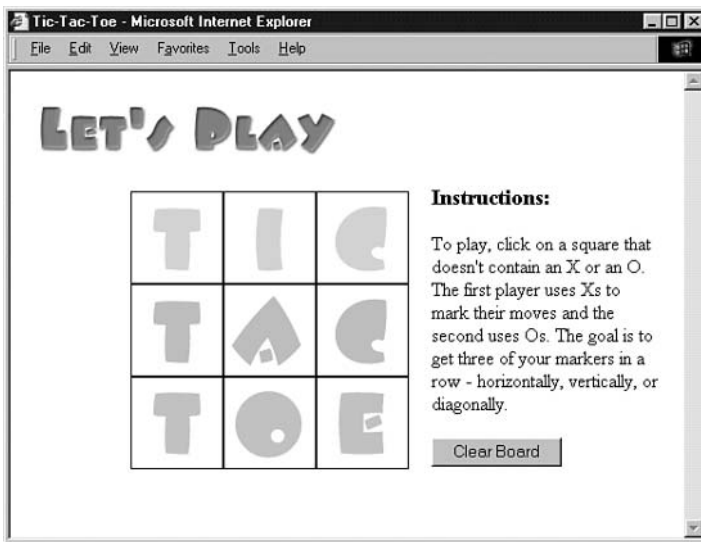


Abbildung 15.9:
Das Spiel Tic-Tac-Toe
als Browser-übergrei-
fende Webseite

Auf der statischen Seite dieses Buches kann man nicht erkennen, dass der Titel von links eingespielt wird.

Erstellen der Grafiken

Grafik ist ein wichtiger Teil der Webseiten. Für dieses Tic-Tac-Toe-Spiel sollte die Seite ein lustiges »Outfit« bekommen, deshalb werde ich sie sehr farbenfroh gestalten.

Nachdem ich das Konzept der Webseite erstellt und schon einmal versuchsweise koordiniert habe, habe ich mich zu einem Grafikkonzept entschlossen. Mein erster Gedanke war, dass das Spielfeld ein großes Quadrat sein sollte. Dieses Quadrat sollte dann in neun Boxen unterteilt werden. Zu Anfang habe ich das auch so erstellt, entdeckte aber dann, dass es einfacher sein würde, neun einzelne Spielfeldgrafiken zu erstellen, die innerhalb eines `<div>`-Elements platziert sein sollten. Da jede Grafik präzise mit CSS platziert werden kann, wird man nicht mehr erkennen können, dass sich das Spielfeld aus neun Einzelteilen zusammensetzt.



Für die Grafik verwendete ich Adobe Photoshop 5 und Adobe ImageReady. Natürlich gibt es auch andere (billigere) Grafikprogramme. Das Konzept der Gestaltung und Optimierung ist ja im Wesentlichen das Gleiche und unabhängig von den Applikationen, die Sie verwenden.

In einem Tic-Tac-Toe Spiel haben wir eine ungerade Anzahl von Tafeln. Um die Tafeln visuell interessant zu gestalten, habe ich für jede Tafel das Maß 77 x 77 Pixel gewählt und in jede Tafel einen Buchstaben des Spiels (T-I-C, etc.) gestellt. Hierbei erscheint »Tic« in rot, »Tac« in schwarz und »Toe« in blau.

Danach habe ich an der Sichtbarkeit des Textes herumgebastelt und zwar so lange, bis die Buchstaben auf ein angemessenes Hintergrund-Level abgestimmt waren. Ich wollte nämlich nicht, dass die Buchstaben zu grell leuchten und die Bestandteile des Spiels in den Schatten stellen.

Schließlich habe ich neun .gif Dateien gleicher Größe erstellt (siehe Bild 15.10) und habe sie nach der jeweiligen Position, die sie im Spielfeld innehaben, benannt (topleft.gif, centercenter.gif usw.).

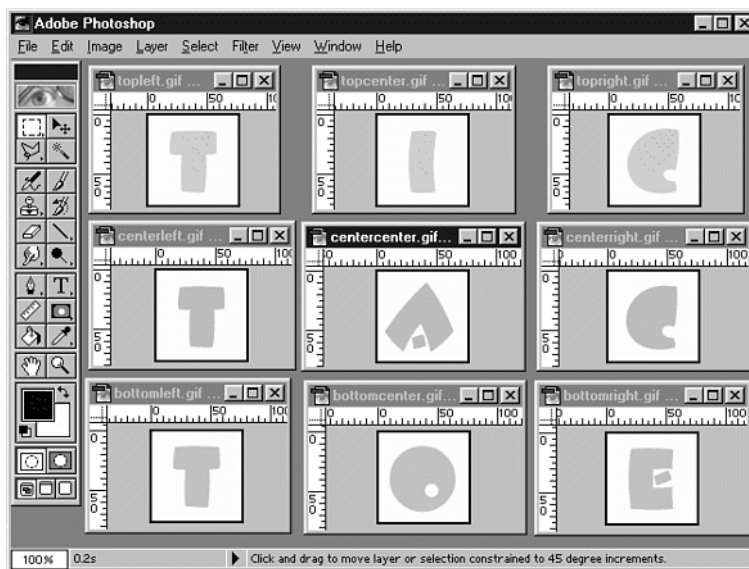


Abbildung 15.10:
Die neun Bildtafeln wurden so verändert, dass die Seite nicht von ihnen dominiert wird.

Ich musste jetzt nur noch zwei Spielgrafiken erstellen, nämlich X und O, welche ebenfalls 77 x 77 Pixels messen und nicht transparent sind. Ich wollte nämlich nicht, dass irgendein Spielteil nach einem Spielzug noch zu sehen ist. Bild 15.11 zeigt X und O mit Photoshop.

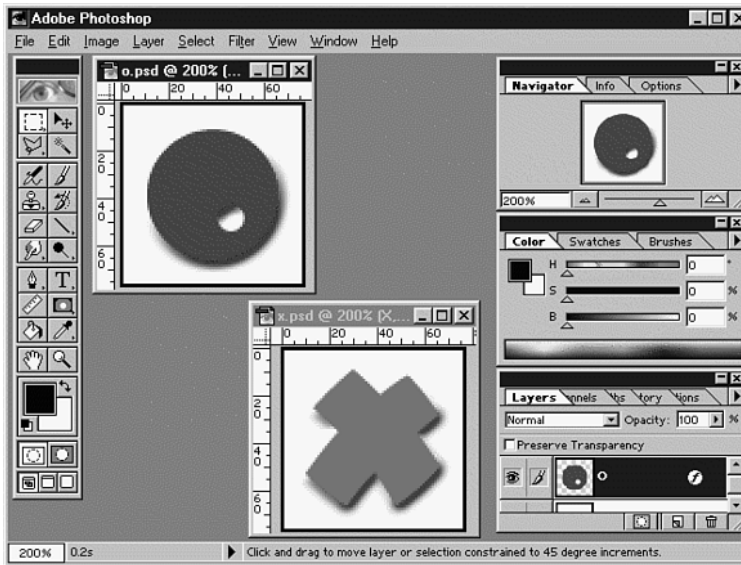


Abbildung 15.11:
Die X und O
Spielsymbole

Als Letztes erstellte ich die Titelgrafik (siehe Bild 15.12), welche in die Seite einge-
spielt wird, sobald diese heruntergeladen wird. Anstatt »Tic-Tac-Toe« zu wiederholen,
wählte ich den Titel »Let es play«.

Die Schriftart mit der ich den gesamten Text erstellt habe ist Beeskness ITC. Für ei-
nen anderen Text würde ich eine Schriftart wählen, die etwas einfacher zu lesen ist,
aber für diesen Zweck schien mir Beeskness ITC geeignet.

Die Grafik wurde erstellt, machen wir uns also ans Kodieren.

HTML mit Stil schreiben

Beginnen Sie mit dem Deklarieren der HTML-Version, die Sie verwenden, und erstel-
len Sie dann den Dokumentkopf und -rumpf, um eine Umgebung zu formen.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">
```

```
<html>
<head>
<title>Tic-Tac-Toe</title>
```

```
<meta http-equiv="Content-Script-Type" content="text/javascript" />

<!-- Most JavaScript will go here -->

</head>
<body>

<!-- HTML will go here -->

</body>
</html>
```



Abbildung 15.12:
Titelgrafik, die in
die Seite einge-
spielt wird

Ab hier, arbeiten Sie innerhalb des Dokumenttrumpfes. Erstellen Sie zuerst den Grafiktitel, den Sie später animieren werden. Geben Sie ein `<div>`-Element ein und geben Sie diesem eine `id`, damit Sie es über das Script, das Sie später erstellen werden, manipulieren können. Verwenden Sie absolute Positionierung, um den Titel auf der Webseite zu fixieren. Da der Titel außerhalb des Bildschirms beginnen soll, weisen Sie ihm negative Werte mit `left`- und `top`-CSS-Eigenschaften zu. Die `z-index`-Eigenschaft legt fest, dass der Titel oberhalb sämtlicher CSS Ebenen (Layers) sein wird. Fügen Sie das Bild, das das `<img>`-Element enthält, dem Inhalt des `<div>`-Elements hinzu und schließen Sie das `div`-Element mit dem abschließendem `</div>`-Tag.



```
<div id="titleimage"
  style="position: absolute;
    left: -100;
    top: -100;
    z-index: 5">

</div>
```

Das war doch gar nicht so schwer, und es wird Ihnen leicht fallen, die nächsten Elemente Ihres Spiels zu erstellen: Spielfeld und Spielsymbole.

Das Spielfeld besteht aus neun einzelnen `<div>`-Elementen, jedes mit einem eigenen Bild versehen. Sämtliche Abmessungen wurden in Photoshop bestimmt, um die Koordinaten der `top`- und `left`-Eigenschaften zu bekommen. Da das Spielfeld die unterste Ebene sein soll, weisen Sie jedem Teil die `z-order 0` zu. Außerdem sollte jedes Quadrat sichtbar bleiben. Hierfür müssen Sie die `visibility`-Eigenschaft einfügen. Sobald ein Spieler das Spielfeld anklickt, wollen Sie natürlich, dass ein Ereignis stattfindet. Verwenden Sie deshalb das `onclick`-Attribut jedes `<div>`-Elements, um eine Variable zu setzen, die die Position der Tafeln definiert und die `update()` Funktion aufruft, die das Spielsymbol auf der Tafel sichtbar macht. Wenn Sie nun den HTML-Code ansehen, der das Spielfeld erstellt hat, werden Sie feststellen, dass jedes Quadrat eine eindeutige ID-, `top`- und `left`- Eigenschaft hat. Wenn der Anwender mit der Maus auf das Spielfeld klickt, setzt dieses die `position`-Variable auf die korrekte Position auf dem Spielfeld. Dies werden Sie später beim Scripten der `update()` Funktion verwenden.



```
<!-- begin game board divisions -->

<div id="topleft"
  style="position: absolute;
    top: 100;
    left: 100;
    z-index: 0;
    visibility: visible"
  onclick="position = 'topleft'; update();">

</div>

<div id="topcenter"
```

```

        style="position: absolute;
            top: 100;
            left: 177;
            z-index: 0;
            visibility: visible"
        onclick="position = 'topcenter'; update();">

</div>

<div id="topright"
    style="position: absolute;
        top: 100;
        left: 254;
        z-index: 0"
    onclick="position = 'topright'; update()">

</div>

<div id="centerleft"
    style="position: absolute;
        top: 177;
        left: 100;
        z-index: 0"
    onclick="position = 'centerleft'; update()">

</div>

<div id="centercenter"
    style="position: absolute;
        top: 177;
        left: 177;
        z-index: 0"
    onclick="position = 'centercenter'; update()">

</div>

<div id="centerright"
    style="position: absolute;
        top: 177;
        left: 254;
        z-index: 0"
    onclick="position = 'centerright'; update()">

</div>

```

```

<div id="bottomleft"
  style="position: absolute;
    top: 254;
    left: 100;
    z-index: 0"
  onclick="position = 'bottomleft'; update()">

</div>

<div id="bottomcenter"
  style="position: absolute;
    top: 254;
    left: 177;
    z-index: 0"
  onclick="position = 'bottomcenter'; update()">

</div>

<div id="bottomright"
  style="position: absolute;
    top: 254;
    left: 254;
    z-index: 0"
  onclick="position = 'bottomright'; update()">

</div>

<!-- end game board divisions -->

```

Bild 15.13 zeigt Ihnen, wie weit wir bis jetzt schon gekommen sind.

Das Spielfeld ist nun erstellt und Sie können jedes Spielsymbol miteinbeziehen. Was das Design betrifft, dachte ich, es wäre am einfachsten die X und O Spielsymbole so zu erstellen, dass sie auf jeder Tafel des Spielfelds liegen, vorerst aber verdeckt bleiben. Wenn man eine Spieltafel anklickt, bestimmt das Script auf welcher Seite X oder O ist und die passende Grafik dann sichtbar wird.

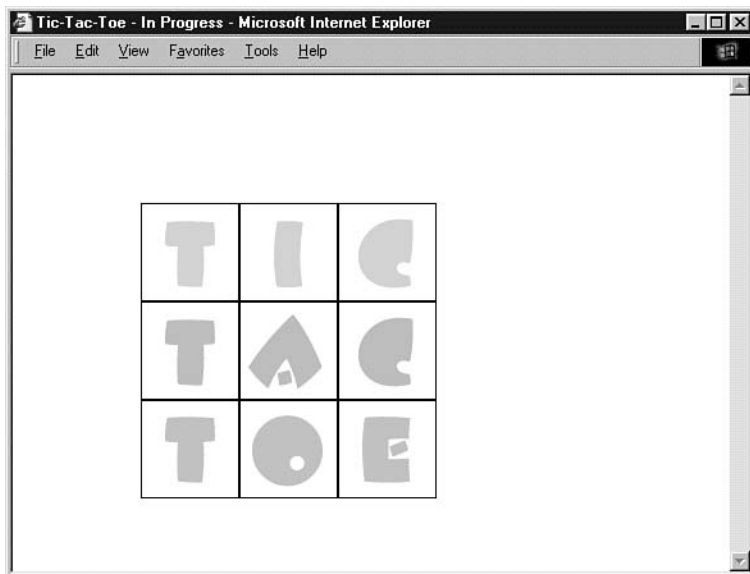


Abbildung 15.13:
Der außerhalb des
Bildschirms posi-
tionierte Titel
und das Spielfeld



```
<!-- begin hidden o markers -->

<div id="toplefto"
  style="position: absolute;
        top: 100;
        left: 100;
        z-index: 3;
        visibility: hidden">

</div>

<div id="topcentero"
  style="position: absolute;
        top: 100;
        left: 177;
```

```

        z-index: 3;
        visibility: hidden">

</div>

<div id="toprighto"
    style="position: absolute;
        top: 100;
        left: 254;
        z-index: 3;
        visibility: hidden">

</div>

<div id="centerlefto"
    style="position: absolute;
        top: 177;
        left: 100;
        z-index: 3;
        visibility: hidden">

</div>

<div id="centercentero"
    style="position: absolute;
        top: 177;
        left: 177;
        z-index: 3;
        visibility: hidden">

</div>

<div id="centerrighto"
    style="position: absolute;
        top: 177;
        left: 254;
        z-index: 3;
        visibility: hidden">

</div>

<div id="bottomlefto"
    style="position: absolute;
        top: 254;
        left: 100;
        z-index: 3;
        visibility: hidden">

```

```

</div>
```

```
<div id="bottomcentero"
  style="position: absolute;
    top: 254;
    left: 177;
    z-index: 3;
    visibility: hidden">

</div>
```

```
<div id="bottomrighto"
  style="position: absolute;
    top: 254;
    left: 254;
    z-index: 3;
    visibility: hidden">

</div>
```

```
<!-- end hidden o markers -->
```

```
<!-- begin hidden x markers -->
```

```
<div id="topleftx"
  style="position: absolute;
    top: 100;
    left: 100;
    z-index: 2;
    visibility: hidden">

</div>
```

```
<div id="topcenterx"
  style="position: absolute;
    top: 100;
    left: 177;
    z-index: 2;
    visibility: hidden">

</div>
```

```
<div id="toprightx"
  style="position: absolute;
    top: 100;
```



```

        left: 254;
        z-index: 2;
        visibility: hidden">

</div>

<div id="centerleftx"
    style="position: absolute;
        top: 177;
        left: 100;
        z-index: 2;
        visibility: hidden">

</div>

<div id="centercenterx"
    style="position: absolute;
        top: 177;
        left: 177;
        z-index: 2;
        visibility: hidden">

</div>

<div id="centerrightx"
    style="position: absolute;
        top: 177;
        left: 254;
        z-index: 2;
        visibility: hidden">

</div>

<div id="bottomleftx"
    style="position: absolute;
        top: 254;
        left: 100;
        z-index: 2;
        visibility: hidden">

</div>

<div id="bottomcenterx"
    style="position: absolute;
        top: 254;
        left: 177;

```

```

        z-index: 2;
        visibility: hidden">

</div>

<div id="bottomrightx"
    style="position: absolute;
        top: 254;
        left: 254;
        z-index: 2;
        visibility: hidden">

</div>

<!-- end hidden x markers -->

```

Das war ja ganz schön lang. Jetzt ist es an der Zeit einige Spielieranweisungen zu erstellen. Hier kommt das `<div>`-Element wieder ins »Spiel«. Sie werden ebenfalls ein Formular anwenden, das eine CLEAR-BOARD-Schaltfläche (um das Spiel wieder neu beginnen zu können) anzeigt.



```

<div id="instructions"
    style="position: absolute;
        top: 95;
        left: 350">
<h3>Instructions:</h3>
<p>To play, click on a square that doesn't contain an X or an O. The first player
uses Xs to mark their moves and the second uses Os. The goal is to get three of
your markers in a row - horizontally, vertically, or diagonally.</p>

<form name="clearboard">
<input type="submit" value="Clear Board" onclick="window.location.reload()" />

</form>

```

Bild 15.14 zeigt, was bis jetzt erstellt wurde.

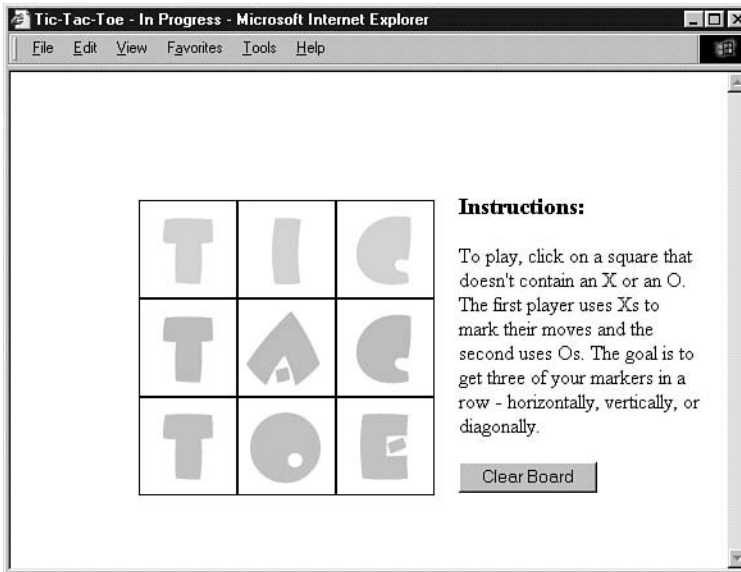


Abbildung 15.14:
Hinzufügen der
Spieleranweisung
und der CLEAR-
BOARD-Schalt-
fläche.

Weil ich eine komplette Kontrolle über das endgültige Erscheinungsbild der Seite haben möchte, soll sich dieses Element in einer absoluten Position befinden. Das Formular beinhaltet weder ein `action`- noch ein `method`-Attribut. Es dient lediglich als Container für die CLEAR-BOARD-Schaltfläche. Diese ruft die `window.location.reload()`-Funktion zum erneuten Laden der Seite auf.

JavaScripting bis in die frühen Morgenstunden

Die Webseite ist erstellt und wir können nun beginnen das Script zu schreiben, das die gesamte DHTML-Arbeit machen wird.

Zuerst müssen Sie das `<script>`-Element öffnen und einige Variablen für später erstellen. Gleich nach dem einleitenden `<script>`-Tag ist es notwendig, ein HTML-Kommentar `<!--` zu benutzen, um den Code vor nicht Script-fähigen Browsern zu verstecken. Versichern Sie sich, dass Sie auch ein abschließendes Kommentartag `-->` am Ende des Scripts stehen haben. JavaScript Kommentare haben zwei Schrägstriche (`//`).



```
<script language="javascript" type="text/javascript">
<!-- Hide JavaScript
```

```
// Declaring a few variables
var player1 = "x";
var player2 = "o";
var currentplayer;
var position;
var yhop = 3;
var ygoal = 20;
var xhop = 3;
var xgoal = 20;
var delay = 1;
var xpos;
var ypos;
```

Als Nächstes müssen Sie die Browser »aufschnüffeln« und erkennen. Da diese Webseite bis zu einem gewissen Punkt Browser-übergreifend sein soll, ist es wichtig, die Browser erst einmal zu überprüfen.

Hierbei verwende ich einen modifizierten Browser-übergreifenden Code von Netscape, der diverse Informationen der Browser auswertet. Darunter zum Beispiel die Haupt- und Nebenversion des Browsers, die Plattform auf der dieser ausgeführt wird, usw.. Diesen Code müssen Sie in den HTML-Dokumentenkopf eingeben und im `<script>`-Tag verpacken.



```
// Trimmed down browser sniffer that
// detects Navigator 4+ and IE 4+.
// Reference is_nav4up and is_ie4up in other
// portions of the script
var agt = navigator.userAgent.toLowerCase();
var is_major = parseInt(navigator.appVersion);
var is_minor = parseFloat(navigator.appVersion);
var is_nav = ((agt.indexOf('mozilla')!=-1) && (agt.indexOf('es poofers')==-1)
    && (agt.indexOf('compatible') == -1) && (agt.indexOf('opera')==-1)
    && (agt.indexOf('webtv')==-1));
var is_nav4up = (is_nav && (is_major >= 4));
```

```
var is_ie = (agt.indexOf("msie") != -1);
var is_ie4up = (is_ie && (is_major >= 4));
```

Um zu sehen, ob Sie es mit dem Netscape Navigator oder dem MS Internet Explorer in den Versionen 4 oder einer höheren Version zu tun haben, verwenden Sie folgendes Script:

```
if (is_nav4up) {
    do something }
if (is_ie4up) {
    do something else }
```

Die erste Funktion initialisiert das Script und wird aufgerufen, wenn die Seite im Browser geladen wird. Um herauszufinden, welcher Spieler als Nächstes dran ist, setzen Sie die `currentplayer`-Variable (aktueller Spieler) auf `player1`. Damit wird die erste Person, die das Spiel anklickt, als `player1` identifiziert.

Der zweite Teil dieser Funktion testet, ob der Browser nun der Netscape Navigator ist und auf Mausereignisse reagiert. Das ist besonders wichtig, da Netscape das `onclick`-Ereignis im `<div>`-Element erkennt. Rufen Sie dann die `slidetitle()` Funktion auf, um die Titelgrafik zu animieren und um sie auf die Webseite zu bringen.



```
// Initializes first player and
// begin event capturing for Netscape.
// Call from BODY element when loading page.
function init() {
    currentplayer = player1;
    if (is_nav4up == true ) {
        document.captureEvents(Event.CLICK | Event.MOUSEDOWN)
        document.onclick=updatenav
        document.onmousedown=updatenav
    }
    slidetitle();
}
```

Die `slidetitle()` Funktion überprüft welcher Browser gerade die Grafik animiert. Der einzig wirkliche Unterschied besteht darin, welcher Browser das Titelbild aufruft. Netscape verwendet `document.titleimage` und der Internet Explorer `document.all.titleimage.style`. Die `if`-Anweisung überprüft, dass die `left`- und `top`-Werte für die Titelgrafik nicht den später deklarierten `xgoal`- und `ygoal` Variablen entsprechen. Diese Variablen definieren die endgültige Position des Bildes. Wenn keine der Eigenschaften der end-

gültigen Position entsprechen, fügen Sie die `xhop`- und `yhop`-Werte der aktuellen Position hinzu. Diese Werte legen fest, wie viele Pixel das Bild bei jedem Spielzug dieser Routine bewegt.



```
// Slides the title graphic from offscreen.
function slidetitle() {
  if (is_nav4up == true ) {
    if ((parseInt(document.titleimage.left) != xgoal) ||
        (parseInt(document.titleimage.top) != ygoal)) {
      document.titleimage.left = document.titleimage.left + xhop;
      document.titleimage.top = document.titleimage.top + yhop;
      timeoutID = setTimeout("slidetitle()", delay); }
  }
  if (is_ie4up == true ) {
    if ((parseInt(document.all.titleimage.style.left) != xgoal) ||
        (parseInt(document.all.titleimage.style.top) != ygoal)) {
      document.all.titleimage.style.left =
parseInt(document.all.titleimage.style.left) + xhop;
      document.all.titleimage.style.top =
parseInt(document.all.titleimage.style.top) + yhop;
      timeoutID = setTimeout("slidetitle()", delay); }
  }
}
```

Da der Netscape Navigator nicht auf das `onclick`-Ereignis reagiert, muss eine separate Routine erstellt werden, um das Spielfeld auf den neuesten Stand zu bringen sobald ein Spieler ein Quadrat anklickt.

Mit den `e.pageX`- und `e.pageY`-Eigenschaften erhalten Sie zunächst einmal die Position des Mauszeigers. Der Code innerhalb jeder Spieler-Routine ist identisch und bewirkt, dass die X Bilder für Spieler 1 und die O Bilder für Spieler 2 sichtbar sind. Ein erweiterter Positionscheck ist notwendig, damit festgelegt werden kann, welches Quadrat gerade angeklickt wird. Ein weiterer wichtiger Punkt ist, die Sichtbarkeit des Gegenspielers, der gerade am Zug ist, zu überprüfen. Denn man muss festlegen können, ob das Quadrat schon markiert wurde oder nicht. Ist das Spielfeld des Spielers, der gerade am Zug ist, sichtbar, geschieht nichts.



```
// Updates when Navigator detects a mouse event.
// IE appears to ignore in favor of onClick.
function updatenav(e) {
    xpos = e.pageX;
    ypos = e.pageY;
    if (currentplayer == player1) {
        if (xpos >= 100 && xpos <= 176 && ypos >= 100 && ypos <= 176 &&
document.topleftto.visibility == "hide") {
            document.toplefttx.visibility = "visible"; currentplayer = player2;}
        if (xpos >= 177 && xpos <= 253 && ypos >= 100 && ypos <= 176 &&
document.topcentero.visibility == "hide") {
            document.topcenterx.visibility = "visible"; currentplayer = player2;}
        if (xpos >= 254 && xpos <= 331 && ypos >= 100 && ypos <= 176 &&
document.toprightto.visibility == "hide") {
            document.toprighttx.visibility = "visible"; currentplayer = player2;}
        if (xpos >= 100 && xpos <= 176 && ypos >= 177 && ypos <= 253 &&
document.centerleftto.visibility == "hide") {
            document.centerlefttx.visibility = "visible"; currentplayer = player2;}
        if (xpos >= 177 && xpos <= 253 && ypos >= 177 && ypos <= 253 &&
document.centercentero.visibility == "hide") {
            document.centercenterx.visibility = "visible"; currentplayer = player2;}
        if (xpos >= 254 && xpos <= 331 && ypos >= 177 && ypos <= 253 &&
document.centerrightto.visibility == "hide") {
            document.centerrighttx.visibility = "visible"; currentplayer = player2;}
        if (xpos >= 100 && xpos <= 176 && ypos >= 254 && ypos <= 331 &&
document.bottomleftto.visibility == "hide") {
            document.bottomlefttx.visibility = "visible"; currentplayer = player2;}
        if (xpos >= 177 && xpos <= 253 && ypos >= 254 && ypos <= 331 &&
document.bottomcentero.visibility == "hide") {
            document.bottomcenterx.visibility = "visible"; currentplayer = player2;}
        if (xpos >= 254 && xpos <= 331 && ypos >= 254 && ypos <= 331 &&
document.bottomrightto.visibility == "hide") {
            document.bottomrighttx.visibility = "visible"; currentplayer = player2;}
    }
    else {
        if (xpos >= 100 && xpos <= 176 && ypos >= 100 && ypos <= 176 &&
document.toplefttx.visibility == "hide") {
            document.topleftto.visibility = "show"; currentplayer = player1;}
        if (xpos >= 177 && xpos <= 253 && ypos >= 100 && ypos <= 176 &&
document.topcenterx.visibility == "hide") {
            document.topcentero.visibility = "show"; currentplayer = player1;}
        if (xpos >= 254 && xpos <= 331 && ypos >= 100 && ypos <= 176 &&
```

```
document.toprightx.visibility == "hide") {
    document.toprighto.visibility = "show"; currentplayer = player1;}
    if (xpos >= 100 && xpos <= 176 && ypos >= 177 && ypos <= 253 &&
document.centerleftx.visibility == "hide") {
    document.centerlefo.visibility = "visible"; currentplayer = player1;}
    if (xpos >= 177 && xpos <= 253 && ypos >= 177 && ypos <= 253 &&
document.centercenterx.visibility == "hide") {
    document.centercentro.visibility = "visible"; currentplayer = player1;}
    if (xpos >= 254 && xpos <= 331 && ypos >= 177 && ypos <= 253 &&
document.centerrightx.visibility == "hide") {
    document.centerrighto.visibility = "visible"; currentplayer = player1;}
    if (xpos >= 100 && xpos <= 176 && ypos >= 254 && ypos <= 331 &&
document.bottomleftx.visibility == "hide") {
    document.bottomlefo.visibility = "visible"; currentplayer = player1;}
    if (xpos >= 177 && xpos <= 253 && ypos >= 254 && ypos <= 331 &&
document.bottomcenterx.visibility == "hide") {
    document.bottomcentro.visibility = "visible"; currentplayer = player1;}
    if (xpos >= 254 && xpos <= 331 && ypos >= 254 && ypos <= 331 &&
document.bottomrightx.visibility == "hide") {
    document.bottomrighto.visibility = "visible"; currentplayer = player1;}
}
}
```

Als Nächstes kommen wir zu der Funktion, die vom Internet Explorer benutzt wird, um das Spielfeld auf den neuesten Stand zu bringen. Überprüfen Sie zunächst wer gerade spielt (Spieler 1 oder 2), legen Sie das Spielersymbol fest und bestimmen Sie den Spieler, der nach einem gewonnenen Spiel beginnen darf (gewinnt Spieler 1, beginnt Spieler 2 beim nächsten Spiel).

Als Nächstes legen Sie jede Positon fest und legen die `visibility`-Eigenschaft für den Spieler fest, der markieren muss. Sie müssen die Koordinaten nicht überprüfen, da diese Routine von der Stelle innerhalb des Spielfelds, die gerade angeklickt wird, automatisch aufgerufen wird.



```
// Updates with the IE onClick event. Navigator
// ignores this event so no need to detect.
function update() {
    if (currentplayer == player1) {
        playermarker = "x";
        currentplayer = player2;
    }
    else {
```



```

    playermarker = "o";
    currentplayer = player1;
}
if (is_ie4up == true) {
    if (position == "topleft") {
        eval("document.all.topleft"+playermarker+
".style.visibility = 'visible'");}
    if (position == "topcenter") {
        eval("document.all.topcenter"+playermarker+
".style.visibility = 'visible'");}
    if (position == "topright") {
        eval("document.all.topright"+playermarker+
".style.visibility = 'visible'");}
    if (position == "centerleft") {
        eval("document.all.centerleft"+playermarker+
".style.visibility = 'visible'");}
    if (position == "centercenter") {
        eval("document.all.centercenter"+playermarker+
".style.visibility = 'visible'");}
    if (position == "centerright") {
        eval("document.all.centerright"+playermarker+
".style.visibility = 'visible'");}
    if (position == "bottomleft") {
        eval("document.all.bottomleft"+playermarker+
".style.visibility = 'visible'");}
    if (position == "bottomcenter") {
        eval("document.all.bottomcenter"+playermarker+
".style.visibility = 'visible'");}
    if (position == "bottomright") {
        eval("document.all.bottomright"+playermarker+
".style.visibility = 'visible'");}
    }
}
//--> end hide JavaScript
</script>

```

Der letzte Teil des Scripts erscheint im HTML Code selbst. Stellen Sie sicher, dass Sie die `onload`-Funktion, wie folgt, im einleitendem `<body>`-Tag aufrufen.

```
<body onload="init()">
```

Das Resultat

Das war ein ganzes Stück Arbeit. Bis zu diesem Punkt ist es aber nicht offensichtlich, ob Sie wirklich alle drei DHTML-Technologien – HTML, CSS, Scripting – verwenden müssen, damit auch alles funktioniert.

Hier ist nun das endgültige Listing:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">

<html>
<head>
<title>Tic-Tac-Toe</title>
<meta http-equiv="Content-Script-Type" content="text/javascript" />

<script language="javascript" type="text/javascript">
<!-- Hide JavaScript

// Declaring a few variables
var player1 = "x";
var player2 = "o";
var currentplayer;
var position;
var yhop = 3;
var ygoal = 20;
var xhop = 3;
var xgoal = 20;
var delay = 1;
var xpos;
var ypos;

// Trimmed down browser sniffer that
// detects Navigator 4+ and IE 4+.
// Reference is_nav4up and is_ie4up in other
// portions of the script
var agt = navigator.userAgent.toLowerCase();
var is_major = parseInt(navigator.appVersion);
var is_minor = parseFloat(navigator.appVersion);
var is_nav = ((agt.indexOf('mozilla')!=-1) && (agt.indexOf('es poofer')==-1)
    && (agt.indexOf('compatible') == -1) && (agt.indexOf('opera')==-1)
    && (agt.indexOf('webtv')==-1));
var is_nav4up = (is_nav && (is_major >= 4));
var is_ie = (agt.indexOf("msie") != -1);
var is_ie4up = (is_ie && (is_major >= 4));

// Initializes first player and
// begin event capturing for Netscape.
// Call from BODY element when loading page.
function init() {
    currentplayer = player1;
    if (is_nav4up == true ) {
```

```

document.captureEvents(Event.CLICK | Event.MOUSEDOWN)
document.onclick=updatenav
document.onmousedown=updatenav
}
slidetitle();
}

// Slides the title graphic from offscreen.
function slidetitle() {
  if (is_nav4up == true ) {
    if ((parseInt(document.titleimage.left) != xgoal) ||
        (parseInt(document.titleimage.top) != ygoal)) {
      document.titleimage.left = document.titleimage.left + xhop;
      document.titleimage.top = document.titleimage.top + yhop;
      timeoutID = setTimeout("slidetitle()", delay); }
  }
  if (is_ie4up == true ) {
    if ((parseInt(document.all.titleimage.style.left) != xgoal) ||
        (parseInt(document.all.titleimage.style.top) != ygoal)) {
      document.all.titleimage.style.left =
parseInt(document.all.titleimage.style.left) + xhop;
      document.all.titleimage.style.top =
parseInt(document.all.titleimage.style.top) + yhop;
      timeoutID = setTimeout("slidetitle()", delay); }
  }
}

// Updates when Navigator detects a mouse event.
// IE appears to ignore in favor of onClick.
function updatenav(e) {
  xpos = e.pageX;
  ypos = e.pageY;
  if (currentplayer == player1) {
    if (xpos >= 100 && xpos <= 176 && ypos >= 100 && ypos <= 176 &&
document.topleftto.visibility == "hide") {
      document.toplefttx.visibility = "visible"; currentplayer = player2;}
    if (xpos >= 177 && xpos <= 253 && ypos >= 100 && ypos <= 176 &&
document.topcenterto.visibility == "hide") {
      document.topcenterx.visibility = "visible"; currentplayer = player2;}
    if (xpos >= 254 && xpos <= 331 && ypos >= 100 && ypos <= 176 &&
document.toprightto.visibility == "hide") {
      document.toprighttx.visibility = "visible"; currentplayer = player2;}
    if (xpos >= 100 && xpos <= 176 && ypos >= 177 && ypos <= 253 &&
document.centerleftto.visibility == "hide") {
      document.centerlefttx.visibility = "visible"; currentplayer = player2;}
  }
}

```

```
    if (xpos >= 177 && xpos <= 253 && ypos >= 177 && ypos <= 253 &&
document.centercentero.visibility == "hide") {
        document.centercenterx.visibility = "visible"; currentplayer = player2;}
    if (xpos >= 254 && xpos <= 331 && ypos >= 177 && ypos <= 253 &&
document.centerrighto.visibility == "hide") {
        document.centerrightx.visibility = "visible"; currentplayer = player2;}
    if (xpos >= 100 && xpos <= 176 && ypos >= 254 && ypos <= 331 &&
document.bottomlefto.visibility == "hide") {
        document.bottomleftx.visibility = "visible"; currentplayer = player2;}
    if (xpos >= 177 && xpos <= 253 && ypos >= 254 && ypos <= 331 &&
document.bottomcentero.visibility == "hide") {
        document.bottomcenterx.visibility = "visible"; currentplayer = player2;}
    if (xpos >= 254 && xpos <= 331 && ypos >= 254 && ypos <= 331 &&
document.bottomrighto.visibility == "hide") {
        document.bottomrightx.visibility = "visible"; currentplayer = player2;}
}
else {
    if (xpos >= 100 && xpos <= 176 && ypos >= 100 && ypos <= 176 &&
document.topleftx.visibility == "hide") {
        document.toplefto.visibility = "show"; currentplayer = player1;}
    if (xpos >= 177 && xpos <= 253 && ypos >= 100 && ypos <= 176 &&
document.topcenterx.visibility == "hide") {
        document.topcentero.visibility = "show"; currentplayer = player1;}
    if (xpos >= 254 && xpos <= 331 && ypos >= 100 && ypos <= 176 &&
document.toprightx.visibility == "hide") {
        document.toprighto.visibility = "show"; currentplayer = player1;}
    if (xpos >= 100 && xpos <= 176 && ypos >= 177 && ypos <= 253 &&
document.centerleftx.visibility == "hide") {
        document.centerlefto.visibility = "visible"; currentplayer = player1;}
    if (xpos >= 177 && xpos <= 253 && ypos >= 177 && ypos <= 253 &&
document.centercenterx.visibility == "hide") {
        document.centercentero.visibility = "visible"; currentplayer = player1;}
    if (xpos >= 254 && xpos <= 331 && ypos >= 177 && ypos <= 253 &&
document.centerrightx.visibility == "hide") {
        document.centerrighto.visibility = "visible"; currentplayer = player1;}
    if (xpos >= 100 && xpos <= 176 && ypos >= 254 && ypos <= 331 &&
document.bottomleftx.visibility == "hide") {
        document.bottomlefto.visibility = "visible"; currentplayer = player1;}
    if (xpos >= 177 && xpos <= 253 && ypos >= 254 && ypos <= 331 &&
document.bottomcenterx.visibility == "hide") {
        document.bottomcentero.visibility = "visible"; currentplayer = player1;}
    if (xpos >= 254 && xpos <= 331 && ypos >= 254 && ypos <= 331 &&
document.bottomrightx.visibility == "hide") {
        document.bottomrighto.visibility = "visible"; currentplayer = player1;}
}
}
```

```
// Updates with the IE onClick event. Navigator
// ignores this event so no need to detect.
function update() {
    if (currentplayer == player1) {
        playermarker = "x";
        currentplayer = player2;
    }
    else {
        playermarker = "o";
        currentplayer = player1;
    }
    if (is_ie4up == true) {
        if (position == "topleft") {
            eval("document.all.topleft"+playermarker+".style.visibility = 'visible'");
        }
        if (position == "topcenter") {
            eval("document.all.topcenter"+playermarker+".style.visibility = 'visible'");
        }
        if (position == "topright") {
            eval("document.all.topright"+playermarker+".style.visibility = 'visible'");
        }
        if (position == "centerleft") {
            eval("document.all.centerleft"+playermarker+".style.visibility = 'visible'");
        }
        if (position == "centercenter") {
            eval("document.all.centercenter"+playermarker+".style.visibility = 'visible'");
        }
        if (position == "centerright") {
            eval("document.all.centerright"+playermarker+".style.visibility = 'visible'");
        }
        if (position == "bottomleft") {
            eval("document.all.bottomleft"+playermarker+".style.visibility = 'visible'");
        }
        if (position == "bottomcenter") {
            eval("document.all.bottomcenter"+playermarker+".style.visibility = 'visible'");
        }
        if (position == "bottomright") {
            eval("document.all.bottomright"+playermarker+".style.visibility = 'visible'");
        }
    }
}
//--> end hide JavaScript
</script>

</head>

<body onload="init()">
```

```
<div id="titleimage"
  style="position: absolute;
    left: -100;
    top: -100;
    z-index: 5">

</div>

<!-- begin game board divisions -->

<div id="topleft"
  style="position: absolute;
    top: 100;
    left: 100;
    z-index: 0;
    visibility: visible"
  onclick="position = 'topleft'; update();">

</div>

<div id="topcenter"
  style="position: absolute;
    top: 100;
    left: 177;
    z-index: 0;
    visibility: visible"
  onclick="position = 'topcenter'; update();">

</div>

<div id="topright"
  style="position: absolute;
    top: 100;
    left: 254;
    z-index: 0"
  onclick="position = 'topright'; update()">

</div>

<div id="centerleft"
  style="position: absolute;
    top: 177;
    left: 100;
    z-index: 0"
  onclick="position = 'centerleft'; update()">

</div>
```

```

<div id="centercenter"
  style="position: absolute;
    top: 177;
    left: 177;
    z-index: 0"
  onclick="position = 'centercenter'; update()">

</div>

<div id="centerright"
  style="position: absolute;
    top: 177;
    left: 254;
    z-index: 0"
  onclick="position = 'centerright'; update()">

</div>

<div id="bottomleft"
  style="position: absolute;
    top: 254;
    left: 100;
    z-index: 0"
  onclick="position = 'bottomleft'; update()">

</div>

<div id="bottomcenter"
  style="position: absolute;
    top: 254;
    left: 177;
    z-index: 0"
  onclick="position = 'bottomcenter'; update()">

</div>

<div id="bottomright"
  style="position: absolute;
    top: 254;
    left: 254;
    z-index: 0"
  onclick="position = 'bottomright'; update()">

</div>

<!-- end game board divisions -->

```

```
<!-- begin hidden o markers -->

<div id="toplefto"
  style="position: absolute;
    top: 100;
    left: 100;
    z-index: 3;
    visibility: hidden">

</div>

<div id="topcentero"
  style="position: absolute;
    top: 100;
    left: 177;
    z-index: 3;
    visibility: hidden">

</div>

<div id="toprighto"
  style="position: absolute;
    top: 100;
    left: 254;
    z-index: 3;
    visibility: hidden">

</div>

<div id="centerlefto"
  style="position: absolute;
    top: 177;
    left: 100;
    z-index: 3;
    visibility: hidden">

</div>

<div id="centercentero"
  style="position: absolute;
    top: 177;
    left: 177;
    z-index: 3;
    visibility: hidden">

</div>
```



```

<div id="centerrighto"
  style="position: absolute;
    top: 177;
    left: 254;
    z-index: 3;
    visibility: hidden">

</div>

<div id="bottomlefto"
  style="position: absolute;
    top: 254;
    left: 100;
    z-index: 3;
    visibility: hidden">

</div>

<div id="bottomcentero"
  style="position: absolute;
    top: 254;
    left: 177;
    z-index: 3;
    visibility: hidden">

</div>

<div id="bottomrighto"
  style="position: absolute;
    top: 254;
    left: 254;
    z-index: 3;
    visibility: hidden">

</div>

<!-- end hidden o markers -->

<!-- begin hidden x markers -->

<div id="topleftx"
  style="position: absolute;
    top: 100;
    left: 100;
    z-index: 2;
    visibility: hidden">

```

```

</div>
```

```
<div id="topcenterx"
  style="position: absolute;
    top: 100;
    left: 177;
    z-index: 2;
    visibility: hidden">

</div>
```

```
<div id="toprightx"
  style="position: absolute;
    top: 100;
    left: 254;
    z-index: 2;
    visibility: hidden">

</div>
```

```
<div id="centerleftx"
  style="position: absolute;
    top: 177;
    left: 100;
    z-index: 2;
    visibility: hidden">

</div>
```

```
<div id="centercenterx"
  style="position: absolute;
    top: 177;
    left: 177;
    z-index: 2;
    visibility: hidden">

</div>
```

```
<div id="centerrightx"
  style="position: absolute;
    top: 177;
    left: 254;
    z-index: 2;
    visibility: hidden">

```

```

</div>

<div id="bottomleftx"
  style="position: absolute;
    top: 254;
    left: 100;
    z-index: 2;
    visibility: hidden">

</div>

<div id="bottomcenterx"
  style="position: absolute;
    top: 254;
    left: 177;
    z-index: 2;
    visibility: hidden">

</div>

<div id="bottomrightx"
  style="position: absolute;
    top: 254;
    left: 254;
    z-index: 2;
    visibility: hidden">

</div>

<!-- end hidden x markers -->

<div id="instructions"
  style="position: absolute;
    top: 95;
    left: 350">
<h3>Instructions:</h3>
<p>To play, click on a square that doesn't contain an X or an O. The first player
uses Xs to mark their moves and the second uses Os. The goal is to get three of
your markers in a row - horizontally, vertically, or diagonally.</p>

<form name="clearboard">
<input type="submit" value="Clear Board" onclick="window.location.reload()" />
</form>

</div>
</body>
</html>

```

Das Ergebnis dieses Codes können Sie auf den Bildern 15.15 und 15.16 sehen (endlich!). Obwohl es beim Kodieren der Seite erhebliche Unterschiede gibt, abhängig davon, ob man mit dem Internet Explorer oder dem Netscape Navigator arbeitet, sehen die Seiten nicht nur gleich aus, sondern haben in beiden Browsern auch dasselbe Verhalten.

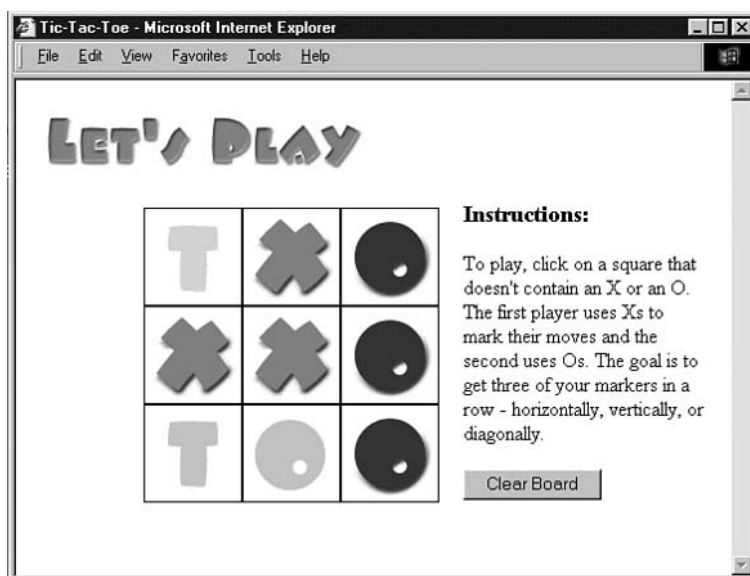


Abbildung 15.15:
Das Resultat im
Microsoft Internet
Explorer 5

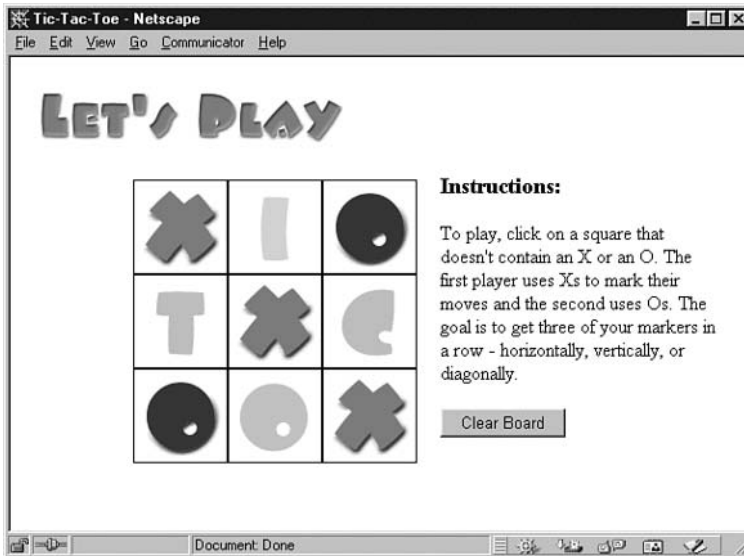


Abbildung 15.16:
Dieselbe Seite im
Netscape Naviga-
tor 4.61

Fahren wir mit Ihrer DHTML-Ausbildung fort

Ich empfehle Ihnen, die DHTML gewidmeten Web-Sites von Microsoft und Netscape zu besuchen. Beide Sites bieten interessante Perspektiven.

Die Microsoft-Web-Site (msdn.microsoft.com/workshop/author/default.asp) hat eine Fülle von Informationen für den Gebrauchs von DHTML – Fähigkeiten.

Netscape stellt sich als führend in Browser-übergreifender Kompatibilität heraus und verfügt über ausführliche Artikel und technische Anmerkungen unter devedge.net-scape.com.

Weitere Sites, die nützliche Informationen liefern sind:

- ▶ WebReference Dynamic HTML Lab (www.webreference.com/dhtml/)
Enthält Beispiele und Codes, von Drag-and-Drop mit DHTML bis zu erweiterbaren Menüs und Outlines.

- ▶ Macromedia es Dynamic HTML Zone (www.dhtmlzone.com) bietet DHTML Artikel, Quellcodes und Tutorials zu vielen DHTML Themen.
- ▶ Webmonkey (www.hotwired.com/webmonkey/dynamic_html/) hat eine ausführliche Sektion dem DHTML gewidmet.
- ▶ Brainjar (www.brainjar.com) – verfügt über eine JavaScript-Funktions-Bibliothek, die hauptsächlich Informationen über das Integrieren von DHTML in Webseiten liefert.
- ▶ The Dynamic Duo: Cross Browser Dynamic HTML (www.dansteinman.com/dynduo/index.html) – liefert Projekte, Tutorials und Sample Codes.

Zusammenfassung

Heute haben Sie gelernt, dass DHTML aus drei Techniken besteht: HTML, Style Sheets und Scripting. Jede davon ist für DHTML wichtig und spielt eine besondere Rolle. HTML liefert die Grundlage für die Webseite. Style Sheets ermöglichen Ihnen die Elemente so zu formatieren und zu positionieren, wie es Ihnen gefällt. Scripting macht DHTML erst »Dynamisch«.

Leider tauchen bei DHTML viele Probleme auf. Die Inkonsequenz der Browser machen das Kodieren der Browser-übergreifenden DHTML zu einer Herausforderung.

Dieses Kapitel hat Ihnen dennoch nur einen Überblick von DHTML und den verschiedenen Techniken vermitteln können. Wie schon gesagt, eine intensivere Behandlung der Thematik würde den Rahmen dieses Buches sprengen.

Workshop

Sie haben heute gelernt, wie Sie die verschiedenen Techniken – bekannt als Dynamic HTML – anwenden müssen, und dass Ihnen diese Techniken bei der Gestaltung Ihrer Webseite und beim Erstellen von Seiten, die auf Anwenderereignisse reagieren, helfen. Unser Workshop heute befasst sich mit Fragen über die wichtigsten Punkte, die in diesem Kapitel behandelt wurden.

Fragen und Antworten

F Löhnen sich der Zeitaufwand und die Bemühungen bei DHTML wirklich?

A Das ist schwierig zu beantworten und die Versuchung ist groß einfach »nein« zu sagen. Aber vielleicht brauchen (oder wollen) Sie Elemente erstellen, die nur mit DHTML erzeugt werden können. Wenn Sie dieser Kategorie angehören, dann lohnen sich natürlich all die Bemühungen und der Zeitaufwand. Wenn Sie jetzt aber keine ultimative DHTML-Webseite erstellen wollen, und diese auch nicht veröffentlichen möchten, dann sollten Sie weiterhin mit den Ihnen bekannten Grundlagen arbeiten und sich stufenweise Ihren Weg durch DHTML bahnen.

F Wir wurden ja wirklich mit Informationen bombardiert. Wie kann ich das nur alles lernen?

A Fangen Sie zunächst mit HTML an und werden Sie ein HTML-Experte. Danach können Sie auf zweierlei Art und Weise vorgehen: alles »meistern«, oder nur die Dinge lernen, die Sie auch wirklich brauchen. Alles meistern zu wollen, wird Sie von HTML zu CSS führen, und von dort zu JavaScript. Sie können nur Fortschritte machen, wenn Sie kompetent mit den Ihnen zur Verfügung stehenden Technologien umgehen können. Wenn dies erreicht ist, untersuchen Sie, wie diese drei Techniken miteinander in Verbindung stehen. Erst dann sollten Sie sich an DHTML heranwagen. Wenn Sie nur das lernen, was Sie auch wirklich benötigen, kommen Sie natürlich schneller voran. Suchen Sie sich eine spezifische Technik aus, die Sie verwenden wollen, wie etwa dynamische Veränderung der Sichtbarkeit eines Objekts, und studieren Sie dann genau, wie man das macht. Danach können Sie sich die nächste, für Sie interessante Technik vornehmen. Mit dieser Verfahrensweise werden Sie alle Technologien Stück für Stück kennen lernen.

F Wo ist DHTML angesiedelt?

A Das W3C arbeitet an einem Standard DOM, dem DOM Level 2. Der Microsoft Internet Explorer 5 setzt viele dieser Ideen um, und die nächste Version des Netscape Navigators (Version 5) beabsichtigt dies ebenfalls zu tun. Das nächste offizielle DOM verspricht also etwas anders als das Aktuelle zu sein.

F Was hat es mit dem Netscape <layer>-Element und anderen Browser-spezifischen DHTML-Features auf sich?

A Wenn Sie mit Ihren Webseiten das größtmögliche Publikum erreichen wollen, sollten Sie sich auf eine relativ universelle Gestaltung Ihrer Webseiten konzentrieren. Natürlich sind die DHTML-Features sehr verlockend und mit einigen von ihnen erzielt man auch tolle Ergebnisse. Es gibt aber nichts frustrie-

renderes als wochenlang eine DHTML-Webseite zu erstellen, um danach feststellen zu müssen, dass die meisten Besucher sie überhaupt nicht lesen können.

Quiz

1. Mit welchen drei Techniken arbeitet DHTML?
2. Was ist ein Document Object Model?
3. Kann man VBScript oder eine andere Scriptsprache zur Erstellung von DHTML verwenden?
4. Was ist das wichtigste Element der Browser-übergreifenden DHTML?

Antworten

1. HTML, Style-Sheets und Scripting.
2. DOM ist die »Sprache«, die Sie verwenden, wenn Sie sich per Scripting auf die Objekte auf einer Webseite beziehen. Vergessen Sie nicht, dass der Microsoft Internet Explorer und der Netscape Navigator unterschiedliche DOM-Implementierungen haben. Deshalb benötigen Sie unterschiedliche Anweisungen, um sich auf dasselbe Objekt beziehen können.
3. Ja, allerdings sollten Sie wissen, dass VBScript ohne ein spezielles Plugin vom Netscape Navigator nicht unterstützt wird. Andere Scriptsprachen werden sogar noch weniger unterstützt. JavaScript ist also noch immer am besten.
4. Ohne jeden Zweifel, die Sniffer-Funktion. Mit dem Sniffer wird jeder Browser identifiziert, der Ihre Seite herunterlädt. Somit können Sie alternative Codes für die Browser erstellen, die Ihre Seite unterstützen sollen.

Übungen

1. Verbessern Sie Ihr Tic-Tac-Toe Spiel und geben Sie einen Code ein, der einen Spieler gegen den Computer spielen lässt.
2. Laden Sie eine oder mehrere Browser-übergreifende DHTML-APIs und integrieren Sie sie im Tic-Tac-Toe Spiel. Erstellen Sie dabei verschiedene Effekte oder benutzen Sie die APIs, um den existierenden Code zu ersetzen.



Schreiben und Gestalten von Webseiten: Tipps

**Woche
3**

Sie werden in diesem Kapitel nichts über irgendwelche HTML-Tags erfahren oder darüber, wie Sie Dateien aus einem seltsamen Format in ein anderes konvertieren. Sie sind mit dem HTML-Teil der Gestaltung von Webseiten fast fertig; als Nächstes folgt das nicht direkt Greifbare, die Sachen, die Ihre Dokumente von denen eines anderen Web-Autoren unterscheiden, der bloß die Tags kennt und mit Text und Grafik um sich wirft und das dann eine Site nennt.

Gewappnet mit den Informationen aus den letzten fünf Tagen könnten Sie das Buch eigentlich beiseite packen und damit loslegen, fröhlich Webseiten zu erzeugen, wie es Ihr Herz erfreut. Ausgerüstet sowohl mit jenen Informationen wie auch mit dem, was Sie heute lernen, können Sie *bessere* Webseiten erzeugen. Brauchen Sie noch mehr Anreize, um weiterzulesen?

Dieses Kapitel enthält Hinweise zur Erstellung gut geschriebener und gut gestalteter Webseiten und beleuchtet, was Sie dafür tun und was Sie unterlassen sollten.

- ▶ Wie Sie das Wirrwarr klären, wann man HTML-3.2-Standard-Tags, neuere HTML 4.0-Tags, Style Sheets oder HTML-Erweiterungen oder eine Kombination von mehreren einsetzen soll
- ▶ Wie Sie Ihre Web-Dokumente so schreiben, dass sie schnell überflogen und gelesen werden können
- ▶ Was bei Entwurf und Gestaltung Ihrer Webseiten wichtig ist
- ▶ Wann und weshalb Sie Verknüpfungen erzeugen sollten
- ▶ Wie Sie Bilder effektiv einsetzen
- ▶ Was es sonst noch an Leckerbissen und Tipps gibt

Die Anwendung von HTML-Erweiterungen

Früher, bevor alle Browser-Hersteller ihre eigenen neuen HTML-Kennungen herausbrachten, war es einfach, ein Web-Designer zu sein. Die einzigen HTML-Tags, mit denen Sie sich abgeben mussten, waren die von HTML 2.0, und die große Mehrzahl der Browser im Web konnte Ihre Seiten problemlos lesen. Inzwischen ist es viel komplizierter geworden. Denn jetzt haben sie verschiedene Gruppen von Kennungen:

- ▶ HTML-2.0-Kennungen
- ▶ HTML-3.2-Kennungen wie Tabellen, Unterteilungen, Hintergründe und Farben, die von einigen, aber nicht von allen Browsern unterstützt werden.
- ▶ HTML 4.0 und damit verbundene Features, wie zum Beispiel Cascading Style Sheets, Dynamic HTML und Framesets.

- ▶ XHTML 1.0, die vorgeschlagene Erweiterung von HTML 4.0, erweitert die HTML-Tags und erstellt Ihre Webseite so, dass sie zukünftigen Ansprüchen gerecht wird.
- ▶ Plugins und andere eingebettete Objekte, die Dateien und Browser-externe Daten verwenden.
- ▶ Browser-spezifische Kennungen (von Netscape und dem Internet Explorer) die als Teil der offiziellen HTML-Spezifikationen gelten werden, oder auch nicht, und deren Unterstützung von Browser zu Browser unterschiedlich ist.
- ▶ Andere Tags, die für zukünftige HTML-Spezifikationen vorgeschlagen wurden, und nur wenige oder keine Browser unterstützen.

Wenn Sie alles dies ziemlich verwirrend finden, dann stehen Sie nicht allein da. Autoren und Entwickler wie Sie versuchen allesamt, sich in der Unordnung zurechtzufinden und Entscheidungen darüber zu treffen, die auf ihren Vorlieben für das Aussehen ihrer Seiten beruhen.

Mit Cascading Style Sheets und Dynamic HTML verfügen Sie, was Layouts und Inhalte betrifft, in HTML 4.0 und XHTML 1.0 über mehr Flexibilität.

Die HTML-Erweiterungen geben Ihnen zwar mehr Flexibilität beim Layout, beschränken aber das Publikum, das diese Seiten auf eine Weise sehen kann, wie Sie es beabsichtigen.

Eine Strategie über den Einsatz von HTML-Erweiterungen ist eine der bedeutenderen Designentscheidungen, die Sie treffen, wenn Sie mit Ihren Webseiten anfangen. Es kann Ihnen helfen, wenn Sie die Ihnen zur Verfügung stehenden Wahlmöglichkeiten als eine Art kontinuierliche Skala betrachten, die sich vom konservativen bis zum experimentierfreudigen Web-Autor erstreckt (siehe Abbildung 16.1).



Betrachten Sie die Endpunkte der Skala nicht als Wertungen; konservativ ist nicht schlimmer als experimentell oder umgekehrt. Es gibt Vorteile an beiden Polen und beträchtliche Vorteile in der Mitte.

Bevor HTML 4.0 auf den Markt kam, konnte man eine kontinuierliche Entwicklung beobachten. Diese Entwicklung sehen Sie auf der linken Seite des Dreiecks in Bild 16.1.

Die konservativen Web-Entwickler wünschen sich das größtmögliche Publikum für ihre Webseiten. Sie halten sich strikt an die HTML-2.0-Kennungen. Ich behaupte nicht, dass der konservative Web-Developer langweilig ist. Im Gegenteil, mit älteren Tags können Sie tolle Web-Inhalte erstellen. Ein weiterer großer Vorteil ist, dass die Mehrheit aller Browser die Seiten problemlos unterstützt.

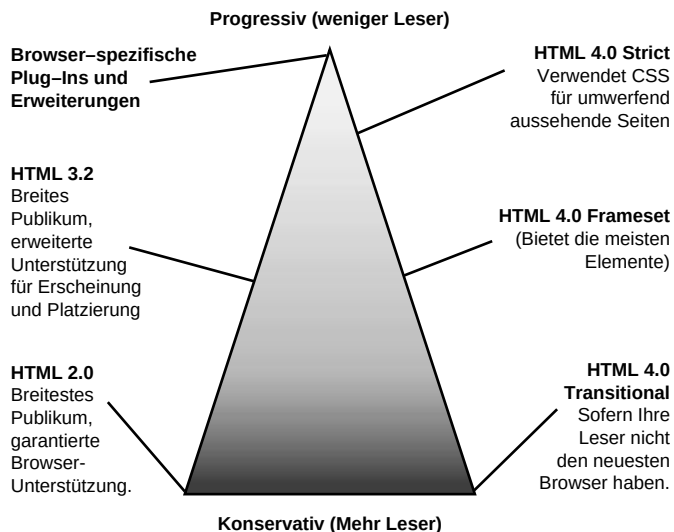


Abbildung 16.1:
Das Kontinuum der
Web-Autoren

Die damals geläufigen HTML 3.2 Tags waren der Standard und kannten die gleiche Entwicklung wie heute die XHTML 1.0 Tags. Obwohl es sich um einen akzeptierten Standard handelte, hinkten die Browser wieder einmal hinterher und die Web-Developer mussten warten, bis diese auf dem neuesten Stand waren. Dennoch wollte man das Risiko eingehen, und hoffte dabei, dass die Mehrheit der Personen, die ihre Site besuchten, wenigstens einen der beiden Hauptbrowser verwenden würden, die, schon zu dieser Zeit, die wichtigsten HTML 3.2 Tags unterstützten.

Experimentierfreudige Web-Entwickler andererseits wünschen sich das Maß an Kontrolle über das Layout, das die fortgeschritteneren Kennungen bieten. Die Seiten des experimentierfreudigen Web-Designers sind auf einen einzigen Browser (oder höchstens auf zwei bis drei) hin konzipiert, in einem einzigen getestet und können sogar eine große Ankündigung auf den Seiten haben, die besagt: »Diese Seiten müssen mit dem Browser X betrachtet werden.« Diese Seiten dann mit einem anderen Browser zu betrachten, kann das Design unlesbar machen oder wenigstens verwirrend – aber es kann auch gutgehen. Um diese unterschiedlichen Szenarios auch zu vereinbaren, enthalten die HTML 4.0- und XHTML 1.0-Definitionen Spezifikationen, die uns im gesamten Buch beschäftigen werden. Die drei neuen Style-Spezifikationen sind auf der rechten Seite des Dreiecks in Bild 16.1 aufgezeigt und deren kurze Beschreibung lautet wie folgt:

- HTML 4.0 oder XHTML 1.0 Transitional ist auf konservative Web-Developer ausgerichtet, die so viele Browser wie möglich unterstützen wollen. Von Browsern wird erwartet, dass sie zumindest HTML 2.0-Tags unterstützen. Da die HTML

3.2-Browser weiter »sehen«, entspricht es der Vernunft, die HTML 3.2 Spezifikation als Basis zu betrachten. Die Mehrheit der Seiten, die Sie in diesem Buch erstellt haben, fallen in diese Kategorie.

- ▶ HTML 4.0 oder XHTML 1.0 Frameset sind die empfohlenen Methoden für Web-Developer, die Ihre Seiten für HTML 3.2-Browser gestalten, aber auch Ihre Web-Sites in Framesets präsentieren wollen. Meiner Meinung nach, handelt es sich hier um die heutige Durchschnittsmethode. Obwohl mit der Frameset-Methode mehr Tags als mit der Transitional-Methode unterstützt werden, gibt es noch viele Browser, die keine Frames unterstützen.
- ▶ HTML 4.0 oder XHTML 1.0 Strict ist für den progressiven Web-Developer geeignet, der seine Seiten nur mit der HTML 4.0- oder XHTML 1.0 Spezifikation gestalten will. Das bedeutet, dass man die so genannten »verworfenen« Tags überhaupt nicht mehr anwenden darf und stattdessen Cascading Style Sheets für die Präsentation des Dokuments verwenden muss.

Obwohl die HTML 4.0- und die XHTML 1.0 Spezifikationen Meilensteine des Web-Designs sind und so ziemlich jeden Web-Developer zufrieden stellen, gibt es da immer noch eine Gruppe, eine kleine Elitegruppe, die mit Features experimentiert, die über die formellen Spezifikationen hinaus gehen. Solange Browser-Firmen fortfahren, neue Features zu implementieren, wird es auch eifrige Entwickler geben, die sie anwenden möchten. Solche Features werden von den neuesten und besten Versionen ihres Lieblings-Browsers unterstützt und verwenden Browser-spezifische Tags für das Design ihrer Seiten.

Die beste Position hinsichtlich der Wahl zwischen interessantem Design und einem breiten Publikum ist ein Mittelweg zwischen den beiden Arten von Web-Entwicklern. Wenn Sie im voraus den Effekt von HTML-Erweiterungen auf Ihre Seiten abschätzen können, sowohl für Browser, die diese Erweiterungen unterstützen, als auch für die, die dies nicht tun, können Sie leichte Modifikationen an Ihrem Design durchführen, mit denen Sie die Vorteile beider Seiten nutzen können. Ihre Seiten sind dann in älteren Browsern und über eine breite Auswahl von Browsern immer noch nützlich und lesbar, können aber die fortgeschritteneren Möglichkeiten der neueren Browser anwenden.

Bisher habe ich in diesem Buch immer wieder erklärt, welche Kennungen Teil von HTML 3.2 oder 4.0 sind und welche in welchen der wichtigsten Browser verfügbar sind. Ich habe auch für jede Kennung die Alternativen angegeben, die Sie einsetzen können, wenn ein Browser diese Kennung nicht versteht. Mit diesen Informationen sollten Sie dazu in der Lage sein, mit jeder Kennung in verschiedenen Browsern zu experimentieren, um herauszufinden, was für eine Auswirkung sie auf Ihr Design hat.

Die wichtigste Strategie, die ich für die Anwendung der HTML-Erweiterungen vorschlagen kann, wenn Sie die Kompatibilität mit anderen Browsern aufrechterhalten wollen, ist das Testen Ihrer Dateien in diesen anderen Browsern. Die meisten Browser können kostenlos oder als Shareware heruntergeladen werden, sodass Sie sie nur zu installieren brauchen. Wenn Sie einen Unix-Zugang haben, ist die Lynx-Software wahrscheinlich installiert, sodass Sie sie ebenfalls verwenden können. Indem Sie Ihre Seiten testen, können Sie eine Vorstellung davon bekommen, wie die verschiedenen Browser die unterschiedlichen Kennungen interpretieren, und schließlich werden Sie ein Gefühl dafür bekommen, welche Erweiterungen die meiste Flexibilität bieten, welche speziellen Kodierungen als Alternativen in älteren Browsern benötigt werden und welche Kennungen unbesorgt ohne Komplikationen bei allen Browsern eingesetzt werden können.

Schreiben für Online-Leser

Das Schreiben für das Web ist nicht anders als das Schreiben in der realen Welt. Obgleich das für das Web Geschriebene nicht auf Papier verkauft wird, so ist es doch »veröffentlicht« und spiegelt Sie und Ihre Arbeitsweise wider. Genaugenommen müssen Sie, weil Sie online schreiben und Ihre Texte daher für Ihre Leser viel vergänglicher sind, den Regeln für gutes Schreiben viel eher Folge leisten, weil Ihre Leser weniger nachsichtig sein werden.

Wegen der Unmengen der im Web verfügbaren Informationen werden Ihre Leser nicht allzuviel Geduld aufbringen, wenn Ihre Seite voller Rechtschreibfehler steckt oder schlecht organisiert ist. Viel eher neigen sie dazu, nach den ersten paar Sätzen aufzugeben und zur Seite eines anderen Anbieters zu wechseln. Und schließlich gibt es einige Millionen Seiten da draußen. Keine Zeit, sie mit schlechten Seiten zu vergeuden.

Dies bedeutet nun nicht, dass Sie losgehen und erst ein professioneller Autor werden müssen, um gute Webseiten erschaffen zu können. Aber hier folgen ein paar Hinweise, um Ihre Webseite leichter lesbar und verständlicher zu machen.

Schreiben Sie kurz und bündig

Solange Sie nicht das große abendländische Web-Epos schreiben, werden Ihre Leser Ihre Seite nicht aufsuchen, um sich liebevoll bei Ihren Wörtern aufzuhalten. Einer der besten Wege, den Text Ihrer Seite wirkungsvoll zu gestalten, ist, so klar und prägnant zu schreiben, wie Sie können, Ihre Gesichtspunkte zu benennen und dann aufzuhören. Das, was Sie sagen wollen, mit zusätzlichen Wörtern auszuwalzen, macht es für Ihre Leser bloß schwer, herauszukriegen, was Sie eigentlich sagen wollen.

Wenn Sie noch kein Exemplar von Strunk und Whites »*The Elements of Style*« besitzen, legen Sie dieses Buch gleich weg, kaufen jenes und lesen es. Und dann lesen Sie es erneut, vergegenwärtigen sich den Inhalt, inhalieren ihn, schlafen mit ihm unter dem Kopfkissen, zeigen es all Ihren Freunden, zitieren es auf Parties, machen es zum Teil Ihres Lebens. Es gibt keine bessere Anleitung für die Kunst, gut und klar zu schreiben, als jenes Buch. (Ein Tipp für die deutschen Leser: Werner Raith, *Gut schreiben, ein Leitfaden*; Campus, Frankfurt/Main, 1988.)

Organisieren Sie Ihre Dokumente leicht überschaubar

Selbst wenn Sie die klarste, knappste und schillerndste Prosa schreiben, die man je im Web gesehen hat, ist die Chance ziemlich hoch, dass Ihre Leser nicht am Anfang Ihrer Webseite beginnen und dann sorgsam jedes Wort bis zum Ende der Seite lesen.

Das »Überfliegen« ist in diesem Zusammenhang ein erster schneller Blick, den Ihre Leser auf die Seite werfen, um einen allgemeinen Eindruck vom Inhalt zu bekommen. Je nachdem, was Ihre Leser von den Dokumenten erwarten, mögen sie jene Teile überfliegen, die ihnen entgegenspringen (Überschriften, Verknüpfungen, andere hervorgehobene Wörter), oder vielleicht lesen sie einige zusammenhängende Absätze und wandern dann weiter. Indem Sie Ihre Dokumente im Hinblick auf ein solches »Überfliegen« schreiben und organisieren, können Sie Ihren Lesern helfen, die Informationen, die sie suchen, so schnell wie möglich zu finden.

Um die Überfliegbarkeit Ihrer Webseiten zu verbessern,

- ▶ verwenden Sie Überschriften, um Themen zusammenzufassen. Auch dieses Buch verwendet Überschriften und Untertitel, sodass Sie es schnell durchblättern und die Teile finden können, die Sie interessieren. Das Gleiche gilt für Webseiten;
- ▶ verwenden Sie Listen. Listen eignen sich wunderbar für das Zusammenstellen verwandter Themen. Jedesmal, wenn Sie sich dabei ertappen, etwas Ähnliches wie »jedes Modul besteht aus vier Teilen« oder »machen Sie die folgenden Schritte, um dies zu tun« zu schreiben, sollte der weitere Inhalt nach diesen Ausdrücken in einer geordneten oder ungeordneten Liste stehen;
- ▶ vergessen Sie keine Verknüpfungs-Menüs. Als eine der Listen-Formen haben Verknüpfungs-Menüs nicht nur all die Vorzüge von Listen, sondern sind auch ausgezeichnet zum Navigieren geeignet;
- ▶ verbergen Sie keine wichtigen Informationen im Text. Wollen Sie etwas Wesentliches ansprechen, so machen Sie es möglichst am Anfang der Seite oder am Beginn eines Absatzes. Lange Absätze sind schwerer zu lesen, und es ist schwieriger, die Informationen daraus zusammenzutragen; je weiter unten innerhalb des Absatzes Sie Ihre Anliegen ansprechen, desto eher werden sie übersehen.

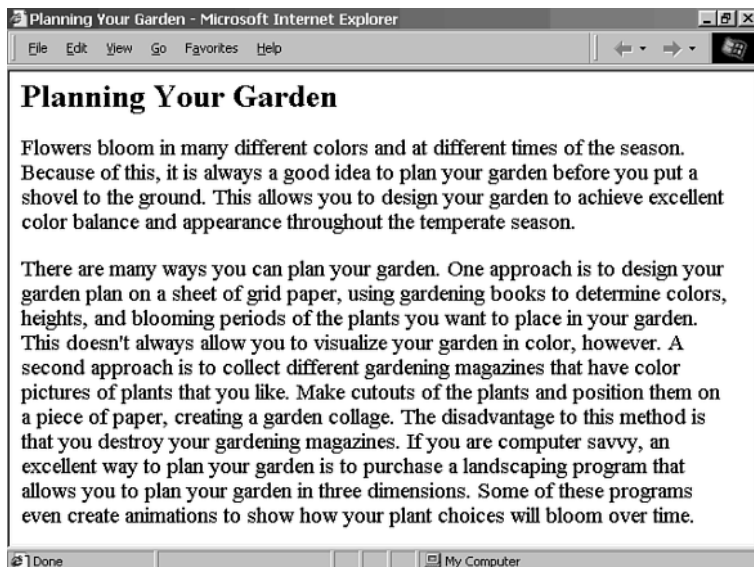


Abbildung 16.2:
Eine Webseite,
die schwer zu
überfliegen ist

Da alle Informationen dieser Seite in Absatzform dargeboten werden, müssen Ihre Leser alle drei Absätze der Reihe nach lesen, um das herauszufinden, was sie wollen und wohin sie als Nächstes gehen möchten.

Wie würden Sie dieses Beispiel verbessern? Versuchen Sie, den Absatz neu zu schreiben, sodass die wesentlichen Gesichtspunkte dem Text besser entnommen werden können.

Bedenken Sie, dass

- ▶ es eigentlich zwei verschiedene Themen in den drei Absätzen gibt,
- ▶ die vier Bestandteile des Getränks eine gute Liste abgeben würden.

Abbildung 16.3 zeigt, wie man diese Seite verbessern könnte.

Lassen Sie jede Seite für sich selbst sprechen

Denken Sie, während Sie schreiben, daran, dass Ihre Leser von irgendwoher in jede Ihrer Seiten einsteigen können. Obgleich Sie Ihre Sachen so strukturiert haben mögen, dass Abschnitt vier folgerichtig auf Abschnitt drei folgt und es keine anderen Verknüpfungen dorthin gibt, könnte jemand, den Sie nicht einmal kennen, eine Verbindung mit der Seite, auf der Abschnitt vier beginnt, erzeugen, und ein Leser könnte sich dann in Abschnitt vier wiederfinden, ohne überhaupt zu wissen, dass es einen Abschnitt drei gibt.

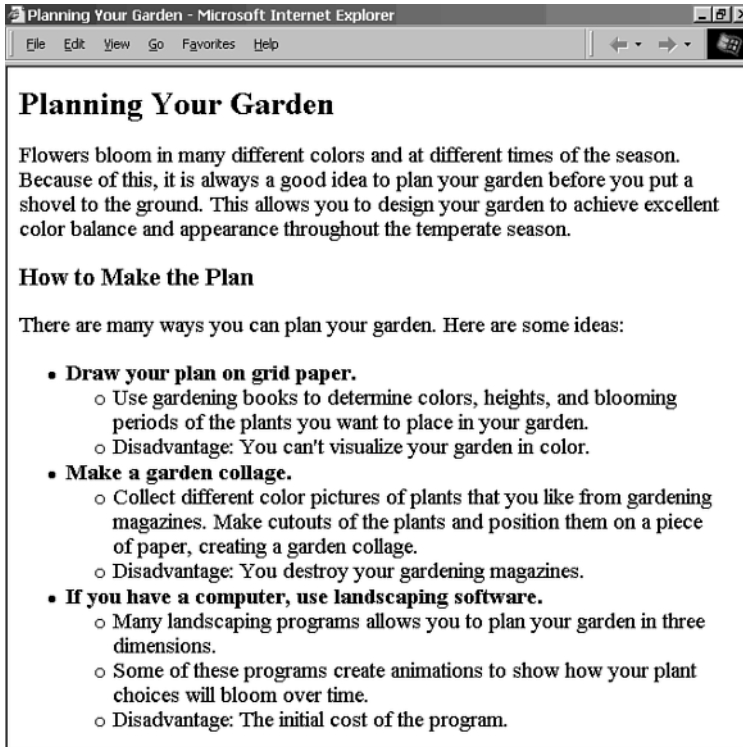


Abbildung 16.3:
Verbesserungsvor-
schlag für die
schwierige Web-
seite

Achten Sie darauf, jede Seite so zu schreiben, dass sie für sich selbst zu stehen vermag. Die folgenden Richtlinien mögen Ihnen dabei helfen:

- ▶ Verwenden Sie beschreibende Titel. Der Titel sollte nicht bloß das direkte Thema der Seite enthalten, sondern auch seinen Bezug zum Rest der Seiten in der Präsentation.
- ▶ Wenn das Dokument von einem anderen davor abhängt, bieten Sie eine Navigationsverbindung zurück zu diesem anderen Dokument an (und vorzugsweise auch eine hoch zur obersten Ebene).
- ▶ Vermeiden Sie einleitende Sätze wie: »Sie können jene Probleme umgehen, indem ...«, »Nachdem Sie das erledigt haben, machen Sie ...« oder »Die Vorzüge dieser Methode sind ...«. Die Informationen, auf die durch »jene«, »das« und »dieser« Bezug genommen wird, stehen woanders, auf einer anderen Seite. Wenn jene Sätze das Erste sind, was Ihre Leser sehen, werden sie dadurch verwirrt.

Seien Sie zurückhaltend mit Hervorhebungen

Benutzen Sie Hervorhebungen in Ihren Texten nur sparsam. Absätze mit einer Menge **fetter** und *kursiver* Schrift oder Wörtern in GROSSBUCHSTABEN sind schwer zu lesen, sowohl wenn Sie eines davon mehrfach in einem Absatz verwenden, als auch dann, wenn Sie größere Teile des Textes hervorheben. Am besten, Sie heben nur kurze Wörter (wie, und, dies, oder, aber) hervor.

Verknüpfungs-Text ist ebenfalls eine Form der Hervorhebung. Benutzen Sie einzelne Wörter oder kurze Ausdrücke als Verknüpfungs-Text. Verwenden Sie keine ganzen Passagen oder Absätze für Verknüpfungen.

Abbildung 16.4 illustriert ein besonders schlechtes Beispiel für zu viele Hervorhebungen, die den Rest des Textes schwer verständlich machen.

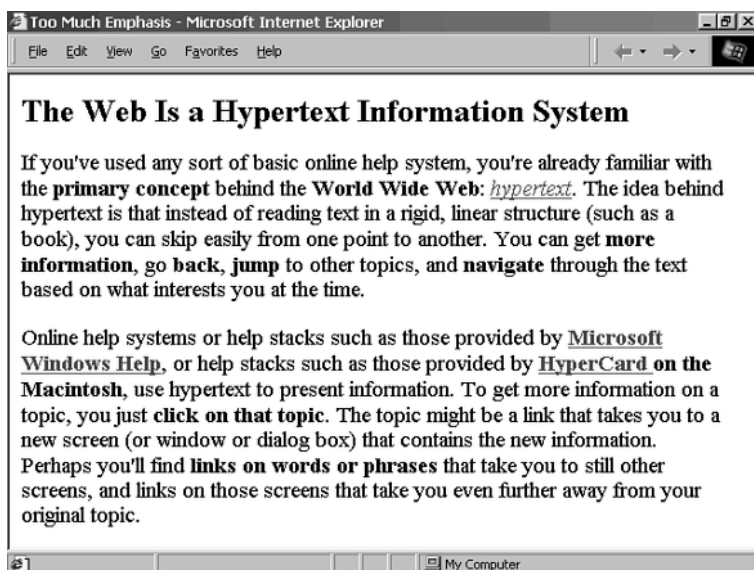


Abbildung 16.4:
Zu viel Hervorhebung

Indem Sie einige der fetten Hervorhebungen entfernen und weniger Text für Ihre Links verwenden, können Sie die Unruhe innerhalb der Seite ganz deutlich verringern. Abbildung 16.5 zeigt das.

Seien Sie besonders zurückhaltend mit Hervorhebungen, die Bewegung oder Veränderungen verwenden, wie zum Beispiel rollende Textfahnen, blinkender Text oder Animation. Wenn die Animation nicht der Schwerpunkt der Seite ist, sollten Sie Bewegung und Sound extrem sparsam einsetzen, um nicht vom Rest der Seite abzulenken.

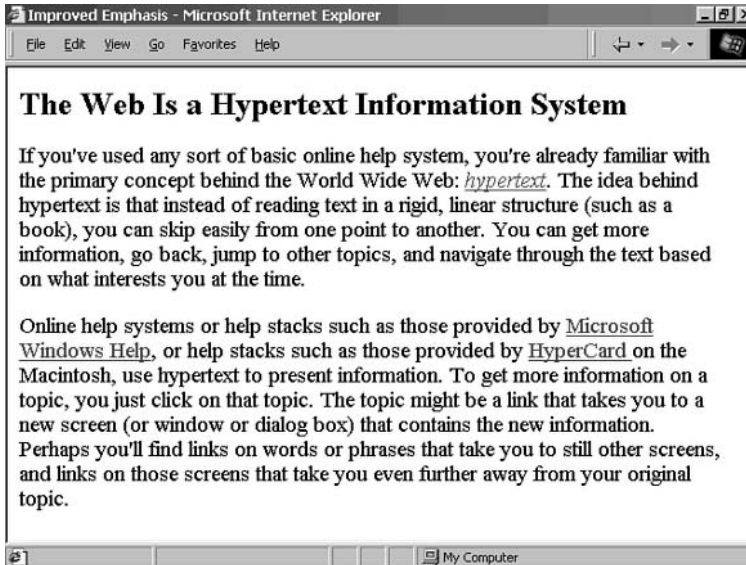


Abbildung 16.5:
Weniger Hervor-
hebung

Vermeiden Sie Browser-spezifische Terminologie

Vermeiden Sie in Ihrem Text, sich auf spezielle Möglichkeiten eines bestimmten Browser zu beziehen.

Benutzen Sie zum Beispiel keine Formulierungen wie:

- ▶ *Hier klicken.* Was ist, wenn Ihr Leser einen Browser ohne Maus benutzt? Ein allgemeinerer Ausdruck wäre: »Wählen Sie diese Verbindung.« (Natürlich sollten Sie in erster Linie das »Hier-Syndrom« vermeiden, wodurch dieses Problem auch einfach umgangen wird.)
- ▶ *Um das Dokument zu speichern, wählen Sie aus dem Datei-Menü den Punkt »Sichern«.* Jeder Browser hat andere Menüs und unterschiedliche Wege, um die gleiche Aktion auszuführen. Soweit irgend möglich, beziehen Sie sich in Ihren Webseiten nicht auf Browser-Befehle.
- ▶ *Benutzen Sie den Zurück-Button, um zur vorherigen Seite zurückzukehren.* Wie schon beim letzten Punkt der Hinweis, dass jeder Browser andere Methoden hat, um »zurück«-zugehen. Wenn Sie wollen, dass Ihre Leser die Möglichkeit haben, zu einer vorhergehenden Seite zurückzugehen oder zu einer anderen bestimmten Seite, dann bieten Sie ihnen dafür ausdrücklich Verknüpfungen an.

Rechtschreibung prüfen und Korrekturlesen

Das Überprüfen der Rechtschreibung und Korrekturlesen mag als überflüssige Empfehlung erscheinen, doch bei der Anzahl von Dokumenten, die ich im Web gesehen habe und bei denen dies offensichtlich nicht beherzigt wurde, scheint der Rat angemessen.

Das Entwerfen einer Reihe von Webseiten und deren Veröffentlichung im Web ist wie das Publizieren eines Buchs, die Herstellung einer Zeitschrift oder die Veröffentlichung eines Produkts. Es ist natürlich spürbar einfacher, Webseiten zu publizieren, als es bei Büchern, Zeitschriften oder anderen Produkten der Fall ist, doch dass es einfach ist, bedeutet nicht, dass man dabei nachlässig sein darf.

Womöglich Tausende von Leuten könnten die Inhalte, die Sie anbieten, lesen und durchgehen. Schreibfehler und schlechte Grammatik werfen ein schlechtes Licht auf Ihre Arbeit, auf Sie und auf die Inhalte, die Sie beschreiben. Ein schlechter Schreibstil mag Ihre Leser schon so weit abschrecken, dass sie sich kaum weiter in Ihre Seiten vertiefen werden, selbst wenn das Thema, über das Sie schreiben, faszinierend ist.

Lesen Sie jede Seite Korrektur, und prüfen Sie die Rechtschreibung. Wenn möglich, lassen Sie jemand anderes lesen – andere Leute entdecken oft Fehler, die Sie als Autor übersehen. Selbst einfaches Redigieren kann viele Dokumente stark verbessern, ihre Lesbarkeit erhöhen und das Navigieren vereinfachen.

Gestaltung und Seitenlayout

Mit dem Einführen von Technologien, wie etwa Style Sheets und Dynamischem HTML, wurde Personen, die eigentlich gar keine Ahnung von Design hatten, die Möglichkeit gegeben, eine Site zu erstellen. In den meisten Fällen sieht diese aber ganz einfach schrecklich aus.

Die wohl wichtigste Regel, die man stets befolgen sollte, wenn man Webseiten gestaltet, ist diese: Halten Sie die Gestaltung so einfach wie möglich. Reduzieren Sie die Anzahl der Elemente (Abbildungen, Überschriften, Trennlinien), und achten Sie darauf, dass das Auge zunächst vom wichtigsten Teil der Seite angezogen wird.

Behalten Sie diese Grundregel im Hinterkopf, während Sie die nächsten Abschnitte lesen, die noch ein paar weitere Vorschläge für den grundlegenden Entwurf und die Gestaltung von Webseiten anbieten.

Benutzen Sie Überschriften als Überschriften

Überschriften werden in grafikfähigen Browsern oft in einer größeren oder fetteren Schrift dargestellt. Daher ist es häufig verlockend, eine Überschriften-Kennung zu verwenden, um irgendwelche Warnungen, Hinweise oder andere Hervorhebungen im regulären Text darzustellen, siehe Abbildung 16.6.

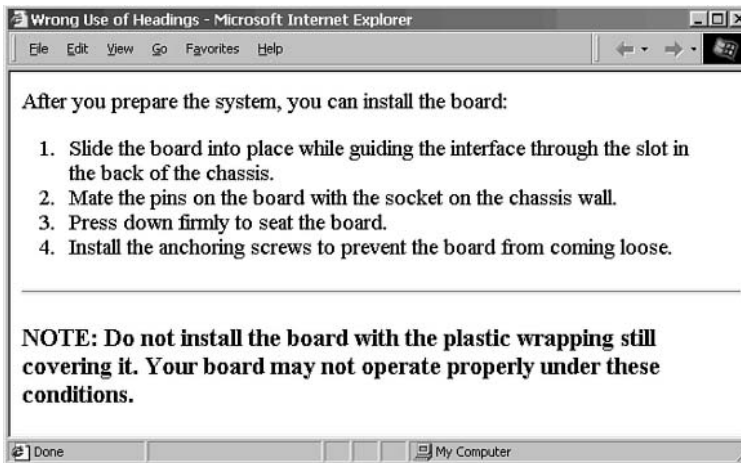


Abbildung 16.6:
Der falsche Weg,
Überschriften zu
verwenden

Überschriften wirken am besten, wenn sie auch als Überschriften benutzt werden, weil sie vom Text abgesetzt sind und den Anfang eines neuen Themas signalisieren. Wenn Sie tatsächlich einen bestimmten Textabschnitt hervorheben wollen, erwägen Sie stattdessen die Benutzung von Trennlinien und einem kleinen Symbol oder einer anderen Methode der Hervorhebung. Abbildung 16.7 zeigt ein Beispiel mit demselben Text wie in Abbildung 16.6, aber einer anderen Art visueller Hervorhebung.

Fassen Sie zusammengehörige Informationen optisch zusammen

Verwandte Informationen zusammenzugruppieren, ist eine Aufgabe sowohl beim Entwurf als auch beim Schreiben. Indem Sie solche Informationen unter Überschriften zusammenfassen, wie ich in den Hinweisen zum Schreiben vorschlug, erhöhen Sie die Lesbarkeit jener Informationen. Jeden dieser Abschnitte optisch vom anderen abzusetzen, macht jeden Abschnitt eigenständig und hebt die Verwandtschaft der Informationen hervor.

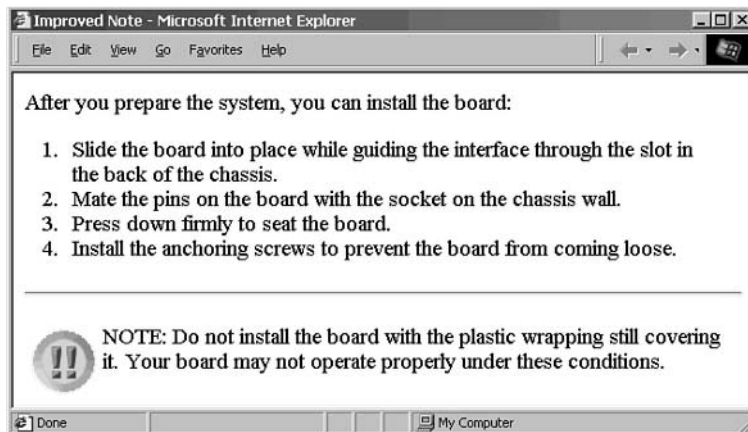


Abbildung 16.7:
Eine Alternative
zu falschen Über-
schriften

Enthält eine Webseite verschiedene Abschnitte mit Informationen, überlegen Sie sich einen Weg, diese Abschnitte optisch voneinander abzutrennen, beispielsweise mit einer Überschrift oder einer Trennlinie (<hr>), vgl. Abbildung 16.8.

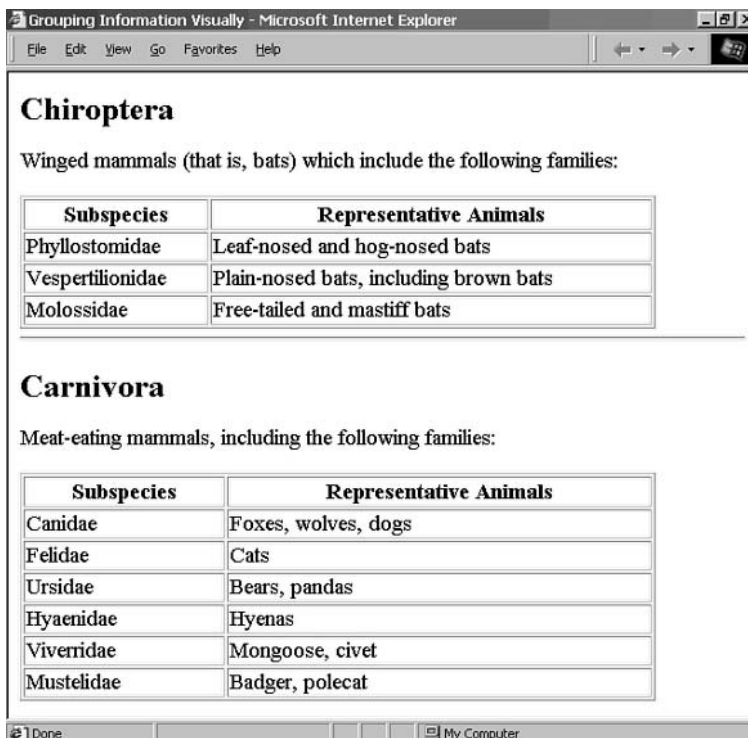


Abbildung 16.8:
Abschnitte
optisch untertei-
len

Verwenden Sie ein durchgehendes Layout

Wenn Sie ein Buch oder eine Zeitschrift lesen, haben jede Seite und jeder Abschnitt üblicherweise dasselbe Layout. Die Seitennummern sind da, wo Sie sie erwarten, und das erste Wort auf jeder Seite beginnt an derselben Stelle.

Dieselbe Art gleichmäßigen Layouts eignet sich genauso gut für Webseiten. Ein gleichmäßiges Erscheinungsbild aller Seiten in Ihrer Web-Präsentation ist für Ihre Leser angenehm. Nach zwei oder drei Seiten werden sie wissen, was die einzelnen Elemente auf jeder Seite bedeuten und wie sie zu finden sind. Bei einer gleichmäßigen Gestaltung können Ihre Leser die Informationen, die sie möchten, finden und sich durch Ihre Seiten bewegen, ohne auf jeder Seite innezuhalten, um herauszukriegen, wie die Dinge liegen.

Eine gleichmäßige Gestaltung kann bestehen aus:

- ▶ Wiederkehrenden Seitenelementen. Wenn Sie auf einer Seite Überschriften zweiten Grades (<h2>) verwenden, um die Hauptthemen zu bezeichnen, dann benutzen Sie Überschriften zweiten Grades auf allen Ihren Seiten. Wenn Sie eine Überschrift und eine Trennlinie am Beginn Ihrer Seite haben, so verwenden Sie auf jeder Seite die gleiche Gestaltung.
- ▶ Wiederkehrenden Formen der Navigation. Platzieren Sie Ihre Navigations-Menüs an derselben Stelle jeder Seite (üblicherweise am Anfang oder Ende jeder Seite), und verwenden Sie dieselbe Anzahl von ihnen. Falls Sie sich anschicken, Navigations-Symbole zu verwenden, achten Sie darauf, dieselben Symbole in der gleichen Reihenfolge auf jeder Seite zu benutzen.
- ▶ Das Verwenden von externen Style Sheets. Wenn Sie bei reinem HTML 4.0 bleiben wollen, können Sie ein Master Style Sheet erstellen, das Hintergrund-Eigenschaften, Text- und Linkfarben, Schriftart und -größe, Seitenränder und noch vieles mehr definiert. Das Design Ihrer Seite wird in Ihrer Site konsistent sein.

Verknüpfungen anwenden

Ohne Verknüpfungen wären Webseiten ziemlich eintönig, und es wäre fast unmöglich, irgendetwas Interessantes auf dem Web zu finden. Die Qualität Ihrer Verknüpfungen kann in vieler Weise genauso wichtig sein wie das Schreiben und Design Ihrer eigentlichen Seiten. Hier ist etwas gutgemeinter Rat über das Erstellen und die Anwendung von Verknüpfungen.

Verwenden Sie Verknüpfungs-Menüs mit beschreibendem Text

Wie ich bereits des Öfteren erwähnt habe, sind Verknüpfungs-Menüs eine wunderbare Art, wie Sie den Inhalt und die Verknüpfungen auf einer Seite organisieren können. Indem Sie Ihre Verknüpfungen in Form von Listen oder anderen menüartigen Strukturen anordnen, können Ihre Leser die Optionen für die Seite schnell und einfach überfliegen.

Oft reicht es jedoch nicht, einfach nur Ihre Verknüpfungen in Menüs anzuordnen. Vergewissern Sie sich, dass die Beschreibungen der Verknüpfungen in Ihren Menüs nicht zu knapp sind. Es ist verlockend, Menüs, bestehend aus Datei-Namen oder anderen wenig eingängigen Verknüpfungen, zu erstellen, wie in Abbildung 16.9 gezeigt wird:

Nun ja, das ist ein Verknüpfungs-Menü, und die Verbindungen beschreiben das eigentliche Dokument, auf das sie verweisen, doch sie sind nicht aussagekräftig genug. Woher sollen Ihre Leser wissen, was auf der anderen Seite der Verbindung ist, und wie sollen sie anhand der begrenzten Information, die Sie ihnen gegeben haben, entscheiden, ob sie daran interessiert sind oder nicht? Von den drei Verbindungen hier gibt nur die letzte ([pesto.recipe.txt](#)) einen Hinweis darauf, was die Leser sehen werden, wenn sie zur Datei springen.



Abbildung 16.9:
Einfaches Verknüpfungs-Menü

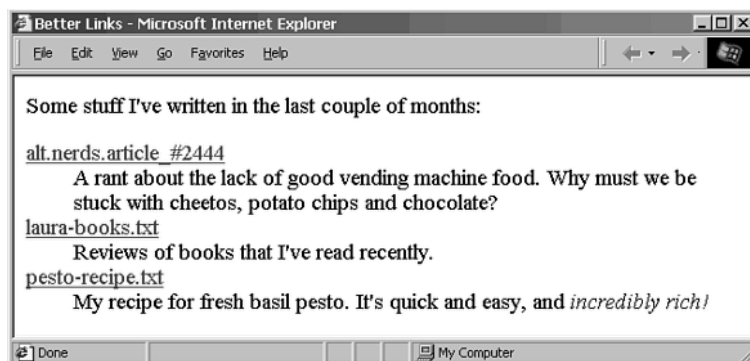


Abbildung 16.10:
Ein besseres Verknüpfungs-Menü

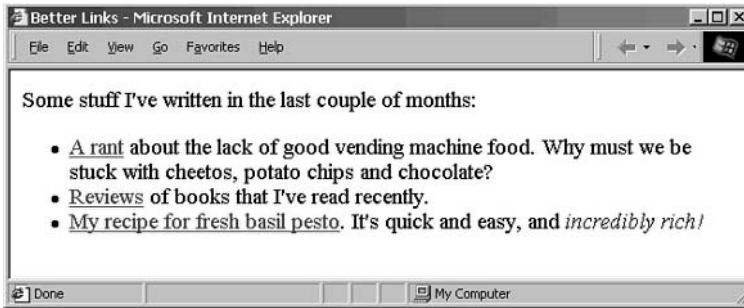


Abbildung 16.11:
Ein weiteres bes-
seres Verknüp-
fungs-Menü

Besser wäre es, etwas zusätzlichen Text aufzunehmen, der den Inhalt der Dateien beschreibt (vgl. Abbildung 16.10), oder die Datei-Namen ganz zu vermeiden (wen kümmert es ?) und in dem Menü einfach den Inhalt der Dateien zu beschreiben, in dem der entsprechende Text hervorgehoben wird (vgl. Abbildung 16.11).

Verwenden Sie Verknüpfungen im Text

Der beste Weg, Verbindungen innerhalb des Textes anzubieten, besteht darin, den Text zunächst ohne die Verknüpfungen zu schreiben und dann später die passenden Wörter, die auf die Verbindung hinweisen, hervorzuheben. Achten Sie darauf, dass Sie den Fluss des Dokuments nicht unterbrechen, wenn Sie eine Verknüpfung einfügen. Der Gedanke hinter dem Integrieren von Verbindungen in den Text ist der, dass der Text für sich allein stehen kann, sodass die Verknüpfungen zusätzliche oder weiterreichende Informationen bieten und Ihre Leser nach eigener Laune entscheiden können, ob Sie sie ignorieren oder ihnen folgen.

Hier ist ein weiteres Beispiel für die Benutzung von Verknüpfungen innerhalb des Textes (dargestellt in Abbildung 16.12), jedoch eines, in dem der Text selbst nicht besonders wichtig ist, er dient lediglich dazu, die Verknüpfungen zusammenzufassen. Wenn Sie Text auf diese Weise benutzen, nur um Verknüpfungen zusammenzustellen, sollten Sie besser ein Verknüpfungs-Menü anstelle eines Absatzes verwenden. Es wird für Ihre Leser leichter, die Information zu finden, die sie wollen. Anstatt den ganzen Absatz lesen zu müssen, können sie die Seite danach durchsehen, was sie interessiert.

In Bild 16.13 sehen Sie eine Methode, wie man das vorherige Beispiel umstrukturieren kann. Die wichtigsten Elemente dieser Seite sind der Name der Konferenz, die Ereignisse und die Daten, wann sie stattfinden. Sie können die Seite also so umstrukturieren, dass diese Informationen aus dem Text hervorstechen. Wie Sie in Bild 16.13 sehen können, stechen die wichtigen Informationen aus dem Text heraus, wenn diese in einer vorformatierten Texttabelle präsentiert werden.

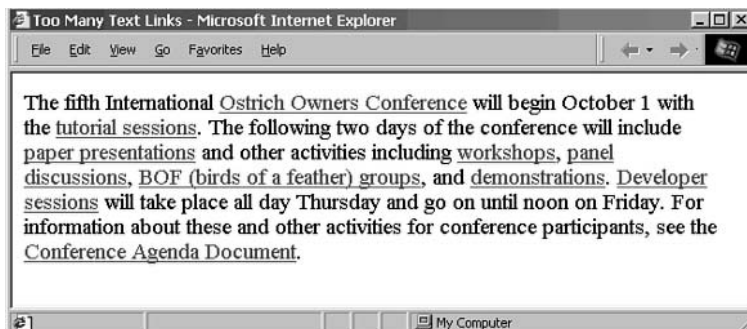


Abbildung 16.12:
Verknüpfungen
im Text, die nicht
gut aussehen

Der wahrscheinlich einfachste Weg herauszufinden, ob Sie die Links richtig im Text erstellen ist, die formatierte Webseite einfach auszudrucken. Liest sich die Seite komisch auf dem Papier, wird sie auch Online komisch zu lesen sein. Die Revisionen müssen nicht immer genauso unterschiedlich wie in diesem Beispiel sein. Manchmal hilft auch oft nur eine einfache Umformulierung der Sätze, um den Text Ihrer Seite leserlicher zu machen (sowohl Online, als auch auf Papier).



Abbildung 16.13:
Restrukturierung der Ver-
knüpfung im Text

Vermeiden Sie das »Hier«-Syndrom

Ein verbreiteter Fehler, den viele Web-Autoren beim Einfügen von Verknüpfungen in den Text machen, ist das so genannte »Hier«-Syndrom. Das »Hier«-Syndrom bezeichnet die Neigung, Verknüpfungen mit einem einzeln hervorgehobenen Wort (»hier«) zu erzeugen und die Verbindung selbst irgendwo anders im Text zu erklären. Hier ein paar Beispiele (wobei der unterstrichene Text die Verknüpfungen angibt):

Information about ostrich socialization is contained here.
Select this link for a tutorial on the internal combustion engine.

Da Verknüpfungen ja auf der Webseite hervorgehoben werden, springen diese Verbindungen mehr als der umgebende Text ins Auge. Ihre Leser werden die Verknüpfung zuerst wahrnehmen, bevor sie den Text lesen. Probieren Sie es aus. Abbildung 16.14 zeigt einige besonders unglücklich gewählte Verknüpfungen. Schließen Sie die Augen, und öffnen Sie sie dann schnell, suchen Sie sich ein beliebiges »Hier« heraus, und schauen Sie, wie lange Sie brauchen, um herauszubekommen, für was das »Hier« überhaupt steht.

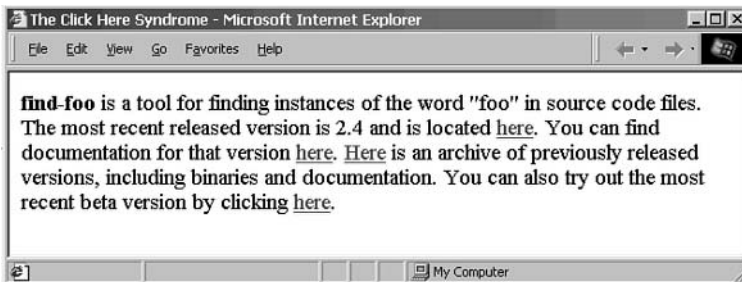


Abbildung 16.14:
Das Hier-Syndrom

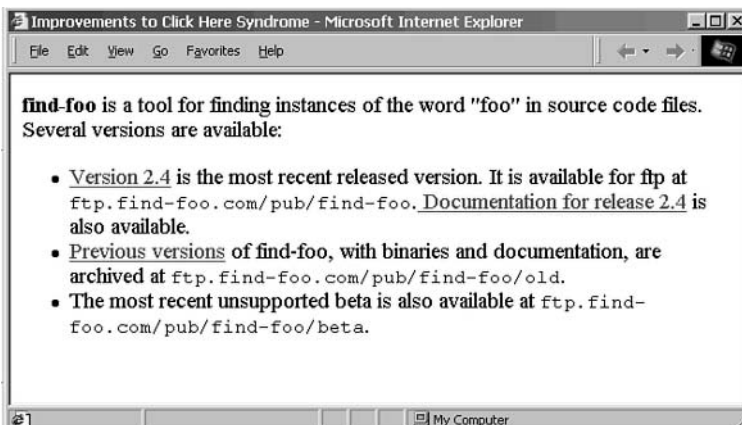


Abbildung 16.15:
Die gleiche Seite,
anders gestaltet

Nun versuchen Sie das Gleiche mit einem gut organisierten Menü, das dieselben Informationen zur Verfügung stellt; siehe Abbildung 16.15.

Da »Hier« nichts darüber aussagt, wozu die Verbindung überhaupt dient, müssen Ihre armen Leser den Text vor und hinter der Verknüpfung selbst daraufhin durchsuchen, was das »Hier« überhaupt bedeuten soll. In Absätzen, die viele »Hier« oder andere undurchschaubare Verknüpfungen enthalten, wird es schwierig, die Verknüpfungen selbst mit dem in Verbindung zu bringen, wofür sie stehen, sodass Sie Ihre Leser zwingen, hart zu arbeiten, um die Dinge zu verstehen.

Anstelle einer Verknüpfung wie dieser:

Information about ostrich socialization is contained [here](#).

lautet eine bessere Formulierung ungefähr so:

The Palo Alto Zoo has lots of information about ostrich socialization.

oder einfach:

The Palo Alto Zoo has lots of information about [ostrich socialization](#).

Verknüpfen oder nicht?

Genauso wie bei Abbildungen sollten Sie sich auch jedesmal, wenn Sie eine Verknüpfung erzeugen wollen, fragen, weshalb Sie zwei Dokumente oder Abschnitte miteinander verbinden. Ist die Verknüpfung für Ihre Leser nützlich? Wird sie ihnen zusätzliche Informationen bieten oder sie näher an ihr Ziel führen? Ist die Verbindung in irgendeiner Weise wichtig für den aktuellen Inhalt?

Jede Verknüpfung sollte einem Zweck dienen. Verknüpfen Sie nur aus einem wichtigen Grund. Bloß weil Sie das Wort »Kaffee« tief in einer Seite über ein anderes Thema erwähnen, müssen Sie das Wort noch nicht mit der Kaffee-Homepage verbinden. Es mag schlaun erscheinen, doch wenn eine Verknüpfung für den aktuellen Inhalt unbedeutend ist, verwirrt sie Ihre Leser nur.

Dieser Abschnitt beschreibt einige Kategorien von Verbindungen, die in Web-Dokumenten nützlich sind. Wenn Ihre Verknüpfungen in keine dieser Kategorien passen, sollten Sie sich fragen, weshalb Sie sie in Ihr Dokument aufnehmen.



An dieser Stelle möchte ich Nathan Torkington für seine »Taxonomy of Tags«, veröffentlicht in der [www-talk-Mailing-Liste](#) danken, welche mir Anregungen für diesen Abschnitt gab. *Explizite Navigationsverbindungen* sind Verknüpfungen, die auf bestimmte Wege verweisen, die man durch Ihre Web-Dokumente nehmen kann: vorwärts, zurück, hoch, zurück zum Anfang (auf die Homepage). Diese Verbindungen werden häufig durch Navigations-Symbole bezeichnet (vgl. Abbildung 16.16).

Implizite Navigationsverbindungen (Abbildung 16.17) unterscheiden sich von expliziten insofern, als der Verknüpfungs-Text eine Verbindung enthält, aber nicht direkt auf sie hinweist. Verknüpfungs-Menüs sind das beste Beispiel hierfür; die Hervorhebung des Verknüpfungs-Textes macht es offensichtlich, dass Sie weitere Informationen zu dem Thema erhalten, wenn Sie die Verbindung anwählen, doch der Text selbst drückt dies nicht unbedingt aus. Der Hauptunterschied zwischen expliziten und impliziten Navigationsverbindungen besteht darin, dass Sie nach dem Ausdrucken einer Seite mit diesen beiden die impliziten Verbindungen womöglich nicht mehr erkennen können.

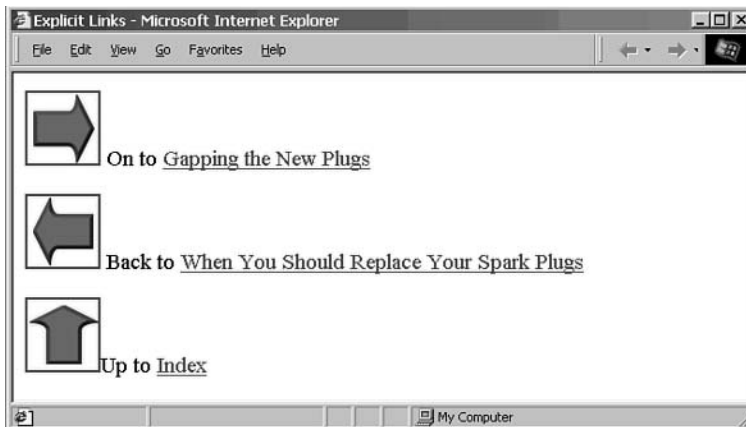


Abbildung 16.16:
Explizite Navigationsverbindungen

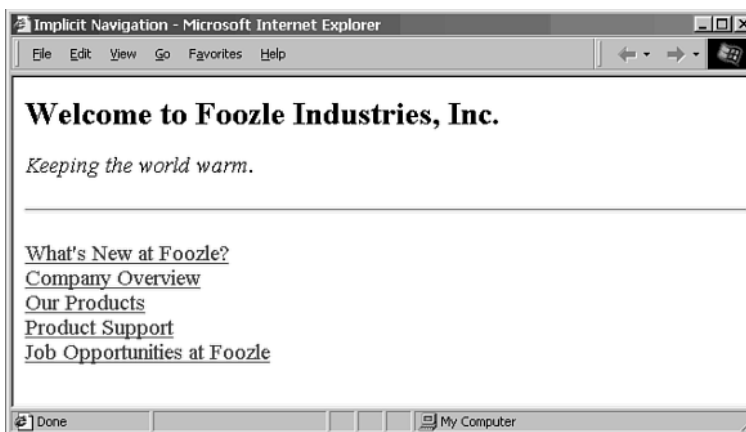


Abbildung 16.17:
Implizite Navigationsverbindungen

Implizite Navigationsverbindungen können auch aus Strukturen bestehen wie beispielsweise einem Inhaltsverzeichnis oder aus ähnlichen Übersichten, die vollständig aus Verknüpfungen bestehen.

Wort- oder Begriffsdefinitionen ergeben ausgezeichnete Verbindungen, insbesondere wenn Sie ein großes Netzwerk aus Dokumenten erzeugen, die Wörterbücher enthalten. Indem Sie beim ersten Auftauchen eines Worts dieses mit der entsprechenden Definition irgendwo anders verbinden, können Sie den Lesern, die es nicht kennen, die Bedeutung des Worts erklären, ohne jene Leser abzulenken, die sie bereits wissen (vgl. Abbildung 16.18).

Schließlich sind Verbindungen mit ähnlichen und *verwandten Informationen* wertvoll, wenn deren Inhalt vom Hauptzweck des Dokuments ablenken würde. Stellen Sie sich solche Verbindungen ähnlich wie Fußnoten in einem gedruckten Text (vgl. Abbildung 16.19) vor; sie können sich auf Zitate aus anderen Werken beziehen oder auf zusätzliche Informationen, die interessant sind, aber nicht unbedingt mit dem Thema Ihrer Webseite zu tun haben.

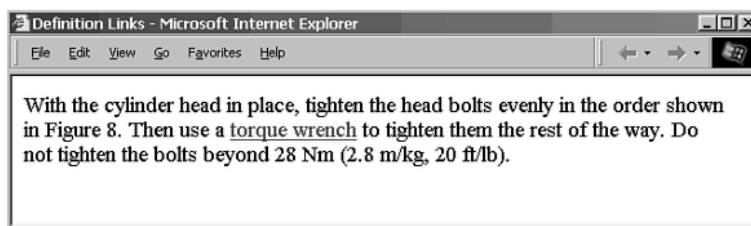


Abbildung 16.18:
Definitionsverbindungen

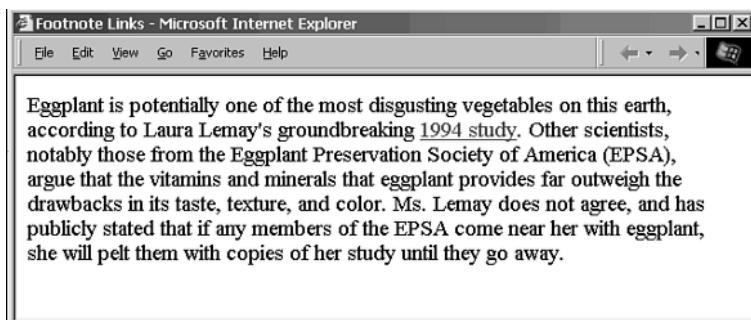


Abbildung 16.19:
Fußnoten-Verbindungen

Sehen Sie sich vor, dass Sie es mit den Verbindungen zu ähnlichen und verwandten Informationen nicht übertreiben. Es ist möglich, so viele Verknüpfungen zu verwandten Themen aufzustellen, dass Ihre Leser zuviel Zeit damit verbringen und schließlich den Faden verlieren. Widerstehen Sie der Versuchung, bei jeder Gelegenheit zu verknüpfen, sondern tun Sie es nur, wenn es erforderlich ist.

Vermeiden Sie es auch, zum selben Thema mehrfach zu verknüpfen – beispielsweise indem Sie jedes Auftauchen der Buchstaben »WWW« auf Ihrer Seite mit der Homepage des WWW-Projekts in der Schweiz verknüpfen. Wenn Sie zweimal oder häufiger auf

dieselbe Stelle auf derselben Seite verweisen, überlegen Sie, ob Sie nicht die meisten zusätzlichen Verknüpfungen entfernen können. Ihre Leser haben die Möglichkeit, eine der Verknüpfungen anzuwählen, wenn sie an den Informationen interessiert sind.

Bilder verwenden

Am Tag 7 haben Sie alles über das Erzeugen und die Verwendung von Bildern auf Webseiten gelernt. Dieser Abschnitt fasst vieles von dem zusammen, was Sie bereits über Bilder gelernt haben.

Übertreiben Sie es mit den Abbildungen nicht

Nehmen Sie sich davor in Acht, Unmengen von Abbildungen in Ihre Webseite aufzunehmen. Abgesehen von dem Umstand, dass jede Abbildung die Zeit verlängert, die zum Laden des Dokuments benötigt wird, kann das Aufnehmen vieler Abbildungen Ihr Dokument aufdringlich und überladen erscheinen lassen und von dem ablenken, worauf Sie eigentlich hinweisen wollen (vgl. Abbildung 16.20).

Erinnern Sie sich an die Tipps, die ich Ihnen in Kapitel 7 gab. Fragen Sie sich mittlerweile, wofür Sie die jeweilige Abbildung brauchen, bevor Sie sie in Ihre Seite einfügen, und lassen Sie sie weg, wenn Sie keine direkte Bereicherung des Inhalts darstellt?

Verwenden Sie Alternativen zu Abbildungen

Natürlich muss ich, sobald ich auf Abbildungen zu sprechen komme, wieder darauf hinweisen, dass nicht alle Browser diese Abbildungen darstellen können. Um Ihre Dokumente für das größtmögliche Publikum verfügbar zu machen, müssen Sie auch die textbasierten Browser berücksichtigen, wenn Sie Ihre Webseiten entwerfen. Diese beiden Anregungen sollen Ihnen dabei helfen:

- ▶ Verwenden Sie das `alt`-Attribut der `<img>`-Kennung, um in textbasierten Browsern automatisch einen passenden Text anstelle der Abbildungen einsetzen zu lassen. Benutzen Sie entweder eine passende Bezeichnung, um die Vorgabe `[IMAGE]` zu ersetzen, die anstelle einer internen Abbildung erscheint, oder benutzen Sie eine leere Zeichenfolge (`" "`) (keine typographischen Anführungszeichen, sondern gerade), um die Abbildung komplett zu ignorieren.
- ▶ Wenn es zuviel Arbeit wird, eine einzelne Seite für sowohl grafische als auch textbasierte Browser anzubieten und das Ergebnis Ihnen nicht gefällt, erwägen Sie die Erzeugung separater Seiten für jeden von beiden: eine für Farb- und grafikfähige Browser gestaltete Seite und eine für die textbasierten Browser. Auf Ihrer Homepage bieten Sie dann die Möglichkeit an, eine der beiden anzuwählen.

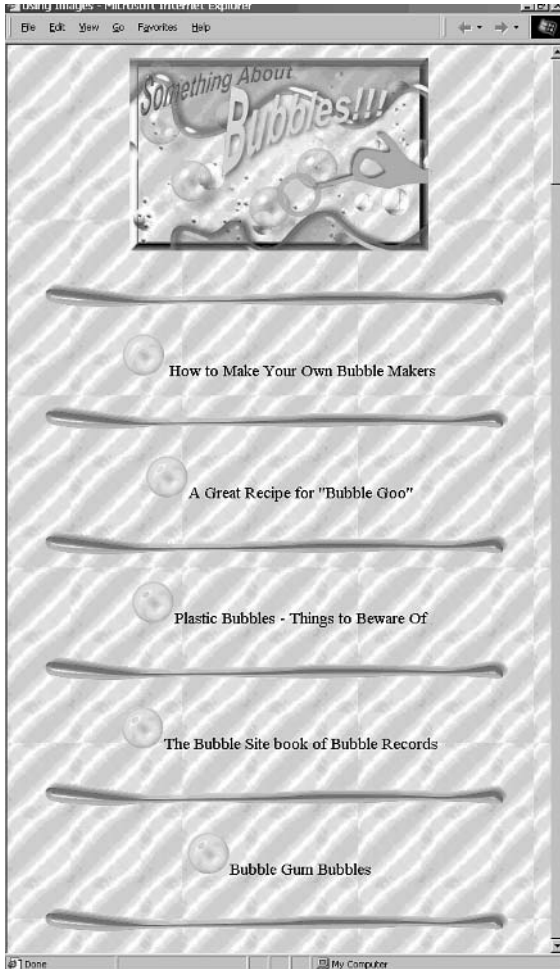


Abbildung 16.20:
Zu viele Abbildungen

Halten Sie Ihre Abbildungen klein

Bedenken Sie, wenn Sie Bilder einsetzen, dass jedes eine separate Netzwerk-Verbindung benötigt und Zeit braucht, um über das Netzwerk geladen zu werden; mit anderen Worten, jedes Bild verlängert die Zeit, die benötigt wird, bis Ihre Seite geladen ist. Versuchen Sie, die Anzahl der Bilder auf einer Seite geringzuhalten und halten Sie die Bilder sowohl in ihrer Dateigröße als auch in ihren physikalischen Dimensionen klein. Behalten sie insbesondere die folgenden Hinweise im Hinterkopf:

- ▶ Es gilt als Faustregel für große Bilder, dass Ihre Seite bei einer 14,4-Kbps-Modemverbindung mit ungefähr 1 Kbyte pro Sekunde geladen wird. Die gesamte Seite (Text und Bilder) sollte nicht länger als 30 Sekunden zum Laden brauchen, oder Sie riskieren es, Ihre Leser zu verärgern und zu verlieren, ohne dass Ihre Seite gelesen wird. Dies beschränkt Sie auf eine Gesamtsumme von 30 Kbyte pro Seite. Streben Sie diese Größe an, indem Sie Ihre Bilder kleinhalten.
- ▶ Für größere Bilder sollten Sie die Verwendung von Thumbnails in Betracht ziehen, die dann zu einer größeren Version verknüpft sind, anstatt gleich das größere Bild intern auf der Seite einzufügen.
- ▶ Verwenden Sie Interlacing für Ihre GIF-Dateien.
- ▶ Testen Sie Ihr Bild als GIF und auch als JPEG, um herauszubekommen, welches für den von Ihnen verwendeten Bildtyp die kleinste Dateigröße ergibt.
- ▶ Bei GIF-Dateien wird die Dateigröße kleiner, wenn Sie weniger Farben verwenden; deshalb sollten Sie so wenig Farben wie möglich einsetzen, um auch Probleme mit der systemspezifischen Farbzurordnung zu vermeiden.
- ▶ Sie können die Ausmaße Ihres Bildes durch Ausschneiden (die Verwendung eines kleineren Teils des ursprünglichen Bildes) oder durch Skalierung (Verkleinerung) des Originalbildes vermindern. Beachten Sie jedoch, dass durch Skalierung wahrscheinlich einige der Feinheiten des Originalbildes verlorengehen.
- ▶ Sie können die Attribute `WIDTH` und `HEIGHT` verwenden, um das Bild zu vergrößern, ohne seine Dateigröße zu beeinflussen. Ursprünglich waren dies reine Netscape-Erweiterungen. Inzwischen sind sie Teil von HTML 3.2. Beachten Sie aber, dass das skalierte Ergebnis nicht Ihren Erwartungen entsprechen könnte. Testen Sie es, bevor Sie sich darauf verlassen.

Sehen Sie sich die Bilder der Seite ein zweites Mal an und denken Sie dabei an die eben gemachten Vorschläge. Sie haben Ihre Bilder wirklich ins Herz geschlossen und wollen sich von keinem trennen. Wie kann man also die in Bild 16.20 gezeigte Seite auf Diät setzen und somit das gesamte Erscheinungsbild verbessern?

Was die Bildgröße betrifft, könnte die Grafik zu Beginn der Seite, die ein Site-Logo darstellt, eine Größenreduzierung vertragen. Logos sind meistens nur Banner und enthalten keine Links. Sie können die Grafik also auf die Hälfte der ursprünglichen Größe reduzieren. Die Ladezeit für die Grafik wird somit um die Hälfte verringert.

Ein weiteres Problem, das hier noch erwähnt werden muss, ist, dass der Seitentitel, (in diesem Fall der Site-Name) im Gegensatz zum Text, nicht an irgendeiner Stelle der Seite erscheinen kann. Diejenigen, die Ihre Site besuchen und die Grafiken nicht angezeigt haben, werden nicht einmal den Namen der Site kennen! Das muss unserer verbesserten Version hinzugefügt werden.

Die horizontalen Linien sind wirklich ein großes Problem. Erstens, gibt es zu viele davon. Zweitens, überragen sie das Banner-Bild, da sie ja viel länger sind. Und drittens, lenken sie von der Link-Liste ab. Also werden wir die Anzahl und die Größe dieser Bilder verringern müssen. Und schon haben wir wieder Ladezeit gespart.

Die Merkpunkte, die vor jeder Item-Liste stehen, sind viel zu groß. Wir können sie ruhig um 50 Prozent verkleinern! In der Regel sind Merkpunkte 30 x 30 Pixel groß. (Sie können auch kleiner sein.)

Die Merkpunkte und der Text wurden zentriert. Dadurch wirkt die Link-Liste sehr unorganisiert. Es gibt verschiedene Methoden, um eine Liste mit Blickfangpunkten zu erstellen. Entweder Sie gehen den offiziellen Weg mit dem `<ul>`-Tag. Dabei können Sie mit dem `src`-Attribut das Bild für den Blickfangpunkt festlegen. HTML 3.2 und frühere Browser-Versionen können nur eine normale Liste mit den Standardblickfangpunkten anstelle der Bilder darstellen. Eine weitere Methode wäre, die Bilder und Liste mit einer rahmenlosen Tabelle zu gestalten. Sämtliche Verbesserungsvorschläge sind in Bild 16.21 zu sehen.



Abbildung 16.21:
Bessere Verwen-
dung der Bilder

Vorsicht bei zu hohen Systemvoraussetzungen

Viele Web-Entwickler bereiten ihren Lesern Probleme, indem Sie einige leichtfertige Annahmen über die Hardware anderer Leute treffen. Wenn Sie Webseiten entwickeln, seien Sie so nett und denken Sie an diese beiden Punkte:

- ▶ Gehen Sie nicht davon aus, dass alle die gleichen Bildschirm- oder Browser-Dimensionen wie Sie haben.
- ▶ Nur weil das riesenhafte GIF, das Sie angefertigt haben, auf Ihren Bildschirm passt, heißt das noch lange nicht, dass es auch auf die Browser-Bildschirme anderer passt. Und es kann ärgerlich für Ihre Leser sein, auf ein Bild zu stoßen, das zu breit ist, weil sie dann entweder ihre Fenster vergrößern oder scrollen müssen.
- ▶ Damit Ihre Bilder in die Mehrzahl der Browser-Fenster hineinpassen, versuchen Sie, die Breite der Bilder kleiner als 450 Pixel zu halten (die meisten Macintosh-Browser haben eine Bildschirmbreite von ca. 465 Pixel).
- ▶ Unterstellen Sie nicht, dass jeder alle Farben hat.
- ▶ Testen Sie Ihre Abbildungen auch in Auflösungen mit wenigen Farben (Sie können dies meist in Ihrem Bildverarbeitungsprogramm tun). Viele Ihrer Leser haben wahrscheinlich Display-Systeme, die nur in 16 Farben, Grautönen oder sogar nur in Schwarzweiß abbilden. Die Ergebnisse könnten Sie überraschen: Farben verschwinden oder erzeugen in Grautönen oder Schwarzweiß seltsame Muster.
- ▶ Vergewissern Sie sich also, dass Ihre Bilder bei allen Auflösungen zu erkennen sind, und bieten Sie Alternativen für hohe und niedrige Auflösungen der Bilder auf der Seite an.

Seien Sie vorsichtig bei Hintergründen und Verknüpfungs-Farben

Mit HTML-Erweiterungen können Sie Hintergrundfarben und -muster einsetzen und die Farben des Textes auf Ihrer Seite verändern. Die Anwendung dieses Merkmals kann sehr verlockend sein, seien Sie aber sehr vorsichtig, wenn Sie sich dafür entscheiden. Die Fähigkeit, die Farbe Ihrer Seiten und Schrift zu verändern und schicke Hintergrundbilder einzusetzen, könnte auch bewirken, dass Sie Ihre Seiten schnell und einfach völlig unlesbar machen. Hier folgen einige Hinweise, wie Sie dies vermeiden können:

- ▶ Vergewissern Sie sich, genügend Kontrast zwischen den Hintergrund- und Vordergrundfarben (Textfarben) zu verwenden. Bei zu wenig Kontrast kann der Text schwer zu lesen sein. Außerdem ist heller Text auf dunklem Hintergrund schwerer zu lesen als dunkler Text auf hellem Hintergrund.

- ▶ Vermeiden Sie es gänzlich, die Verknüpfungs-Farben zu verändern. Da Ihre Leser semantische Assoziationen zu den Voreinstellungsfarben haben (Blau bedeutet nichtverfolgte Verknüpfung, Violett oder Rot bedeutet bereits aktivierte Verknüpfung), kann der Wechsel dieser Farben sehr verwirrend wirken.
- ▶ Manchmal kann die Vergrößerung der Schrift mit `<basefont>` den Text auf dem Hintergrund lesbarer gestalten. Jedoch werden sowohl der Hintergrund als auch die Vergrößerung des Textes in Browsern fehlen, die diese Netscape-Kennungen nicht verstehen.
- ▶ Wenn Sie Hintergrundmuster verwenden, vergewissern Sie sich, dass diese Muster den Text nicht weniger lesbar gestalten. Einige Muster können isoliert betrachtet interessant aussehen, es aber erschweren, den Text, der daraufsteht, zu lesen. Vergessen Sie nicht, dass Hintergründe auch im Hintergrund bleiben sollen. Subtile Muster sind immer besser als wilde. Ihre Leser kommen doch wegen des Inhalts Ihrer Seiten, nicht um Ihre Fähigkeit zu bewundern, künstliche Marmormuster mit Ihrem Bildverarbeitungsprogramm zu kreieren.

In Zweifelsfall bitten Sie einfach einen Freund, sich Ihre Seiten anzusehen. Da Sie den Inhalt und Text bereits kennen, fällt Ihnen vielleicht gar nicht auf, wie schwer Ihre Seiten zu lesen sind. Jemand, für den dies neu ist, wird Ihre Vorurteile nicht teilen und dazu in der Lage sein, zu beurteilen, ob Ihre Farben zu ähnlich sind oder das Hintergrundmuster den Text beeinträchtigt. Vorausgesetzt, der Freund ist aufrichtig.

Andere gute Angewohnheiten und Tipps

In diesem Abschnitt habe ich einige Tipps und Hinweise beim Arbeiten mit mehreren Webseiten zusammengetragen. Hinzu kommen einige Bemerkungen darüber, welchen Umfang jede Seite Ihrer Präsentation haben sollte und wie Sie Ihre Dokumente kennzeichnen.

Eine Verbindung mit der Homepage

Erwägen Sie für jede einzelne Seite Ihrer Präsentation eine Verbindung, die Ihre Leser zurück zur obersten Ebene oder Ihrer Homepage führt. Das Angebot einer solchen Verknüpfung erlaubt Ihren Lesern, schnell aus den Tiefen Ihrer Inhalte zu entkommen. Die Benutzung einer Verbindung zur Homepage ist für Ihre Leser viel einfacher, als zu versuchen, durch eine Hierarchie zurückzunavigieren oder, schlimmer noch, mit der »Zurück«-Möglichkeit ihres Browser all ihre Schritte einzeln zurückzugehen.

Verteilen Sie Themen nicht über mehrere Seiten

Jedes Web-Dokument wirkt am besten, wenn es ein Thema in seiner Gesamtheit behandelt. Verteilen Sie Ihre Themen nicht über mehrere Seiten; selbst wenn Sie diese miteinander verknüpfen, mögen die Übergänge verwirrend sein. Und erst recht irritierend wird es sein, wenn jemand auf die zweite oder dritte Seite springt, die das Thema behandelt, und sich dann fragt, worum es eigentlich geht.

Wenn Sie meinen, ein Thema werde zu groß für ein einzelnes Dokument, dann überlegen Sie, wie Sie den Inhalt, anstatt das Dokument an willkürlichen Stellen aufzuteilen, so reorganisieren, dass Sie das Thema in Unterthemen zerlegen können. Dies funktioniert besonders gut in hierarchischen Organisationen, da sie Ihnen erlauben, genau zu erkennen, in welche Ebene der Hierarchie eine bestimmte Detailebene gehört und wie groß und vollständig jede Seite sein sollte.

Erzeugen Sie nicht zu viele und nicht zu wenige Dokumente

Es gibt keine Regel dafür, wie viele Seiten Sie in Ihrer Web-Präsentation haben müssen, und keine Regel, wie lang jede Seite sein sollte. Sie können eine einzige Seite haben oder einige Tausend, je nach der Menge an Inhalten, die Sie behandeln wollen und wie Sie diese organisiert haben.

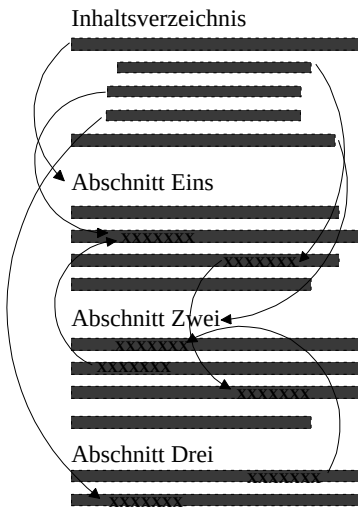


Abbildung 16.22:
Ein großes Dokument

Dies im Sinn, können Sie sich für das eine oder das andere Extrem entscheiden, jedes hat Vorzüge und Nachteile. Nehmen wir zum Beispiel an, Sie schreiben all Ihre Inhalte auf eine große Seite und erzeugen Verknüpfungen mit den einzelnen Abschnitten des Dokuments (wie in Abbildung 16.22 illustriert).

Vorzüge:

- ▶ Eine Datei ist leichter zu pflegen, und die Verknüpfungen darin werden nie »zerbrechen«, wenn Sie die Datei verschieben oder umbenennen.
- ▶ Es spiegelt die Dokumenten-Struktur der realen Welt wider; falls Sie die Dokumente sowohl gedruckt als auch online vertreiben, vereinfacht eine einzelne Datei beide Produktionen.

Nachteile:

- ▶ Eine umfangreiche Datei braucht sehr viel Zeit beim Laden, insbesondere bei langsamen Netzwerk-Verbindungen und besonders, wenn sie zahlreiche Grafiken enthält.
- ▶ Die Leser müssen viel blättern, um zu finden, was sie suchen. Das Zugreifen auf einzelne Teile der Informationen kann ermüdend werden. Das Navigieren an anderen Stellen als am Anfang und am Ende wird nahezu unmöglich.
- ▶ Die Struktur ist überaus streng. Ein einzelnes Dokument ist schon von Natur aus linear. Obgleich Sie zwischen den Abschnitten in dem Dokument herumspringen können, spiegelt die Struktur doch noch immer die einer gedruckten Seite wider und verzichtet auf die Vorteile der Flexibilität kleinerer in nichtlinearer Struktur miteinander verknüpfter Dokumente.

Andererseits könnten Sie eine Reihe kleiner Dokumente erzeugen und zwischen ihnen viele Verbindungen haben, wie es Abbildung 16.22 zeigt.

Vorzüge:

- ▶ Kleine Dokumente werden sehr schnell geladen.
- ▶ Oft passt die gesamte Seite auf einen Bildschirm, sodass die Informationen in dem Dokument von Ihren Lesern schnell überflogen werden können.

Nachteile:

- ▶ Die Pflege aller Verknüpfungen kann zum Alptraum werden. Allein schon das Einfügen irgendeiner Art von Navigationsstruktur in so viele Dokumente kann Tausende von Verbindungen erzeugen.
- ▶ Wenn Sie zu viele Sprünge zwischen den Dokumenten machen, kann das Springen unangenehm werden. Der rote Faden geht verloren, wenn Ihre Leser mehr Zeit damit verbringen herumzuspringen, als tatsächlich zu lesen.

Was also wäre also Lösung? Häufig werden die Inhalte, die Sie beschreiben, die Größe und Anzahl der Seiten, die Sie dafür brauchen, vorgeben, insbesondere wenn Sie dem Vorschlag »ein Thema pro Seite« folgen. Oft können Sie auch erkennen, ob ein Dokument zu groß ist, wenn Sie Ihre Webseiten auf verschiedenen Plattformen und mit unterschiedlichen Netzwerk-Geschwindigkeiten testen. Wenn Sie viel Zeit damit verbringen herumzublättern oder wenn beispielsweise mehr Zeit, als Sie erwartet haben, zum Laden benötigt wird, dürften die Seiten zu groß sein.

Unterzeichnen Sie Ihre Dokumente

Jedes Dokument sollte irgendeine Art von Information am Ende der Seite enthalten, die als »Unterschrift« des Dokuments dient. Ich sprach in Kapitel 6, »Noch mehr Textformatieren mit HTML«, ausführlich bei der Beschreibung der `<address>`-Kennung darüber; diese besondere Kennung ist genau für diesen Zweck gedacht.

Hier ist eine Liste mit einigen nützlichen Informationen, die Sie möglicherweise in die `<address>`-Kennung auf jeder Seite schreiben können:

- ▶ Angaben dazu, wer die Seite erzeugt hat oder verantwortlich für sie ist – umgangssprachlich als »Webmaster« bezeichnet. Dies sollte wenigstens den Namen und vorzugsweise eine E-Mail-Adresse enthalten.
- ▶ Der Status des Dokuments. Ist es fertig? Ist es noch in Arbeit? Ist die Seite absichtlich leer geblieben? ähl Punkt3
- ▶ Wann das Dokument zuletzt überarbeitet wurde. Dies ist besonders für Dokumente wichtig, die häufig geändert werden. Fügen Sie in jedes Dokument ein Datum ein, sodass die Leute wissen, wie alt es ist.
- ▶ Urheberrechts- und Warenzeichen-Angaben, soweit dies notwendig ist. ähl Punkt3
- ▶ Die URL des Dokuments. Das Einfügen einer geschriebenen URL in das Dokument, das mit dieser URL gefunden werden kann, mag etwas übertrieben erscheinen, doch was passiert, wenn jemand die Seite ausdruckt und sie in einem Papierberg auf dem Schreibtisch liegen läßt? Woher kam das bloß? (Mir ist das viele Male passiert, und ich wünschte mir oft, dass die URL in dem Dokument selbst geschrieben stünde.

Abbildung 16.23 zeigt ein nettes Beispiel für einen `ADDRESS`-Block.

Ein netter Zug auf Ihrer Webseite wäre es, den Text, der die E-Mail-Adresse des Webmaster enthält, wie hier mit einer `mailto`-URL auszuzeichnen:

```
<address>
Laura Lemay <a href="mailto:lemay@netcom.com"> lemay@netcom.com </a>
</address>
```



Abbildung 16.23:
Ein Adressen-Bei-
spiel

Dies erlaubt jenen Lesern des Dokuments, die Browser benutzen, welche die `mailto`-URL unterstützen, einfach diese Verknüpfung anzuwählen und Post an die für die Seite verantwortliche Person zu schicken, ohne die Adresse für ihr Mail-Programm abschreiben zu müssen.



Das funktioniert nur mit Browsern, die `mailto`-URLs unterstützen. Doch selbst in Browsern, die das nicht akzeptieren, wird der Verknüpfungs-Text wie gewohnt dargestellt, also schadet es nicht, diese Verknüpfung in jedem Fall einzusetzen.

Falls Sie indes nicht jede Seite mit persönlichen Kontaktmöglichkeiten oder siedend-heißen Urheberrechts-Informationen vollstopfen mögen, so wäre hier eine einfache Lösung, für diese zusätzlichen Informationen eine eigene Seite zu erzeugen und dann die Unterschrift mit jener Seite zu verbinden wie hier:

```
<address>
<a href="copyright.html">Copyright</a> and
<a href="webmaster.html">contact</a> information is available.
</address>
```

Bieten Sie auch Nicht-Hypertext-Versionen Ihrer Hypertext-Seiten an

Obgleich das Web einen Weg bietet, Dokumente auf neue und aufregende Weise zu erzeugen, mögen doch manche Leser noch immer viele Dinge lieber »offline« lesen, im Bus oder am Frühstückstisch. Solche Leser haben echte Probleme mit Hypertext-Dokumenten, denn wenn Sie einmal damit beginnen, Hypertext für die Organisation Ihrer Dokumente zu verwenden, wird es schwierig, einfach sagen zu können »Druck das ganze Ding aus.« Der Browser kennt nämlich nur die Grenzen individueller Seiten.

Wenn Sie das Web dazu benutzen, irgendetwas zu publizieren, das auch außerhalb des Web lesbar und nützlich sein könnte, dann ziehen Sie die Erzeugung einer einzelnen Text- oder PostScript-Datei für Ihre Web-Dokumente in Betracht, die Sie als externes

Dokument zum »Download« zur Verfügung stellen. Das ermöglicht Ihren Lesern, das Dokument sowohl online mit ihrem Browser zu lesen als auch das Ausdrucken des Dokuments, wenn sie das wollen, um es offline zu lesen. Sie können diese Textversion sogar in eine Verknüpfung am Beginn der Hypertext-Version aufnehmen wie hier:

```
Eine <a href="ftp://mein.heim.de/pub/meinverzeichnis/dasdokument.ps">
PostSript-Version (500 KB) </a> dieses Dokumentes ist per FTP vom System
mein.heim.de als Datei /pub/meinverzeichnis/dasdokument.ps beziehbar.
Und selbstverständlich macht sich ein kleiner Hinweis in der Textversion gut, der
auf die Hypertext-Version hinweist:
"Dieses Dokument ist im World Wide Web mit dem URL
http://mein.heim.de/pub/meinverzeichnis/dasdokument.index.html
auch in Hypertext-Form zu finden."
```

Zusammenfassung

Sie sollten die folgenden wichtigen Gesichtspunkte beim Gestalten von Webseiten beachten:

- ▶ Verdeutlichen Sie sich den Unterschied zwischen HTML 2.0, HTML 3.2 und die verschiedenen Spezifikationen von HTML 4.0 und XHTML 1.0 und entscheiden Sie sich für eine Designstrategie.
- ▶ Bieten Sie, wenn möglich, Alternativen zu HTML-Erweiterungen an.
- ▶ Testen Sie Ihre Seiten in verschiedenen Browsern.
- ▶ Schreiben Sie Ihre Dokumente klar und bündig.
- ▶ Organisieren Sie den Text Ihres Dokuments so, dass die Leser ihn auf interessante Informationen hin überfliegen können.
- ▶ Schreiben Sie keine Webseiten, die von den Seiten vor oder nach ihnen strukturell abhängen. Schreiben Sie Seiten, die alleine dastehen können.
- ▶ Benutzen Sie nicht zuviel Hervorhebungen (Fettschrift, kursiv, Großbuchstaben, Verknüpfungs-Text, blinken, Textfahnen). Verwenden Sie Hervorhebungen zurückhaltend und nur, wenn sie absolut nötig sind.
- ▶ Prüfen Sie die Rechtschreibung, und lesen Sie Korrektur.
- ▶ Benutzen Sie keine Überschriften, um etwas hervorzuheben.
- ▶ Gruppieren Sie verwandte Informationen sowohl nach ihrer Bedeutung (mit Hilfe der Organisation des Inhalts) als auch visuell (durch die Verwendung von Überschriften oder die Unterteilung der Abschnitte mit Trennlinien).

- ▶ Verwenden Sie ein gleichmäßiges Layout über all Ihre Seiten hinweg.
- ▶ Verwenden Sie Verknüpfungs-Menüs, um Ihre Verknüpfungen für schnelles Überfliegen zu organisieren. Verwenden Sie außerdem aussagekräftige Namen für Ihre Verknüpfungen.
- ▶ Vermeiden Sie das »Hier«-Syndrom bei Verknüpfungen.
- ▶ Suchen Sie gute Gründe für Verbindungen. Verknüpfen Sie kein irrelevantes Material.
- ▶ Bieten Sie nicht mehrfach auf derselben Seite Verbindungen zu denselben Stellen an.
- ▶ Halten Sie das Layout einfach.
- ▶ Blähen Sie die Seite nicht mit Unmengen hübscher, aber unnötiger Abbildungen auf.
- ▶ Bieten Sie für textbasierte Browser Alternativen zu den Abbildungen.
- ▶ Halten Sie Ihre Bilder klein, sodass Sie schneller übers Netzwerk laden.
- ▶ Seien Sie vorsichtig bei Hintergründen und farbigem Text, sodass Ihre Seiten zwar auffällig, aber unlesbar werden.
- ▶ Bieten Sie stets eine Verbindung zu Ihrer Homepage.
- ▶ Erstellen Sie für jedes Thema eine Seite.
- ▶ Verteilen Sie keine einzelnen Themen über verschiedene Seiten.
- ▶ Verwenden Sie am Ende jeder Seite einen Unterschriftsblock oder eine Verknüpfung zu einer Kontaktadresse.
- ▶ Bieten Sie eine eigene Textversion von linearen Dokumenten an.

Workshop

Es ist wieder einmal an der Zeit, unsere Denkkappe aufzusetzen. It's Workshop-time!

Fragen und Antworten

- F** Statistiken im Web ergaben, dass die meisten Leute im Web den Netscape oder den Internet Explorer benutzen. Warum sollte ich also weiterhin meine Seiten für andere Browser gestalten und sie in anderen Browsern testen, wenn doch sowieso die halbe Welt einen dieser zwei Browser benutzt?

A Natürlich können Sie Ihre Seiten ausschließlich für Netscape, den Internet Explorer oder für beide Browser gestalten – es sind ja schließlich Ihre Seiten, also ist es auch Ihre Entscheidung. Aber angesichts der Tatsache, dass man nur mit geringen Änderungen anderen Browsern ermöglicht, Ihre Seiten zu sehen und zu lesen, ohne dabei viel von Ihrem Design einbüßen zu müssen, warum also den übrigen Teil Ihres Publikums ausschließen? Denken Sie daran: Wenn man den Schätzungen des Webs Glauben schenken soll, steigt diese Minderheit ständig an und beträgt inzwischen schon über eine Millionen Menschen

F **Ich konvertiere vorhandene Dokumente in Webseiten. Diese Dokumente sind sehr textlastig und dazu gedacht, vom Anfang bis zum Ende gelesen und nicht bloß schnell überflogen zu werden. Ich kann den Inhalt nicht neu strukturieren oder gestalten, um den Richtlinien, die Sie in diesem Kapitel vorgeschlagen haben, besser zu folgen – dafür bin ich nicht zuständig. Was kann ich tun?**

A Manche Inhalte sind halt so, besonders, wenn ein Dokument, das in Papierform geschrieben ist, online lesbar gemacht werden soll. Im Idealfall wären Sie in der Lage, es für die Online-Präsentation neu zu schreiben und zu strukturieren, doch realistischerweise kann man häufig nicht viel mehr mit dem Inhalt tun, als ihn einfach online verfügbar zu machen.

Doch ist noch nicht alles verloren. Sie können die Gesamtpräsentation dieser Dokumente noch immer verbessern, indem Sie vernünftige Indizes für den Inhalt anbieten (Zusammenfassungen, Seiten mit Inhaltsverzeichnissen, Themenverzeichnisse und so weiter) und indem Sie einige Standard-Verbindungen einfügen, um aus den textlastigen Seiten zurückzukehren. Mit anderen Worten: Sie können einen leicht bedienbaren Rahmen um die Dokumente selbst herum erzeugen, der Sie ein ganzes Stück voranbringt bei der Verbesserung des Inhalts, der andernfalls online nur schwer zu lesen wäre.

F **Ich habe einen Standard-Unterschriftsblock, der meinen Namen und meine E-Mail-Adresse enthält, Bearbeitungsvermerke für das Dokument und eine Reihe von Zeilen mit Urheberrechtsangaben, auf die der Anwalt meiner Firma besteht. Ist das nicht ein wenig übertrieben, besonders bei kleinen Seiten, auf denen die Signatur größer ist als die Seite selbst!**

A Wenn der Anwalt Ihrer Firma nichts dagegen hat, überlegen Sie sich eine separate Seite mit den Kontakt- und Urheberrechtsangaben, die Sie dann mit jeder Seite verbinden können, anstatt den Block jedes Mal zu kopieren. Auf diese Weise werden Ihre Seiten nicht mit juristischen Einzelheiten überfrachtet, und wenn die Unterschrift einmal geändert wird, muss sie nicht auf jeder einzelnen Seite geändert werden.

Quiz

1. Welches sind die drei Spezifikationen von HTML 4.0 und XHTML 1.0, und welche dieser Spezifikationen wird am meisten angewandt?
2. Nennen Sie einige Methoden, mit denen Sie ihre Seiten so organisieren können, sodass der Leser sie leichter erfassen kann.
3. Richtig oder falsch: Sie können die Ladezeit eines Bildes reduzieren, wenn Sie die `width`- und `height`-Attribute des `<img>`-Tags skalieren.
4. Richtig oder falsch: Die Tags für die Definition von Überschriften stellen eine ideale Methode dar, um gewisse Informationen mit größerer Schrift und fett hervorzuheben.
5. Was sind die Vor- und Nachteile einer einzigen, großen Webseite, im Gegensatz zu vielen kleinen Webseiten?

Antworten

1. Die drei Spezifikationen von HTML 4.0 und XHTML 1.0 sind Transitional (für ein größtmögliches Anwenderpublikum und für diejenigen geeignet, die ältere Browser benutzen), Frameset (enthält sämtliche Tags der Transitional-Spezifikation und die der Framesets), Strict (für diejenigen, die weiterhin mit reiner HTML 4.0 oder XHTML -Tags und -Attributen arbeiten wollen).
2. Die drei Methoden sind: Überschriften (um Ihre Themen zusammenzufassen), Listen (um Informationen zu organisieren und anzuzeigen), Link-Menüs (für die Navigation). Außerdem können Sie wichtige Informationen von langen Absätzen trennen.
3. Eine knifflige Frage – Wenn sie die `width`- und `height`-Attribute dazu verwenden, um ein Bild Ihrer Seite kleiner erscheinen zu lassen, wird dadurch zwar die Anzeigegröße reduziert, aber die Ladezeit wird dieselbe bleiben.
4. Falsch. Sie sollten Überschriften als Überschriften verwenden. Den Text auf einer Seite können Sie auch auf andere Art und Weise hervorheben, oder aber Sie verwenden eine Grafik, um die Aufmerksamkeit auf einen wichtigen Punkt Ihrer Seite zu ziehen.
5. Die Vorteile einer großen Seite sind, dass eine Seite leichter zu warten ist, Sie sich nicht um die Verlinkung zwischen den einzelnen Teilseiten kümmern müssen und sie die reale Struktur eines Dokuments widerspiegelt. Die Nachteile sind, dass Sie eine längere Ladezeit haben, Leser viel scrollen müssen und die Struktur zu steif und zu linear ist.

Übungen

1. Versuchen Sie doch einmal das Beispiel in Bild 16.2 zu erstellen. Organisieren Sie die Informationen in einer Liste mit Blickfangpunkten oder in einer Tabelle. Erleichtern Sie dem Leser das Erfassen der wichtigen Punkte in der Seite.
2. Versuchen Sie das Beispiel in Bild 16.4 zu erstellen. Arrangieren Sie die Informationen dieser Seite so, dass die wichtigen Punkte und Links leicht gefunden werden können.

A tilted, light gray calendar grid is visible in the background of the text area. The top row of the grid contains the letters 'M', 'D', 'M', 'D', 'I', and 'S' in white. The rest of the grid is empty.

Beispiele für gutes und schlech- tes Web-Design

**Woche
3**

In diesem Kapitel werden Sie einige einfache Beispiele von Seiten und Präsentationen kennen lernen, die Sie im Web finden könnten. (Tatsächlich werden Sie sie natürlich nicht im Web finden; ich habe diese Beispiele speziell für dieses Kapitel entwickelt.) Jede dieser Web-Präsentationen ist entweder typisch für eine Art, in der heutzutage Informationen im Web angeboten werden, oder sie zeigt eine Methode, um bestimmte Probleme zu lösen. Im Einzelnen lernen Sie die folgenden Web-Präsentationen kennen:

- ▶ Ein Firmenprofil der Foozle-Pullover-Fabrik
- ▶ Eine Enzyklopädie der Motorräder mit Bildern, Geräuschen und anderen Medien-Ausschnitten
- ▶ Den Katalog einer kleinen Baumschule, in dem Sie sowohl blättern als auch Kakteen und Sukkulanten bestellen können

Der Code und die Bilder für diese Beispiele sind ebenfalls in der Support-Site im Web zu finden.

Beispiel eins: ein Firmenprofil

Foozle Industries, Inc. ist auf Strickwaren spezialisiert. Besucher der Foozle Industries Web-Site, werden zuerst die Homepage dieser Site angezeigt bekommen (siehe Bild 17.1). Für die Gestaltung seiner Webseiten verwendet der Web-Autor auf der gesamten Site Cascading Style Sheets.

Zusätzlich zu den Style Sheets verwendet der Autor auch andere Elemente, die regelmäßig auf den Seiten vorkommen. Jede der Seiten der Site enthält ein kleines Banner, welches den Titel auf jeder Hauptseite anzeigt. Der untere Teil einer jeden Seite enthält eine Navigationsleiste, Copyright-Informationen und eine E-Mail-Adresse, die angeschrieben werden kann, falls man Fragen hat oder Kommentare zur Site abgeben möchte.

Von dieser nett gestalteten Homepage aus haben Sie mehrere Möglichkeiten, die in einem Verknüpfungs-Menü angeordnet sind. Sie werden in diesem Abschnitt nicht alle besuchen, sondern nur die, die interessante Eigenheiten bieten.



Abbildung 17.1:
Die Homepage
der Foozle GmbH

Was gibt es Neues bei Foozle?

Wählt man den »Was gibt es Neues?«-Link, werden Sie auf die entsprechende Seite verwiesen (siehe Bild 17.2). Dieses Link ist das erste Link der Homepage und das zweite (nach dem Home-Link) in der Navigationsleiste.

In rückwärtslaufender chronologischer Folge (vom neuesten Geschehnis an rückwärts) organisiert, enthält die Neuigkeiten-Seite Informationen über Interessantes aus der Foozle Inc. und ihrer Umgebung.

Diese Seite dient dazu, den Web-Kunden neue Produkte vorzustellen oder ihnen einfach Informationen über den Site, die Firma und andere Firmeninformation zur Verfügung zu stellen. Neuigkeiten-Seiten sind im Allgemeinen nützlich für Sites, die oft und regelmäßig besucht werden, da sie es Ihren Lesern ermöglichen, neue, Ihren Site betreffende Informationen schnell und einfach zu finden, ohne danach suchen zu müssen.

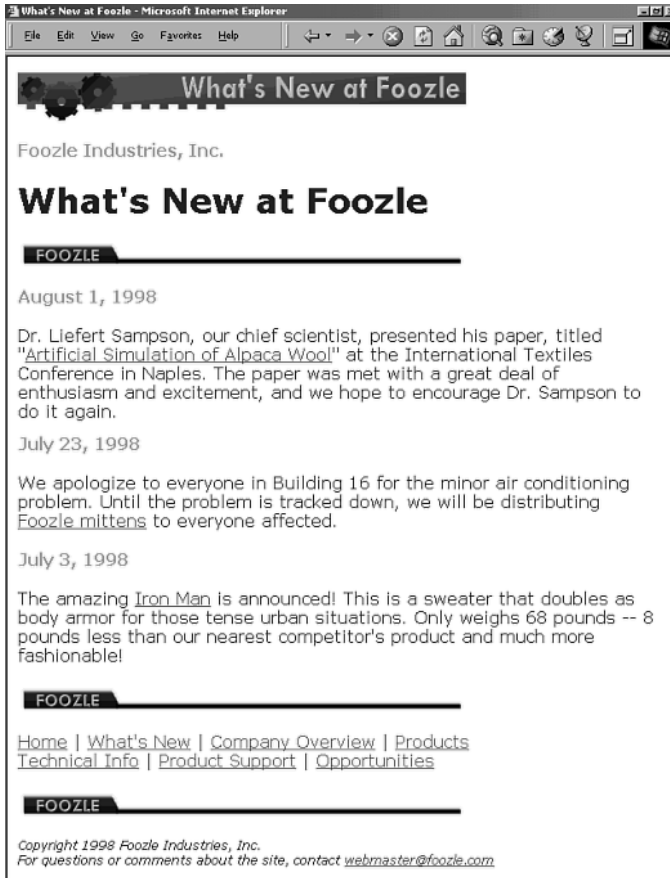


Abbildung 17.2:
Die Neuigkeiten-Seite
von Foozle

Der oberste Punkt in der Liste der Neuigkeiten ist der Hinweis auf ein Papier über die Behandlung von Alpakawolle, das vom wissenschaftlichen Direktor von Foozle auf einer Konferenz in Neapel präsentiert wurde. Dieser Punkt ist an eine Verknüpfung gebunden, was darauf hinweist, dass sich das Papier selbst auf der anderen Seite der Verknüpfung befindet, und siehe da, dort finden wir es auch (Abbildung 17.3)!

Alpakawolle ist etwas Faszinierendes, doch wie geht es nun weiter? Die Links im unteren Teil der Seite sind hier anders als in der Hauptseite. Auf der Hauptseite hat der Leser mehrere Navigationsmöglichkeiten. Er oder sie kann zur Foozle Homepage (mit dem Home-Link), zur technischen Info-Seite (mit dem technischen Info-Link) oder zu irgendeiner anderen Seite navigieren. Die Titel einer jeden Seite sind kurz in der Navigationsleiste aufgelistet. Diese Navigationsleiste erscheint auf allen technischen Seiten innerhalb dieser Sektion und ermöglicht somit eine einfache Navigation zwischen diesen technischen Seiten.

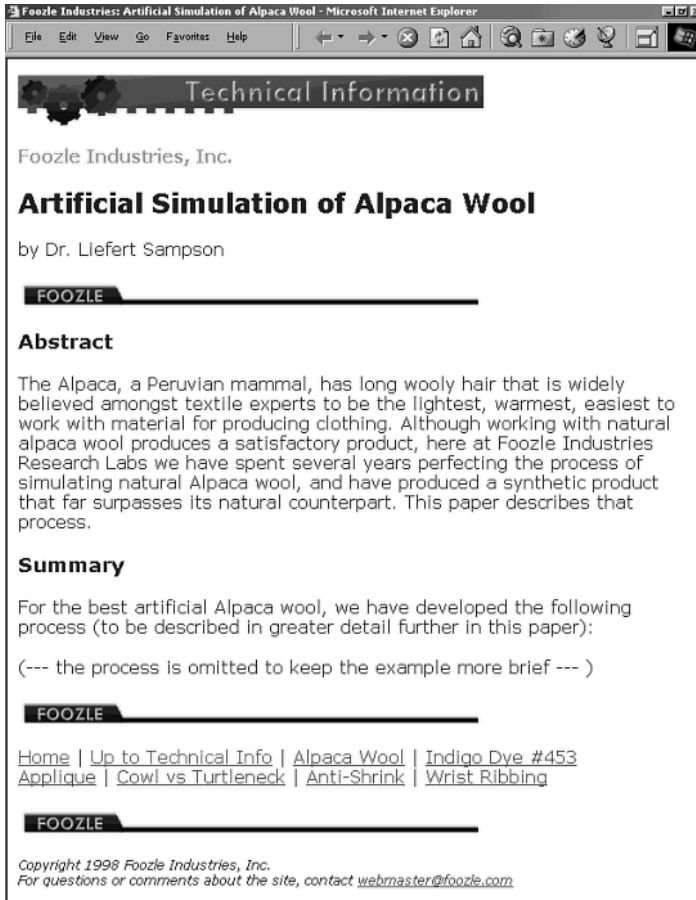


Abbildung 17.3:
 Alles über
 Alpakawolle von Fooz-
 le

Technische Informationen

Der Abschnitt mit technischen Informationen (Abbildung 17.4) besteht aus einer einfachen Liste der verfügbaren Dokumente, die Foozle publiziert hat und die technische Fragen der Herstellung von Pullovern behandeln. (Sie hätten wohl nicht geglaubt, dass es so etwas gibt, nicht wahr?). Jeder Link dieser Seite bringt Sie zu dem Dokument mit den entsprechenden technischen Informationen.

Von hier aus könnten Sie sich die Hierarchie herunterbewegen und nun jedes der Dokumente lesen oder zurück zur Homepage gehen. Das Zurückgehen würde Ihnen erlauben, die anderen Bereiche dieser Webseite zu erkunden, die Sie noch nicht besucht haben: einen Überblick über die Firma, Produktbeschreibungen oder eine Liste mit offenen Stellen.

Die Company Overview Seite

Diese Seite bietet eine Link-Liste zu anderen Seiten an, die Ihnen zusätzliche Informationen über die Firma liefern. Wählen Sie diese Seite, können Sie mehr über die Firma erfahren, was sie tut, welche Mission sie verfolgt, den Werdegang der Firma, wo der Hauptsitz liegt, usw... Eine solche Seite sehen Sie in Bild 17.5.

Die Product Seite

Wählen Sie das entsprechende Link in der Navigationsleiste, gelangen Sie auf die Fozzle Produktseite. Von dort können Sie zu den Katalogseiten gelangen (ähnlich wie das Shopping-Katalog-Beispiel, das etwas später in diesem Kapitel gezeigt wird). Diese Seite ist für Bilder, Beschreibungen und Preisinformationen für alle Produkte der Firma gedacht. Die Produktseite sehen Sie in Bild 17.6.



Abbildung 17.4:
Der Technik
Informationsbereich



Abbildung 17.5:
 Die »Company
 Overview«-Seite

Die Product Support Seite

Ein Kunde kann viele Fragen bezüglich der Pflege und des Gebrauchs von FOOZLE Produkten haben. FOOZLE Industries Inc. hat eine produktunterstützende Seite eingerichtet, die Sie auf Bild 17.7 sehen können. Diese Seite enthält eine Liste oft gestellter Fragen bezüglich der Produkte dieser Firma. Sollte eine Kundenfrage auf dieser Seite nicht beantwortet werden, so kann sich der Kunde, mittels der angezeigten E-Mail-Adresse, direkt an den Kundendienst dieser Firma wenden.

Von der produktunterstützenden Seite aus, können Sie auf jede andere Seite gelangen und, je nach Interesse, eine der Hauptseiten der Web-Site wählen.

Die Open Opportunities Seite

Ein modernes Unternehmen wie FOOZLE, hält ständig Ausschau nach talentierten Mitarbeitern und ist immer dankbar für Anfragen, selbst dann, wenn momentan keine Stelle frei ist. Bild 17.8 zeigt eine Seite, die Stellenangebote auflistet und Kontaktinformationen für diejenigen liefert, die eine Anstellung innerhalb der Firma suchen.

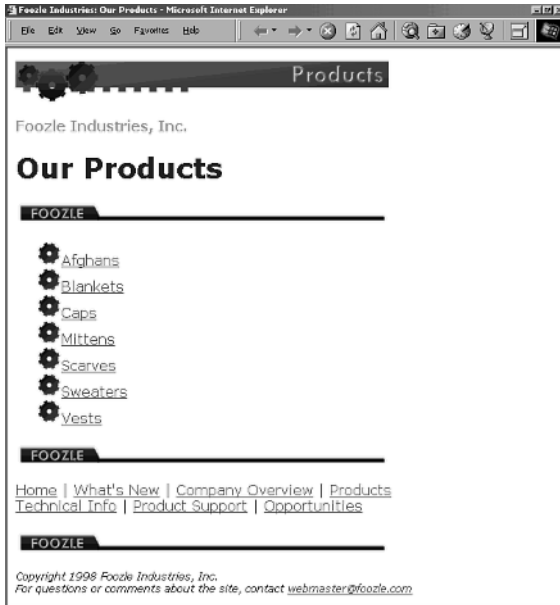


Abbildung 17.6:
Die Fozzle Products Seite



Abbildung 17.7:
Die Product Support Seite

Features dieser Web-Site und einige Worte zur Entwicklung

Das Firmenprofil dieser Web-Site ist bezüglich des Designs ziemlich streng gehalten. Die Struktur entspricht einer einfachen Hierarchie mit Linkmenüs, um zu den entsprechenden Seiten navigieren zu können. Sie können die Navigationsleiste auch oben auf der Seite anbringen, oder aber auch ober- und unterhalb der Seite, falls der Leser nicht die gesamte Seite lesen sollte. Das Navigationssystem ist eine einfache Angelegenheit. Man muss der Hierarchie nur einige »Glieder«, also neue Links, in den oberen Teil der Seite hinzufügen.



Abbildung 17.8:
Die Open Opportunities-
Seite

Achten Sie auf den Pfad, den Sie durch die wenigen Seiten dieser Site genommen haben. In einer klassischen Hierarchie, besuchen sie ein Element nach dem anderen, arbeiten sich nach unten durch und gehen dann zurück, um neue Seiten zu besuchen. Erinnern Sie sich noch an das Link zwischen der »Was gibt es Neues?«- und der »Alpaca Wool«-Seite? Dieses Link ließ Sie sich seitwärts, von einem Element (die »Was gibt es Neues?«-Seite) zum nächsten (die technische Sektion) bewegen. Mit einer anderen Navigationsleiste, mit der Sie zur Homepage navigieren können und, je nachdem, hoch oder seitwärts zu anderen technischen Seiten gelangen, können Sie auch leicht Ihren Weg zurück zu anderen wichtigen Sektionen im Web finden.

In diesem Beispiel, das doch relativ einfach ist, gibt es wenig Anlass zu Verwirrung. Ist eine Hierarchie nun komplizierter, mit vielen Ebenen und Teilnavigationsbäumen, dann können Links, die hierarchische Grenzen durchbrechen durchaus verwirrend sein. Nach einigen querverweisenden Links kann es schwierig werden herauszufinden, in welcher Hierarchie Sie sich gerade befinden. Hier handelt es sich um ein weit verbreitetes Problem mit den meisten Hypertext-Systemen und ist allgemein unter dem Begriff »verloren im Hyperspace« bekannt.

Aber auch hierfür gibt es einige Lösungen. Eine gängige Lösung sind Framesets, die Ihnen die Navigation in komplexeren Sites erleichtert. Allerdings muss man auch hier mit dem Design vorsichtig sein. Zu viele Frames bieten nicht nur Anlass zu Verwirrung, sondern erschweren auch das Lesen der Seiten mit niedrigeren Auflösungen. Um diesem Problem aus dem Weg zu gehen, dürfen Sie in einer Hierarchie nicht zu viele querverweisende Links verwenden. Wenn Sie bei einer »einfachen« Hierarchie bleiben und nur Navigations-Links verwenden, findet sich der Leser normalerweise zu recht. Sollte dies nicht der Fall sein, gibt es da grundsätzlich zwei Lösungen: Gehen Sie in der Hierarchie so lange zurück, bis Sie an einen bekannten Punkt gelangen, oder gehen Sie in der Hierarchie weiter, um detailliertere Informationen zu erhalten.

Beispiel zwei: eine Multimedia-Enzyklopädie

Die Multimedia-Enzyklopädie der Motorräder besteht aus einigen Webseiten, die ausführliche Informationen über Motorräder und ihre Hersteller bieten. Es ist die umfangreichste Sammlung im Web und enthält Fotografien, Klänge (Motorengeräusche!) und Filme von vielen der aufgeführten Motorräder.

Der Index ist alphabetisch organisiert, mit einer Seite pro Buchstabe (A.html, B.html usw.). Die Homepage dieser Präsentation hilft bei der Navigation durch die Enzyklopädie.

Die Überblicks-Seite

Die Überblicks-Seite ist der Haupteintrittspunkt in die Enzyklopädie (Abbildung 17.9).

Diese Seite bietet zwei Wege an, um die Enzyklopädie zu benutzen: durch Anwahl des Anfangsbuchstabens einer Marke, die Sie interessiert, oder indem Sie eine der spezifischen Marken anwählen, die in der Liste selbst aufgeführt werden.



Motorrad- und Sportauto-Fans sprechen von »Marque«, wenn Sie auf einen Hersteller verweisen wollen.

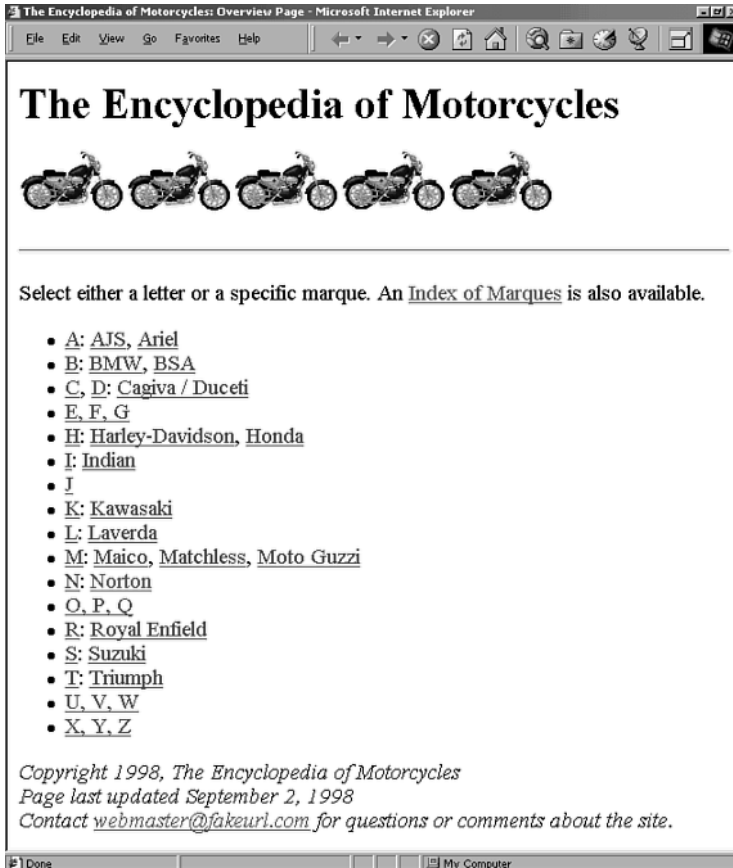


Abbildung 17.9:
Die Überblicks-Seite der Motorrad-Enzyklopädie

Wollten Sie also beispielsweise Informationen über die Motorradfirma Norton haben, so könnten Sie **N** für Norton anwählen und dann bis zu dem gewünschten Eintrag hinunterblättern. Da Norton indes einer der wichtigsten Hersteller und neben der **N**-Verbindung aufgeführt ist, könnten Sie stattdessen diese Verbindung anwählen und so direkt zu dem Eintrag für Norton gehen.

Der Eintrag für Norton

Jede individuelle Seite enthält Einträge für alle Marken, die mit dem jeweiligen Buchstaben beginnen. Wenn der Leser einen bestimmten Hersteller ausgewählt hat, führt die Verknüpfung direkt zu diesem Eintrag (z.B. zum Eintrag für Norton, der in Abbildung 17.10 gezeigt wird). Jeder Eintrag enthält Informationen über die Marke und die verschiedenen Motorräder, die sie in den letzten Jahren produziert hat.

The Encyclopedia of Motorcycles: Norton - Microsoft Internet Explorer
File Edit View Go Favorites Help

The Encyclopedia of Motorcycles

[Overview](#) | [Index of Marques](#)

Norton

The Norton company began in 1902 when James Lansdowne Norton put a French Clement engine into a frame of his own design. In 1907 Norton designed his own engine and competed on it in the TT races of 1909-1911.

Although Norton himself died in 1925, the company's 350cc and 500cc single-cylinder motorcycles went on to win several TT races in the 20's and 30's. Norton met further success in the late 40's due to the development of the 500cc Dominator Twin engine.

The Dominator slowly increased in capacity throughout the 50's and 60's -- from 500cc to 600, 650, and finally 750cc. The Vandervell family acquired the company from Norton, and it later changed hands to AMC, who also owned Matchless and AJS. Following this, the company was acquired by the Manganese Bronze Holding Company, who focused on the 750cc version of the Dominator and renamed it the Atlas.

The Norton Commando was released in 1967, with a new frame that promised to cut down on vibration. The Commando, in 750cc and 850cc models, was to become the flagship model of the Norton Motorcycle Company for ten years.

In the mid-70's the Norton Company ran into financial trouble, and was merged with the Triumph/BSA company to form NVT (Norton-Villiers-Triumph). Since then, the company has struggled along under several different owners. Other than an interesting foray into motorcycles powered by Wankel rotary engines, the company has produced few motorcycles.

Norton Media

The following media files are available on Norton products. Where multiple links appear beneath a description, they are different versions of the same file. Select a version that is compatible with your browser.

Line drawing of the 1926 490cc OHV Engine
[\(49K GIF File\)](#)

850 Commando
[\(46K JPEG File\)](#)
[\(120K GIF File\)](#)

850 Commando Engine
[\(350K AU Sound File\)](#)
[\(479K WAV File\)](#)

3D Rotational Rendering of a Norton Motorcycle

[\(289KB Animated GIF\)](#)
[\(194KB Windows AVI\)](#)
[\(190KB Quick Time Video\)](#)
[\(151KB MPEG-1 Video\)](#)

Norton Manx winning a race in 1938
[\(768K Quicktime Video\)](#)

Copyright 1998, The Encyclopedia of Motorcycles
Page last updated September 2, 1998
Contact webmaster@fakeurl.com for questions or comments about the site.

Abbildung 17.10:
Der Norton-Eintrag

Wo sind nun aber die Bilder? Wir hatten doch angenommen, dies sei eine Multimedia-Enzyklopädie, nicht wahr? Nun, zusätzlich zu dem Text, der Norton selbst beschreibt, enthält der Eintrag eine Liste externer Medien-Dateien. Die Medienliste befindet sich in der rechten Tabellenseite. Da die Hintergrundfarbe der Zelle, die diese Links enthält, eine andere ist, sind die Media-Links visuell vom Text der Seite isoliert. Somit wird verdeutlicht, dass diese Links zu eigenständigen Beiträgen verweisen, die aber dennoch mit der Firma Norton in Zusammenhang stehen. Die Medien Sektion enthält Bilder verschiedener Motorräder, Sound Clips und Filme bekannter Fahrer von Norton Motorrädern.

Jede Medien-Datei ist im Text beschrieben und so verknüpft, dass Sie sie laden können, wenn Sie es mögen. Beispielsweise greift die Verbindung der 850 Commando auf eine JPEG-Abbildung dieses Modells zu (Abbildung 17.11).

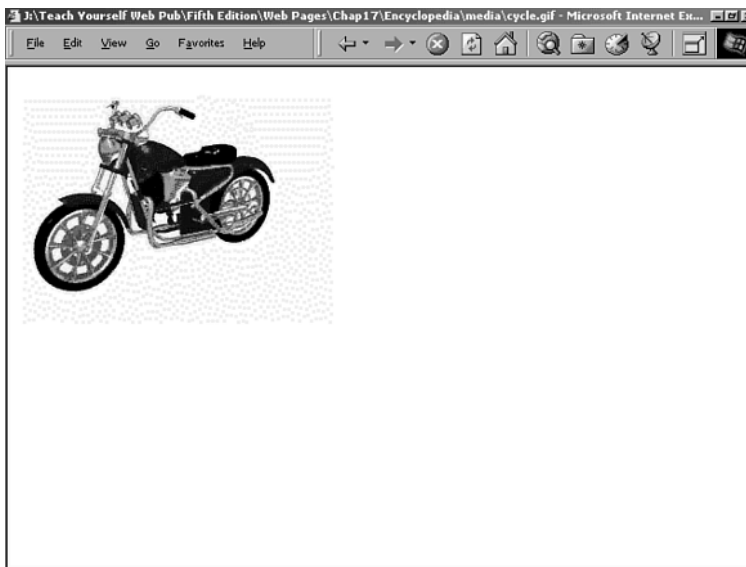


Abbildung 17.11:
Ein Norton
Motorrad, das in
3D angezeigt wird

Achten Sie auch darauf, dass an jenen Stellen im Text, an denen ein anderer Hersteller erwähnt wird, dieser Hersteller mit seinem eigenen Eintrag verbunden ist (Abbildung 17.12).

Sind Sie dann einmal mit dem Blättern fertig, ist es immer wichtig, zu einem bekannten Punkt zurückkehren zu können. Zu diesem Zweck enthält jeder Eintrag in der Enzyklopädie eine »Zurück-zum-Überblick«-Verknüpfung. Die Vervielfältigung dieser Verknüpfung innerhalb eines jeden Eintrags bedeutet, dass der Leser niemals weit zum Anfang oder Ende der Seite blättern muss, um eine Verknüpfung zurück zur Zeile zu finden.

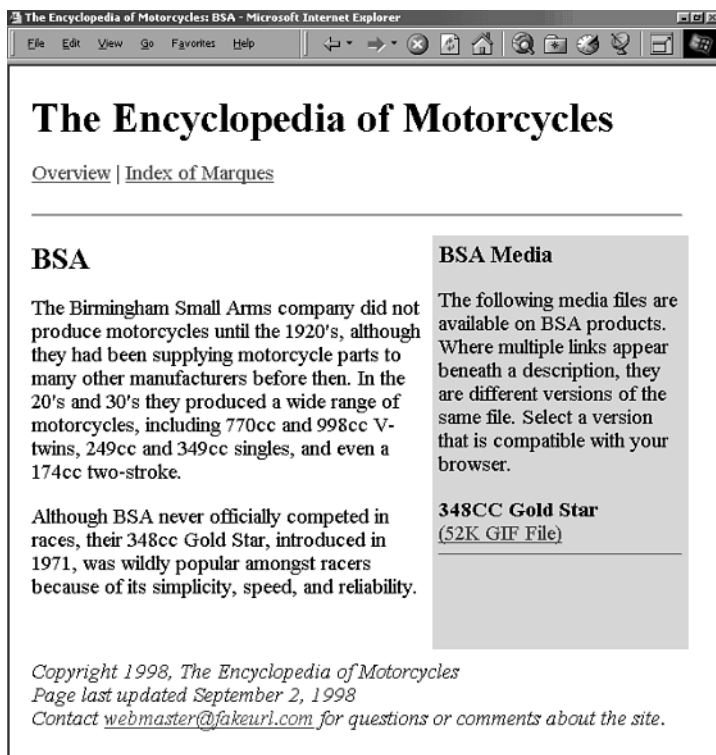


Abbildung 17.12:
Der Eintrag für BSA

Der Marken-Index

Zurück auf der Hauptseite, finden wir noch ein Merkmal, auf das ich hinweisen möchte: Der Überblick enthält auch eine Verbindung zum Marken-Index, einem Verzeichnis aller Hersteller von Motorrädern, die in der Enzyklopädie erwähnt werden (Abbildung 17.13).

Jeder Name im Index ist, wie Sie wohl erwarten, eine Verbindung mit dem Eintrag für diesen Hersteller in der Enzyklopädie selbst, wodurch Ihnen einfach ein anderer Weg offensteht, sich schnell durch das alphabetische Verzeichnis zu bewegen. Eine zusätzliche Erweiterung des Index erscheint im oberen Teil der Seite. Mit der »Springe zu«-Liste der Links müssen die Leser nicht mehr durch das gesamte Index scrollen und haben einen schnellen Zugang zu den einzelnen Marken, die mit dem jeweiligen Buchstaben beginnen. Wenn sie nun zum Beispiel die »O«-Sektion im »Springe zu«-Link wählen, springt der Leser direkt zum »O«-Anker der Seite. Ein anderer Link am Ende der »O«-Sektion (ähnlich dem, das am Ende der »A«-Sektion in Bild 17.13 gezeigt wird), bringt den Leser zum Seitenanfang zurück, wo er oder sie dann einen anderen Buchstaben wählen kann.

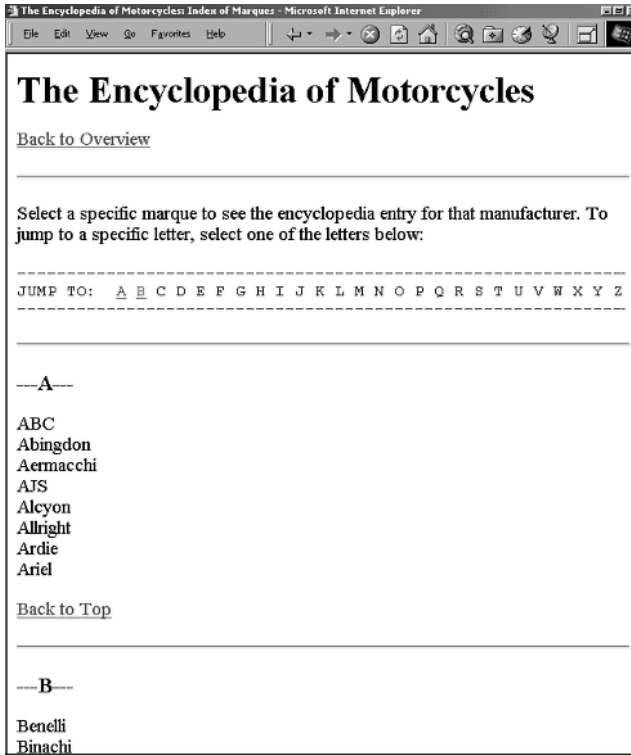


Abbildung 17.13:
Der Marken-Index

Besonderheiten dieser Web-Site und Gesichtspunkte der Entwicklung

Wohl das beste Merkmal bei der Gestaltung dieser Enzyklopädie ist die Überblicks-Seite. In vielen Fällen würde eine Online-Enzyklopädie dieser Art Verbindungen von jedem Buchstaben des Alphabets anbieten und es dabei belassen. Wenn Sie nun etwas über Norton-Motorräder lesen wollten, so würden Sie das »N« anwählen und dann hinunterblättern zu Norton. Indem er die Überblicks-Seite selbst mit Verbindungen zu einigen der beliebteren Motorradhersteller ausstattet, bietet der Autor dieser Webseite eine schnelle Referenz, durch welche die Zeit fürs Blättern abgekürzt wird und die seine Leser direkt dahin bringt, wohin sie wollen.

Das Hinzufügen eines Marken-Index ist gleichfalls ein netter Zug, ermöglicht er es den Lesern doch, direkt zum Eintrag eines bestimmten Herstellers zu springen – auch hier, um die Zeit zu verringern, die benötigt wird, um zu finden, was man sucht. Es ist der gleiche Inhalt wie in der Enzyklopädie. Die Übersichts-Seite liefert den Lesern einfach verschiedene Möglichkeiten, die gesuchte Information zu finden.

Die Enzyklopädie selbst ist nach einem locker geknüpften Gewebemuster strukturiert, das es den Lesern ermöglicht, nahezu überall hineinzuspringen und dann den Querverweisen zu folgen und so durch die verfügbaren Informationen zu streifen, wobei Sie Verbindungen zwischen Motorrädern und Marken und Motorradgeschichte aufdecken können, die in einer gedruckten Enzyklopädie nicht so einfach herauszufinden wären. Indem er alle zusätzlichen Medien extern anbietet, erlaubt es der Autor dieser Web-Site nicht nur, die Enzyklopädie mit grafischen wie textbasierten Browsern gleichermaßen gut zu nutzen., Beachten Sie schließlich, dass jede Marken-Seite (wie Norton und BSA in Abbildung 17.10 und 17.12) eine Verbindung zur Überblicks-Seite enthält. Gäbe es mehr als diese Verbindungen, so würde dies die Seite überhäufen und schlecht aussehen. Da indes die einzige ausdrückliche Navigationsrichtung nach oben führt, ermöglichen diese zwei Verbindungen den Lesern, schnell und einfach aus der Enzyklopädie zurückzukommen, anstatt zum Anfang oder Ende des Dokuments blättern zu müssen.

Der wichtigste Gesichtspunkt beim Entwickeln einer Web-Präsentation dieser Art ist der Aufbau und die Pflege. Je nach der Menge des Materials, die Sie online zur Verfügung stellen wollen, sollten Sie alle tasks sofort ordnen. (Verwenden Sie genau 26 Dateien, eine für jeden Buchstaben des Alphabets? Oder mehr? Oder weniger?) Ein Kompromiss wäre vielleicht, die Marken mit wenig Information (wie etwa die BSA-Seite) auf der Buchstaben-Seite (`b.html`) zu kombinieren, und die bekannteren Marken, wie zum Beispiel Norton, auf einer eigenen Seite zu kennzeichnen.

Glücklicherweise muss eine Enzyklopädie dieser Art nicht allzuoft überarbeitet werden, sodass die Pflege, nachdem die Anfangsarbeit erledigt ist, nicht mehr so schwer ist.

Beispiel drei: ein Einkaufskatalog

Annas Kräuter- und Gewürze-Garten ist eine gewerbliche Baum- und Pflanzenschule, spezialisiert auf die Aufzucht und den Versand von Kakteen und Sukkulenten. Dort werden über 120 Arten von Kakteen und Sukkulenten sowie Bücher und andere kaktusbezogene Artikel angeboten. Abbildung 17.14 zeigt die Homepage von Annas Kräuter-und Gewürze-Garten.

Auch Anna hat Cascading Style Sheets angewandt, um Hintergrund-, Text- und Link-Farben für Ihre Web-Site zu erstellen. Ihr Firmenname erscheint im Titel jeder Webseite, und in Form einer Grafik mit einem Logo-Bild am Anfang einer jeden Seite. Anna hat sich dazu entschlossen, auf die Hauptseiten, in der Nähe des Seitenanfangs, Links zu platzieren. Somit muss der Leser nicht durch die gesamte Seite scrollen, um dann an eine andere Stelle navigieren zu können. Die Navigationsleiste liefert Links zu den Seiten, die von den Lesern am meisten besucht werden: die Homepage, die Browse-Katalog-Seite, die »Wie bestelle ich?«-Seite und die Bestellseite selbst.



Abbildung 17.14:
 Homepage von Annas
 Kräuter- und Gewürze-
 Garten

Sind die Leser erst einmal auf der Homepage, haben sie verschiedene Möglichkeiten: Den Katalog durchsuchen, Informationen über das Bestellen und die ausgewählten Artikel bestellen.

Im Katalog blättern

Die Anwahl der Verbindung zum Blättern bringt die Kunden auf eine andere Menü-Seite, wo sie verschiedene Möglichkeiten haben, *wie* sie den Katalog durchgehen wollen (Abbildung 17.15).

Hier können die Kunden etwas Hintergrundinformationen über die Gärtnerei selbst lesen, Informationen über Sonderangebote und neue Pflanzen bekommen, den Katalog durchblättern, Informationen über das Bestellen erhalten und die Kräuter oder Gewürze, die sie ausgewählt haben, tatsächlich auch bestellen.

Diejenigen, die sich mit Kräutern und Gewürzen auskennen und ...

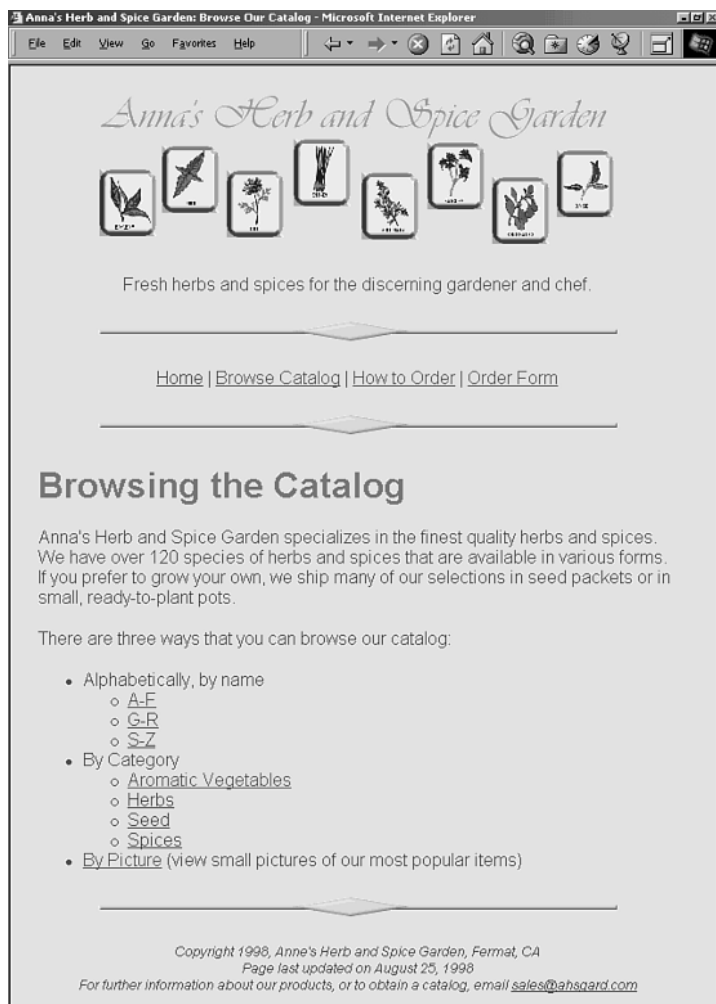


Abbildung 17.15:
Wie im Katalog
geblättert wird

Diejenigen, die einige frische Kräuterpflanzen für ihren eigenen Garten haben möchten, aber noch nicht wissen, welche Kräuter sie wollen, können die Kräuter-Kategorie durchsuchen. Schließlich gibt es noch die Besucher, die keine Pflanzennamen kennen und die eigentlich nur etwas Hübsches kaufen wollen. Für sie gibt es eine Foto-Galerie, die sie besuchen können. Die alphabetischen Verbindungen (A-F, G-R, S-Z) führen die Kunden zu einem alphabetischen Verzeichnis der zum Kauf angebotenen Pflanzen. Abbildung 17.16 zeigt ein Beispiel aus dem alphabetischen Katalog. In diesem Fall werden die Seiten A-F angezeigt.



Abbildung 17.16:
Der Katalog, A-F
alphabetische Ver-
knüpfungen

Ich habe nicht alle Links auf dieser Seite vervollständigt, nehme aber einfach an, dass jedes der Links den Leser zu der Seite bringen wird, die mehr Informationen über spezielle Kräuter oder Gewürze enthält. Wenn der Leser zum Beispiel mehr über *Salbei* erfahren möchte (siehe Abbildung 17.16), wird er oder sie zu einer Seite verwiesen, die dieses Gewürz genauer beschreibt. Siehe hierzu Bild 17.17.



Abbildung 17.17:
Eine Produkt-Seite

Auf jeder Produkt-Seite wird ein kleines Foto der Pflanze gezeigt (hier als Line-Art-Bild für Repräsentationen angezeigt). Neben dem Foto befindet sich der Name der Pflanze in Englisch und Latein. Danach folgt eine kurze Beschreibung über den üblichen Gebrauch, den Geschmack und weitere Informationen, die nennenswert sind. Danach kommt eine Liste, die dem Leser anzeigt, wie man die Pflanze bestellen kann und wieviel sie kostet.

Wenn der Leser eines der Links in der Kategorie-Sektion der Katalog-Seite anklickt, gelangt er oder sie zu einer anderen Seite des Menüs. Wenn der Leser, wie in diesem Fall die Kräuter-Kategorie wählt, erscheint die in Bild 17.18 gezeigte Seite.



Abbildung 17.18:
 Ansicht einer Kate-
 gorie

Wenn die Kräuter-Kategorie gewählt wird, wird eine Liste der erhältlichen Kräuter angezeigt. Auch hier habe ich nicht alle Links der Seite vervollständigt, aber jeder Link bringt den Leser zu einer speziellen Produkt-Seite (siehe Beispiel in Bild 17.17). Vom Kategorie-Index aus gelangen Besucher zurück zur Katalog-Seite. Auf der Katalog-Seite kann der Besucher die Produkte auch in einer Foto-Galerie sehen.

Die Foto-Galerie bietet dem Kunden die Möglichkeit, eine bestimmte Pflanze zu suchen, ohne dass er dabei den Namen kennen muss. Hat ein Kunde einmal eine Pflanze in Nachbars Garten gesehen und kann sich nicht mehr an den Namen erinnern, ist die Foto-Galerie die ideale Stelle, nach dieser Pflanze zu suchen. Dieses Feature ist natürlich nur für grafische Browser gedacht.

Bestellungen

Nachdem die Kunden das Durchblättern des Katalogs abgeschlossen haben und wissen, welche Art von Kaktus sie bestellen wollen, können sie zurück zur Homepage von Annas Kräuter- und Gewürze-Garten springen, um herauszufinden, wie sie bestellen können.

Die Seite zum Bestellen ist einfach etwas Text (Abbildung 17.20): Informationen darüber, wo man anrufen und wohin man Schecks senden kann, Tabellen mit Versandkosten, Hinweise dazu, wann die Pflanzen verschickt werden und so weiter.

Im Abschnitt zum schriftlichen Bestellen findet sich eine Verbindung zum Bestellformular. Die Bestellung können Sie auf verschiedene Art und Weise erstellen. Wenn Sie sich nicht mit dem Erstellen von Formularen und dem Schreiben von Form-Handlern herumschlagen wollen, können Sie eine einfache Textdatei erstellen, die ein Bestellformular enthält. Kunden können das Bestellformular herunterladen oder in ihren Browsern anzeigen (siehe Bild 17.21). Sie können dieses Formular ausdrucken, es ausfüllen und es der Gärtnerei zuschicken.

Es gibt verschiedene Wege, externe Versionen dieses Dokuments anzubieten. Sie können zum Beispiel eine Postscript-Datei, eine Adobe-Acrobat-Reader-Datei (mit der Endung .PDF) oder sogar ein Textverarbeitungsformat, wie etwa Microsoft Word oder Word Perfect anbieten. Speichern Sie diese Dateien, wie jede andere Seite auch, in Ihrer Web-Site, und fügen Sie diesen Dateien Links hinzu. Wenn der Leser keine Helfer-Applikation besitzt, die dieses Dokument als eine Webseite anzeigen kann, wird er die Seite unverzüglich herunterladen wollen.

Sie können das Bestellformular auch Online gestalten. Ein Beispiel hierzu sehen Sie in Bild 17.22. Allerdings müssen Sie hierfür einiges beachten:

Sollten Sie Online-Bestellungen annehmen, müssen Sie sich darüber im Klaren sein, dass viele Kunden sehr vorsichtig sind, was Informationen ihrer Kreditkarte betrifft. Hier können Sie Ihren Online-Katalog in einen sicheren Server geben, der Kundeninformationen schützt.

Falls sich Ihr Katalog auf keinem sicheren Server befindet, müssen Sie den Kunden darüber informieren und alternative Bestellmethoden anbieten. Bieten Sie eine Telefonnummer an oder verwenden Sie ein Text-Bestellformular, wie in Bild 17.21. Die Kunden können somit ihre Bestellungen auf traditionellere Art und Weise aufgeben.

Eine andere Alternative wäre, dass Ihr Online-Bestellformular alles, außer der Kreditkarten-Information enthält. Wenn Sie eine Bestellung von einem Kunden erhalten, können Sie ihm oder ihr eine Bestätigung dieser Bestellung via E-Mail senden. Auf der Bestätigung sollten sich eine Referenznummer sowie eine Telefonnummer befinden, damit der Kunde sich über die Zahlungsbedingungen erkundigen kann.



Abbildung 17.19:
Die Foto-Galerie-
Seite

Sie möchten wissen ob die Online-Bestellung eines Kunden automatisch in eine Datenbank eingefügt wird. Das ist durchaus im Bereich des Möglichen, verlangt aber eine fortgeschrittene Web-Programmierung und eine gewisse Kenntnis über Datenbank-Strukturen sowie auch über andere Techniken, die in diesem Buch nicht erwähnt wurden.



Abbildung 17.20:
Pflanzen bestellen

Kehren wir zu Annas Kräuter- und Gewürz-Garten-Site zurück. Der dritte Punkt dieser Homepage ist ein direkter Link zur Bestellformular-Datei. Dieser Link wird schon an dieser Stelle angezeigt. Somit muss der Leser nicht wieder zur Bestell-, Versand- und Bezahl-Seite zurücknavigieren.

O R D E R F O R M

SHIP TO: _____

QTY	ITEM NO.	DESCRIPTION	PRICE EA.	TOTAL
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____

SUBTOTAL _____

CALIFORNIA RESIDENTS PLEASE ADD SALES TAX _____

SHIPPING:
 For orders \$15.00 or less, add \$3.00
 For orders \$15.01 to \$30.00, add \$5.00
 For orders \$30.01 to \$50.00, add \$7.00
 For orders \$50.01 to \$100.00, add \$10.00
 For orders \$100.00 and up, add \$12.00 _____

TOTAL _____

Please enclose check or money order for the above amount,
 or provide your credit card number, expiration date,
 and signature below. Allow 2-4 weeks for delivery.

CREDIT CARD NO. _____ EXP. DATE _____ SIGNATURE _____

Mail order to:
 Anna's Herb and Spice Garden
 2374 Sunrise Lane
 Fermet, CA 93481

Abbildung 17.21:
Text des Bestellformulars

Besonderheiten und Gesichtspunkte der Entwicklung dieser Web-Site

Bei jedem Online-Einkaufsangebot bestehen die Ziele der Präsentation darin, es den Kunden zu ermöglichen, die zum Kauf angebotenen Dinge durchzublättern und diese dann zu bestellen. Beim Blättern ergeben sich wiederum verschiedene Anforderungen: Was ist, wenn ein Leser eine bestimmte Sache möchte? Kann sie schnell und einfach gefunden werden? Was ist, wenn jemand einfach nur die zum Verkauf stehenden Dinge durchsehen möchte, bis er etwas Interessantes findet?

Diese beiden Anforderungen beim Blättern in einem Online-Verzeichnis scheinen einander zu widersprechen, doch in diesem besonderen Beispiel wurde es ziemlich gut gehandhabt – mit Hilfe der unterschiedlichen Zugangsmöglichkeiten zum Inhalt des Katalogs (alphabetisch, über Kategorien, über Bilder). Diese mehrfachen Zugangswege sind in gewisser Hinsicht Umwege (ein zusätzliches Menü zwischen der obersten Ebene und dem Inhalt des Katalogs selbst), doch dieser kleine Umweg ist eine Verzweigung, die den unterschiedlichen Typen von Kunden hilft, ihre jeweiligen Absichten zu verwirklichen.

Ich habe absichtlich Einiges in dieser Site ausgelassen, um Ihnen demonstrieren zu können, wie man die Navigation verbessern kann. Katalog-Browsing, zum Beispiel, könnte erweitert werden, wenn man für jede Methode, mit der man den Katalog besuchen kann, ein eigenes Menü erstellt. Nehmen wir das erste Site-Beispiel dieses Kapitels, in der die Hauptseite eine Navigationsleiste hat, die auf alle Hauptseiten im Web verweist (siehe Bild 17.1). Beim Navigieren zu dem Bereich mit technischen Informationen, fanden Sie eine weitere Navigationsleiste vor, die auf Seiten mit den technischen Informationen untereinander verknüpfte (siehe Bild 17.13). Ein ähnliches System würde Annas Katalog sehr gut erweitern.

Für Anna ist es jedoch am wichtigsten, mit Ihrer Site den Umsatz zu steigern. Wie könnte Sie hierfür Ihre Navigation verbessern? Am besten ist es, ein Link der auf die »Wie bestelle ich?«-, zur »Bestellformular«-Seite oder auf beide Seiten verweist. Das sollte sie auf jeder Seite, die Informationen Ihrer Produkte enthält, tun. Sollten zum Beispiel Leser zur Salbei-Seite (siehe Bild 17.17) kommen und sich dazu entschließen, diese Pflanze zu kaufen, könnte er oder sie bis dahin vergessen haben, dass die »Wie bestelle ich?«-Seite direkt zum Bestellformular führt. Geben Sie einen Bestell-Link also gleich unter die Preisinformationen – ein Klick und der Kunde gehört Ihnen.

Vermutlich der härteste Teil bei Aufbau und Pflege von Webseiten wie diesen ist die Pflege des Katalogs selbst, besonders, wenn einzelne Einträge hinzugefügt oder entfernt werden müssen oder wenn die Preise häufig geändert werden. Hätte die Pflanzenschule nur eine Katalog-Ansicht (die alphabetische), so wäre das nicht so schlimm, weil die Änderungen direkt in den Katalog-Dateien durchgeführt werden könnten. Mit zusätzlichen Ansichten und den Verknüpfungen zwischen ihnen wird die Pflege des Katalogs jedoch wesentlich aufwendiger.

Diese Art der Einteilung wäre vermutlich am leichtesten zu pflegen, wenn der Katalog tatsächlich in einer Art Datenbank außerhalb der eigentlichen Webseiten gespeichert wäre. Datenbanken sind darauf konzipiert, mit solcher Art Information umzugehen und auf Abruf verschiedene Ansichten herzugeben. Wie aber verbinden Sie eine Datenbank mit den Webseiten?

Anna's Herb and Spice Garden - Order Form - Microsoft Internet Explorer

File Edit View Go Favorites Help

Anna's Herb and Spice Garden



Order Form

Ship to:

Name:

Street:

City, State, Postal Code:

Country:

Items Ordered:

Qty	Item No	Description	Price Ea.	Total

Subtotal:

California Residents,
please add sales tax:

Shipping, for orders:
 \$15.00 or less, add \$3.00
 \$15.01 - \$30.00, add \$5.00
 \$30.01 - \$50.00, add \$7.00
 \$50.01 - \$100.00, add \$10.00
 \$100.00 and up, add \$12.00

Total

Credit Card Information:

Type	Number	Expiration Date
Visa		

You will receive a confirmation of your order by email. Enter your email address here:

To submit your order, press the Submit button below. Allow 2-4 weeks for delivery. Thank you for shopping at Anna's Herb and Spice Garden.

Abbildung 17.22:
Ein Online-Bestellformular

Wenn Sie über genügend Programmierfähigkeiten (und Datenbank-Kenntnisse) verfügen, könnten Sie ein Programm erstellen, das Datenbank Suchfragen über eine Webseite laufen lässt, die dann in Form formatierter Produktlisten zu Ihnen zurückkommen. Wird auf der Webseite eine alphabetische Auflistung verlangt, werden Sie automatisch eine generierte Liste erhalten, die auf dem neuesten Stand der Datenbank ist. Hierfür brauchen Sie allerdings eine Datenbank, die mit Ihrem Web-Server kommunizieren kann. Je nachdem mit welchem System Ihr Web-Server läuft, ist es, oder auch nicht, technisch durchführbar. Sie müssen auch über die nötigen Programmierfähigkeiten verfügen: wie etwa CGI, Java, JavaScript, dynamische HTML-Generierung oder andere Möglichkeiten. Intensiver auf diese Möglichkeiten einzugehen, würde den Rahmen dieses Buches sprengen. Mehr Information hierzu erhalten Sie in der einschlägigen Literatur.

Eine alternative Lösung könnte darin bestehen, die Daten in der Datenbank zu lassen und sie dann ab und zu in Text umzuwandeln, der dann wiederum zu HTML konvertiert wird. Die Hauptschwierigkeit ist bei dieser Lösung natürlich, wieviel Arbeit die Konvertierung jedesmal bereitet, wenn gleichzeitig die Querverweise zu den anderen Seiten aufrechterhalten werden sollen. Könnte dieser Vorgang automatisiert werden, und wieviel Anfangsarbeit und tägliche Aufrechterhaltung wären erforderlich?

Bei dieser Art von Anwendung müssen Sie mit diesen Arten von Fragen und technischen Herausforderungen rechnen, wenn Sie Web-Präsentationen erstellen. Manchmal fordert ein Problem halt mehr, als nur Informationen für den Bildschirm zu konzipieren, zu schreiben und zu formatieren.

Zusammenfassung

Ich habe hier nur eine Handvoll Ideen für die Verwendung und Strukturierung von Webseiten präsentiert; unbegrenzt sind die Variationen dieser Themen für die Webseiten, die Sie gestalten werden.

Vermutlich der beste Weg, um Beispiele zu finden für die Art von Webseiten, die Sie gestalten möchten, und wie sie aufgebaut werden, ist, ins Web hinauszugehen und dort herumzustöbern, was es so gibt. Richten Sie, während Sie blättern, Ihr Augenmerk neben der Beurteilung von Anordnung und Gestaltung einzelner Seiten und des Inhalts, den sie beschreiben, darauf, welche Struktur die Leute benutzt haben, um ihre Seiten zu organisieren, und versuchen Sie zu erraten, weshalb sie sich für die jeweilige Organisation entschieden haben. (»Die haben sich überhaupt nichts gedacht«, ist unglücklicherweise ein häufiger Grund für viele erbärmlich organisierte Webseiten.) Kritisieren Sie anderer Leute Webseiten unter dem Gesichtspunkt der Struktur: Ist es leicht, sich in ihnen zu bewegen? Haben Sie sich verirrt? Kommen Sie leicht von einer beliebigen anderen Stelle in der Präsentation auf eine bekannte Seite zurück? Wenn

Sie für diese Präsentation ein Ziel gehabt hätten, haben Sie es erreicht, und wenn nicht, wie hätten Sie die Präsentation umorganisiert?

Das Lernen aus den Fehlern anderer Leute und zu sehen, wie andere Leute schwierige Probleme gelöst haben, kann Ihnen dabei helfen, Ihre eigenen Webseiten besser zu machen.

Workshop

Heute haben Sie gelernt, wie man das Design und die Funktionalität Ihrer Webseiten verbessern kann. Wie immer, dient der Workshop dazu, Ihnen die wichtigsten Punkte in diesem Kapitel noch einmal ins Gedächtnis zu rufen.

Fragen und Antworten

F Diese Web-Präsentationen sind echt toll. Wie lauten ihre URLs?

A *Wie ich zu Anfang des Kapitels bemerkte, sind die Web-Präsentationen, die ich hier beschrieben habe, Attrappen möglicher Web-Präsentationen, die es geben könnte (und Sie finden diese Attrappen auf der CD-ROM, die diesem Buch beiliegt). Obwohl viel von der Gestaltung und Organisation, die ich hier benutzt habe, durch real existierende Webseiten inspiriert wurde, gibt es diese Seiten nicht im Web.*

F Die Beispiele hier haben eine Form hierarchischer Organisation benutzt. Sind Hierarchien so gebräuchlich, und muss ich sie benutzen? Kann ich nicht auch etwas anderes machen?

A *Hierarchien sind außerordentlich gebräuchlich im Web, doch das heißt nicht, dass sie schlecht sind. Hierarchien sind ein ausgezeichneter Weg, seinen Inhalt zu organisieren, besonders, wenn die Informationen, die man präsentiert, schon von Haus aus zu einer hierarchischen Organisation neigen.*

Gewiss können Sie irgendwas anderes nehmen, um Ihre Präsentation zu organisieren. Doch die Schlichtheit von Hierarchien erlaubt es, sie schnell zu strukturieren, einfach zu durchmessen und leicht zu pflegen. Warum sich und den Lesern mehr Arbeit machen und den Versuch unternehmen, eine komplizierte Struktur über ansonsten einfache Informationen zu stützen?

Quiz

1. Wie können Sie dem Leser helfen, sich besser auf Ihrer Web-Site zurechtzufinden?
2. Welche vorbeugenden Maßnahmen können Sie treffen, damit der Leser in Ihrer Web-Site nicht verloren geht?
3. Welches sind die besten Lösungen für die Entwicklung einer Site, wenn diese Site viele Informationen enthält (zum Beispiel eine Online-Enzyklopädie)?
4. Welches sind die Vorteile, wenn man viele Browsing-Methoden durch einen Einkaufskatalog anbietet?
5. Was ist bei der Annahme von Bestellungen über das Web am wichtigsten?

Antworten

1. Stellen Sie Navigationsleisten zur Verfügung, die auf die Hauptseiten in Ihrem Web verweisen. Für gewöhnlich befinden sich diese Navigationsleisten in jeder Seite an der gleichen Stelle. Navigationsleisten können oberhalb, unterhalb, oder auf der linken Seite einer Webseite positioniert werden. Eine »Oft gestellte Fragen«- und eine »Was gibt es Neues?«-Seite, sowie eine Inhaltstabelle sind ebenfalls Links zu Seiten Ihrer Site.
2. Vermeiden Sie querverweisende, Hierarchie-durchlaufene Links, und bleiben Sie bei einer rigiden hierarchischen Struktur. Der Leser wird sich somit gut in Ihrer Site zurechtfinden können.
3. Bei Webseiten, die viele Informationen enthalten, ist es am schwierigsten die Organisation und die Anordnung der Informationen festzulegen. Außerdem stellt das Erstellen von Medien und der querverweisenden Links zu Seiten einen enormen Zeitaufwand dar. Da es jedoch eher unwahrscheinlich ist, dass sich der Inhalt einer Enzyklopädie ändert, wird die Wartung dieser Site nicht sehr zeitaufwendig sein.
4. Einen Online-Einkaufskatalog einzurichten, der Produkte in verschiedenen Kategorien aufzeigt (Produkttyp, Produktnummer, usw) helfen dem Leser, Ihre Produkte leichter zu finden. Der Logik entsprechend, sollten Sie mit dem Organisieren Ihrer Produkte in Kategorien beginnen. Sind Ihre Produkte dem Kunden bekannt, werden sie sicherlich den Vorteil, den das Suchen nach Produktnamen oder -nummern begrüßen. Andere wiederum bevorzugen es, sich bei der Suche nach dem Produkt, sich am Aussehen des Produkts zu orientieren (Farbe, Größe, usw).

5. Die sichere Verwalten von Kundeninformationen, wenn die Kunden Online bestellen, ist ein wichtiger Punkt. Viele Kunden möchten zum Beispiel keine Kreditkartenauskünfte Online geben. Daher sollten Sie alternative Bestellmöglichkeiten zur Verfügung stellen und versuchen, Ihre Bestellformulare in einem sicheren Server zu lokalisieren.

Übungen

1. Nehmen Sie die Motorrad Enzyklopädie als Vorlage und planen oder erstellen Sie ein Frameset mit zwei oder drei Frames. Wie würden Sie ein Interface erstellen, das den Leser leichter von Sektion zu Sektion navigieren lässt?
2. Sie können einige der Navigations- und Designtipps dieses Kapitels in Ihrer eigenen Homepage anwenden. Foozle Industries Inc. hat einige gute Beispiele für Navigationsleisten und Methoden, mit denen auf andere Seiten verwiesen werden kann. Die Motorrad Enzyklopädie enthält gute Tipps, wenn es darum geht, bestimmte Themen zu verknüpfen. Wenn Sie Bilder oder andere Medien für Beschreibungen verwenden wollen, sollten Sie sich ebenfalls auf diese Enzyklopädie beziehen.



**Willkommen in
der Wirklichkeit:
Web-Design für
den tatsächlichen
Gebrauch**

**Woche
3**

Bis jetzt haben Sie ja schon einige Hinweise für das Planen einer Web-Site oder für die Gestaltung der Seiten bekommen. Außerdem haben wir Ihnen gezeigt, was man dabei tun oder unterlassen sollte und was eine gute Web-Site ausmacht. Da gibt es aber noch einen wichtigen Faktor, den Sie in Betracht ziehen müssen, nämlich wie die Webseite für den tatsächlichen Gebrauch aussehen soll.

In Wirklichkeit sieht es nämlich so aus: Es gibt viele Anwender mit verschiedenen Betriebssystemen, die unterschiedliche Browser verwenden. Eine Tatsache, die wir bis jetzt außer Betracht gelassen haben, ist, dass Anwender unterschiedliche Präferenzen und mehr oder weniger große Erfahrung im Umgang mit Computern oder mit dem Internet haben. Wenn Sie die Präferenzen der Anwender kennen, können Sie auch dementsprechend Ihre Seiten gestalten. In diesem Kapitel werden Sie lernen, wie man diese Präferenzen ausfindig macht.

- ▶ Dinge, die Sie in Betracht ziehen müssen, wenn Sie versuchen die Präferenzen Ihres Publikums herauszufinden.
- ▶ Wie füge ich Features ein, die neuen Anwendern von Nutzen sind?
- ▶ Verschiedene Methoden, die dem Anwender Ihrer Site hilfreiche Anweisungen geben
- ▶ Ein HTML Code, der die gleiche Webseite in jedem der XHTML 1.0- Spezifikationen aufzeigt (Transitional, Frameset, Strict).

Wie sieht denn der tatsächliche Gebrauch überhaupt aus?

Mit Windows 95 oder 98, Windows NT, Macintosh oder einem ähnlichem System im Internet zu surfen, ist für Sie wahrscheinlich nichts Neues. Sie konnten sich auch sicherlich schon einen ersten Eindruck darüber verschaffen, wie Webseiten, zu welchem Zweck auch immer, auszusehen haben.

Hoffentlich ist Ihnen im Verlaufe dieses Buches bewusst geworden, dass die Webseite, die Sie angezeigt bekommen, nicht jedermann gleich sehen kann. Nicht jeder bekommt im Web die Seite auf die gleiche Art und Weise angezeigt.

Es gibt nämlich so viele verschiedene Computer und unterschiedliche Betriebssysteme, dass dies gar nicht möglich sein kann. Selbst wenn Sie versuchen Ihre Seite für den gängigsten Browser zu erstellen, gibt es da immer noch einen Faktor, den Sie nicht vorhersehen können: Anwender haben nämlich gewisse Vorlieben. Einige dieser Präferenzen möchten wir Ihnen mit folgenden Beispielen illustrieren:

- ▶ Bill arbeitet für das Wirtschaftsmagazin Fortune 100 und hat eine leitende Position in diesem Unternehmen. Seine Firma verfügt über ein eigenes Intranet. Fast alle in der Firma verwenden das gleiche Betriebssystem und den gleichen Browser. Für Bill sollten Webseiten zum größten Teil aus Text bestehen, mit hier und da einmal einem Bildchen, das eine Information hervorheben soll – kurzum, wenig Multimedia und viel Information. Extravagante Webseiten betrachtet er als unangemessen und was das Laden betrifft als unpraktisch.
- ▶ Bills Frau, Susan, hat keine Erfahrung mit Computern, wollte aber schon immer lernen, wie man damit umgeht. In ihrer Freizeit beschäftigt sie sich mit Stammbaumforschung. Sie hat erfahren, dass das Internet diesbezüglich viele Informationen enthält. Außerdem möchte sie ihren Familienstammbaum und ihre Familiengeschichte im Web veröffentlichen. Als sie und ihr Mann zuhause den neuen Computer anschließen, ist sie wirklich aufgeregt, und schon bald will sie von ihrem Mann wissen, ob man nicht mehr auf den Bildschirm »packen« könne, ob man die Größe der Buchstaben nicht verändern könne, wo denn die Bilder wären, warum denn die Musik abgeschaltet wurde, usw...Sie hat jetzt schon eine ganz andere Vorstellung vom Internet als ihr Mann.
- ▶ Bill und Susan haben einen Sohn, der noch zur Schule geht. Tom ist ein eifriger »Zocker« und steht natürlich auf Spezialeffekte. Der Ton wird bis zum Anschlag aufgedreht und je lauter umso besser. Browser »X« ist seiner Meinung nach besser als Browser »Y«, natürlich nur, weil alle seine Freunde Browser »X« für virtuelle online Spiele benutzen. Tom möchte eine Web-Site gestalten, die Tipps und Tricks für eines seiner Lieblings-Online-Spiele enthält.
- ▶ Toms älteste Schwester Jill, studiert Werbedesign und interessiert sich besonders für Skulpturen und Fotografie. Sie beabsichtigt den neuen Computer für Hausaufgaben zu verwenden. Für sie ist das Web also ausschließlich von virtuellem Interesse. Ihre Seiten sollten in True-Color, mit der höchstmöglichen Auflösung gezeigt werden.
- ▶ Dann gibt es da noch die Großeltern, die vor kurzem vom Land zu ihrer Familie gezogen ist. Mit Computern können sie nicht viel anfangen und diese stellen für sie ein absolutes Mysterium dar. Sie würden gerne mit dem Computer umgehen können. Die beiden teilen Susans Interesse der Stammbaumforschung, allerdings sieht Großvater nicht mehr so gut. Er braucht einen speziellen Browser, mit dem er den Text sowohl sehen als auch hören kann.

Die gesamte Familie benutzt den gleichen Computer und das gleiche Betriebssystem. Alle, außer Tom, verwenden den gleichen Browser. Dieses Beispiel sollte Ihnen verdeutlichen, dass Sie bei der Gestaltung Ihrer Seiten auch auf die Präferenzen der Anwender selbst achten müssen. Hierbei handelt es sich um Präferenzen verschiedenster Art, und einige davon sind leichter zu erfüllen als andere. Folgende Überlegungen müssen gemacht werden, bevor Sie mit dem Design beginnen:

Wie gut kennt sich der Anwender aus?

Unsere Familienmitglieder haben eine mehr oder minder große Erfahrung mit dem Web, und obwohl sich jeder dafür interessiert, haben einige von ihnen noch nie zuvor einen Web-Browser gesehen. Sie müssen sich also beim Design ihrer Site bewusst sein, dass die Besucher Ihrer Site unterschiedliche Web-Erfahrungen und variierende Ansprüche haben.



Abbildung 18.1:
Seite mit
Gebrauchsanwei-
sungen für den
Leser

Ist das Thema Ihrer Site von Interesse für Leute deren Web-Erfahrungen von unterschiedlichem Niveau sind? Wenn ja, dann könnten Sie ja einige Features einbauen, damit sich die Besucher Ihrer Site besser zurechtfinden. Sie haben ja schon im Kapitel des 17. Tages »Schreiben und Gestalten von Webseiten: Tipps« einige Tipps für die Gestaltung von Navigationssystemen erhalten, die verhindern, dass man im Cyberspace verloren geht. Das ist sicherlich ein guter Anfang, aber vielleicht ist es gar keine so schlechte Idee, Ihr Navigationssystem genauer, auf einer zusätzlichen Web-Site, zu erklären.

Auf Bild 18.1 wird die obere Hälfte einer Seite angezeigt, die den Lesern helfen soll mehr über Ihre Site zu erfahren. Links am Anfang der Seite verweisen den Leser auf verschiedene Sektionen der Seite. Diese Sektionen heißen »Über die Site«, »Das Navigationssystem«, »Browser Empfehlungen« und »Zusätzliche Dateien, die sie vielleicht brauchen werden«. Jede dieser Sektionen wiederum verweist auf Seiten, die detailliertere Beschreibungen des Inhalts Ihrer Site enthalten.

Erstellen Sie eine Liste, die Beschreibungen Ihrer Site enthält

Die »Über die Site«-Sektion enthält Verweise zu Seiten und liefert somit dem Leser einen schnellen Überblick über den Inhalt Ihrer Web-Site. Außerdem können die Leser erfahren, was seit ihrem letzten Besuch der Site, Neues dazugekommen ist. Bild 18.2 zeigt wie ein solches Link-Menü aussehen könnte. Folgendes sind einige Beispiele dafür, was diese Kategorie enthalten sollte.

- ▶ Eine »Was gibt es Neues?«-Seite, die auf neu Hinzugefügtes Ihrer Site verweist
- ▶ Eine Inhaltstabelle, welche die Titel der Seiten auflistet und eventuell eine kurze Beschreibung der Seiten Ihrer Web-Site geben
- ▶ Eine Seite, die oft gestellte Fragen von Besuchern Ihrer Seite enthält und Antworten auf diese Fragen gibt.

Diese Seiten werden normalerweise erstellt, wenn es darum geht dem Besucher den Weg durch Ihre Web-Site zu erleichtern. Vielleicht haben Sie ja noch zusätzliche, dem Thema Ihrer Site entsprechende Ideen. Auf alle Fälle sollten Sie diese Seiten immer auflisten.

Beschreiben Sie Ihr Navigationssystem

Navigationssysteme sind von Site zu Site verschieden und sind für den Web-Einsteiger nicht immer leicht zu verstehen. Man unterscheidet drei typische Navigationssysteme: einfache Text-Navigationssysteme, Bilder-Navigationssysteme und Frameset-Navigationssysteme. Sollten Sie der Ansicht sein, dass Ihr Navigationssystem zusätzlicher Erklärungen bedarf, liefern Sie Beschreibungen oder Links, die dem Leser Anweisungen darüber geben, wie sie es benutzen müssen. Bild 18.3 zeigt ein Link-Menü, das dem Leser verschiedene Navigationssysteme einer Web-Site beschreibt.

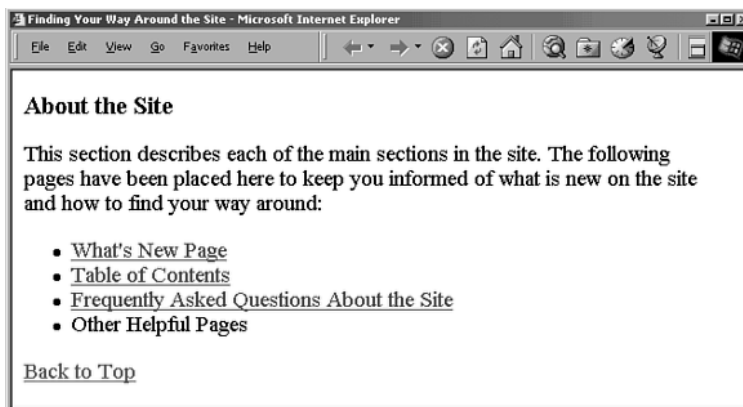


Abbildung 18.2:
Verweise auf Seiten, die dem Leser den Inhalt Ihrer Web-Site beschreiben

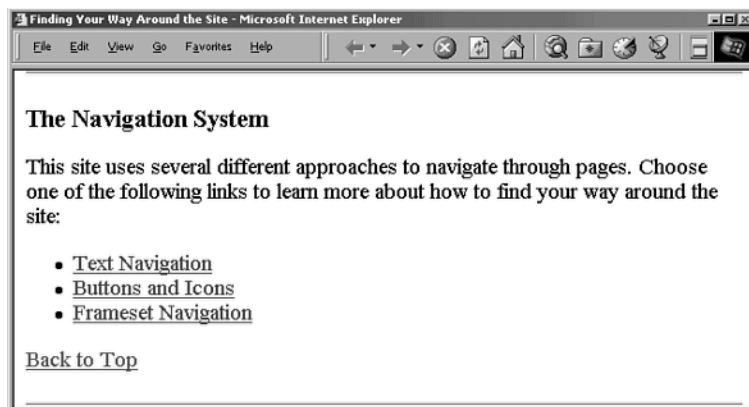


Abbildung 18.3:
Beschreibung
Ihres Navigations-
systems

Bei der Beschreibung Ihres Navigationssystems sollten Sie darauf achten, dass diese auch verständlich ist. Nehmen wir an, dass die Text-Navigationsleiste wie im folgenden Beispiel aussieht und auf allen Hauptseiten im Web-Site angezeigt wird.

Home | Contents | FAQ | What es New | Email | Guest Book | Links

Diese Menüleiste mag vielleicht für diejenigen, die Ihre Site schon besucht haben leicht verständlich sein, aber diejenigen, die neu auf Ihrer Site oder neu im Web sind, können nicht viel damit anfangen. Folgende Beschreibungen können für neue Anwender von großem Nutzen sein:

- ▶ **Home** – »Verwenden sie das Home-Link, um zur Homepage dieser Web-Site zurückzukehren.«
- ▶ **Inhalte** / Contents – »Die Inhaltstabelle der Seiten enthält Links zu allen Seiten der Web-Site. Beginnen Sie mit der Inhaltstabelle, wenn Sie zum ersten Mal im Site sind.«
- ▶ **FAQ** – »Jemand, der neu im Internet ist, weiß vielleicht nicht, was FAQ bedeutet. Die FAQ-Seite listet einige der Fragen von Anwendern auf und liefert die dazugehörigen Antworten.«
- ▶ **E-Mail** – Dieses Link muss ebenfalls erklärt werden, denn es ist nicht ersichtlich, ob es sich hier um ein Link handelt, das den Leser auf eine erhaltene E-Mail verweist, oder ob es sich um ein Link handelt, das Ihnen E-Mails schickt? Geben Sie also eine exakte Beschreibung des Links, damit der Leser auch genau weiß, wofür es steht. Diese Beschreibung könnte wie folgt aussehen: Verwenden Sie das E-Mail-Link, um Fragen oder Kommentare über diese Seite zu Ihrem Webmaster zu senden.

Sollten Sie in Ihrem Navigationssystem Bilder und Icons verwenden, müssen Sie wissen, dass Icons nicht immer leicht zu identifizieren sind. Vielleicht erinnern Sie sich noch an das Beispiel »verregneter Tag«, das im Kapitel des 12. Tages »Frames und Linked Windows« gezeigt wurde? Das Frameset wird zur Auffrischung noch einmal in Bild 18.4 aufgezeigt. Der linke Frame enthält Bilder und Icons für die Navigation und der untere Frame zeigt eine entsprechende Textnavigations-Leiste.



Abbildung 18.4:
Anwendung von
Bildern, Icons
und Frameset
Navigation

In manchen Fällen ist es nicht immer ganz klar, wofür die Icons auf den Navigations-Buttons stehen. Die Seite im Hauptframe zeigt eine kurze Beschreibung jedes Icons. Neue Anwender benötigen aber wahrscheinlich genauere Beschreibungen. Hierfür können Sie, ähnlich wie für die Textnavigations-Leiste, jedes Icon auflisten und eine entsprechende Beschreibung liefern.

Framesets sind für neue Anwender manchmal sehr schwer zu verstehen. Auch hier können genauere Erklärungen über Framesets von großem Nutzen sein. Eine Möglichkeit Framesets zu erklären ist, einen kurzen *Screenshot/Bildschirmausschnitt* dazu auf einer Webseite zu illustrieren (etwa so wie auf Bild 18.5). Geben Sie jedem Frame Ihres Framesets einen Namen, den sich der Leser leicht merken kann (zum Beispiel Links, Oben, Haupt und Unten) und platzieren Sie die Beschreibung jedes Frames auf der gleichen Seite. Die Frames in Bild 18.5 könnten wie folgt beschrieben werden:

- Verwenden Sie die sich im linken Rahmen befindenden Icons um Seiten aufzurufen, die Sie im Hauptframe angezeigt haben wollen.

- ▶ Der obere Frame des Framesets zeigt ständig das Site-Logo, damit Sie auch immer wissen, auf welcher Site Sie sich befinden.
- ▶ Der Hauptframe zeigt sämtliche Seiten an, die Sie im linken oder unterem Frame aufgerufen haben. Wenn Sie ein Link des Hauptframes anklicken, wird die Seite, auf welche das Link verweist, ebenfalls im Hauptframe angezeigt.
- ▶ Die Text-Links im unteren Frame haben die gleiche Funktion wie die Icons im linken Frame. Diese Links sind für diejenigen, die keine Bilder in ihren Browsern sehen können oder Text-Links bevorzugen.

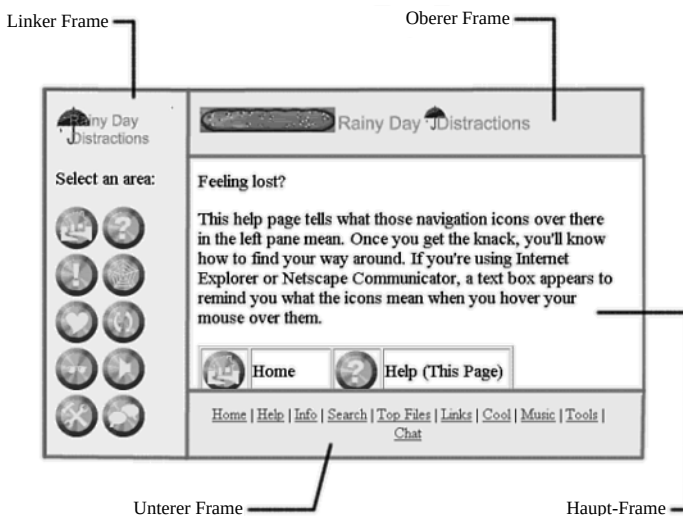


Abbildung 18.5:
Beschreibung
des Framesets

Fügen Sie Browser-Empfehlungen in Ihre Site ein

Nennen Sie den Browser für den Sie die Site gestaltet haben. Bild 18.6 zeigt Ihnen wie das aussehen könnte.

In diesem Beispiel erfährt der Leser, dass die Site für HTML 3.2 und für solche Browser gestaltet wurde, die Framesets unterstützen (HTML 4.0 Frameset Spezifikation). Eine Liste informiert den Leser über die Browser, die Sie verwendet haben, um Ihre Site zu testen, und bietet Links zum Herunterladen dieser Browser. Schließlich will der Web-Autor wissen, ob Anwender mit anderen Browsern Probleme hatten. Eine E-Mail-Adresse wird gestellt, damit der Autor mehr über die Problematik in anderen Browsern, die er nicht getestet hat, erfahren kann.

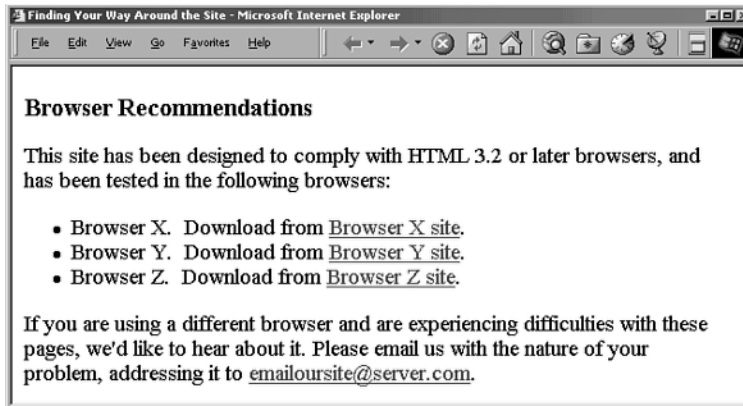


Abbildung 18.6:
Browser Empfehlungen

Erstellen Sie eine Liste mit Dateien, die der Anwender möglicherweise gebrauchen könnte

Sie sollten den Anwendern nicht nur Browser empfehlen, sondern auch über Plugins oder andere Dateien informieren, die sie vielleicht herunterladen möchten. Wenn Sie Dateien auf Ihrer Site haben, die nicht im HTML-Format sind (wie etwa komprimierte Dateien, Textverarbeitungs-Dokumente, GIF oder JPG Bilder, usw), sollten Sie den Leser darüber informieren, dass er oder sie eventuell einen speziellen Viewer braucht. Erstellen Sie eine Liste der Viewer oder externen Applikationen, die man benötigt.



Abbildung 18.7:
Liste anderer
Dateien, die dem
Leser vielleicht
von Nutzen sein
könnten

Ein Beispiel eines Link-Menüs für externe Dateien wird in Bild 18.7 gezeigt. Der Web-Autor listet in diesem Beispiel einige externe Applikationen auf, die man eventuell braucht, um eine Datei der Site anzuzeigen oder anzuwenden. Die fiktive Site enthält

einige Adobe Acrobat Reader – (.PDF) und Zip-komprimierte (.ZIP) Dateien, die auf verschiedenen Seiten erscheinen. Wenn Sie Links bei allen für den Leser wichtigen und bei allen externen Applikationen einfügen (wie etwa Adobe Acrobat Reader und WinZip), erhält der Leser sofort einen Überblick über alle wichtigen Dateien, die für eine optimale Benutzung der Site wichtig sind.

Finden Sie die Präferenzen der Anwender heraus

Zusätzlich zu den unterschiedlichen Erfahrungen der Leser im Web, hat auch jeder Leser Vorlieben dafür, wie er Ihre Webseite angezeigt haben will. Hier stellt sich nun die Frage, wie man es denn allen recht machen könnte? Leider kann man das nicht, aber Sie können immerhin versuchen ein möglichst großes Publikum anzusprechen. Zum guten Web-Design gehört herauszufinden, was die Leser auf Ihrer Web-Site sehen wollen. Das wird umso schwieriger, wenn die Themen Ihrer Web-Site ein großes Publikum interessieren.

Bill, Susan, Tom, Jill und die Großeltern unseres Beispiels hatten jeweils unterschiedliche Vorstellungen von der Art und Weise wie eine Webseite auszusehen hat. Das hängt sowohl von den unterschiedlichen Interessen, als auch von den speziellen Bedürfnissen eines jeden ab.

Ein Thema wie etwa »Wie repariere ich meinen 300cc Motorrad Motor?« wird natürlich nur für einen kleinen Kreis von Interesse sein. Hier ist es relativ einfach zu wissen, was dieser Leserkreis auf Ihrer Site sehen möchte. Man könnte zum Beispiel Schritt für Schritt Anweisungen zu Reparaturen geben, Bilder oder Multimedia könnten Techniken aufzeigen, die mit Text allein schwierig zu beschreiben wären.

»Die sieben Wunder der Antike« wiederum, könnten sowohl Studenten, Lehrkräfte, Archäologen, Historiker und jeden, der sich mit Geschichte befasst, interessieren. Sie haben ein größeres Publikum, eine größere Altersspanne und verschiedene Bildungsniveaus. Hier kann es durchaus schwierig werden eine Site zu erstellen, die jedem gefallen soll.

In solchen Fällen ist es vielleicht hilfreich Ihr Zielpublikum schon vorher festzulegen. Eine Möglichkeit hierbei wäre, Ihre Site nur für eine spezifische Anwendergruppe zu erstellen.

► **Schüler und Lehrkräfte von Grundschulen** – Diese Site müsste über ein einfaches Navigationssystem verfügen. Der Inhalt sollte Grundwissen vermitteln und einfach zu lesen sein. Farbenfrohe Bilder und Animationen tragen dazu bei, die Aufmerksamkeit jüngerer Leser zu wecken.

- ▶ **Realschüler, Gymnasiasten und deren Lehrkräfte** – Hier können Sie ein etwas schwierigeres Navigationssystem verwenden. Mit Multimedia und neuesten Web-Technologien könnten Sie diese Zielgruppe dazu bewegen, Ihre Site öfters zu besuchen.
- ▶ **Studenten und Lehrkräfte an Universitäten** – Der Inhalt Ihrer Web-Site muss dem höherem Bildungsniveau dieser Zielgruppe angepasst werden. Multimedia mag hier nicht so wichtig sein, eine Online Enzyklopädie hingegen wäre eine gute Idee.
- ▶ **Archäologen und Historiker** – Bei dieser Site spielt der Textinhalt eine große Rolle, Multimedia ist von geringer Bedeutung.

Es ist nicht immer möglich, die Zielgruppe einer Web-Site so leicht zu definieren. Ist dies der Fall, müssen Sie erst einmal von Ihren eigenen Vorlieben ausgehen. Besuchen Sie andere Sites, die einen ähnlichen Inhalt haben wie die, die Sie selbst erstellen möchten. Fragen Sie sich dabei, was Sie von diesen Sites erwarten, ob die Informationen gut angezeigt sind, ob genügend Hilfe angeboten wird, ob die Site mit zu viel oder zu wenig Multimedia arbeitet, usw...

Erstellen Sie schon einmal einige Versuchsseiten und fordern Sie Freunde, Familienangehörige und Kollegen auf, diese Site zu besuchen und sie »auseinander zu nehmen«. Vergessen Sie aber nicht, dass, wenn Sie schon um konstruktive Kritik bitten, auch Sachen hören werden, die Sie vielleicht nicht unbedingt hören wollten. Dennoch ist diese Vorgehensweise wichtig, denn oft werden Sie hiermit zu neuen Ideen angeregt, die Ihnen dabei helfen werden, Ihre Site zu verbessern.

HTML 4.0 oder XHTML 1.0?

In den vorhergehenden Artikeln haben Sie einiges über die Vorzüge von HTML 4.0 und XHTML 1.0 erfahren und wie sie auf Anwender älterer oder neuerer Browser ausgerichtet werden.

- ▶ **HTML 4.0 oder XHTML 1.0 Transitional** – Für alle, die ältere Browser unterstützen wollen
- ▶ **HTML 4.0 oder XHTML 1.0 Frameset** – Für alle, die Frameset-Navigation zusätzlich zur Unterstützung der Befehle in HTML 4.0 Transitional verwenden wollen
- ▶ **HTML 4.0 oder XHTML 1.0 Strict** – Für alle, die bei der Erstellung ihrer Seiten sich streng an HTML 4.0- oder XHTML 1.0 Spezifikation halten und keine verworfenen Elemente oder Attribute verwenden.

HTML 4.0 und XHTML 1.0 Transitional

Wenn Sie damit rechnen, dass Ihre Leser viele unterschiedliche Browser verwenden, ist es von Vorteil Ihre Webseiten mit HTML 4.0 oder XHTML 1.0 Transitional-Spezifikation zu gestalten. Hierbei ist die Kompatibilität mit älteren Browsern gesichert. Die Transitional-Spezifikation macht Sie flexibler und stellt Befehle zur Verfügung, die die HTML 4.0 Strict-Spezifikation nicht hat. Sie können ohne weiteres HTML 3.2 Präsentations-Tags verwenden (wie etwa das <center>-Tag oder das align-Attribute für die Textausrichtung oder die bgcolor- und color-Attribute für Hinter- und Vordergrundfarben).

Greifen wir noch einmal unser »Halloween House of Terror«-Beispiel auf, das Sie im Kapitel »Verwenden von Abbildungen, Farben und Hintergründe« des 7. Tages erstellt haben. Im folgenden Code werden wir diese Seite einem »Lifting« unterziehen. Die Seite verwendet HTML 3.2-kompatible Tags, um die Präsentation der Seite aufzuzeigen. Schriftart, Farben und Ausrichtung werden mit Tags formatiert, die mit der Strict-Spezifikation nicht möglich waren.

Diese Tags und Attribute werden im folgenden Code-Beispiel kursiv geschrieben. Zusätzlich wird eine Tabelle verwendet, um das gestaffelte Layout der Links, Beschreibungen und Bilder der Seite zu erstellen. Bild 18.8 zeigt das Resultat des Codes im Internet Explorer.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd"><html>
<head>
<title>Welcome to the Halloween House of Terror</title>
</head>
<body bgcolor="#ff9933" link="#990000">
<h1 align="center">
  <font face="Arial, sans-serif">
    The Halloween House of Terror!!</font></h1>
<div align="center">
  <p>
    
    
    
  </p>
</div>
<hr />
<p>
```

```

<font face="Arial, sans-serif">
  Voted the most frightening haunted house three years in a row, the
  <font color="#cc0000"><b>Halloween House of Terror</b></font>
  provides the ultimate in Halloween thrills. Over 20 rooms of thrills and
  excitement to make your blood run cold and your
  hair stand on end!
</font>
</p>
<hr />
<p><font face="Arial, sans-serif">
  Don't take our word for it ... preview some images of what awaits!
</font>
</p>
<div align="center">
<table border="0" width="75%" cellspacing="5" cellpadding="5">
  <tr>
    <td width="30%">
      <font face="Arial, sans-serif">
        
      </font>
    </td>
    <td width="40%" bgcolor="#CC0000"> <font face="Arial, sans-serif">
      Watch out for Esmerelda. You never know what she has in her cauldron.
    </font> </td>
    <th width="30%" bgcolor="#FF6600"> <b><font face="Arial, sans-serif">
      <a href="entry.gif">The Entry Way</a> </font></b> </th>
    </tr>
    <tr>
      <th width="30%" bgcolor="#FF6600"> <b><font face="Arial, sans-serif">
        <a href="bedroom.gif">The Master Bedroom</a> </font></b> </th>
      <td width="40%" bgcolor="#CC0000"> <font face="Arial, sans-serif">
        Don't open the closet door, whatever you do!</font></td>
      <td width="30%">
        <font face="Arial, sans-serif">
          
        </font>
      </td>
    </tr>
    <tr>
      <td width="30%">
        <font face="Arial, sans-serif">
          
        </font></td>
      <td width="40%" bgcolor="#CC0000"> <font face="Arial, sans-serif">
        More than a few innocents have been cast in chains for eons. They just
        aren't the same anymore.</font></td>
    </tr>
  </table>
</div>

```

```

<th width="30%" bgcolor="#FF6600"><b> <font face="Arial, sans-serif">
  <a href="galley.gif">The Galley</a></font></b></th>
</tr>
<tr>
<th width="30%" bgcolor="#FF6600"> <b><font face="Arial, sans-serif">
  <a href="dungeon.gif">The Dungeon</a> </font> </b></th>
<td width="40%" bgcolor="#CC0000"> <font face="Arial, sans-serif">
  Better listen to the tour guides, or you'll get lost! </font> </td>
  <td width="30%">
    <font face="Arial, sans-serif">
      
    </font></td>
  </tr>
</table>
</div>
<hr />
<p><font face="Arial, sans-serif"> The
  <font color="#cc0000">Halloween House of Terror</font>
  is open from October 20 to November 1st, with a gala celebration
  on Halloween night. Our hours are: </font></p>
<ul>
  <li>
    <font face="Arial, sans-serif">
      Mon-Fri 5PM-midnight</font>
    </li>
    <li>
      <font face="Arial, sans-serif">
        Sat & Sun 5PM-3AM</font>
      </li>
    <li>
      <font face="Arial, sans-serif">
        Halloween Night (31-Oct): 3PM-??</font></li>
  </ul>
<p align="center">
<font face="Arial, sans-serif"> The
  <font color="#cc0000">Halloween House of Terror</font>
  is located at:<br />
  The Old Waterfall Shopping Center<br />
  1020 Mirabella Ave<br />
  Springfield, CA 94532</font></p>
</body>
</html>

```



Abbildung 18.8:
Beispiel eines HTML 4.0
Transitional-Codes im
Internet Explorer

HTML 4.0 und XHTML 1.0 Framesets

Wenn Sie für das Navigieren in Ihrer Site Framesets bevorzugen, so ist die logischste Wahl, die Sie treffen können, HTML 4.0 oder XHTML 1.0 mit Frameset-Spezifikation. Hier können sie alle Befehle und Tags, die in HTML 4.0 oder in der XHTML 1.0 Transitional-Spezifikation erlaubt sind, verwenden. Zusätzlich können Sie sämtliche Tags und Attribute, die Framesets oder Frames betreffen, verwenden.

Der Schlüssel zum guten Frameset-Design ist, nur eine geringe Anzahl von Frames zu erstellen. Somit wird Ihr Navigationssystem leicht verständlich sein. Es wird jedoch schwieriger sein zu bestimmen, welche Bildauflösung Sie für Ihr Frameset haben wollen, da der Browser in eine Vielzahl von Sektionen unterteilt ist. Je mehr Frames das Frameset enthält, umso kleiner werden die Seiten in den jeweiligen Frames sein.

Bild 18.9 zeigt die »Halloween House of Terror«-Webseite, die in ein Frameset konvertiert wurde. Der Hauptframe zeigt genau dieselbe Seite wie in Bild 18.8, allerdings verweisen hier die Links des linken Frames den Leser zu den Bildern und Beschreibungen eines jeden Zimmers des Spukhauses. Der Leser muss also nicht mehr seine Browser-Buttons zuhelfe nehmen, um eine kleine virtuelle Tour im Spukhaus machen zu können. Die Navigation wurde also erheblich vereinfacht.

Der Code für das Frameset unterteilt das Browser-Fenster in zwei Frames: left und right. Die navigation.html Seite wird in den linken Frame, und die Hauptseite (main.html) wird in den rechten Frame geladen. Der Code für das Frameset sieht wie folgt aus:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/frameset.dtd">
<html>
<head>
<title>Halloween House of Terror Frameset</title>
</head>
<frameset cols="170,*">
  <frame name="left" src="navigation.html" />
  <frame name="right" src="main.html" />
</frameset>
<body>
<p> ... insert noframes content here ... </p>
</body>
</noframes>
</frameset>
</html>
```

Der Code für den linken Frame des Framesets (`navigation.html`) zeigt, sobald der Anwender einen der Image-Links aufruft, jede Seite im rechten Frame an (definiert durch das `target="right"`-Attribut im `<base>`-Tag). Jeder Image-Link zeigt einen Text auf, sobald der Anwender mit seiner Maus auch nur den Button streift, oder aber wenn Bilder im Browser ausgeschaltet werden. Der Code für den linken Frame sieht wie folgt aus:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">
<html>
<head>
<title>The Halloween House of Terror</title>
<base target="right" />
</head>
<body bgcolor="#ff9933" text="#000000">
<p><font face="Arial">The Halloween<br />
House of<br />
Terror</font></p>
<p><a href="main.html" target="right">
  
</a></p>
<p><a href="entry.html" target="right">
  
</a></p>
<p><a href="bedroom.html" target="right">
  
</a></p>
<p><a href="galley.html" target="right">
  
</a></p>
<p><a href="dungeon.html" target="right">
  
</a></p>
<p><a href="location.html" target="right">
  
</a></p>
</body>
</html>
```



Abbildung 18.9:
Beispiel des
HTML 4.0 Frame-
set Codes im
Internet Explorer

HTML 4.0 und XHTML 1.0 Strict

Die Transitional- und Frameset-Spezifikation ermöglicht es Ihnen, ältere Browser zu unterstützen (alles vor dem Netscape Navigator oder dem Microsoft Internet Explorer in den Versionen 4.0). HTML 4.0 oder die XHTML 1.0 Strict-Spezifikation jedoch bietet Ihnen nicht nur eine Vielzahl von Anzeige-Optionen mittels Cascading Style Sheets (CSS) sondern auch die Möglichkeit mit anderen Web-Technologien, die Sie im Rahmen dieses Buches schon kennen gelernt haben, zu arbeiten. Sobald nun neue Web-Browsing Hardware eingeführt wird, wird man in naher Zukunft immer mehr die Strict-Spezifikation verwenden müssen. Derzeit jedoch vermeidet man HTML 4.0 oder XHTML 1.0 Strict anzuwenden, da wahrscheinlich noch einige Zeit vergehen muss, bis die Mehrheit der Anwender Ihre Site mit kompatiblen Browsern besuchen wird.

Wenn Sie mit HTML 4.0 oder XHTML 1.0 Strict arbeiten, müssen Sie sämtliche HTML-Tags und Attribute, die als verworfen gelten, vermeiden. Verwenden Sie für Ihre Seitenpräsentation stattdessen Cascading Style Sheets oder andere Methoden, wie etwa Dynamic HTML.

Sollten Sie sich auf eine bestimmte Zielgruppe festgelegt haben, was gleichzeitig ein weniger großes Publikum bedeutet, bietet HTML 4.0 oder die XHTML 1.0 Strict-Spezifikation einige Vorteile. Am 10. Tag haben Sie ja schon erfahren, dass Sie mit CSS-Technologien eine bessere Kontrolle über Seiten-Layout und das Design haben. Sie können dabei Ihre Seiten fast genauso gestalten, wie Sie es mit einem Seitenlayout-

oder Textverarbeitungsprogramm Programm tun würden. Die Cascading Style Sheets Level 2 (CSS2) sind ebenfalls von großem Vorteil, da sie zusätzliche Features für nicht visuelle Browser enthalten. Das bedeutet, dass Sie Seiten gestalten können, die von Sehgeschädigten trotzdem erkannt werden können.

Das folgende Codebeispiel illustriert, wie Sie, die in Bild 18.8 gezeigte Seite mit der XHTML 1.0 Strict-Spezifikation erstellen können. In diesem Beispiel wurden beim »Halloween House of Terror« eingebettete Style-Sheet- Eigenschaften und -Werte zur Formatierung des Textes und der Bilder angewandt. Die Farben für den Text, die Links und die Tabellenzellen wurden am Anfang der Seite, in der Style-Sektion, definiert. Außerdem wurden oberhalb, unterhalb, rechts und links der Webseite Ränder definiert. Anstelle der Standard Bullets, werden im oberen Teil der Seite vom Style Sheet definierte Image – Bullets angezeigt. Das Resultat des Codes wird in Bild 18.10 gezeigt.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/strict.dtd">
<html>
<head>
<title>Welcome to the Halloween House of Terror</title>
<style type="text/css">
<!--
body { background-color: #ff9933;
      color: #000000;
      font-family: Arial, sans-serif;
      font-size: 12pt;
      margin-left: 20px;
      margin-right: 20px;
      margin-top: 10px;
      margin-bottom: 10px }
a:link{ color: #990000 }
a:visited{ color: #CC00CC }
a:active{ color: #CC0000 }
h1{ font-family: Arial, sans-serif;
    font-size: 24pt }
table { text-align: center }
th { background-color: #ff6600;
     color: #000000;
     font-family: Arial, sans-serif;
     font-size: 12pt;
     font-weight: bold }
td { color: #000000;
```

```

    font-family: Arial, sans-serif;
    font-size: 12pt }
td.red { background-color: #cc0000 }
ul { list-style-image: url("bullet.gif") }
.bloodred { color:#CC0000 }
.bloodredbold { color: #CC0000; font-weight: bold }
.center { text-align: center }
->
</style>
</head>
<body>
<h1 class="center">The Halloween House of Terror!!</h1>
<div class="center">
<dl>
<dd>



</dd>
</dl>
</div>
<hr />
<p>Voted the most frightening haunted house three years in a row,
the <span class="bloodredbold">Halloween House of Terror</span>
provides the ultimate in Halloween thrills. Over 20 rooms of
thrills and excitement to make your blood run cold and your hair
stand on end!</p>
<hr />
<p>Don't take our word for it ... preview some images of what
awaits!</p>
<table border="0" summary="House of Terror" width="75%" cellspacing="5"
cellpadding="5">
<tr>
<td>

</td>
<td class="red">Watch out for Esmerelda. You never
know what she has in her cauldron.</td>
<th><a href="code/entry.gif">The Entry Way</a></th>
</tr>
<tr>
<th><a href="code/bedroom.gif">The Master Bedroom</a></th>
<td class="red">Don't open the closet door, whatever
you do!</td>
<td>
</td>

```

```

</tr>
<tr>
  <td></td>
  <td class="red">More than a few innocents have been
  cast in chains for eons. They just aren't the same anymore.</td>
  <th><a href="code/galley.gif">The Galley</a></th>
</tr>
<tr>
  <th><a href="code/dungeon.gif">The Dungeon</a></th>
  <td class="red">Better listen to the tour guides, or
  you'll get lost!</td>
  <td></td>
</tr>
</table>
<hr />
<p>The <span class="bloodred">Halloween House of Terror</span> is
open from October 20 to November 1st, with a gala celebration on
Halloween night. Our hours are:</p>
<ul>
<li>Mon-Fri 5PM-midnight</li>
<li>Sat & Sun 5PM-3AM</li>
<li>Halloween Night (31-Oct): 3PM-??</li>
</ul>
<p class="center">The <span class="bloodred">Halloween House of
Terror</span> is located at:<br />
The Old Waterfall Shopping Center<br />
1020 Mirabella Ave<br />
Springfield, CA 94532</p>
</body>
</html>

```

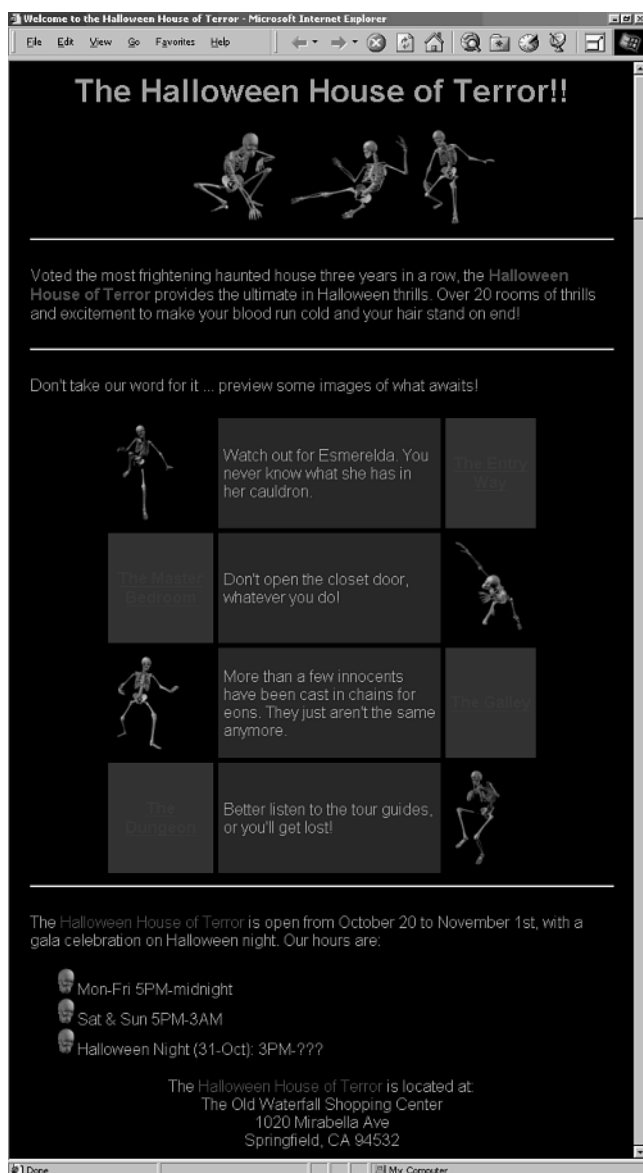


Abbildung 18.10:
Beispiel des HTML 4.0
Strict Codes im Internet
Explorer

Zusammenfassung

HTML 4.0 und XHTML 1.0 bieten drei verschiedene Möglichkeiten für das Web-Site-Design. Ich hoffe Sie haben verstanden, dass Sie bei der Wahl einer dieser Möglichkeiten, die Präferenzen Ihrer Leser berücksichtigen sollten. Das A und O jedoch bleibt herauszufinden, welche Bedürfnisse und Ansprüche Ihre Leser haben könnten, und das Design Ihrer Web-Site danach zu richten. Nicht jede Site muss unbedingt mit den neuesten Web-Technologien erstellt werden. Andererseits gibt es aber Themen, die nach einem höherem Niveau bezüglich der Seitengestaltung verlangen. Berücksichtigen Sie also die Präferenzen Ihrer Leser, und Sie können sichergehen, dass diese Ihre Site wieder besuchen werden.

Workshop

Frage-Antwort

- F Ich bekomme viele Anregungen und Ratschläge von Besuchern meiner Site. Einige wollen mehr Medien, andere wiederum weniger. Gibt es eine Zwischenlösung, die beide Seiten zufriedenstellen könnte?**
- A** *In diesem Kapitel haben Sie gelernt, dass man mit Links auf externe Medien-Dateien verweisen kann. Das wäre die beste Möglichkeit für Leser, die nur wenige diese Dateien sehen wollen, da nur durch Aufrufen eines Links die Datei angezeigt wird. Sie können aber auch Scripting-Methoden anwenden (wie etwa JavaScript oder VBScript) um herauszufinden, welchen Browser Ihre Leser benutzen. Nachdem der Browser-Typ ausfindig gemacht wurde, kann das Script die Leser automatisch zu den Seiten führen, die mit diesem Browser kompatibel sind. Dafür muss beim Design etwas mehr Zeit aufgebracht werden, denn Sie müssen nicht nur das Script schreiben, sondern eventuell auch verschiedene Versionen Ihrer Webseite erstellen. Falls Sie die Browser nun nicht automatisch auffindbar machen wollen, können Sie mit Links auf Ihrer Homepage die Leser zu den Seitenarten, die sie angezeigt haben wollen, führen. Verwenden Sie Ihre Homepage als eine Art Portal, nach dem der Leser zwei Richtungen einschlagen kann: eine einfache oder eine medien-intensive Site.*
- F Ich verwende viele externe Dateien auf meiner Web-Site, die man von unterschiedlichen Seiten aus laden kann. Wäre es nicht sinnvoll, ein Link auf die jeweilige Seite zu setzen, der den Download des jeweiligen Viewers ermöglicht?**

- A** Obwohl es für den Leser viel einfacher ist, externe Dateien und die jeweilige Reader- oder Helfer-Applikation von derselben Seite herunterzuladen, würde es für Sie schwieriger sein Ihre Seiten zu pflegen, wenn sich die URL der Applikation ändert. Ein guter Kompromiss wäre, eine Download-Seite mit sämtlichen Links zu allen Helfer-Applikationen, die für die Dateien auf Ihrer Web-Site benötigt werden. Wenn der Anwender nun die auf Ihrer Site zur Verfügung stehende externe Datei herunterlädt, kann er oder sie auf der Download-Seite navigieren. Von dort kann der Leser dann die Helfer-Applikation, die er zum Anzeigen der Datei braucht, laden.

Quiz

1. In welchen Dingen gehen die Bedürfnisse der Anwender im Web auseinander?
2. Wie sollte Ihre Site sein, damit auch Computer- bzw. Interneteinsteiger sich auf ihr zurechtfinden?
3. Wie hilft Ihnen die Anwendung von HTML 4.0 oder XHTML 1.0 Spezifikationen den Präferenzen mehrerer Personen gerecht zu werden?
4. Richtig oder falsch? Es ist besser viele Frames in einem Frameset zu haben, denn dadurch können viele Informationen zur selben Zeit in den Browser-Fenstern angezeigt werden.
5. Welches sind die Vor- und Nachteile bei der Anwendung von HTML 4.0 oder XHTML 1.0 Strict?

Antworten

1. Nicht alle Anwender kennen sich gleich gut im Web aus. Deswegen unterscheiden sich auch die Browser. Manche der Anwender bevorzugen Bilder und interaktive Medien, andere wiederum bevorzugen einfachere, einen Prozess oder Technik demonstrierende Bilder. Vieles, was Anwender bevorzugen, hängt aber hauptsächlich von den unterschiedlichen Interessen der Leser ab.
2. Die Seiten Ihrer Site sollten auf alle Fälle Informationen enthalten, die dem Leser bei Ihrer Suche helfen. Außerdem sollten die Site Anweisungen darüber enthalten, wie sich der Leser am besten auf Ihrer Site zurechtfindet.
3. HTML 4.0 oder XHTML 1.0 Transitional wurde zur sicheren Unterstützung älterer Browser erschaffen. Außerdem können Sie mit der Transitional-Spezifikation Tags und Attribute, die in der Strict-Spezifikation nicht vorkommen, verwenden.

4. Falsch! Zu viele Frames können neue Anwender verwirren. Außerdem werden Seiten, die sich in einem zu kleinem Bereich befinden, mit einer niedrigeren Farbauflösung angezeigt.
5. Der Nachteil von HTML 4.0 oder XHTML 1.0 Strict ist, dass es noch zu lange dauern wird, bis die Mehrheit der Leser Browser benutzt, die vollkommen der HTML- bzw. XHTML-Spezifikation genügen. Die Vorteile sind, dass HTML 4.0 oder XHTML 1.0 Strict eine Unterstützung für spezielle Präsentationsfähigkeiten bietet (zum Beispiel für diejenigen, die keine visuellen Browser verwenden).

Übungen

1. Erstellen Sie ein einfaches Navigationssystem für eine Web-Site und beschreiben Sie es so, dass es für Sie einen Sinn ergibt. Bitten Sie dann jemand anderen, Ihre Beschreibungen zu lesen und stellen Sie sicher, dass Ihre Erklärungen auch gut verstanden wurden.
2. Erstellen Sie eine Liste von Themen, über die auf Ihrer Web-Site diskutiert werden sollen. Gehen Sie Ihre Liste ein zweites Mal durch und versuchen Sie die Präferenzen der Leute herauszufinden, die sich für diese Themen interessieren könnten. Gehen Sie die Liste ein drittes und letztes Mal durch und machen Sie sich Notizen darüber, was Sie für die jeweiligen Anwender-Zielgruppen benötigen.



**Woche
3**

In der letzten Woche haben Sie Ihre Webseiten auf Ihrem lokalen Rechner gestaltet und getestet, in Ihrem eigenen Browser. Vielleicht hatten Sie Ihren Rechner noch nie an ein Netzwerk angeschlossen. Und zu diesem Zeitpunkt haben Sie wahrscheinlich eine Web-Präsentation mit einer gut organisierten Struktur zusammengebaut, aussagekräftige Abbildungen eingefügt (und sorgfältig ausgewählten Text für das ALT-Attribut beigefügt), Ihren Text mit Verstand und Sorgfalt geschrieben, nur relative Verbindungen benutzt und das Ganze ausgiebig auf Ihrem eigenen System getestet.

Es ist nun, am letzten Wochentag, endlich an der Zeit, das alles online verfügbar zu machen, sodass andere Leute auf dem Web Ihre Seiten betrachten und Verknüpfungen dazu herstellen können. In diesem und dem nächsten Kapitel lernen Sie alles, was Sie brauchen, um damit zu beginnen, die Arbeit, die Sie sich gemacht haben, zu veröffentlichen. Heute lernen Sie Folgendes:

- ▶ Was ein Web-Server tut, und warum Sie einen brauchen.
- ▶ Wo Sie einen Web-Server finden können, auf den Sie Ihre Präsentation packen können.
- ▶ Wie Sie Ihre Web-Präsentation installieren.
- ▶ Wie Sie Ihren URL herausfinden.
- ▶ Wie Sie Ihre Webseiten testen.
- ▶ Methoden zur Werbung für Ihre Präsentation.
- ▶ Wie Sie Protokolldateien und Zähler einsetzen können, um herauszufinden, wer Ihre Seiten betrachtet.

Web-Server?

Um Webseiten zu publizieren, brauchen Sie einen Web-Server. Ein Web-Server ist ein Programm, das sich auf einer Maschine im Netz befindet und auf einen Web-Browser wartet, der eine Anfrage wegen einer Datei übermittelt. Ist die Anfrage einmal übers Kabel gekommen, sucht und übermittelt der Server die Datei. So einfach ist das.

Server und Browser kommunizieren mit Hilfe des HyperText Transfer Protocol (HTTP), einer besonderen »Sprache«, die speziell für die Übermittlung von Hypertext-Dokumenten durch das Web entwickelt wurde. Daher werden Web-Server häufig auch HTTPD-Server genannt.



Das »D« in HTTPD steht für »Daemon«. Ein *Daemon* ist ein Unix-Ausdruck für ein Programm, das im Hintergrund sitzt und auf Anfragen wartet. Erhält er eine Anfrage, so »wacht« er auf, bearbeitet die Anforderung und legt sich

dann wieder »schlafen«. Sie müssen nicht unbedingt Unix haben, um ein Programm zu benutzen, das sich wie ein Daemon verhält, daher werden Web-Server auch auf anderen Plattformen HTTPD genannt. Ich nenne sie meistens Web-Server.

Andere Dinge, die Web-Server tun

Obwohl die Hauptaufgabe eines Web-Servers darin besteht, Anfragen eines Browsers zu bearbeiten, ist er auch noch für einiges mehr verantwortlich. Einiges davon werden Sie heute erfahren.

Datei- und Medientypen

In Kapitel 13 haben Sie etwas über Inhaltstypen gelernt und wie Browser und Server Dateinamenerweiterungen verwenden, um den Dateityp zu ermitteln. Server sind dafür verantwortlich, den Browser wissen zu lassen, um was für einen Inhaltstyp es sich bei der Datei handelt. In »Fragen zum Webmaster«, später in diesem Kapitel behandelt, werden Sie einige wichtige Problematiken kennen lernen, die Sie Ihrem Web-Provider oder Webmaster unbedingt vor der Veröffentlichung Ihrer Seiten stellen sollten.

Dateiverwaltung

Der Web-Server ist auch für eine sehr eingeschränkte Dateiverwaltung zuständig; hauptsächlich, indem er herausfindet, wo eine Datei liegt, und dann auch Buch darüber führt, wohin die Datei weitergeschickt wird. Wenn ein Browser eine Datei anfordert, die nicht existiert, ist es der Web-Server, der eine Seite mit der Botschaft "404: Datei nicht gefunden" zurückschickt. Server können auch dazu konfiguriert werden, Parallelbezeichnungen (Alias) für Dateien zu erstellen (dieselbe Datei, die aber unter einem anderen Namen erreicht wird), oder sie können Dateien an unterschiedliche Orte weiterleiten (indem Sie einen Browser auf einen neuen URL hinweisen, wenn eine Datei »umgezogen« ist). Eine andere Funktion der Web-Server besteht darin, dass sie eine Standarddatei oder eine Verzeichnisliste zurückschicken, wenn ein Browser einen URL anfordert, der mit einem Verzeichnisnamen endet.

Schließlich können Server Protokolldateien (Log-Files) darüber aufstellen, wie oft jede Datei auf einer Site angefordert wurde, einschließlich des Sites, der sie anforderte, des Datums und bei einigen Servern auch des Browsertyps und des URL der Seite, von der die Verknüpfung ausging.

CGI-Scripte, Programme und Formularverarbeitung

Eine der interessanteren und komplexeren Dinge, die ein Server tun kann, ist es, externe Programme auf dem Server-Rechner laufen zu lassen, die auf Eingaben beruhen, die Ihre Leser über ihre Browser machen. Diese speziellen Programme werden meistens CGI-Programme genannt und liefern die Basis für die Erstellung von interaktiven Formen und anklickbaren Imagemaps (Bilder mit verschiedenen aktiven Bereichen, die verschiedene Funktionen ausführen, je nachdem, welcher Bereich angeklickt wurde). CGI-Scripte werden oft zur Verarbeitung von anklickbaren serverseitigen Imagemaps verwendet, was Sie bereits in Tag 9 gelernt haben.

CGI-Scripten können auch dazu benutzt werden, einen Web-Server mit einer Datenbank oder einem anderen Informationssystem zu verbinden.

Datenverarbeitung auf der Server-Seite

Einige Server haben die Fähigkeit, Dateien zu bearbeiten, bevor Sie sie zum Browser zurückschicken. In einfachen Fällen sind dies so genannte Server-Side-Includes, bei denen ein Datum oder ein Stück vorgefertigter Text in eine Seite eingefügt wird oder die auch ein Programm ausführen können (viele der Zugangszähler, die Sie auf Seiten sehen, funktionieren auf diese Weise). Serverseitige Verarbeitung kann auch in viel raffinierteren Formen vor sich gehen, um Dateien auf die Schnelle für verschiedene Browser umzuändern oder um kleine Teile eines Scripts auszuführen

Authentifizierung und Sicherheit

Einige Web-Sites erfordern es, dass Sie sich bei deren Service registrieren lassen, und veranlassen Sie, sich auf der Site jedes Mal mit Namen und Passwort einzuklinken. Dies wird *Authentifizierung* genannt (authentication) und ist etwas, wozu die meisten Web-Server heutzutage in der Lage sind. Bei Authentifizierungen können Sie User mit Passwörtern versehen und den Zugang zu bestimmten Dateien und Verzeichnissen einschränken. Sie können den Zugang zu Dateien oder zu einem ganzen Site auch auf Grundlage der Site-Namen oder der IP-Adressen beschränken – so können Sie zum Beispiel jegliche Person außerhalb Ihrer Firma davon abhalten, Dateien anzusehen, die nur für die interne Verwendung gedacht sind.



Authentifizierung ist die Fähigkeit, Dateien oder Verzeichnisse auf Ihrem Web-Server so zu schützen, dass der Server von den Lesern einen Namen und ein Passwort verlangt, bevor die Dateien angesehen werden dürfen.

Einige Server benutzen zur Sicherheit jetzt einen Mechanismus, der sichere Verbindungen und Transaktionen mit dem SSL-Protokoll von Netscape durchführt. SSL bietet Authentifizierung des Servers (um zu beweisen, dass der Server derjenige ist, für den er sich darstellt) und eine verschlüsselte Verbindung zwischen dem Server und dem Browser, sodass vertrauliche Informationen zwischen den beiden geheim gehalten werden.

Einen Server finden

Bevor Sie Ihre Web-Präsentation im Web anbieten können, müssen Sie einen Web-Server finden. Abhängig davon, auf welche Weise Sie Zugang zum Internet haben, kann dies sehr einfach oder nicht ganz so einfach sein.

Der Server Ihrer Schule/Universität etc. oder Ihrer Firma

Wenn Sie Ihre Internet-Verbindung durch Ihre Schule/Universität etc. oder Ihre Arbeit bekommen, wird diese Organisation Ihnen es wahrscheinlich auch erlauben, Webseiten auf deren eigenem Web-Server zu publizieren. Wenn man bedenkt, dass diese Organisationen normalerweise schnelle Verbindungen zum Internet haben und auch spezielle Personen, die den Site für Sie verwalten, dann haben Sie damit eine ideale Situation.

Wenn Sie sich in dieser Situation befinden, werden Sie Ihren Systemverwalter, Computerberater, Web-Master oder Netzwerk-Anbieter fragen müssen, ob sie einen Web-Server verfügbar haben, und wenn ja, wie Sie Ihre Seiten dort installieren können. Sie werden später in diesem Kapitel erfahren, welche Fragen Sie stellen müssen.

Nutzung eines kommerziellen Web-Dienstes

Wenn Sie einen Internet Service Provider (ISP) oder einen kommerziellen Online-Dienst für Ihren Zugang zum Internet bezahlen, können Sie unter Umständen Ihre Webseiten durch diesen Anbieter publizieren. Dies kann Sie jedoch zusätzliches Geld kosten, und es kann Einschränkungen darin geben, was für Seiten Sie publizieren dürfen oder ob Sie CGI-Scripts ausführen können oder nicht. Kontaktieren Sie die technische Hilfe Ihres Anbieters oder Online-Gruppen und Konferenzen, die sich mit Internet-Anbietern beschäftigen, wegen der Einzelheiten, wie das Publizieren auf dem Web dort vor sich gehen soll.

Im letzten Jahr sind überall im Lande Firmen entstanden, die Ihnen nichts anderes als Web-Veröffentlichungsdienste anbieten. Es wird Ihnen angeboten, Ihre Dateien auf deren Rechner zu übertragen (meistens mit FTP), und sie stellen den Plattenplatz und

die Netzwerk-Verbindung zur Verfügung. Sie haben auch professionelle Site-Verwalter auf der Site, um sicherzustellen, dass die Server zu jeder Zeit richtig funktionieren. Normalerweise wird Ihnen ein festgelegter monatlicher Betrag berechnet, wobei zusätzliche Kosten auftreten, wenn Sie große Mengen von Speicherkapazität benötigen oder wenn Sie besonders beliebte Seiten haben, die einen großen Teil der Netzwerk-Bandbreite vereinnahmen. Einige Anbieter werden es Ihnen sogar erlauben, CGI-Scripts für Formulare und Imagemaps einzurichten und Sie auch bezüglich deren Einrichtung beraten; einige andere werden Ihnen sogar Ihren eigenen Host-Namen auf deren Server einrichten, sodass es so aussieht, als besäßen Sie Ihren eigenen Web-Server. Solche Angebote können ein kommerzielles Web-System zu einer recht verlockenden Möglichkeit machen.

Die meisten aktuellen Internet-Provider bieten den Ihnen verfügbaren Platz im Web in einem Unterverzeichnis auf ihrem Server an, und Ihre URL sollte in etwa so aussehen: <http://www.ispname.net/-youraccountname>. Sie können auch einen virtuellen Webserver bei einem Internet Service Providers erhalten, eine sehr preisgünstige Option.

Um einen virtuellen Webserver zu erstellen, müssen Sie Ihren Domain-Namen bei InterNIC oder DENic registrieren lassen. Die Anfangskosten für die Registrierung und den Erhalt Ihres Domain-Namens und Ihrer IP-Adresse beträgt für die ersten zwei Jahre unter 70 Dollar. Damit Ihr Domain-Name gültig bleibt, müssen Sie eine jährliche Abgabe bezahlen.

Ein virtueller Webserver kann von der ganzen Welt aus erreicht werden. Sie können diese Möglichkeit zu allen erdenklichen Zwecken verwenden. Die Funktionalität ist mit einer Web-Site Ihres eigenen Servers vergleichbar. Ihre Site wird eine Adresse haben, die in etwa so aussieht: <http://www.mygreatsite.com>

Viele ISPs und Web-Anbieter lassen Ihren Domain-Namen bei InterNIC registrieren. Sie können Ihren Domain-Namen aber auch bei Network Solutions – dem Privatunternehmen, das im Moment die InterNIC-Registrierungen in den USA verwaltet- unter www.networksolutions.com registrieren lassen.



Die ultimative Web-Host Liste unter <http://webhostlist.internetlist.com/> ist eine gute Quelle, um Web-Hosting-Service zu finden und zu evaluieren.

Beachten Sie, dass Anbieter, die Webseiten publizieren, sich nicht unbedingt in Ihrer Stadt oder sonstigen näheren Umgebung befinden müssen. Sie müssen sich nur in der physischen Nähe Ihres Hauptanbieters für Internet-Dienste befinden, aber die Sites, die Ihre Webseiten publizieren, können sich irgendwo im Internet befinden, sodass Sie sich nach den billigsten Preisen und dem besten Service umschaun können, ohne sich um den geographischen Ort sorgen zu müssen.

Den eigenen Server einrichten

Wenn Sie die Spitzenklasse der Web-Publikation erreichen wollen, dann ist die Einrichtung Ihres eigenen Web-Servers der beste Weg. Wenn Sie Ihren eigenen Site verwalten, können Sie nicht nur so viel veröffentlichen, wie es Ihnen passt und auch jede Art von Inhalt anbieten, sondern Sie können auch Formulare, CGI-Skripte, Image-maps und viele andere spezielle Möglichkeiten ausnutzen, die die meisten Web-Publikationsanbieter nicht zulassen. Jedoch können die Kosten, die Zeit für Instandhaltung und das technische Wissen, die erforderlich sind, um Ihren eigenen Server zu unterhalten, entmutigend sein, sodass es auf keinen Fall für jeden zu empfehlen ist. Sie werden noch mehr über die Aufrechterhaltung Ihres eigenen Servers im nächsten Kapitel erfahren.

Organisieren und Installieren Ihrer HTML-Dateien

Haben Sie erst einmal Zugriff auf einen Web-Server, können Sie die Web-Präsentation installieren, an deren Erstellung Sie so hart gearbeitet haben. Doch bevor Sie alles tatsächlich an seinen Ort schaffen, ist es am besten, Ihre Dateien zu organisieren und zu überlegen, was wohin gehört, damit Sie keine Dateien vergessen oder Verbindungen zwischen ihnen zerbrechen.

Was Sie Ihren Web-Master fragen sollten

Der Web-Master ist die Person, die Ihren Web-Server administriert, was auch Ihr Systemverwalter, Ihre technische Hilfsperson oder Ihr Netzwerkverwalter sein kann. Bevor Sie Ihre Dateien publizieren können, sollten Sie von Ihrem Web-Master einige Einzelheiten über den Server erfragen. Diese Liste wird Ihnen auch später im Buch nützlich sein, wenn es nämlich darum geht, herauszufinden, was Sie genau mit Ihrem Server anfangen können.

- **Wo kann ich auf dem Server meine Dateien ablegen?** In vielen Fällen wird Ihr Web-Master Ihnen Ihr eigenes Verzeichnis einrichten. Sie sollten wissen, wo es sich befindet und wie Sie Zugang dazu erlangen.

In einigen Fällen, vor allem auf Unix-Rechnern, können Sie ein spezielles Verzeichnis in Ihrem persönlichen Verzeichnis einrichten und dort Ihre Web-Dateien aufbewahren. Wenn dies der Fall ist, wird Ihr Web-Master Ihnen den Namen dieses Verzeichnisses vorgeben.

- **Wie lautet der URL des Wurzel-Verzeichnisses?** Dieser URL kann sich vom tatsächlichen Pfad-Namen zu Ihren Dateien unterscheiden.

- ▶ **Wie heißt die Standard-Indexdatei des Systems?** Dies ist die Datei, die standardmäßig geladen wird, wenn ein URL mit einem Verzeichnisnamen endet. Normalerweise ist dies `index.html`, kann aber auch manchmal `default.html`, `Homepage.html` oder sonst was sein.
- ▶ **Kann ich CGI-Scripts ausführen?** Abhängig von Ihrem Server kann die Antwort darauf glatt »nein« lauten, oder Sie können auch auf bestimmte Programme und Fähigkeiten beschränkt sein. Momentan brauchen Sie noch keine ausführlichen Details über CGI zu erfahren; das kommt noch später.
- ▶ **Unterstützen Sie spezielle Plugins oder Dateitypen?** Wurde Ihre Site für spezielle Medientypen erstellt (Real Audio, Shockwave, Shockwave Flash oder ähnliche Plugins oder Web-Erweiterungen), könnte es sein, dass Ihr Web-Master den Server konfigurieren muss, um diese Dateitypen anpassen zu können. Überprüfen Sie, ob Ihr Server spezielle Medientypen unterstützt, bevor Sie diese erstellen.
- ▶ **Unterstützen Sie FrontPage-Server-Erweiterungen?** Microsoft FrontPage, eine der bekanntesten Web-Tools für die Windows-Plattform (ebenso für die Macintosh-Plattform) ermöglicht Ihnen das Entwickeln komplexer Webseiten, die Formulare und andere Features für Fortgeschrittene enthalten. Um viel dieser Features zu verwenden, müssen auf dem Server jedoch die FrontPage-Server-Erweiterungen installiert sein. Wenn Sie an der Verwendung von FrontPage zum Designen von Webseiten interessiert sind, gehen Sie sicher und fragen Sie zuerst Ihren ISP, ob dieser die Server-Erweiterungen unterstützt.
- ▶ **Ist meine Site begrenzt, was die Datenmenge angeht?** Manche Provider limitieren Web-Sites auf eine bestimmte Inhaltsmenge (zum Beispiel Seiten, die nur zu Arbeitszwecken gedacht sind) oder erlauben Ihnen nur wenige Seiten. Diese Sites sollten darauf aufmerksam machen, dass nur eine bestimmte Besucheranzahl zur gleichen Zeit nicht überschritten werden darf und sollte auch darauf hinweisen, wie Sie Ihre Seite veröffentlichen können. Gehen Sie sicher, dass Sie mit den Einschränkungen dieses Systems vertraut sind, damit Sie damit arbeiten können.
- ▶ **Gibt es eine Beschränkung bezüglich des Download-Volumens?** Diese Frage schließt gleich an der letzten Frage an. Ihr Anbieter darf zwar kein Seitenlimit setzen, aber Sie müssen mit zusätzlichen Kosten rechnen, wenn Sie ein bestimmtes Download-Volumen im Monat überschreiten. Bevor Sie also 10MB oder mehr Inhalt auf Ihre Webseite packen und Dutzende von riesigen Grafiken, Sound-Dateien und Video-Clips auf Ihre Webseite laden, fragen Sie lieber zuerst Ihren Provider, ob es ein Limit gibt. Das Downloadvolumen ergibt sich aus der Menge der Daten, die von Ihrer Site zu den Besuchern und von den Besuchern zu Ihrer Site übertragen wurde.

- **Bieten Sie auch Scripts an, die ich für meine Webseite verwenden kann?** Wenn Sie nicht Ihre eigenen Scripts zum Hinzufügen von Features auf Ihre Seite schreiben, können Sie Ihren Service-Provider fragen, ob diese die für Sie entsprechenden Scripts anbieten. Viele ISPs bieten zum Beispiel Scripts an, die es Ihnen ermöglichen, Seitenzähler für Ihre Homepage einzurichten. Andere bieten auch formularverarbeitende Scripts an.

Organisieren Sie Ihre Dateien in Verzeichnissen

Wohl der beste Weg, jede Ihrer Präsentationen zu organisieren, ist das Zusammenfassen aller Dateien einer solchen Präsentation in einem einzelnen Verzeichnis. Wenn Sie viele zusätzliche Dateien haben – für Ihre Abbildungen beispielsweise –, können Sie diese in einem Unterverzeichnis dieses Hauptverzeichnisses ablegen. Das Ziel besteht darin, all Ihre Dateien an einem einzigen Ort aufzubewahren, anstatt sie über das ganze Laufwerk zu verteilen. Haben Sie nun Ihre Dateien abgelegt, können Sie all die Verbindungen in Ihren Dateien so einrichten, dass sie relativ zu diesem Verzeichnis sind. Folgen Sie diesen Tipps, so haben Sie gute Chancen, dieses Verzeichnis auf verschiedene Server zu verschieben, ohne dabei die Verbindungen zu zerbrechen.

Führen Sie eine Standard-Indexdatei mit korrekten Dateinamen

Web-Server haben normalerweise eine Standard-Indexdatei, die geladen wird, wenn ein URL mit einem Verzeichnis anstelle eines Dateinamens endet. Im vorigen Abschnitt war eine der Fragen an den Web-Master, welchen Namen diese Datei hat. Für die meisten Web-Server wird diese Datei `index.html` genannt (`index.htm` bei DOS). Ihre Homepage oder die höchste Ebene im Index jeder Präsentation sollte diesen Namen führen, damit der Server weiß, welche Seite er als Standardseite schicken soll. Jedes Unterverzeichnis sollte ebenfalls, wenn es HTML-Dateien enthält, eine Standarddatei besitzen. Wenn Sie diesen Standarddateinamen einsetzen, wird das auch den URL dieser Seite verkürzen, da Sie nicht den tatsächlichen Dateinamen anzugeben brauchen. Wenn Ihr URL zum Beispiel `http://www.meinserver.com/www/index.html` ist, wäre der verkürzte nur noch `http://www.meinserver.com/www/`.

Jede Ihrer Dateien sollte zudem eine passende Endung besitzen, die anzeigt, um welche Art von Datei es sich handelt, sodass der Server sie dem passenden Dateityp zuordnen kann. Wenn Sie dem Buch so weit gefolgt sind, sollten alle Ihre Dateien schon diese gewisse Endung haben, sodass hier kein Problem entsteht.

Die Liste in Tabelle 19.1 zeigt Ihnen noch einmal alle gängigen Dateierweiterungen, die Sie für Ihre Dateien und Media verwenden können.

Format	Endung
HTML	.html, .htm
ASCII-Text	.txt
PostScript	.ps
GIF	.gif
JPEG	.jpeg, .jpg
AU Audio	.au
WAV Audio	.wav
MPEG Audio	.mp2
MPEG Video	.mpeg, .mpg
QuickTime Video	.mov
AVI Video	.avi

Tabelle 19.1: Datei-Typen und -Endungen

Wenn Sie spezielle Medien in Ihrer Web-Präsentation verwenden, die in dieser Liste nicht enthalten sind, müssen Sie Ihren Server unter Umständen so konfigurieren, dass er mit diesem Dateityp umzugehen kann. Darüber werden Sie später in diesem Kapitel noch mehr erfahren.

Die Dateien installieren

Haben Sie alles organisiert? Dann müssen Sie nur noch alles auf dem Server an Ort und Stelle bringen. Wenn der Server die Dateien erreichen kann, sind Sie offiziell auf dem Web publiziert. Das ist alles.

Aber wo ist der angemessene Ort auf Ihrem Server? Das hätten Sie von Ihrem Web-Master erfahren sollen. Von ihm hätten Sie auch erfahren sollen, wie Sie zu diesem speziellen Ort gelangen, ob Sie einfach Dateien kopieren müssen, ob Sie sie mit FTP auf den Server senden sollen oder eine andere Methode verwenden müssen.

Hinweise zum Verschieben von Dateien zwischen verschiedenen Systemen

Wenn Sie einen Web-Server nutzen, den jemand anderes eingerichtet hat, dann müssen Sie wahrscheinlich Ihre Web-Dateien von Ihrem System auf das andere übertragen, mit FTP, ZModem oder einer anderen Methode. Obgleich die HTML-Tags innerhalb Ihrer Dateien vollständig portabel sind, hat doch das wirkliche Verschieben von Plattform zu Plattform so seine Tücken. Im Einzelnen achten Sie auf die folgenden Dinge:

- ▶ **Übertragen Sie alle Dateien im Binär-Modus.** Ihr FTP- oder Datei-Transferprogramm kann Ihnen die Möglichkeit, Dateien im binären oder im Textmodus zu senden (oder noch ganz andere Möglichkeiten) zur Verfügung stellen. Transferieren Sie immer alles – also alle HTML-Dateien, Bilder und alle anderen Medien – im binären Format (auch die Dateien, die ja eigentlich Text sind; Sie können Textdateien problemlos im binären Modus übertragen).

Wenn Sie einen Macintosh verwenden, wird Ihr Übertragungsprogramm Ihnen wahrscheinlich alle möglichen Optionen mit solchen Namen wie MacBinary, AppleDouble oder anderen seltsamen Namen geben. Vermeiden Sie sie alle. Die Option, die Sie wollen, heißt einfach `binary` oder `raw data`. Wenn Sie Dateien in irgendeinem anderen Format übertragen, werden Sie vielleicht nicht funktionieren, wenn sie auf die andere Seite gelangen.

- ▶ **Beachten Sie Beschränkungen in Bezug auf Dateinamen.** Wenn Sie Ihre Dateien zu oder von DOS-Systemen bewegen, müssen Sie auf die gefürchteten 8.3 Acht geben – die DOS-Regel, die besagt, dass Dateinamen nur acht Zeichen lang sein und eine drei Zeichen lange Endung haben dürfen. Wenn Ihr Server ein PC ist und Sie Ihre Dateien auf einem anderen System geschrieben haben, müssen Sie Ihre Dateien und die dahin führenden Links möglicherweise umbenennen, um sich an die richtigen Dateinamenkonventionen zu halten. (Dateien andersherum, von einem PC auf ein anderes System zu bewegen, ist meist kein Problem.)

Verlagern Sie Dateien von einem Macintosh auf ein anderes System, dann dürfen die Dateinamen keine Leerzeichen oder andere komische Zeichen enthalten. Halten Sie Ihre Dateinamen so kurz wie möglich, und verwenden Sie lediglich Buchstaben und Ziffern, dann ist alles in Ordnung.

- ▶ **Achten Sie auf Groß- und Kleinbuchstaben** Einige Betriebssysteme und Dateiverarbeitungsprogramme zeigen Dateinamen alle in Kleinbuchstaben an (wie zum Beispiel `myfile.html`). Eigentlich sollten jedoch Dateinamen, die auf DOS basieren, alle in Großbuchstaben geschrieben werden (wie zum Beispiel `MYFILE.HTML`). Wenn die Links im Code Ihrer Webseite in Kleinbuchstaben geschrieben sind (was normalerweise der Fall ist), werden Sie gebrochene Links erhalten,

wenn Sie die Webseiten zu einem Server schicken, in dem die URLs Groß- und Kleinschreibung unterscheiden (wie zum Beispiel UNIX-Server). Prüfen Sie Ihre Dateien auf diesen Punkt hin, bevor Sie sie zu Ihrer Site übertragen.

- **Achten Sie auf Wagenrücklauf und Zeilenvorschub.** Unterschiedliche Systeme benutzen verschiedene Methoden zum Beenden einer Zeile; der Macintosh verwendet den Wagenrücklauf (*carriage return*, CR), Unix den Zeilenvorschub (*linefeed*, LF), und DOS nimmt beides zusammen (CR/LF). Wenn Sie Dateien vom einen System auf ein anderes übertragen, so werden die Zeilenende-Markierungen häufig passend umgesetzt, bisweilen leider aber auch nicht. Das kann dazu führen, dass Ihre Dateien auf dem System, auf das sie übertragen wurden, mit doppeltem Zeilenabstand erscheinen oder komplett in einer Zeile. Meistens ist das nicht weiter wichtig, da Browser falsche Wagenrücklauf- und Zeilenvorschub-Zeichen in Ihren HTML-Dateien ignorieren. Das Vorhandensein oder die Abwesenheit eines der beiden ist nicht so furchtbar wichtig. Wo es jedoch wesentlich sein kann, ist ein Textabschnitt, den Sie mit `<pre>` ausgezeichnet haben; es kann sein, dass Ihr sorgfältig formatierter Text, der auf der einen Plattform so gut aussah, nach dem Verschieben auf eine andere Plattform nicht mehr ganz so gut formatiert aussieht. Wenn Sie Zeilenende-Probleme haben, haben Sie verschiedene Möglichkeiten, diese zu beheben. Einige Text-Editoren erlauben Ihnen, ASCII-Dateien in einem Format für eine andere Plattform zu speichern. Wenn Sie wissen, auf welcher Plattform Sie sie bewegen wollen, können Sie Ihre Dateien für diese Plattform vorbereiten, bevor Sie sie dorthin übertragen. Wenn Sie HTML-Dateien auf ein Unix-System verschieben wollen, können kleine Filter-Programme zum Konvertieren der Zeilenenden namens `dos2unix` und `unix2dos` diese Aufgabe für Sie erledigen. Macintosh-Dateien können unter Unix mit dem folgenden Programmaufruf ins Unix-Format konvertiert werden:

```
tr '\015' '\012' < altdatei.html > neudatei.html
```

Dabei ist `altdatei.html` die Original-Datei mit dem Zeilenende-Problem, und `neudatei.html` ist der Name der neu erzeugten Datei.

Remote-Management-Tools

Neue Tools ermöglichen Ihnen die Verwaltung und Aktualisierung der Inhalte Ihrer Seiten aus der Ferne auf einem entfernten Server (Remote Server). Am weitesten verbreitet sind Tools von Netscape und Microsoft.

FrontPage von Microsoft ist ein Web-Entwicklungs-Tool, das auf kleine bis mittlere Web-Sites ausgerichtet ist. FrontPage bietet einen WYSIWYG-Seiten-Editor, einen Site-Manager, um die Dokumentenstruktur und die Links zu verwalten, sowie einige Server-Erweiterungen, die mit einer ganzen Reihe von Servern verwendet werden können (von Servern auf Windows-Basis von Microsoft und Netscape bis hin zu Unix-Servern).

Diese Erweiterungen erlauben es dem Web-Master, seiner Site eine Vielzahl von Merkmalen, wie zum Beispiel interaktive Diskussionsgruppen und weitere interaktive Features, hinzuzufügen. Diese Erweiterungen erlauben es aber auch, FrontPage für den Up-Load von Dateien auf den Server, sobald Sie Änderungen am Inhalt Ihrer Site vornehmen, zu verwenden. FrontPage ermöglicht Ihnen das Veröffentlichen Ihrer Web-Site bei Verwendung eines Remote-Servers, unabhängig davon, ob er die FrontPage-Server-Erweiterungen installiert hat oder nicht.

Ganz ähnlich ist das LiveWire von Netscape es LiveWire enthaltene Tool mit dem Namen SiteManager, das es erlaubt, neue Inhalte auf einen entfernten Server zu laden. Anders als FrontPage ist LiveWire für die Verwendung mit FastTrack Netscape und Enterprise-Web-Servern entwickelt worden. LiveWire bietet eine Scriptsprache für den serverseitigen Einsatz, die auf JavaScript basiert und nur mit Netscape-Servern zusammenarbeitet. Das SiteManager-Tool kann dagegen zur Verwaltung von Dokumentstrukturen und Links auf beliebigen Sites verwendet werden und verfügt auch über die Möglichkeit, Inhalte via FTP auf einen Server zu übertragen.

Andere Tools zur Entwicklung und Verwaltung von Sites, wie zum Beispiel Fusion von NetObjects (<http://www.netobject.com>), haben die Möglichkeit, offline zu entwickeln und anschließend die Inhalte auf einem entfernten Server zu aktualisieren.

Wie lautet meine URL?

Nun ist es so weit, Sie haben einen Web-Server. Ihre Webseiten sind installiert und fertig zum Abruf. Alles, was noch bleibt, ist, die Welt auf Ihre Präsentation loszulassen. Alles, was Sie jetzt noch brauchen, ist ein URL.

Wenn Sie einen kommerziellen Web-Server benutzen oder einen Server, den jemand anderes betreibt, dürften Sie Ihren URL recht leicht herausbekommen, indem Sie einfach den Systemverwalter fragen (und das war ja auch einer der Punkte, über den Sie Ihren Web-Master fragen sollten). Andernfalls müssen Sie es selbst herausbekommen. Glücklicherweise ist das nicht so schwer.

Wie ich in Kapitel 5, »Alles über Verknüpfungen«, bemerkt habe, bestehen URLs aus drei Teilen: dem Protokoll, dem Systemnamen und dem Pfad zu der Datei. Um jeden dieser drei Teile zu bestimmen, stellen Sie sich die folgenden Fragen:

- **Was verwenden Sie, um Ihre Dateien anzubieten?** Wenn Sie einen richtigen Web-Server benutzen, ist Ihr Protokoll `http`. Wenn Sie FTP oder Gopher verwenden, wäre das Protokoll `ftp` bzw. `gopher`. (Ist es nicht einfach?)

- **Wie heißt Ihr Server?** Dies ist der Netzwerk-Name der Maschine, auf welcher der Web-Server läuft, und er beginnt üblicherweise mit `www`; zum Beispiel `www.mein-system.de` oder `www.netcom.com`. Beginnt der Name nicht mit `www`, ist das auch nicht schlimm; es hat keinen Einfluss darauf, ob die Leute an Ihre Dateien kommen können oder nicht. Beachten Sie, dass der Name, den Sie benutzen werden, der volle Systemname ist, das heißt der Name, den die Leute, die sich anderswo auf dem Web befinden, benutzen würden, um zu Ihrem Server zu gelangen, was sich von dem Namen unterscheiden kann, mit dem Sie zu Ihrem Web-Server gelangen. Dieser Name wird normalerweise verschiedene Teile haben und auf `.com`, `.edu` oder Ihren Staaten-Code enden (z.B. `.de`, `.uk`, `.fr` usw.). Bei manchen SLIP- oder PPP-Verbindungen haben Sie womöglich nicht einmal einen Netzwerknamen, sondern nur eine Nummer – so etwas ähnliches wie `193.96.188.67`. Sie können auch das als Netzwerk-Namen benutzen.

Wenn der Web-Server für einen anderen Port als 80 installiert ist, müssen Sie diese Nummer ebenfalls kennen. Ihr Web-Master wird sich da auskennen.

- **Wie lautet der Pfad zu meiner Homepage?** Der Pfad zu Ihrer Homepage beginnt häufig in dem Verzeichnis, in dem die Webseiten gespeichert sind (dies wird bei der Server-Installation festgelegt) und kann die oberste Ebene des Dateisystems sein oder auch nicht. Wenn Sie zum Beispiel Ihre Dateien in das Verzeichnis `/meinsystem/www/dateien/meine_dateien` gesteckt haben, kann der Pfad-Name zu Ihrem URL einfach nur `/meine_dateien` sein. Dies hängt davon ab, wie Ihr Server konfiguriert ist, wenn Sie es also nicht herausfinden können, sollten Sie Ihren Systemverwalter fragen. Ist Ihr Web-Server so eingerichtet, dass Sie Ihr Home-Verzeichnis (bzw. ein Unterverzeichnis davon) zum Speichern von Webseiten benutzen können, so können Sie sich mit Hilfe der Tilde (`~`) auf diese Dateien in Ihrem Home-Verzeichnis beziehen. Den Namen des Unterverzeichnisses, das Sie dafür eingerichtet haben, müssen Sie nicht mit im URL angeben. Ist also zum Beispiel die Webseite `heim.html` in einem Unterverzeichnis namens `public_html` in Ihrem Home-Verzeichnis abgelegt, so wäre der Pfad zu dieser Datei in Ihrem URL: `/~ihurname/home.html`

Sie kennen nun alle drei Bestandteile, und so können Sie den URL zusammenbauen. Sie werden sich aus Kapitel 5 erinnern, dass ein URL etwa so aussieht:

```
protokoll://maschinen_name.de:port/pfad
```

Sie sollten nun in der Lage sein, Ihre Werte für jedes dieser Elemente an die passende Stelle der URL-Struktur zu setzen. Zum Beispiel:

```
http://www.meine_kiste.de/www/einstieg/index.html
ftp://ftp.netcom.com/pub/le/lemay/index.html
http://www.commercialweb.com:8080/~lemay/index.html
```

Testen, testen und nochmals testen

Nun sind Ihre Seiten also im Netz verfügbar, und Sie können die Gelegenheit wahrnehmen, um sie auf so vielen Plattformen wie möglich zu testen und mit so vielen Browsern, wie Sie kriegen können. Erst, wenn Sie gesehen haben, wie Ihre Dokumente auf unterschiedlichen Plattformen aussehen, werden Sie begreifen, wie wichtig es ist, Ihre Dokumente so zu gestalten, dass sie auf so vielen Plattformen und in so vielen Browsern wie möglich gut aussehen.

Probieren Sie es und schauen Sie ... Die Ergebnisse könnten Sie überraschen. In Kapitel 21 werden Sie lernen, wie man Fehler auf einer Site behebt, die bereits in einem Remote-Server veröffentlicht ist. Wir werden in diesem Kapitel aber über oft vorkommende Probleme in der Praxis sprechen.

Fehler finden

Was passiert, wenn Sie alle Ihre Dateien zum Server geladen haben, Ihre Homepage in Ihrem Browser laden und etwas schiefgeht? Suchen Sie zuerst hier.

Der Server ist unerreichbar

Wenn Ihr Browser nicht mal den Server erreichen kann, können Sie dieses Problem wahrscheinlich nicht selber beheben. Vergewissern Sie sich, dass Sie den richtigen Server-Namen verwenden und dass es ein vollständiger Host-Name ist (der meist mit .com, .edu, .net oder einer anderen gebräuchlichen Endung aufhört). Versichern Sie sich auch, dass Sie den URL korrekt eingetippt haben und dass Sie das richtige Protokoll verwenden. Wenn Ihr Web-Master Ihnen mitgeteilt hat, dass Sie eine Portnummer im URL mitangeben müssen, dann vergewissern Sie sich, dass Sie die Portnummer nach dem Host-Namen im URL aufnehmen.

Testen Sie auch, ob Ihre Netzwerkverbindung funktioniert. Können Sie andere Server erreichen? Können Sie die oberste Ebene Ihrer Sites erreichen?

Wenn keine dieser Methoden das Ziel erreicht, ist vielleicht Ihr Server nicht funktionsfähig oder reagiert nicht. Rufen Sie den Web-Master an, und bitten Sie um Aufklärung.

Die Dateien sind unerreichbar

Was ist, wenn alle Ihre Dateien als `Not Found` oder `Forbidden` (nicht gefunden; Zugang verwehrt) erscheinen? Überprüfen Sie zuerst Ihren URL. Wenn Sie einen URL mit einem Verzeichnisnamen verwenden, ersetzen Sie letzteren mit einem echten Dateinamen, und sehen Sie, ob Sie damit mehr Glück haben. Überprüfen Sie nochmals den Pfad zu Ihren Dateien; erinnern Sie sich, dass der Pfad im URL sich vom Pfad auf dem tatsächlichen Laufwerk unterscheiden kann. Behalten Sie auch im Hinterkopf, dass Groß- und Kleinschreibung einen Unterschied machen. Wenn Ihre Datei `MeineDatei.html` heißt, prüfen Sie, ob Sie nicht vielleicht `meinedatei.html` oder `Meinedatei.html` getippt haben.

Wenn der URL korrekt zu sein scheint, prüfen Sie als Nächstes die Zugriffsberechtigungen (permissions). Auf Unix-Systemen sollten alle Ihre Verzeichnisse von der ganzen Welt ausführbar sein (world-executable), und Ihre Dateien sollten für jedermann lesbar sein (world-readable). Sie können die Zugriffsberechtigungen korrigieren, indem Sie die folgenden Befehle eingeben:

```
chmod 755 dateiname
chmod 755 verzeichnisname
```

Die Bilder sind unerreichbar

Sie können Ihre HTML-Dateien gut erreichen, aber alle Ihre Bilder erscheinen als Bildsymbole oder als gebrochene Bildsymbole. Vergewissern Sie sich zuerst, dass die Verweise auf Ihre Bilder korrekt sind. Wenn Sie relative Pfad-Namen eingesetzt haben, sollte dies kein Problem darstellen. Wenn Sie absolute Pfad-Namen oder File-URLs angegeben haben, kann es passieren, dass die Verweise auf Ihre Bilder nicht mehr aktuell sind, als Sie die Dateien auf den Server geschoben haben (ich habe Sie ja gewarnt...).

Bei einigen Browsern (Netscape ist da ein gutes Beispiel) können Sie ein Bild mit dem rechten Maus-Button auswählen (beim Macintosh halten Sie den Maus-Button heruntergedrückt) und erhalten dadurch ein Pop-up-Fenster. Der Menü-Punkt »Dieses Bild betrachten« wird versuchen, das Bild direkt zu laden, wodurch Sie den URL des Bildes herausfinden können (oder den URL, wo der Browser das Bild vermutet), der sich von dem Ort unterscheiden kann, an dem Sie das Bild wähen. Mit diesem Trick können Sie oft seltsame Probleme lösen, die bei Pfad-Namen auftauchen.

Wenn die Verweise alle gut aussehen und die Bilder auf Ihrem lokalen System einwandfrei funktionierten, ist die einzige verbleibende Möglichkeit ein Irrtum bei der Übertragung der Dateien von einem System zum anderen. Wie bereits erwähnt, sollten Sie alle Bilddateien in binärem Format übertragen. Wenn Sie sich auf einem Mac

befinden, übertragen Sie alle Daten als `raw data` oder einfach `data`; lassen Sie die Finger vom MacBinary- oder AppleDouble-Format, oder Sie werden auf der anderen Seite Schwierigkeiten bekommen.

Die Verknüpfungen funktionieren nicht

Wenn Ihre HTML- und Bilddateien einwandfrei funktionieren, Ihre Links jedoch nicht, dann haben Sie wahrscheinlich Pfad-Namen für diese Verknüpfungen eingesetzt, die nur auf Ihrem lokalen System funktionieren – zum Beispiel haben Sie möglicherweise absolute Pfad-Namen oder File-URLs eingesetzt, um die Dateien, die Sie erreichen wollen, darzustellen. Wie ich schon bei den Bildern erwähnt habe, sollte dies kein Problem hervorrufen, wenn Sie relative Pfad-Namen einsetzen und File-URLs vermeiden.

Die Dateien werden falsch dargestellt

Nehmen wir mal an, Sie haben eine HTML-Datei oder eine Datei in einem anderen Format, die auf Ihrem lokalen System gut angezeigt wird und/oder deren Verknüpfung einwandfrei funktioniert. Wenn Sie die Datei jedoch auf den Server laden und versuchen, sie zu betrachten, zeigt der Browser Ihnen wirre Zeichen an – beispielsweise stellt er den HTML-Code selbst dar, anstatt die HTML-Datei anzuzeigen, oder er versucht, eine Bild- oder Mediendatei als Text abzubilden.

Es gibt zwei Fälle, bei denen das auftreten kann. Der erste tritt auf, wenn Sie die falschen Dateierendungen für Ihre Dateien benutzt haben. Vergewissern Sie sich, eine der richtigen Dateierendungen mit der korrekten Groß- und Kleinschreibung zu verwenden.

Die andere Möglichkeit, bei der dies vorkommen kann, ist, wenn der Server nicht richtig für die Dateien konfiguriert wurde. Wenn Sie zum Beispiel auf einem DOS-System arbeiten, auf dem alle Ihre HTML-Dateien auf `.htm` enden, dann kann Ihr Server unter Umständen nicht verstehen, dass `.htm` eine HTML-Datei sein soll (die meisten modernen Server verstehen es, einige ältere jedoch nicht). Oder vielleicht verwenden Sie einen neuen Medientyp, den Ihr Server noch nicht kennt. So oder so versucht Ihr Server, seinen Standard-Inhaltstyp auf Ihre Dateien anzuwenden (meistens `text/plain`), den Ihr Browser dann zu verstehen versucht (was ihm oft nicht gelingt).

Um dies zu korrigieren, müssen Sie Ihren Server so konfigurieren, dass er mit den Dateierendungen umgehen kann, die den von Ihnen angewandten Medientyp bezeichnen. Wenn Sie mit einem fremden Server arbeiten, sollten Sie den Web-Master kontaktieren und ihn dazu veranlassen, alles richtig einzurichten. Dazu braucht Ihr Web-Master zwei Arten von Informationen: die Dateierendungen, die Sie einsetzen, und den Inhaltstyp, den Sie den Endungen zugeordnet haben.

Zusammenfassung

In diesem Kapitel haben Sie den letzten Punkt gelernt, der zur Erstellung einer Webseite notwendig ist: das Veröffentlichen Ihrer Arbeit im World Wide Web mit Hilfe eines Web-Servers, der entweder von Ihnen installiert ist oder von einem ISP erhältlich ist. Hier haben Sie gelernt, was ein Web-Server tut, und wo man ihn erhält, wie man Dateien organisiert und sie auf dem Server installiert, und wie Sie Ihre URL finden und sie zum Testen Ihrer Seiten verwenden können.

Workshop

Was Sie von nun an lernen, ist im Grunde genommen nur eine Vertiefung Ihrer erlangten Kenntnisse. Sie werden einfach mehr Features (Interaktivität, Formulare) in Ihre Site integrieren, die Sie bereits ins Web gebracht haben! Herzlichen Glückwunsch!

Fragen und Antworten

F Ich habe meine Seite bei einem ISP veröffentlicht, der mir gut gefällt. Meine URL ist so etwas wie `http://www.thebesisp.com/users/mypages/`. Ich möchte anstelle dieser URL meinen eigenen Kundennamen haben – vielleicht etwas Ähnliches wie `http://www.mypages.com/`. Wie kann ich das erreichen?

A Sie haben zwei Möglichkeiten. Die einfache Möglichkeit ist, Ihren ISP zu fragen, ob Sie Ihren eigenen Domain-Namen haben können. Viele ISPs erstellen Ihr Domain so, dass Sie immer noch damit arbeiten können und den Service nutzen können – nur ihre URL ändert sich. Beachten Sie, dass ein eigener Domain-Name etwas teurer ist, aber wenn Sie diese URL unbedingt brauchen, dann dürfte dieser Weg für Sie in Ordnung gehen. Viele ISPs verlangen für diese Art von Service 20 Dollar im Monat und es kostet momentan 70 Dollar, um sich einen Domain-Namen für zwei Jahre registrieren zu lassen.

Die andere Möglichkeit ist das Erstellen Ihres eigenen Domain-Namens für Ihren eigenen Server. Diese Möglichkeit könnte viel teurer als die erste Möglichkeit sein, und erfordert letztendlich mehr Hintergrundwissen in der Verwaltung von Basic Network. Über das Arbeiten mit Network werden Sie mehr am Tag 20 erfahren.

F Ich habe alle meine Bilddateien auf Macintosh erstellt, sie mittels des Fetch FTP Programms zum UNIX-Server hochgeladen, sie alle getestet und es funktioniert. Nun bekomme ich E-Mails von meinen Besuchern, die mir sagen, dass keines meiner Bilder funktioniert. Was ist hier los?

A Wenn Sie die Dateien mit Fetch uploaden, können Sie normalerweise aus einer Liste den Übertragungsmodus auswählen. Die Standardeinstellung ist MacBinary. Vergewissern Sie sich, dass Sie diese Standardeinstellung auf Raw Data geändert haben.

MacBinary-Dateien funktionieren gut, wenn Sie auf einem Macintosh angezeigt werden. Da Sie, wie ich vermute, Ihre Site mit Macintosh testen, funktionieren die Bilder auch dann; allerdings funktionieren sie in keinem anderen System. Um sicherzugehen zu können, dass Ihre Dateien auch plattformübergreifend funktionieren, sollten Sie sie als Raw Data uploaden.

F Ich habe meine Dateien in einem DOS-System erstellt und die .htm Erweiterung verwendet, wie Sie in diesem Buch bereits sagten. Jetzt habe ich meine Dateien auf einem UNIX-System, das von meinem Arbeitgeber angeboten wird, veröffentlicht. Das Problem ist Folgendes: Wenn ich meine Seiten mit Hilfe meines Browsers betrachten möchte, erscheint nur der HTML-Code für diese Seiten, aber nicht das formatierte Ergebnis! Auf meinem System zu Hause funktionierte alles gut. Was ist schief gelaufen?

A Dieses Problem taucht bei älteren Servern öfter auf. Ihr Server ist nicht so eingestellt, dass er erkennen kann, dass Dateien mit der .htm-Erweiterung HTML-Dateien sind. Deshalb senden diese Browser stattdessen den Standard-Inhaltstyp (text/plain) zurück. Wenn Ihr Browser dann eine Ihrer Dateien vom Server liest, liest er diesen Inhaltstyp und hält dies für eine Textdatei. Ihr Server vermässelt also alles.

Es gibt mehrere Lösungen zu diesem Problem. Das einfachste ist, Ihrem Webmaster mitzuteilen, dass die Serverkonfiguration geändert werden soll, sodass .htm-Dateien als HTML-Dateien erkannt werden. Normalerweise ist dies eine simple Methode, die wie durch ein Wunder das Funktionieren Ihrer Dateien ermöglicht.

Wenn Sie Ihren Webmaster nicht ausfindig machen können, oder er, aus welchen Gründen auch immer, diese Änderung nicht durchführen möchte, bleibt Ihnen nur noch folgende Möglichkeit: Ändern Sie alle Dateinamen, nachdem Sie sie auf das UNIX-System upgeloadet haben. Beachten Sie, dass Sie auch alle Links innerhalb dieser Datei ändern müssen. (Es wäre wohl viel angebrachter, Ihren Webmaster doch noch zu einer Konfigurationsänderung zu überreden...)

Quiz

1. Was ist die Grundfunktion eines Web-Servers?
2. Nennen Sie ein paar Möglichkeiten, wo man Internet – Verbindungen erstellen kann.
3. Was ist eine Standard Index-Datei, und was ist der Vorteil ihrer Verwendung in allen Verzeichnissen?
4. Auf was sollten Sie beim Uploaden Ihrer Internet Site beachten?
5. Was müssen Sie unbedingt nach dem Uploaden Ihrer Webseiten kontrollieren?

Antworten

1. Ein Web-Server ist ein Programm, das auf einer Maschine im Internet (oder Intranet) sitzt. Ein Server hält Daten bereit und verwaltet die Datenübertragung.
2. Sie können eine Internet-Verbindung in der Schule oder Arbeit, im geschäftlichen/öffentlichen Internet oder im Web-Service erstellen oder Ihren eigenen Web-Server installieren.
3. Die Standard-Datei wird geladen, wenn eine URL mit einem Verzeichnisnamen anstelle eines Dateinamens endet. Typische Beispiele oder Standard-Index-Dateien sind index.html, index.htm. oder default.htm. Wenn Sie die Standard-Dateinamen verwenden, können Sie eine URL wie `http://www.mysite.com/` anstelle von `http://www.mysite.com/index.html` verwenden, um zur Homepage im Verzeichnis zu gelangen.
4. Übertragen Sie alle Ihre Dateien im Binärcode, achten Sie auf Dateinamen-Einschränkungen, beachten Sie Groß- und Kleinschreibung und die Carriage Returns / Line Feeds.
5. Kontrollieren Sie, ob Ihr Browser die Webseiten auf dem Server herunterladen kann, ob Sie Zugang zu den Dateien auf Ihrer Webseite haben und ob ihre Links und Bilder so funktionieren, wie Sie es von ihnen erwarten. Nachdem Sie sichergestellt haben, dass alles so funktioniert, wie Sie es gerne hätten, laden Sie Freunde und die Familie ein, Ihre Webseite zu testen.

Übungen

1. Fangen Sie schon einmal an, Angebote für einen virtuellen Web-Server, auf dem Sie Ihre Web-Site gerne veröffentlicht hätten, einzuholen. Vergleichen Sie zwei oder drei ISPs, um festzustellen, welche Vorteile Sie jeweils haben.
2. Uploaden und Testen Sie eine Übungsseite (selbst wenn es eine ganz einfache oder leere Seite ist, der Sie erst später einen Inhalt hinzufügen), um den Vorgang zu lernen. Sie werden vielleicht noch ein paar kleine Dinge dazulernen, bevor Sie Ihr Meisterwerk ins Web uploaden.



**Lassen Sie die
Leute wissen,
dass Sie online
sind**

**Woche
3**

Sie haben nun Ihre Site erstellt. Allerdings wird man sie nicht einfach so besuchen. In der Tat, es gibt ungefähr 1 Million Sites und einige von diesen Sites enthalten Tausende von Dokumenten. Es ist also höchst unwahrscheinlich, dass man rein zufällig über Ihre Site stolpert. Wie können Sie also die Leute dazu bringen, Ihre Site zu besuchen? Was muss man tun, damit man auf Ihre Site aufmerksam wird? In diesem Kapitel werden wir Ihnen einige Tipps hierzu geben. Außerdem werden folgende Themen behandelt:

- ▶ Methoden, wie Sie für Ihre Site werben können
- ▶ Listen Sie Ihre Site in den wichtigsten Web-Verzeichnissen auf
- ▶ Listen Sie Ihre Site in den wichtigsten Web-Indizes auf
- ▶ Verwenden Sie Usenet, um Ihre Site anzukündigen
- ▶ Visitenkarten, Briefköpfe und Broschüren
- ▶ Finden Sie heraus, welche Verzeichnisse und ähnliche Webseiten es gibt
- ▶ Protokoll-Dateien und Zähler

Registrieren Sie Ihre Webseiten und werben Sie dafür

Damit man Ihre Web-Site besucht, müssen Sie zuerst einmal Werbung dafür machen. Je mehr Werbung Sie dafür machen, umso größer ist die Aussicht, dass sie auch besucht wird.



Hits ist in der Web-Sprache die Bezeichnung für die Anzahl von Zugriffen, die auf Ihre Web-Site gemacht werden. Es bezieht sich nicht auf die Anzahl der Personen, sondern darauf, wie oft eine Kopie Ihrer Webseite heruntergeladen wurde.

Man kann auf vielerlei Art und Weise für eine Site werben: Sie können sie in Web-Verzeichnissen eintragen, sie in einem Diskussionsforum ankündigen, die URL auf Visitenkarten setzen, und vieles mehr. Auf den folgenden Seiten werden alle diese Methoden eingehender behandelt.

Verzeichnisse von WWW-Sites

Viele Leute haben Schwierigkeiten, wenn Sie zum ersten Mal mit dem World Wide Web arbeiten, zu begreifen, dass es da draußen viele Web-Sites gibt, die sehr gerne eine Chance hätten, einem Hyperlink zu Ihren Webseiten in ihren eigenen Listen zu beinhalten. Und was sie noch weniger begreifen, ist die Tatsache, dass das meistens keine Kosten nach sich zieht.

Es gibt einen einfachen Grund für die Existenz all dieser anscheinend selbstlosen Individuen. Als das World Wide Web noch jung und frisch war, war die beste Methode für eine Person, die ihre Site bekannt machen wollte, andere Web-Entwickler darum zu bitten, ihre Site auf deren Seiten aufzuführen. Als Dank würde sie den Gefallen erwidern. Dieser Brauch hat sich mit der Zeit ein wenig geändert, aber auch heute noch gibt es viele Sites, die nur zu gerne ein Link zu Ihrer Site aufstellen. Wundern Sie sich also nicht, wenn Sie von Zeit zu Zeit eine E-Mail mit der Bitte erhalten, jemanden in Ihrer Liste von Sites aufzunehmen.

Diese kooperative Haltung ist ein beeindruckendes, einzigartiges Merkmal des Web. Anstatt mit anderen ähnlichen Sites wegen der Besucher zu konkurrieren, führen die meisten Webseiten sogar ein Verzeichnis Ihrer Konkurrenten auf. Leider gibt es trotzdem noch ein Problem dabei, einfach mit anderen Sites Hyperlinks auszutauschen. So wie auch am Anfang der Web-Entwicklung brauchen die Leute immer noch eine einzelne Site als Ausgangspunkt. Dazu wurde eine Art von globalem Internet-Verzeichnis benötigt. Gegenwärtig gibt es keine einzelne Site im World Wide Web, den man als das Internet-Verzeichnis überhaupt bezeichnen könnte, jedoch kommen dem einige der größten Verzeichnisse und Büchereien sehr nahe.

Yahoo!

Das bei weitem bekannteste Verzeichnis von Web-Sites ist die Yahoo!-Site (vgl. Abbildung 20.1), der von David File und Jerry Yang erschaffen wurde und unter <http://www.yahoo.com/> erreicht werden kann. Die Site entstand im April 1994 als eine kleine private Liste von David und Jerrys bevorzugten Web-Sites. Seither wuchs er zu einem hochgeachteten Katalog und Index von Web-Sites, der inzwischen sogar zu einer Firma geworden ist.

Yahoo! benutzt einen eleganten vielschichtigen Katalog, um alle Sites zu katalogisieren. Um den Inhalt einer beliebigen Katalogebene zu betrachten, wählen Sie das Hyperlink der übergeordneten Kategorie, die der von Ihnen gesuchten Information am ehesten entspricht, und folgen dann der Kette von verbundenen Seiten zu einer Aufstellung themenverwandter Web-Sites, die der in Abbildung 20.2 gezeigten ähnelt. Sehen Sie nun die gesamte URL dieser Seite:

http://dir.yahoo.com/Computers_and_Internet/Internet/World_Wide_Web/Announcement_Services/

Die Seite, die in dieser Abbildung gezeigt wird, sollten Sie sich unbedingt anschauen. Sie enthält eine Liste von Ankündigungsdiensten und verwandten Webseiten, die Ihnen dabei helfen können, Ihre neue Web-Site bekannt zu machen.

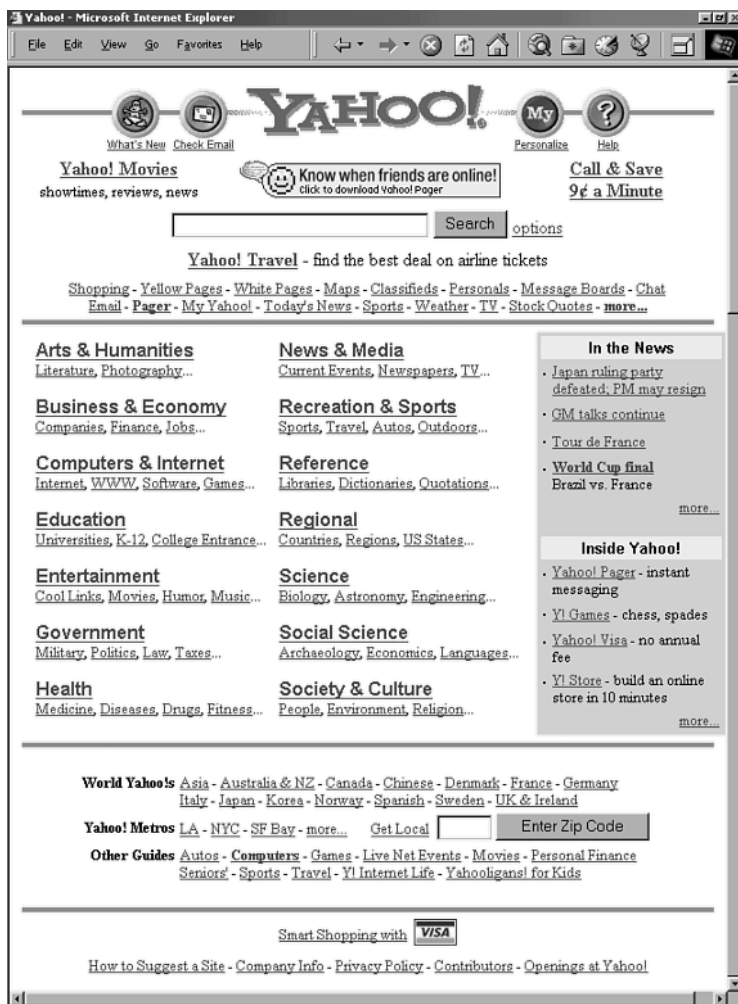


Abbildung 20.1:
Yahoo!

Wenn Sie zum Ende der Seite scrollen, sehen Sie einen Link, der Sie zu einer Site bringt, die sich mit dem Thema »Wie schlage ich eine Site zur Eintragung vor?« beschäftigt. Wenn sie diesen Link anklicken, sehen Sie eine Seite wie in Bild 20.3. (<http://docs.yahoo.com/info/suggest>). Diese Seite liefert komplette Instruktionen und erklärt Ihnen Schritt für Schritt, wie Sie in Yahoo! für Ihre Site werben können. Die Informationsformulare sind einfach zu verstehen.

Ihre Site-Vorschläge werden nach dem Ausfüllen eines vierseitigen Auskunftsfomular von Yahoo! bearbeitet. Schon bald danach wird sich Ihre Site auf der Yahoo!-Liste befinden.

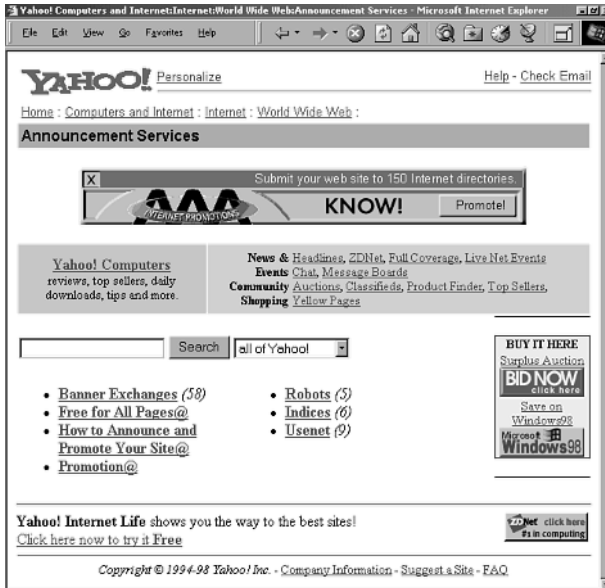


Abbildung 20.2:
Die Kategorie der Ankündigungsdienste in Yahoo!

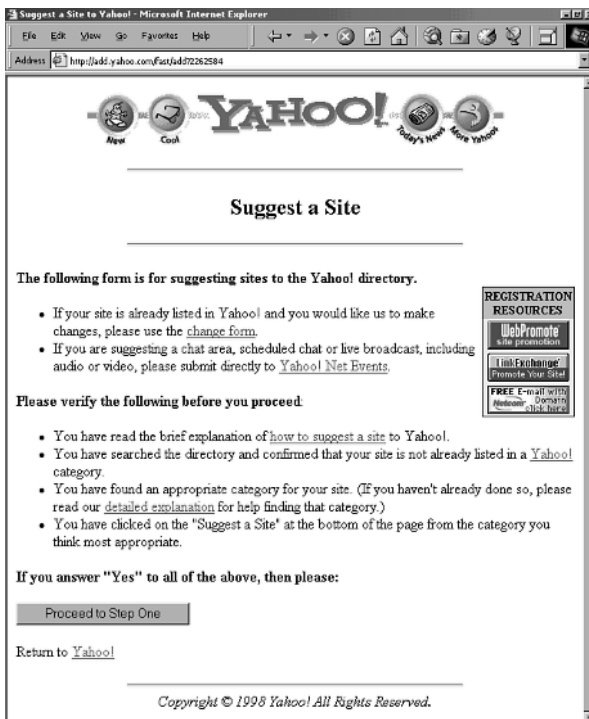


Abbildung 20.3:
Das Formular, mit dem Sie
Yahoo! Ihre URL angeben

Die World Wide Web Virtual Library

Nachdem Sie das Formular abgesendet haben, wird der von Ihnen zuvor ausgewählten Kategorie automatisch ein neues Hyperlink hinzugefügt. Außerdem wird Ihr Site dann in der täglichen und wöchentlichen Neuigkeiten-Liste aufgeführt, die Sie unter <http://www.yahoo.com/New/> finden können.

Die World Wide Web Virtual Library ist da schon etwas selektiver. Nicht jede Site wird hier aufgelistet. Ist Ihre Site nun wirklich sehr gut, sollten Sie es dennoch versuchen. Sie müssen in einer E-Mail Ihre URL der Person unterbreiten, die die Virtual Library verwaltet. Unter <http://conbio.rice.edu/vl/database/output.cfm> erhalten sie die E-Mail Adressen eines jeden Verwalters oder Vlibrarian (Kurzform für Virtual Librarian). Die erste Seite (siehe Bild 20.4) zeigt die ersten 25 Kategorien und enthält ebenfalls ein Link zu den E-Mail-Adressen.

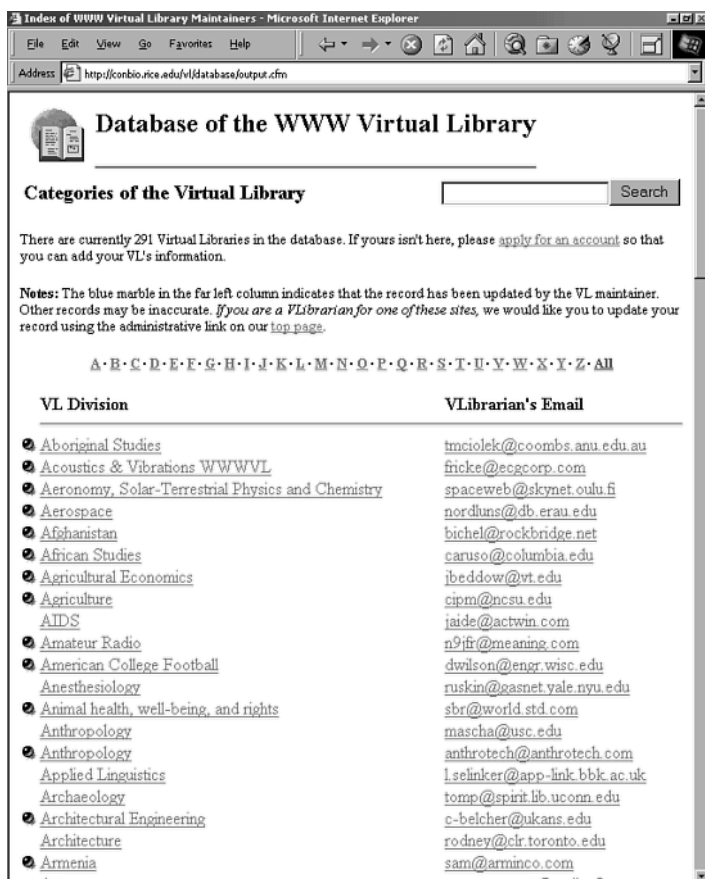


Abbildung 20.4:
Die World Wide Web
Virtual Library

Branchenbuch(Yellow Pages)-Verzeichnisse

Eine weitere beliebte Methode, Ihre Site bekannt zu machen, besteht darin, dass Sie bei einer der wachsenden Branchenbuch-Verzeichnisse registrieren, die jetzt überall im World Wide Web aus dem Boden schießen. Diese Sites kann man sich am besten als das elektronische Äquivalent Ihres örtlichen Branchenbuch-Verzeichnisses vorstellen.

Im Allgemeinen sind Branchenbuch-Verzeichnisse speziell für kommerzielle Web-User gedacht, die ihre Dienste und Fachkenntnisse anbieten wollen. Deshalb bieten die meisten Branchenbücher sowohl kostenlose als auch kostenpflichtige Werbefläche an, wobei die bezahlten Einträge Grafiken, Firmenlogos und erweiterte Layout-Merkmale bieten.

Die GTE Superpages Homepage unter www.superpages.com/ wird in Bild 20.5 gezeigt. Mit dieser Homepage haben Sie Zugang zu zwei Gelbe-Seiten-Verzeichnissen: eines für Business-Informationen, aus den amerikanischen Gelben Seiten entnommen (enthält auch Unternehmen, die keine Web-Sites haben) und eines speziell für Unternehmen, die Web-Sites haben. Beide sind in Kategorien unterteilt und können Ihnen bei der Suche nach Namen und Sitz der Unternehmen behilflich sein.

Private Verzeichnisse

Zusätzlich zu den umfassenden Mainstream-Web-Verzeichnissen gibt es noch viele private Verzeichnisse auf dem Web, die besondere Bedürfnisse erfüllen. Einige dieser Verzeichnisse kümmern sich um die Probleme einzelner, während andere sich mit ganzen Gebieten wie Online-Handel, Bildung, Geschäftsleben und Unterhaltung beschäftigen.

Die beste Methode, solche Verzeichnisse zu finden, besteht darin, ein Internet-Suchwerkzeug wie Lycos (<http://lycos.cs.cmu.edu/>) oder WebCrawler (<http://www.web-crawler.com/>) einzusetzen. Die meisten dieser Verzeichnisse werden auch an Orten wie Yahoo! und der W3 Virtual Library aufgeführt, sodass es sehr hilfreich sein kann, deren Kataloge nach solchen Verzeichnissen zu durchsuchen.

Site-Indizes und Suchmaschinen

Nachdem Sie Ihre Site bei den Hauptverzeichnissen und vielleicht auch bei einigen kleineren bekannt gemacht haben, sollten Sie Ihre Aufmerksamkeit als Nächstes auf die Index- und Suchwerkzeuge richten

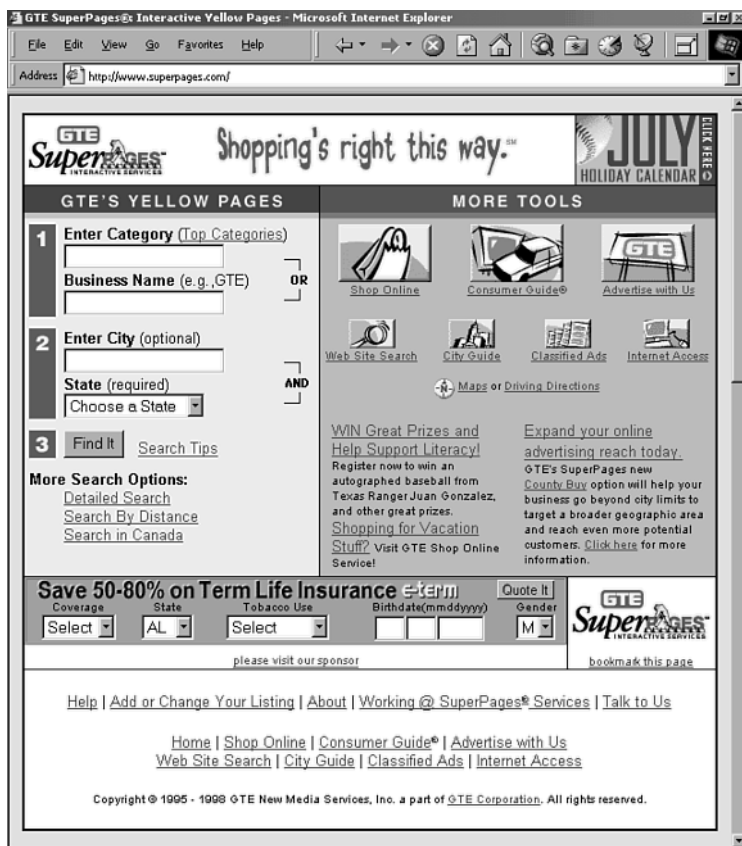


Abbildung 20.5:
Die GTE Super-
Pages-Homepage

Namen und URL der beliebtesten Site-Indizes und Suchmaschinen:

Alta Vist www.altavista.com/

Excite www.excite.com/

HotBot www.hotbot.com/

Lycos www.lycos.com/

Infoseek <http://infoseek.go.com/>

Snap.com www.snap.com/

Anders als Verzeichnisse, die hierarchische Listen von Web-Sites enthalten, die, zu deren Einführung in diese Listen, dem Verzeichnis unterbreitet wurden, verfügen Indizes über großartige Suchmaschinen (manchmal auch Spider genannt). Diese Suchmaschi-

nen durchstreifen das Web und speichern Informationen über jede Site und Seite, die sie auffinden. Die Indizes speichern diese Informationen in einer Site-Datenbank, die Sie durchsuchen können.

Wenn Sie Ihre Site erst im Web publiziert und andere Leute sie besucht haben, können Sie davon ausgehen, dass eine Suchmaschine schließlich auch Ihre Site entdecken und erforschen wird. Sie können die Suchmaschinen jedoch von vornherein wissen lassen, dass Ihre Site existiert und dadurch um so schneller indiziert werden kann. Jede dieser Suchmaschinen liefert Ihnen einen Mechanismus, mit dem Sie den Index Ihrer Site hinzufügen können.

AltaVista

AltaVista ist einer der beliebtesten und schnellsten Web-Indizes. Er wurde von Digital Equipment erstellt und kann unter <http://www.altavista.digital.com/> erreicht werden. AltaVista indiziert einen guten Teil des Web und sticht durch die extrem hohe Geschwindigkeit seines Suchmechanismus hervor. Auf diese Weise wird das Nachschlagen spezifischer Suchbegriffe auf dem Web zu einem schnellen und gründlichen Vorgang.

Sie können Ihre Seite bei AltaVista mit einem Formular unter <http://www.altavista.digital.com/cgi-bin/query?pg=addurl> anmelden (siehe Abbildung 20.6).

Excite

Excite ist als Suchmaschine und Index bekannt geworden, da es eine einzigartige Fähigkeit besitzt: Man kann nach einem Konzept suchen, anstatt einfach nur nach Schlüsselworten. Die Software, die Excite verwendet, um dies zu bewerkstelligen, versucht über einen speziellen Algorithmus die Bedeutung aus der Phrase mit dem zu suchenden Konzept zu extrahieren, um entsprechende Dokumente zu finden. Excite ist im Web unter <http://www.excite.com/> erreichbar. Eigene Seiten können Sie unter http://www.excite.com/Info/add_urlhtml?a-m-t anmelden.

HotBot

Eine Suchmaschine, die seit kurzem sehr beliebt im Web ist. (Unter www.hotbot.com/ zu finden.) Von dieser Seite aus (siehe Bild 20.7) können Sie dann Sites nach Kategorien durchsuchen, verschiedene Site-Sammlungen, Online Shops, sowie Sites geographisch suchen. Mit Shopping Bots kann der Leser auch die Online Shopping Sites aufsuchen, die ihn interessieren.

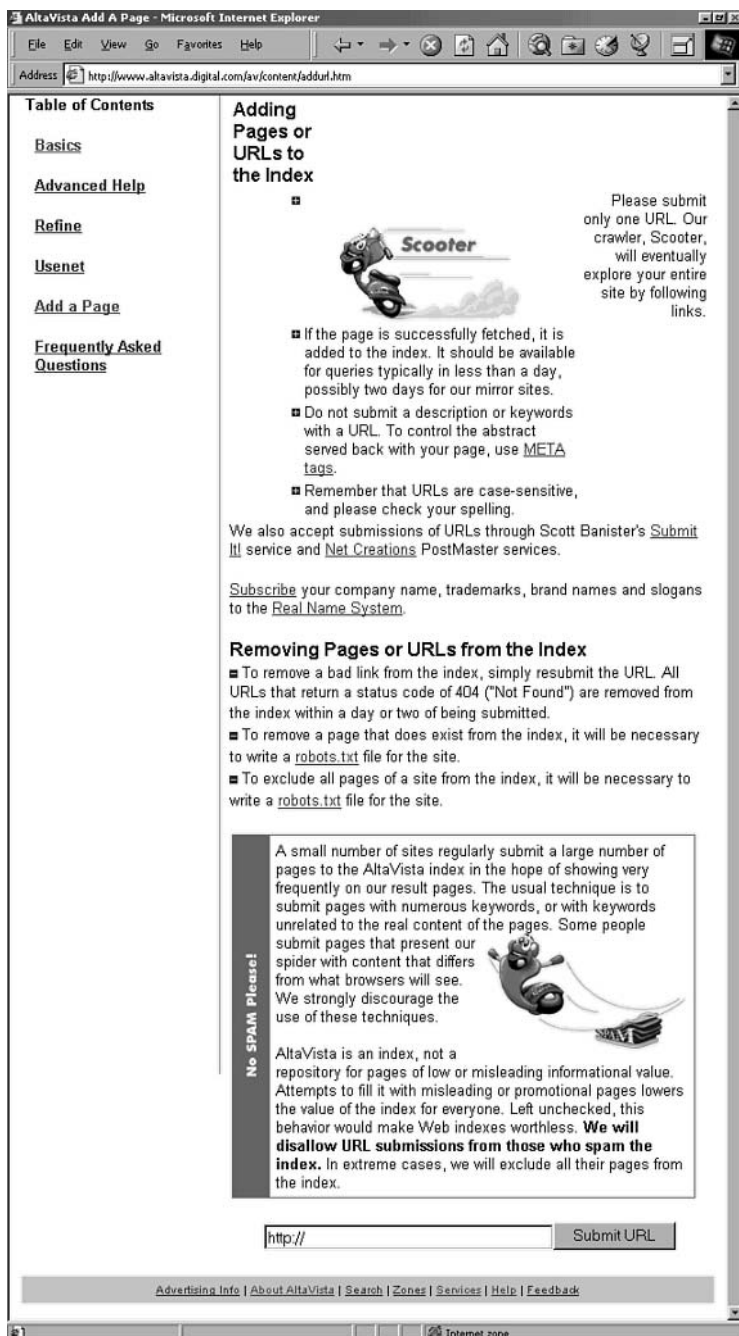


Abbildung 20.6:
Die Seite von Alta-
Vista zum Hinzufü-
gen einer URL

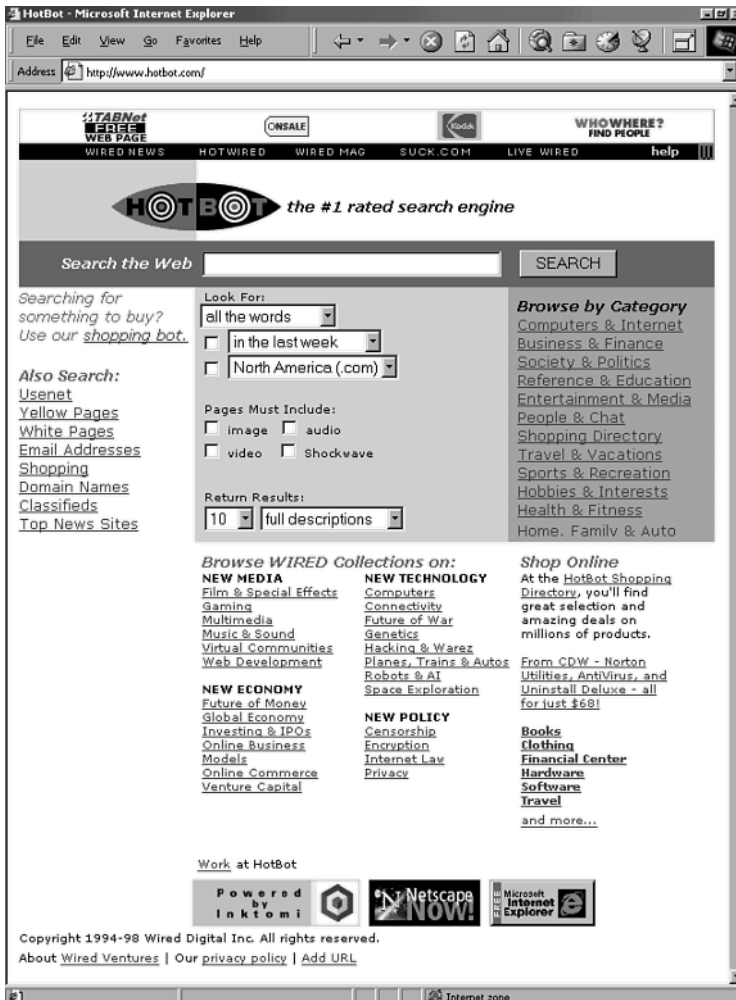


Abbildung 20.7:
Die Homepage von
HotBot

Lycos

Lycos war eine der ersten Suchmaschinen und behauptet immer noch, die umfassendste Abdeckung des Web vorzunehmen. Lycos befindet sich auf <http://www.lycos.com/>, und sein Registrierungsformular, mit dem Sie Ihre Site für die Durchsuchung mittels seiner Suchmaschine anmelden können, befindet sich auf <http://www.lycos.com/register.html>. Für jede Site, die Sie anmelden, müssen Sie die URL und Ihre E-Mail-Adresse angeben.



Abbildung 20.8:
Die Seite zum
Registrieren bei
Lycos

Infoseek

Das Magazin PC Computing verlieh Infoseek (<http://www.infoseek.com/>, siehe Abbildung 20.9) den Titel »Nützlichstes Internet-Werkzeug von 1995« und der Dienst ist seit damals bedeutend gewachsen. So wie Lycos und WebCrawler ist auch Infoseek ein Web-Index-Werkzeug; seine Kraft wird jedoch noch durch die Fähigkeit verstärkt, viele Arten zusätzlicher Dienste und Datenbanken, die nicht Teil des World Wide Web sind, durchsuchen zu können. Diese Funktionalität hat jedoch ihren Preis – nur die Web-Suchmaschine kann kostenlos benutzt werden.


 Abbildung 20.9:
 Infoseek

Werkzeuge, die Ihren URL für Sie einreichen

Neben den Suchwerkzeugen, die gerade behandelt wurden, gibt es noch mindestens 15 andere mit unterschiedlichen Fähigkeiten, und an jede müssen Sie eine eigene Anfrage richten, um sicherzustellen, dass Ihre Site indiziert wird.

Anstatt jedoch die URLs und Einzelheiten dieser Sites aufzuzählen, wird sich diese Diskussion zwei speziellen Webseiten zuwenden, die Ihnen einen guten Teil der Kleinarbeit an den Einreichungsanträgen von Web-Sites an Such-Indizes und -Verzeichnissen abnehmen.

PostMaster2

Die PostMaster2-Site, die in Abbildung 20.10 gezeigt wird und unter <http://www.netcreations.com/postmaster/index.html> gefunden werden kann, ist eine »Alles-inklusive-Einreichungsseite«, die Sie auffordert, alle Einzelheiten auszufüllen, die für ungefähr 25 Web-Indizes und -Verzeichnisse benötigt werden. Darunter fallen viele der Suchmaschinen, die wir schon angesprochen haben. Nachdem Sie das Formular ausgefüllt haben, leitet PostMaster2 Ihre Informationen gleichzeitig an alle diese Sites weiter, sodass Sie nicht jedes einzeln ausfüllen müssen.



Bietet auch eine kommerzielle Version seines Einreichungssystems an, mit der Ankündigung Ihre Site nicht nur an alle Sites der kostenlosen Version, sondern auch an mehr als 200 Magazine, Journale und andere Periodika zu schicken. Der Genuss der kommerziellen Version ist jedoch ein teurer Spaß.

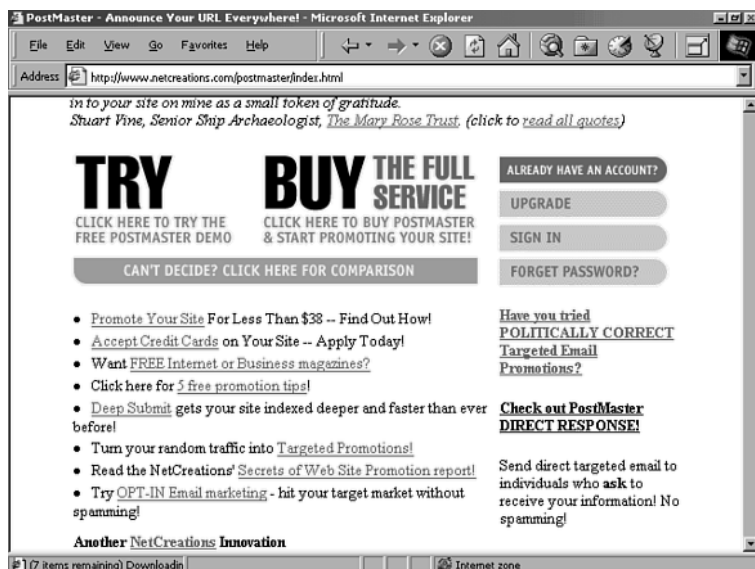


Abbildung 20.10:
PostMaster2

Submit It!

Der Submit It!-Dienst, der von Scott Banister angeboten wird, ähnelt dem PostMaster sehr, da er Ihnen ebenfalls dabei hilft, Ihre URL an verschiedene Verzeichnisse und Such-Indizes weiterzuleiten. Dieser Dienst bietet Ihnen fast die gleichen Leistungen. Was ihn jedoch unterscheidet ist die Art, wie Sie Ihre Informationen einreichen. Bild 20.11 zeigt nur einen Teil aller Suchmaschinen und Verzeichnisse, die derzeit von Submit It! unterstützt werden. Wenn Sie diese Liste ganz sehen wollen, müssen Sie Ihren Browser auf www.submit-it.com/subcats.htm ausrichten.

Submit It! bittet Sie nicht, eine einzige enorme Seite auszufüllen, was etwas ist, das vielen Leuten überwältigend vorkommt. Stattdessen füllen Sie zuerst einige allgemeine Angaben aus, dann wählen Sie nur die Sites aus, an die Sie wirklich Ihre Nachfrage einreichen wollen, und danach füllen Sie die Anträge für einen Site nach der anderen aus.

Um mehr über Submit It! zu erfahren, richten Sie Ihren Web-Browser auf <http://www.submit-it.com/>.

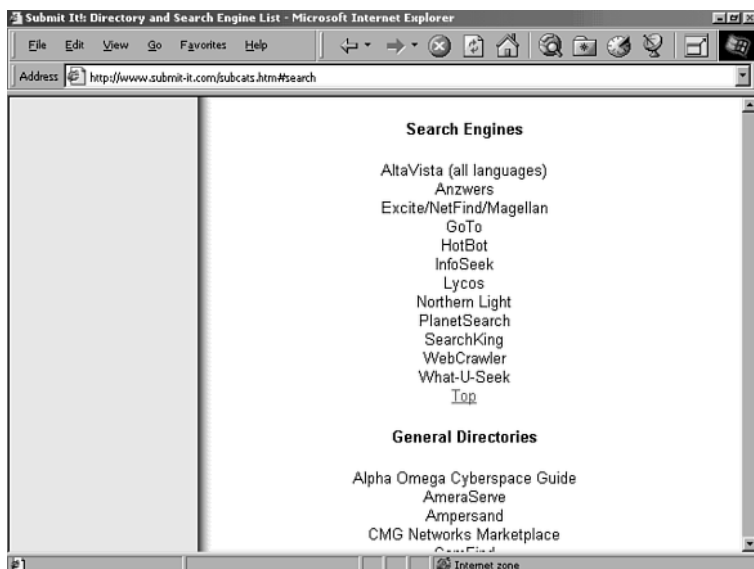


Abbildung 20.11:
Submit It!

Kündigen Sie Ihre Site über das Usenet an

Das World Wide Web ist nicht der einzige Ort auf dem Internet, wo Sie den Start Ihrer neuen Web-Sites bekannt machen können. Viele Leute nutzen die Möglichkeiten einer kleinen Gruppe von Usenet-Newsgroups, die ausdrücklich für Bekanntmachungen gedacht sind. Um diese Newsgroups zu lokalisieren, suchen Sie nach solchen, die auf .announce enden. (Lesen Sie die Dokumentation Ihres Usenet-Newsreaders für Informationen, wie Sie dies machen können.)

Es gibt sogar eine Newsgroup namens comp.infosystems.www.announce, die ausschließlich den Bekanntmachungen über das World Wide Web gewidmet ist (siehe Abbildung 20.12). Wenn Ihr Browser das Lesen von Usenet-News unterstützt und Sie ihn so konfiguriert haben, dass er auf Ihren neuen Server zeigt, dann können Sie die Artikel lesen, die in dieser Newsgroup veröffentlicht werden, und auch Ihre eigenen Ankündigungen hinzufügen, indem Sie den folgenden URL in Ihr Dokumenten-URL-Feld einsetzen: news:comp.infosystems.www.announce

Eine der Botschaften, nach der man in comp.infosystems.www.announce ganz besonders Ausschau halten sollte, ist eine ausgezeichnete FAQ namens: »FAQ: Wie Sie Ihre Web-Site bekannt machen können.« (FAQ: How to Announce Your new Web-Site) Diese FAQ enthält eine aktualisierte Liste der besten und profitabelsten Mittel zur Be-

kanntmachung Ihre Web-Site. Wenn Sie die FAQ in dieser Newsgroup nicht lokalisieren können, können Sie sich auch eine Online-Version unter <http://ep.com/faq/webannounce.html> anschauen.

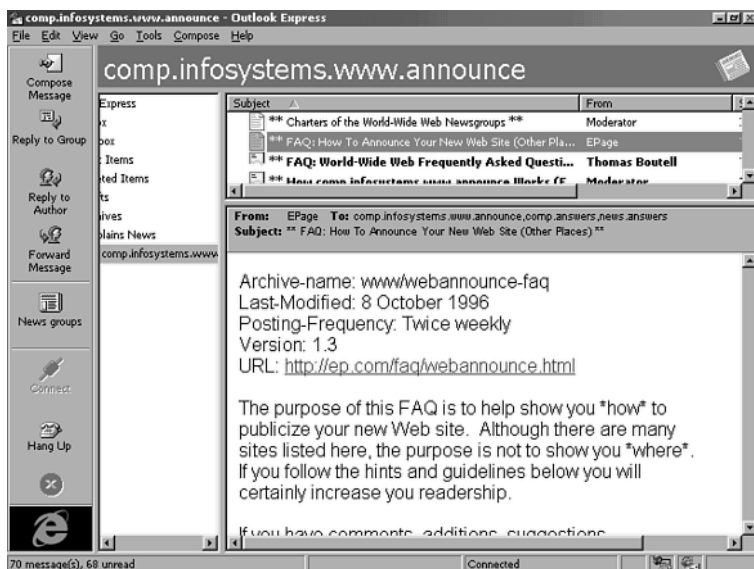


Abbildung 20.12:
Die Newsgroup
comp.info-
systems.www.
announce



comp.infoystems.www.announce ist eine moderierte Newsgroup, bei der alle Veröffentlichungen von einem Moderator zugelassen werden müssen, bevor sie in dem Verzeichnis der Newsgroup auftauchen. Um sicherzustellen, dass Ihre Ankündigung zugelassen wird, sollten Sie das Grundlagendokument lesen, das den Veröffentlichungsprozess skizziert. Sie können dieses Dokument mit Ihrem Browser unter <http://boute11.com/%7Egrant/character.html> lesen.

Web-Ringe

Eine relativ neue Methode für Ihre Web-Site zu werben ist, sie in eine oder mehrere Web-Ringe einzutragen. Ein Web-Ring ist eine Sammlung von Sites, die ein spezifisches Thema haben. Eine immense Anzahl von Web-Ringen zu fast jedem Thema, das Sie sich nur vorstellen können, finden Sie unter www.webring.com/ (siehe Bild 20.13).

Das Konzept hinter einem Web-Ring ist einfach. Um in einem oder mehreren von diesen Web-Ringen zu sein, müssen Sie Ihre URL bei einem Ringmaster, der für den Ring verantwortlich ist, anmelden. Der Ringmaster wird dann von Ihnen verlangen, bestimmten Quelltext (und manchmal auch Bilder) auf einer hervorstechenden Seite Ihrer Site zu integrieren. Meistens handelt es sich um Ihre Homepage, oder um eine Seite die in den Web-Ringen angezeigt werden soll, zu denen Sie gehören.

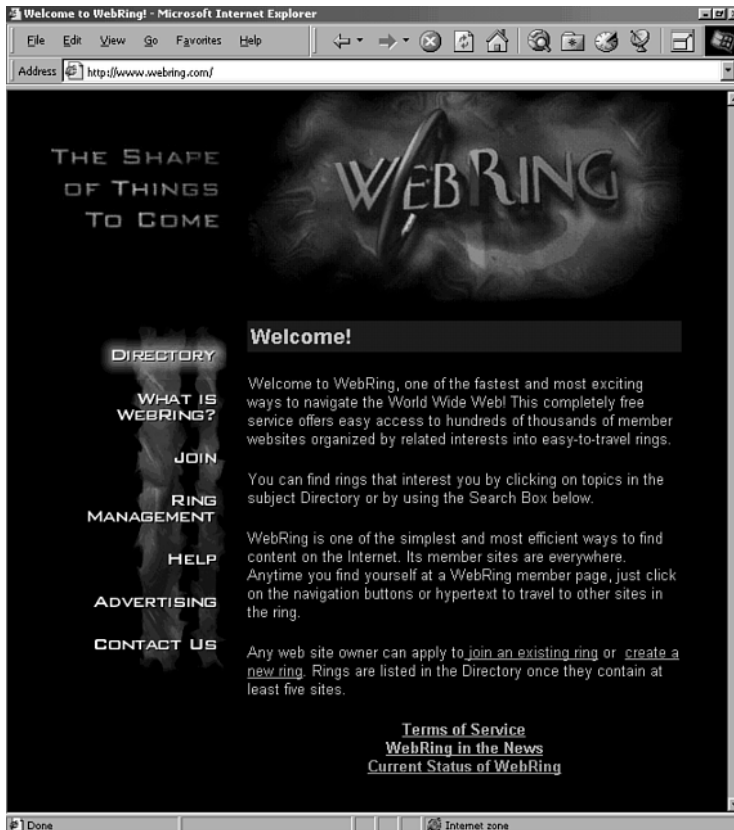


Abbildung 20.13:
Die Web-Ring-
Homepage

Der Code, den Sie auf Ihre Seite geben, bietet ein einfaches Navigationssystem, mit dem der Anwender zu und von anderen Sites im Web-Ring navigieren kann. Auf diesem Weg können Leser Ihre Web-Browsing-Session auf Web-Sites richten, die ein gemeinsames Interesse oder Ziel verfolgen.

Seit 1998 haben Web-Ringe ein phänomenales Wachstum erfahren und es sieht so aus, als ob sich diese Entwicklung ungebrochen fortsetzt. Web-Ringe sind also eine sehr gute Möglichkeit eine große Anzahl von Lesern, die sich für die gleichen Themen interessieren, die Sie auch auf Ihrer Site behandeln, anzuziehen.

Für zusätzliche Informationen über den Beitritt zu einem Web-Ring, finden Sie unter www.webring.com/join.html.

Visitenkarten, Briefköpfe und Broschüren

Obwohl das Internet ein wundervoller Ort ist, um Ihre neue Web-Site bekannt zu machen, gibt es eine andere ausgezeichnete Werbemethode, die von vielen Leuten überhaupt nicht wahrgenommen wird.

Viele Unternehmen geben jedes Jahr eine beträchtliche Summe aus, um Briefköpfe, Visitenkarten und anderes Werbematerial herzustellen. Nur sehr wenige kommen jedoch auf die Idee, ihre E-Mail-Adressen und Homepage-URLs darauf aufzudrucken. Aber warum nicht? Bei mehr als 35 Millionen Leuten im Internet gibt es gute Chancen dafür, dass viele Ihrer Kunden bereits ans Internet angeschlossen sind oder dies zumindest in den nächsten paar Jahren sein werden.

Indem Sie Ihre E-Mail-Adresse und Homepage-URL auf all Ihre Korrespondenz und Ihr Werbematerial drucken, können Sie dadurch völlig verschiedene Gruppen von potentiellen Besuchern Ihrer Site erreichen. Und, wer weiß, vielleicht werden Sie sogar neue Kunden dadurch gewinnen, dass Sie sich die Zeit dazu nehmen, Leuten zu erklären, was all diese neuen Adressen-Informationen bedeuten.

Das Fazit bei der Bekanntmachung Ihrer Web-Site ist, Ihr Denken auf andere Gebiete auszuweiten. Sie müssen jedes erdenkliche Werkzeug benutzen, wenn Sie eine erfolgreiche und aktive Site hervorbringen wollen.

Finden Sie heraus, wer Ihre Webseiten betrachtet

Herzlichen Glückwunsch, jetzt haben Sie Ihre Seiten auf dem Web, bereit, betrachtet zu werden, Sie haben für Ihre Site geworben und sie bekannt gemacht, und die Leute kommen (hoffentlich) scharenweise zu Ihrer Site. Oder etwa nicht? Wie können Sie das herausfinden? Da gibt es verschiedene Möglichkeiten, einschließlich der Protokolldateien und der Zugriffszähler.

Protokolldateien

Die beste Methode, herauszufinden, wie oft und von wem Ihre Seiten betrachtet werden, ist es, zu versuchen, Zugriff auf die Protokolldateien Ihres Servers zu bekommen. Der Server sammelt alle diese Informationen und kann dann, je nachdem, wie beschäftigt er ist, diese Informationen für Wochen oder Monate bereithalten. Viele kom-

merzielle Anbieter für Publikationen auf dem Web stellen Ihnen einen Mechanismus zur Verfügung, mit dem Sie die Web-Protokolle ansehen können oder mit dem Sie sich Statistiken betrachten können, wie viele Personen auf Ihre Seiten zugreifen und von wo aus. Bitten Sie Ihren Web-Master um Hilfe.

Wenn Sie es schaffen, Zugriff auf Ihre unbearbeiteten Protokolldateien zu bekommen, werden Sie wahrscheinlich eine Menge von Zeilen sehen, die ungefähr folgendermaßen aussehen (ich habe diese Zeile in zwei aufgeteilt, sodass sie auf die Seite passt):

```
vide-gate.coventry.ac.uk - - [17/Apr/1996:12:36:51 -0700]  
"GET /index.html HTTP/1.0" 200 8916
```

Was bedeutet dies? Dies ist das standardmäßige Aussehen und der Stil der meisten Protokolldateien. Der erste Teil der Zeile beschreibt die Site, die auf die Datei zugegriffen hat (in diesem Beispiel eine Site aus Großbritannien). Die beiden Striche werden für die Authentifizierung benutzt (wenn Sie Einlog-Namen und Passwörter eingerichtet haben, wird der User-Name der Person, die sich einloggt, und der Name der Gruppe, zu der sie gehört, hier erscheinen). Das Datum und die Uhrzeit, wann auf die Seite zugegriffen wurde, befinden sich innerhalb der eckigen Klammern. Der Teil danach ist der Name der Datei, auf die der Zugriff erfolgte; in unserem Beispiel ist das `index.html` auf der obersten Ebene des Servers. `GET` ist der tatsächliche HTTP-Befehl, den der Browser verwendete; an dieser Stelle sehen Sie meistens `GET`. Die letzten beiden Nummern sind schließlich der HTTP-Status-Code und die Anzahl von Bytes, die übertragen wurden. Der Status-Code kann verschiedene Werte angeben: 200 bedeutet, dass die Datei korrekt gefunden und übertragen wurde; 404 bedeutet, dass die Datei nicht gefunden wurde (jawohl, das ist der gleiche Status-Code, den Sie in den Fehlermeldungen Ihres Browsers bekommen). Schließlich wird die Anzahl der Bytes normalerweise dieselbe Zahl sein wie bei der tatsächlichen Datei; wenn es eine kleinere Zahl ist, bedeutet dies, dass der Leser den Transfer in der Mitte abgebrochen hat.

Zugriffszähler

Wenn Sie aus irgendwelchen Gründen keinen Zugriff auf die Protokolldateien Ihres Servers haben, aber zumindest wissen möchten, wie viele Leute sich Ihre Seiten anschauen, können Sie einen Zugriffszähler auf Ihrer Seite anbringen. Sie haben dies wahrscheinlich schon einige Male beim Web-Surfen gesehen; sie sehen aus wie Kilometerzähler oder kleine Zähler, auf denen steht: »Seit dem 15. Juli 1900 wurde auf diese Seite 5.456.234.432 mal zugegriffen.«

Es gibt viele verschiedene Zugriffszähler, von denen die meisten jedoch erfordern, dass Sie etwas auf dem Server installieren oder im Besitz von etwas sind, das sich Server-Includes nennt. Einige Sites, von denen ich hier drei erwähne, bieten Zugriffszähler an, die keine Server-Installation benötigen (dafür aber teilweise Geld kosten).

Der Zugriffszähler unter <http://www.digits.com/> kann sehr leicht installiert werden und ist sehr beliebt. Wenn Sie eine Site mit weniger als 1000 Hits pro Tag haben, dürfen Sie den Zähler kostenlos benutzen. Darüber hinaus müssen Sie sich dem kommerziellen Plan anschließen, bei dem der Zugriffszähler \$30 oder mehr kostet. Nachdem Sie sich für den [digits.com](http://www.digits.com/)-Zähl-Service registriert haben, bekommen Sie einen URL, den Sie auf Ihren Seiten als Teil eines -Tags einfügen. Wenn dann auf Ihre Seite zugegriffen wird, greift der Browser auf diesen URL auf den [digits.com](http://www.digits.com/) Server zu, der dann ein neues Zugriffszählbild für Sie erzeugt.

Wenn Sie an mehr Informationen über Zugriffszähler im Allgemeinen (sowie an einem riesigen Archiv von Bildern von Zugriffszählern) interessiert sind, besuchen Sie Digit Manias Homepage unter <http://cervantes.learningco.com/kevin/digits/index.html>.

Name	URL
XOOMCounter	www.pagecount.com/
Jcount	http://www.jcount.com/
WebTracker	http://www.fxweb.holowww.com/tracker/
LiveCounter	http://www.chami.com/prog/lc/

Tabelle 20.1: Zugriffszähler

Zusammenfassung

In diesem Kapitel haben Sie verschiedene Methoden kennen gelernt, mit denen Sie für Ihre Site werben können. Außerdem haben Sie erfahren, wie man Protokolldateien verwendet, um die Anzahl der Besucher Ihrer Site herauszufinden. Nun sind Sie endlich im Web und Ihre Site wird besucht. Allerdings gibt es da noch ein wichtiges Thema, das wir bis jetzt noch nicht angesprochen haben: Ihre Site muss aktuell und auf dem neuesten Stand sein. Im nächsten Kapitel werden wir uns damit beschäftigen.

Fragen und Antworten

F Es gibt so wahnsinnig viele Suchmaschinen! Muss ich meine URL denn allen Suchmaschinen vorschlagen?

A Das müssen Sie nicht. Das Internet kann mit einer Autobahn verglichen werden. Wenn Sie auf einer Autobahn fahren, schreien diverse Schilder förmlich nach Ihrer Aufmerksamkeit. Wie viele von diesen Schildern beachten Sie aber in Wirklichkeit? Eigentlich erinnern Sie sich nur an die Schilder, die Sie am häufigsten gesehen haben. Wenn Sie Ihre Seiten in vielen Suchmaschinen auf-

listen, ist das ungefähr so, als würden Sie viele verschiedene Straßen entlang fahren. Die Eintragung in mehrere Suchmaschinen macht Ihre URL für eine größere Anzahl von Anwendern sichtbar.

F Was ist, wenn ich meine URL an Unmengen von Diskussionsforen sende? Ist das nicht eine gute Methode, für meine Site zu werben?

A Ja und nein. Die Besucher von Diskussionsforen sehen es nicht so gerne, wenn sie mit unnützen Informationen überhäuft werden. Wenn Sie Ihre URL in Diskussionsforen anzeigen, sollten Sie etwas diskreter vorgehen und nicht Dutzende von Diskussionsforen mit Ihrer URL oder einer Nachricht bombardieren. Lassen Sie Ihren URL höflich und diskret einer handvoll verbundener Diskussionsforen zukommen. Das ist vielleicht etwas arbeitsaufwendiger, aber Sie werden sich sicherlich weniger Feinde damit machen.

F Was ist, wenn ich keine angemessene Stelle für den Code finde, den die Web-Ringe auf meiner Homepage sehen möchten? Gibt es eine Alternative?

A Das kommt auf den Web-Ring und die Person, die ihn verwaltet an. Jeder Web-Ring hat eine Liste mit Instruktionen, die Ihnen angibt, wie Sie den Code auf Ihrer Seite angeben sollen. Einige der Web-Ringe können diesbezüglich schon sehr eigen sein (zum Beispiel muss der Code in einigen Web-Ringen auf Ihrer Homepage sein), während andere Web-Ringe wiederum, Sie den Code auf eine hervorstechende Seite Ihrer Site platzieren lassen. Dann gibt es da auch noch Web-Ringe, die Sie Grafiken mit einbeziehen lassen oder aber auch Web-Ringe, die nur Text haben wollen.

Falls Ihnen der Ringmaster erlauben sollte, den Web-Ring-Code auf einer anderen Seite als Ihrer Homepage einzugeben, gehen Sie sicher, dass Sie auf Ihrer Homepage ein Link zu Ihrer Web-Ring-Seite haben. Ein einfacher Text, wie etwa »Für eine Auflistung der Web-Ringe, zu denen diese Site gehört, besuchen Sie bitte meine Web-Ring-Seite«, sollte genügen.

Quiz

1. Nennen Sie einige Methoden, mit denen Sie für Ihre Web-Site werben können.
2. Definieren Sie Hit.
3. Welches sind die Vorteile einer Site wie Submilt!, die Ihre Site in einer Vielzahl von Suchmaschinen einträgt, für die Werbung Ihrer Site?
4. Welche Regeln müssen sie befolgen, wenn Sie in einem Diskussionsforum für Ihre Site werben?

Antworten

1. Mit größeren Web-Verzeichnissen und -Indizes, Ankündigungen in Diskussionsforen. Zusätzlich sollten Sie Ihren URL auf Visitenkarten und Briefpapier anbringen. Außerdem können Sie mit Web-Ringen im World Wide Web für Ihre Site werben.
2. Ein Hit ist das Herunterladen einer Datei Ihrer Site. Wird zum Beispiel eine HTML-Seite mit drei Bildern von Ihrer Site abgerufen, dann sind das insgesamt vier Dateien, die heruntergeladen werden. Entsprechend sind das vier Hits.
3. Eine solche Site ermöglicht es Ihnen, Ihren URL in unterschiedliche Site-Promotion-Bereiche und Web-Robotern einzutragen, ohne dies bei jeder dieser Sites einzeln vornehmen zu müssen, das heißt Sie sparen Zeit. Eine kleinere Anzahl von ihnen bieten dies kostenlos an, die Mehrheit aber verlangt eine Gebühr.
4. Die wichtigste Regel ist, Ihren URL nicht einfach blind und auf einen Schlag in eine Vielzahl von Diskussionsforen zu verstreuen. Seien Sie selektiver und bleiben Sie höflich. Besser noch: Adressieren Sie Ihre URL an Diskussionsforen, zu denen Sie bereits gehören.

Übungen

1. Besuchen Sie einige der Sites, über die Sie eine Site in mehrere Suchmaschinen auf einmal eintragen lassen können. Stellen Sie sich daraus eine Liste der Sites zusammen, bei denen Sie für Ihre eigene Site werben wollen. Gehen Sie Ihre Auswahl noch einmal durch und überprüfen Sie, ob es für das Auflisten Ihrer Seite bestimmte Anforderungen gibt.
2. Gestalten Sie eine neue Visitenkarte oder Broschüre, die für Ihren Betrieb und Ihre Web-Site wirbt.

A tilted, light gray calendar grid is visible in the background, showing days of the week (M, D, M, D, I) and empty date boxes.

Testen, Über- arbeiten und Verwalten von Web-Sites

**Woche
3**

Nachdem Sie dieses Buch soweit sorgfältig gelesen haben, haben Sie Ihre eigene Web-Präsentation mit unzähligen praktischen Seiten erzeugt, mit hinreißenden Bildern und mit einem oder zwei Formularen. Und sie ist großartig geworden. Dann haben Sie Tabellen angelegt und Bilder ausgerichtet, mehrere Bilder mit JPEG komprimiert und ein Video-Clip von Ihnen und Ihrer Katze eingefügt.

Cascading Style Sheets geben den Seiten ein einheitliches Aussehen. Mit Dynamic HTML werden Grafiken, Bilder und Text übereinander positioniert und »peppen« Ihre Seite richtiggehend auf. Ihre Seiten könnten wirklich nicht besser aussehen, glauben sie. Sie sind fertig?

Schlechte Neuigkeiten. Sie sind eben noch nicht fertig. Jetzt müssen Sie sich über zwei Dinge Gedanken machen. Sie müssen Ihre Präsentation testen und warten.

Beim Testen stellen Sie sicher, dass Ihre Web-Präsentation funktioniert – nicht nur vom technischen Standpunkt aus gesehen (Ist der HTML-Code korrekt? Funktionieren alle Links), sondern auch von der funktionalen Seite her (Finden die Leser auf Ihren Seiten das, was sie brauchen?). Darüber hinaus sollten Sie sicherstellen, dass Ihre Präsentation in mehreren Browsern gelesen werden kann, insbesondere, wenn Sie die neuen Tags verwenden, die Sie in den vorhergehenden Kapiteln kennen gelernt haben.

Aber selbst, wenn alles getestet ist und wunderbar funktioniert, sind Sie noch nicht fertig. Sobald Sie Ihre erste Präsentation veröffentlicht haben, wollen Sie sie ergänzen und modifizieren, sodass alles interessant und aktuell bleibt. Glauben Sie mir. Im Web ändert sich die Technologie ständig, eine Präsentation wird *nie* fertig. Es gibt immer irgendwelche Seiten, die verändert werden müssen. Nachdem Sie dieses Kapitel gelesen haben, werden Sie alles über die folgenden Themen wissen:

- ▶ Integritätstests, die sicherstellen, dass Ihre Webseiten funktionieren
- ▶ Verwendungstests, die sicherstellen, dass Ihre Seiten so verwendet werden können, wie Sie es erwarten, und dass Ihre Ziele mit der Präsentation erreicht wurden
- ▶ Ergänzen oder Überarbeiten Ihrer Präsentation, ohne bereits Vorhandenes zu zerstören

Integritätstests

Integritätstests haben nichts mit irgendwelchen Steuerhinterziehungen zu tun. Sie stellen einfach nur sicher, dass Ihre Seiten korrekt funktionieren, dass sie ohne Fehler angezeigt werden und dass alle Links auf sinnvolle Positionen zeigen. Dabei wird nichts darüber ausgesagt, ob Ihre Seiten sinnvoll sind oder ob die Leser sie gebrauchen können. Es wird also nur die technische Korrektheit geprüft. Die Integritätstests folgen drei Schritten:

1. Sicherstellen, dass korrekter HTML-Code erzeugt wurde
2. Überprüfen des Aussehens Ihrer Seiten in unterschiedlichen Browsern
3. Sicherstellen, dass Ihre Links funktionieren (sowohl am Anfang als auch nach mehrmonatigem Einsatz)

Prüfen Ihres HTML-Codes

Im ersten Schritt stellen Sie sicher, dass Ihr HTML-Code korrekt ist, dass all Ihre Tags richtig abgeschlossen werden, dass sich die Tags nicht überlappen oder innerhalb von anderen Tags verwendet werden. Jedenfalls sollte es so sein, wenn Ihre Seite in verschiedenen Browsern angezeigt wird, oder? Nicht ganz; Browser sollen Probleme in den HTML-Dateien erkennen und danach versuchen, festzustellen, was der Autor mit seinem Programm ursprünglich bewerkstelligen wollte, und sie zeigen dann irgendetwas an, wenn dies nicht möglich ist. (Erinnern Sie sich noch daran, wie Tabellen in einem Browser aussahen, der keine Tabellen unterstützt? Auch in diesem Beispiel versucht der Browser herauszufinden, was Sie eigentlich wollten.) Dabei sind einige Browser nachsichtiger gegenüber Fehlern als andere. Es kann sein, dass eine fehlerhafte Seite in einem Browser funktioniert, in einem anderen dagegen nicht.

Aber es gibt nur eine gültige HTML-Definition, nämlich die der HTML-Spezifikation. Einige Browser kommen mit dem HTML zurecht, das Sie ihnen möglicherweise fehlerhaft übergeben, aber wenn Sie von vornherein korrektes HTML schreiben, dann funktionieren Ihre Seiten garantiert in allen Browsern, die die von Ihnen verwendete HTML-Version unterstützen.

Um genau zu sein: Die einzig gültige Definition von HTML wird durch die so genannte HTML DTD, also die Document Type Definition, festgelegt. HTML wird durch eine Sprache namens SGML definiert, eine umfassendere Sprache zur Definition anderer Sprachen. Die DTD ist eine SGML-Definition einer Sprache, also ist die HTML DTD die verbindliche technische Definition für HTML.



Aber wie können Sie sicherstellen, dass Ihr HTML-Code korrekt ist? Wenn Sie den Regeln und Beispielen folgen, die ich in den bisherigen Kapiteln vorgestellt habe, dann dürfte es an Ihrem HTML-Code nichts zu bemängeln geben. Aber jeder vergisst einmal ein abschließendes Tag, setzt ein Tag am falschen Platz oder vergisst die Anführungszeichen nach einer HRF. (So etwas passiert ständig und es gibt viele Browser, die damit nicht zurechtkommen.) Die beste Methode, herauszufinden, ob Ihre Seiten korrekt sind, ist die Verwendung eines HTML-Validators, also eines Testprogramms.

Viele HTML-Editoren bieten inzwischen eingeschränkte Möglichkeiten zum Überprüfen Ihres HTML-Codes. Bei Programmen wie HoTMetaL Pro von SoftQuad geht dies sogar soweit, dass die interne Codeüberprüfung Sie davon abhält, Dokumente zu erzeugen, die bei der Überprüfung durchfallen würden. Bei den meisten Editoren ist die Codeüberprüfung allerdings eingeschränkt und unvollständig. Die Krönung ist allerdings, dass einige Editoren es Ihnen nicht nur erlauben, falschen HTML-Code zu produzieren, sondern sogar falschen HTML-Code für Sie erzeugen. Aus diesem Grund ist es sinnvoll, eine externe Codeüberprüfung zu verwenden.

Ein HTML-Validator soll Ihren HTML-Code auf seine technische Korrektheit überprüfen.



Einige Validierer prüfen ältere und neuere HTML-Spezifikationen. Neuere Validierer prüfen die drei Spezifikationen des HTML oder XHTML 1.0: Transitional, Strict oder Frameset.

Damit Ihr HTML auch in Zukunft gelesen werden kann, überprüfen Sie, ob diese auch wirklich korrekt geschrieben ist. Schließlich möchten Sie doch nicht, dass Sie eines Tages vollkommen verzweifelt feststellen müssen, dass Ihr HTML-Code und alles, was Sie mit diesem erstellt haben nicht mehr gelesen werden kann, weil ein ultimatives HTML-Werkzeug auf den Markt gekommen ist.

Aber selbst wenn Ihr HTML-Code korrekt ist, sollten Sie Ihre Seiten in mehreren Browsern testen, um sicherzustellen, dass nicht irgendwelche seltsamen Designentscheidungen getroffen wurden. Ein Validator nimmt Ihnen die Designarbeit nicht ab.

Aber wie können Sie diese HTML-Validatoren einsetzen? Einige davon stehen im Web zur Verfügung, und Sie können sie sich herunterladen und lokal auf Ihrem System ausführen, andere wurden als Webseiten realisiert, wo Sie Ihre URLs in ein Formular eintragen können und der Validator sie im Netzwerk überprüft. Zwei dieser Programme finde ich besonders gut: den HTML Validation Service von WebTech sowie Weblint von Neil Browsern.



Wie es bei allen Webseiten der Fall ist, ändern sich diese Dienste ständig – bieten neue Features und verändern ihr Erscheinungsbild. Aber selbst wenn sie bei Ihrem Besuch anders aussehen sollten als in den hier gezeigten Beispielen, sollten diese Beispiele Ihnen eine gute Vorstellung von der Funktionsweise dieser Dienste vermitteln.

In diesem Abschnitt werden einige Online-Validierer vorgestellt, die Sie im Web verwenden können.

Es gibt auch eigenständige HTML-Validierer, die Sie auf Ihrem Computer verwenden können. Einer dieser Validierer für Windows 95/NT ist der CSE 3310 HTML-Validierer. Dieser eigenständige HTML-Validierer hilft Ihnen, verschiedene Probleme im HTML-Quellcode aufzuspüren und zu korrigieren, wie zum Beispiel Rechtschreibfehler oder ungültige Tags, falsche Tag-Platzierung oder falsche Verschachtelung, und vieles mehr. Sie können mehr über diesen Validierer auf der CSE 3310 Homepage unter www.htmlvalidator.com/htmldownload.html erfahren.

Weblint

Das Programm Weblint führt eine allgemeine Überprüfung durch. Es stellt nicht nur sicher, dass Ihre Syntax korrekt ist, sondern erkennt auch allgemeine Fehler: nicht übereinstimmende schließende Tags, die Angabe von TITLE außerhalb von HEAD, mehrfach vorkommende Elemente, die nur einmal vorhanden sein sollten. Außerdem gibt es noch viele andere Hinweise. Es ist viel benutzerfreundlicher als der HTML-Validierer, aber es ist ungenauer hinsichtlich der strengen HTML-Konformität (und möglicherweise beschwert es sich über ein paar neuere Tags, etwa für Tabellen, weil es sie noch nicht kennt).

Abbildung 21.1 zeigt die Weblint-Seite unter <http://www.unipress.com/cgi-bin/WW-Weblint>. Sie zeigt das Formular, in das Sie Ihre Seiten zur Überprüfung eintragen können.

Mit Weblint können Sie Ihren HTML-Code überprüfen, entweder indem Sie den URL zu der Seite eingeben, die Sie im Internet veröffentlicht haben, oder indem Sie Quellcode in ein Texteingabefeld einfügen. Tippen Sie den folgenden Code in das DATA-Feld des Weblint-Formulars, um einen Eindruck von dem zu bekommen, was Weblint Ihnen über Ihren HTML-Code alles sagen kann.

```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>
<head>
<title>Validating HTML Code</title>
</head>
<body>
Every once in a while I get the urge to be funny. Luckily for
  those around me it usually passes in a few minutes. But sometimes
  I write things down.</p>
</body>
</html>
```

In Bild 21.2 sehen Sie einen Ausdruck eines Sample-Tests, den ich mit derselben Seite, die den fehlenden `<p>`-Tag Fehler hatte, gemacht habe.

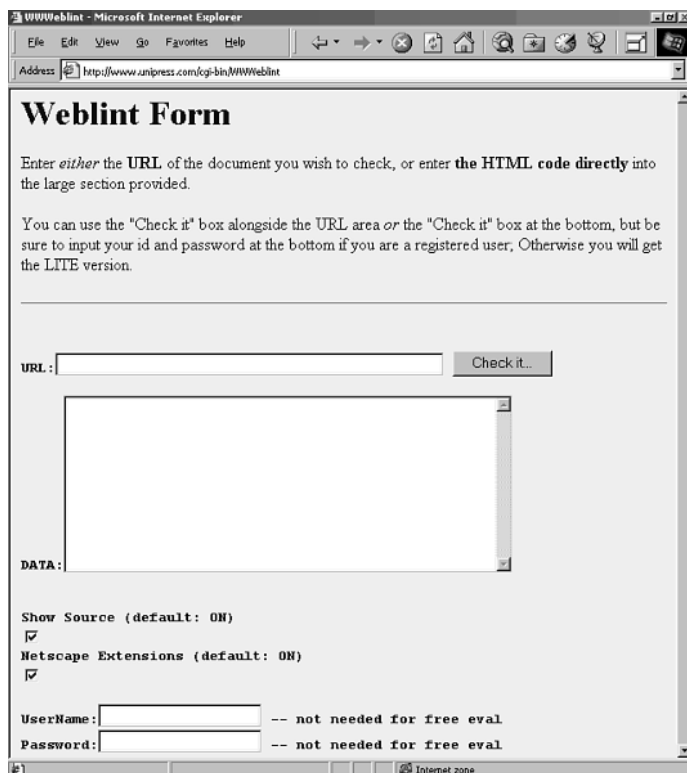


Abbildung 21.1:
Ausgabe von Weblint

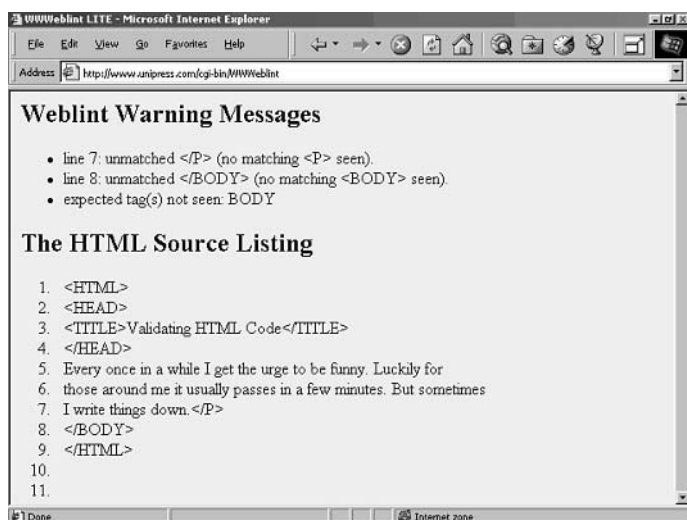


Abbildung 21.2:
Das HTML-Prüfpro-
gramm Weblint

Interessanterweise hat Weblint festgestellt, dass ich ein schließendes `</head>`-Tag vergessen habe, das der Validator übersehen hatte. Dagegen hat es die Tatsache, dass es ein `</p>` ohne entsprechendes `<p>` gibt, völlig ignoriert.

Weblint ist inzwischen kein kostenloser Service mehr. Man kann sich für sechs Monate für eine Gebühr von \$ 16.95 anmelden (Jahresgebühr). Sie können aber auch erst einmal einen kostenlosen Probelauf machen, der nur die ersten 2048 Byte Ihrer Dokumente prüft (das führt bei längeren Dokumenten natürlich zu Fehlern, die eigentlich keine sind, sondern nur aus fehlenden Tags resultieren, die sich im Bereich nach 2048 Byte befinden).

Der W3C HTML Validierungsservice

Unter den verschiedenen *Validierer*, die Ihnen behilflich sind, einen existierenden HTML 4.0 Code zu überprüfen, gibt es nur wenige, die XHTML1.0 unterstützen. Die beste Möglichkeit, Ihre Seite zu validieren, ist, mit dem W3C HTML Validierungsservice, den Sie unter <http://validator.w3.org/> finden, zu arbeiten. Dieser Validierungsservice ist gebührenfrei und ermöglicht es Ihnen, Ihren HTML Code auf verschiedenen DTDs (Document Type Definitions) zu überprüfen. Mit dem W3C-Validierer können Sie außerdem, Netscape- und Internet Explorer spezifische Codes überprüfen. Eine vollständige Liste der DTDs, die dieser Validierer überprüft, finden Sie unter <http://validator.w3.org/sgml.lib/catalog>

Nachdem Sie den URL Ihrer Webseite eingegeben haben, können Sie eine oder mehrere der folgenden Optionen wählen:

- ▶ Include Weblint Results – Wählen Sie diese Option, wird der W3C-Validierer die Weblint-Ergebnisse in Ihre Code-Validierung miteinbeziehen. Die Erklärungen von Weblint sind etwas anwenderfreundlicher und erleichtern somit das Verbessern Ihres Codes.
- ▶ Run Weblint in »pedantic« mode – Wenn Sie mit dieser Option Ihre Seiten vom W3C-Validierer überprüfen lassen, können Sie sichergehen, dass diese aufs Strengste überprüft werden.
- ▶ Show Source Input – Diese Option enthält eine Auflistung der Quellcodes, die der Validierer überprüft. Das kann sehr hilfreich sein, wenn es darum geht, die Fehler in Ihrem Code genau zu lokalisieren, was nur durch die Zeilennummern, die diese Auflistung enthält, möglich ist. Der Validierer gibt Ihnen die Zeilen an, in denen sich die Codefehler befinden.

- ▶ Show an outline of this document – Wenn Sie in Ihrer Seite Überschriften haben, liefert Ihnen diese Option die Gliederung Ihres Web-Dokuments. Das kann sehr hilfreich sein, um herauszufinden, ob Ihre Webseiten-Struktur den Gliederungsstandards entspricht (zum Beispiel H2 muss sich unter H1 befinden, H3 unter H2, usw...)
- ▶ Show parse tree – Als dieses Buch sich im Druck befand, war diese Option noch im Entwicklungsstadium. Der Parsetree hilft Ihnen, die Struktur der HTML-Tags in Ihrem Dokument anzuzeigen.
- ▶ Exclude attributes from the parse tree – Wählen Sie diese Option, werden die Attribute des Parsetree eliminiert.

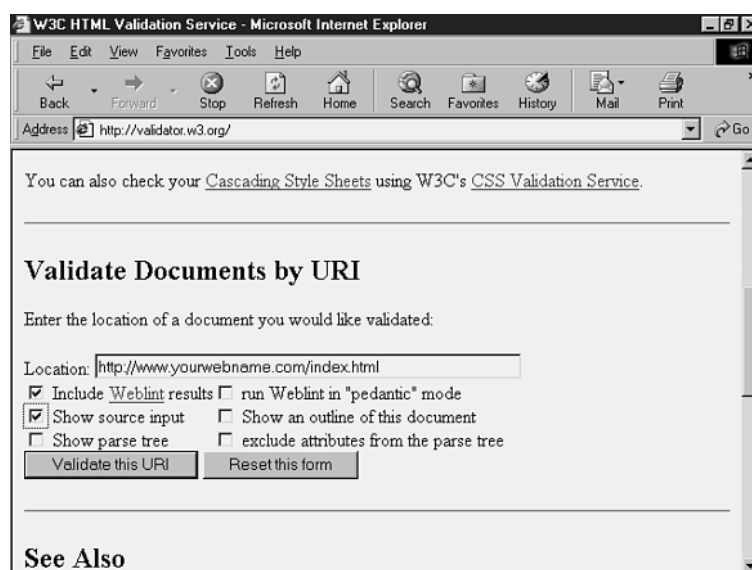


Abbildung 21.3:
Das Validierungs-
formular des
W3C HTML Vali-
dierungsservice

Übung 21.1: Überprüfen einer Beispielseite

Nur um zu zeigen, welche Fehler Weblint und der Validator feststellen können, werde ich hier eine Beispieldatei mit einigen häufig gemachten Fehlern zusammenstellen.

Wir verwenden dazu die Homepage »Susan's Cactus Gardens«, die Sie in Abbildung 21.3 sehen.

Im Internet Explorer funktioniert die Seite ganz gut, und sie sieht auch zufriedenstellend aus. Aber jetzt kommt der Code. Er ist randvoll mit Fehlern. Versuchen Sie, sie selbst zu finden, bevor Sie sich die Ergebnisse der Prüfung ansehen.

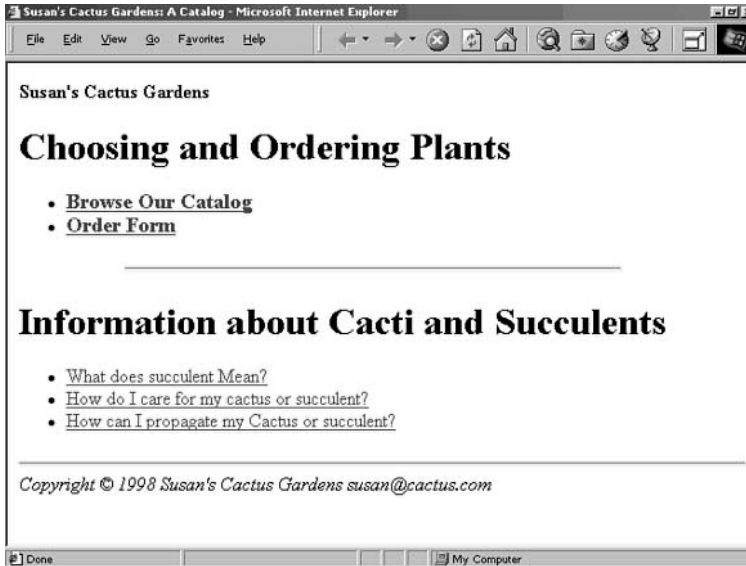


Abbildung 21.4:
Susan's Cactus
Gardens

```
<html>
<head>
<title>Susan es Cactus Gardens: A Catalog</title>
</head>
<body>

<strong>Susan es Cactus Gardens</strong>
<h1>Choosing and Ordering Plants</h3>
<ul>
<li><a href="browse.html">Browse Our Catalog
<li><a href="order.html">How To Order</a>
<li><a href="form.html">Order Form</a>
</ul>
</h3>
<hr width=70% align=center />
<h1>Information about Cacti and Succulents</h1>
<ul>
<li><a href="succulent.html">What does succulent Mean?</a>
<li><a href="caring.html">How do I care for my cactus or succulent?</a>
<li><a href="propogation.html">How can I propagate my Cactus or succulent?</a>
</ul>
<hr />
<address>Copyright &copy; 1994 Susan es Cactus Gardens
susan@catus.com</address>
```

Veröffentlichen Sie diese Seite im Internet und notieren Sie sich ihren URL. Gehen Sie zum W3C-HTML-Validierungsservice, den Sie unter <http://validator.w3.org/> finden und geben Sie den URL Ihrer »fehlerhaften« Webseite in den Lokalisierungsbereich des Formulars ein. Überprüfen Sie die »Include Weblint Results« und »Show Source Input« Optionen (die Ihnen beim Dechiffrieren Ihrer Fehler helfen) und klicken sie dann den »Validate URL«-Button an. Was Sie dann zu sehen bekommen, ist eine Seite, wie sie in Bild 21.4 angezeigt wird.

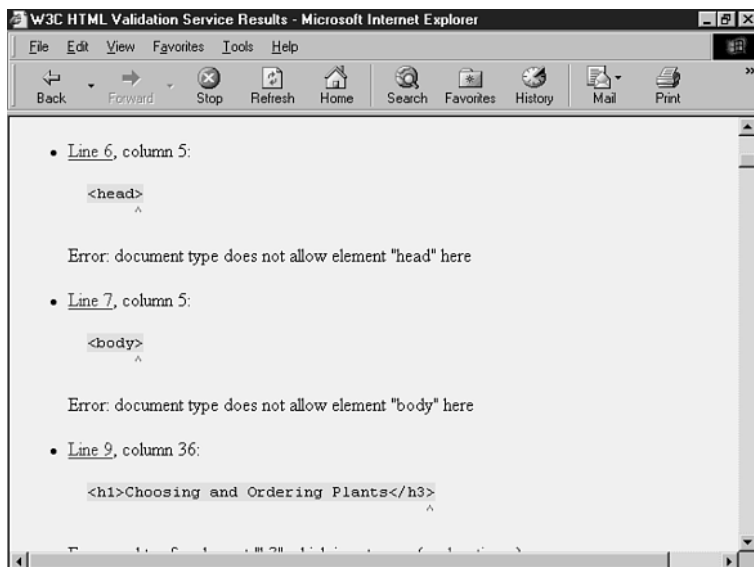


Abbildung 21.5:
Die fehlende
Antwort von
Weblint auf die
fehlerhafte Datei

Beginnen Sie mit den Fehlern, die Ihnen Weblint aufzeigt, da die Beschreibungen der Fehler verständlicher gehalten werden, als die, die Sie vom W3C-Validierungsservice erhalten. Wenn Sie die Weblint-Ergebnis-Sektion runter scrollen, bekommen Sie folgenden Fehler in den Zeilen 6 und 7 angezeigt:

```
line 6: <head> must immediately follow <html>
line 6: tag <head> should only appear once. I saw one on line 4!
line 6: <head> cannot appear in the HEAD element.
line 7: <body> must immediately follow </head>
line 7: <BODY cannot appear in the HEAD element.
```

Im Folgenden sehen Sie die Codezeilen 1 bis 7 erneut:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" http://www.w3.org/
TR/xhtml1/DTD/transitional.dtd">
<html>
<head>
```

```
<title>Susan es Cactus Gardens: A Catalog</title>
<head>
<body>
```

Das Tag in der sechsten Zeile (<head>) sollte </head> sein. Einige Browser haben Schwierigkeiten mit dem Hauptteil des Dokuments, wenn Sie vergessen, den Titel abzuschließen. Diesen Fehler sollten Sie unbedingt beheben.

Anschließend sollten viele der anderen Fehler in der von Weblint erzeugten Liste, die sich darauf beziehen, dass irgendetwas im Titel nicht stehen kann (»X cannot appear in the HEAD element«), verschwinden. Betrachten wir den nächsten Fehler:

Der nächste Fehler befindet sich in Zeile 9:

line 9: malformed heading - open tag is <h1>, but closing is </h3>

Dieser Fehler ist leicht erkennbar. Der Autor hat aus Versehen ein <h1>-Tag mit einem <h3>-Tag abgeschlossen. Die einleitenden und schließenden Tags müssen übereinstimmen. Sie müssen also nur </h3> in <h1> umändern.

Die Ergebnisse von Weblint geben keinen eindeutigen Kommentar zu Zeile 10 ab. Gegen Ende der Fehlermeldung von Weblint erhalten Sie aber doch noch einen Hinweis auf ein Problem:

line 0: no closing seen for on line 10

Wenn Sie nun Zeile 15 untersuchen, können Sie aber sehr wohl ein abschließendes -Tag sehen. Also wo liegt das Problem? Überprüfen Sie nochmals die Zeilen 10 bis 16 und Sie werden feststellen, dass die und <h3>-Tags nicht richtig verschachtelt sind:

```
<ul>
<h3>
<li><a href="browse.html">Browse Our Catalog
<li><a href="order.html">How To Order</a>
<li><a href="form.html">Order Form</a>
</ul>
</h3>
```

Drehen Sie die Reihenfolge der einleitenden und <h3>-Tags um, müsste die Weblint-Fehlermeldung verschwinden. Der überprüfte Code sollte nun wie folgt erscheinen:

```
<h3>
<ul>
<li><a href="browse.html">Browse Our Catalog
<li><a href="order.html">How To Order</a>
<li><a href="form.html">Order Form</a>
</ul>
</h3>
```

In der oben aufgeführten ungeordneten Liste wird ein weiterer Fehler aufgezeigt; die Anzahl der Anführungszeichen in Zeile 13 ist nicht korrekt:

```
line 13:odd number of quotes in element <a href="order.html">
line 13:value for attribute HREF ("order.html") of element A should
be quoted (i.e. HREF="order.html")
```

Der Dateiname `order.html` wurde nicht mit einem Anführungszeichen abgeschlossen. Das funktioniert zwar in älteren Versionen von Netscape oder vom Internet Explorer, aber nur die wenige der anderen Browser kommen damit zurecht, und es handelt sich dabei wirklich um einen der häufigsten Fehler. Der korrigierte Code sollte nun wie folgt aussehen:

```
<li><a href="order.html">How To Order</a>
```

Zeile 13 und 14 bergen den nächsten Fehler:

```
line 13: <a> cannot be nested-</a> not yet seen for <a> on line 12.
line 14: <a> cannot be nested-</a> not yet seen for <a> on line 12.
```

Dies ist eigentlich ein Fehler in Zeile 12:

```
<li><a href="browse.html">Browse Our Catalog
```

Am Ende dieser Zeile fehlt das Tag `</a>`, deshalb die Beschwerde. Man kann innerhalb eines `<a>`-Tags kein weiteres `<a>`-Tag angeben, sodass Weblint verwirrt ist (dieser Fehler kommt in dem Bericht mehrere Male vor). Achten Sie darauf, dass alle `<a>`-Tags am Ende des Verknüpfungstextes abgeschlossen werden. Wenn Sie diesen Fehler korrigieren, wird der damit verbundene Fehler in Zeile 18 ebenfalls eliminiert werden. In Zeile 18 sollte `<a>` innerhalb von `<h1>` sein und nicht umgekehrt. Die Fehler in den Zeilen 20 bis 22, die besagen, dass `<a>` nicht verschachtelt werden kann, da das abschließende `</a>`-Tag hierzu bisher noch nicht in Zeile 12 aufgetaucht ist. Wenn wir in unserer Fehlerliste weitergehen, stellen wir fest, dass Weblint sich über Zeile 17 »beschwert«:

```
line 17: attribute "WIDTH" for <hr /> is extended markup.
line 17: value for attribute WIDTH (70%) of element HR should be quoted
(i.e. WIDTH="70%")
line 17: attribute "ALIGN" for <hr /> is extended markup.
line 17: illegal value for WIDTH attribute of hr (70%).
```

Erweiterte Tags und Attribute sind normalerweise browserspezifisch. Allerdings können Sie auch nicht auf die, in der Strict HTML Definition verworfene Tags verweisen. Wenn Weblint sich also über die erweiterten Tags beschwert, vergleichen Sie diese Zeilen mit den Ergebnissen, die Ihnen im HTML-Validierungsservice aufgezeigt werden. Sollten Sie die HTML 4.0 oder die XHTML 1.0 Transitional Spezifikation verwenden, die Ihnen das Benutzen von erweiterten Tags erlauben, liegt es an Ihnen, ob

Sie die störenden Bereiche in Ihrem Code entfernen wollen oder nicht. Wenn Sie für das Design Ihrer Seiten HTML 4.0- oder XHTML 1.0- Spezifikationen verwenden, kann es vorkommen, dass Sie einen anderen Ansatz wählen müssen. Bei WIDTH-Attributen zum Beispiel, haben Sie CSS-Eigenschaften als Alternative.

Behalten sie aber für das `<hr />`-Tag das WIDTH-Attribut, dürfen Sie nicht vergessen, die Attributwerte in Anführungszeichen zu setzen. Obwohl Weblint sich nicht über den CENTER-Wert beschwert hat, sollten Sie ihn trotzdem in Anführungszeichen setzen. Der korrigierte Code in Zeile 17 müsste nun wie folgt aussehen:

```
line 0: No closing </html> seen for <html> on line 1.
line 0: No closing </head> seen for <head> on line 2.
line 0: No closing </head> seen for <head> on line 4.
line 0: No closing </body> seen for <body> on line 5.
line 0: No closing </h1> seen for <h1> on line 8.
line 0: No closing </ul> seen for <ul> on line 9.
line 0: No closing </h3> seen for <h3> on line 10.
line 0: No closing </a> seen for <a> on line 11.
```

Eine schnelle Überprüfung zeigt, dass am Ende der Datei `</body>` und `</html>` fehlen. Das Problem ist also klar. Durch Ändern von `<head>` in `</head>` und `</h3>` in `</h1>` wird der Fehler behoben.

Aber was ist mit den nächsten beiden? Es gibt eine Beschwerde, dass `<ul>` und `<h3>` keine abschließenden Tags haben, aber sie stehen ganz am Ende der Liste. Beachten Sie jedoch die Reihenfolge. Die Tags `UL` und `H3` überlappen sich hier, sodass das `UL` vor dem `H3` geschlossen wird. Diese beiden Fehler korrigieren wir, indem wir die Tags einfach vertauschen.

Der letzte Fehler besagt, dass ein `</a>`-Tag fehlt. Wir haben ihn bereits behoben.

Das war der erste Durchlauf mit Weblint. Unser korrigiertes Code Beispiel sollte nun wie folgt aussehen:

```
<!doctype html public "-//w3c//dtd xhtml 1.0 transitional//en"
"http://www.w3.org/tr/xhtml1/dtd/transitional.dtd">
<html>
<head>
<title>Susan es Cactus Gardens: A Catalog</title>
</head>
<body>
<strong>Susan es Cactus Gardens</strong>
<h1>Choosing and Ordering Plants</h1>
<h3>
<ul>
<li><a href="browse.html">Browse Our Catalog</a>
<li><a href="order.html">How To Order</a>
```

```
<li><a href="form.html">Order Form</a>
</ul>
</h3>
<hr width="70%" align="center" />
<h1>Information about Cacti and Succulents</h1>
<ul>
<li><a href="succulent.html">What does succulent mean?</a>
<li><a href="caring.html">How do I care for my cactus or succulent?</a>
<li><a href="propagation.html">How can I propagate my Cactus or succulent?</a>
</ul>
<hr />
<address>Copyright &copy; 1999 Susan es Cactus Gardens
susan@cactus.com</address>
</body>
</html>
```

Überprüfen Sie die Seite noch einmal. In Bild 21.6 können Sie sehen, dass der W3C-HTML-Validierungsservice immer noch einige Fehler in der Seite aufzeigt.

Im Gegenteil zum W3C-HTML-Validierungsservice konnte Weblint, zum Zeitpunkt des Buchdrucks, keine Fehler anzeigen, die in Verbindung mit der XHTML 1.0 stehen. Um die Anzahl der Fehler, die sich auf die XHTML 1.0 beziehen, zu reduzieren, überprüfen Sie noch einmal, ob alle Tags korrekt abgeschlossen wurden. Der vor Bild 21.5 aufgezeigte Code enthält sechs -Tags, die nicht abgeschlossen wurden. Diese Tags erscheinen in den Zeilen 12 bis 14 und in den Zeilen 20 bis 22. Schließen Sie sie wie folgt ab:

```
<li><a href="browse.html">Browse Our Catalog</a></li>
<li><a href="order.html">How To Order</a></li>
<li><a href="form.html">Order Form</a></li>

<li><a href="succulent.html">What does succulent mean?</a></li>
<li><a href="caring.html">How do I care for my cactus or succulent?</a></li>
<li><a href="propagation.html">How can I propagate my Cactus or succulent?</a></li>
```

Außerdem enthalten Zeilen 17 und 24 <hr />-Tags, die wie folgt geschlossen werden müssen:

```
<hr width="70%" align="center" />

<hr />
```

Glückwunsch! Die Kaktus-Seite ist jetzt HTML-konform. Und es waren nur zwei Programme und fünf Durchläufe dafür notwendig ...

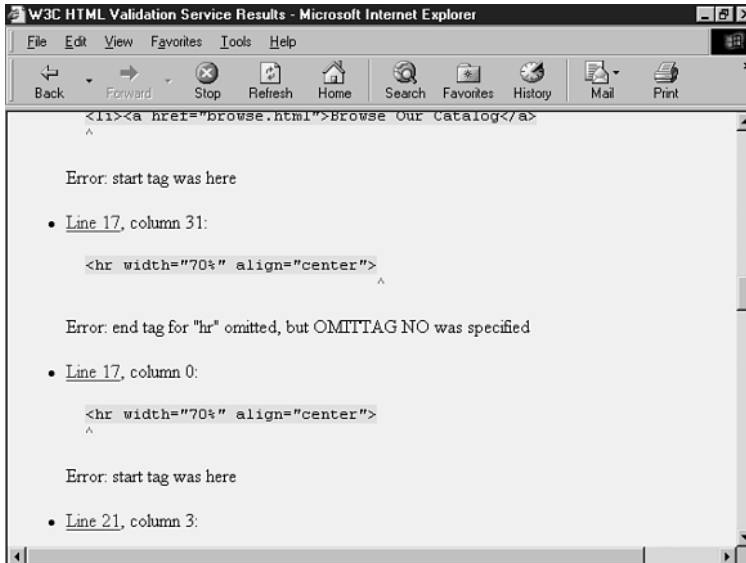


Abbildung 21.6:
Das Ergebnis der
revidierten Seite
des W3C-HTML-
Validierungsser-
vice

Dies war natürlich ein extremes Beispiel. Größtenteils gibt es auf Ihren Seiten auch nicht annähernd so viele Probleme, wie hier gezeigt (und wenn Sie einen HTML-Editor einsetzen, können viele davon gar nicht erst zustandekommen). Beachten Sie jedoch, dass Netscape all diese Fehler ohne jede Beschwerde hingenommen hat. Nicht alle Browser sind so nachsichtig!

Browser-Test

Wie bereits erwähnt, können die HTML-Prüfprogramme nur feststellen, ob Ihr HTML-Code korrekt ist. Sie sagen nichts über das Design aus. Nachdem Sie mit der Überprüfung des HTML-Codes fertig sind, sollten Sie Ihre Seiten mit so vielen Browsern wie möglich ausprobieren, um sicherzustellen, dass der Entwurf passt und Sie keine Elemente verwenden, die in einem Browser wunderschön, in einem anderen dagegen fürchterlich aussehen. Da die meisten Browser einfach aus dem Netz heruntergeladen werden können, sollten Sie mindestens zwei oder drei für Ihre Plattform ausprobieren.

Im besten Fall sollten Sie jede Ihrer Seiten in mindestens drei Browsern testen:

- ▶ In einem der beiden wichtigsten: Netscape oder Internet Explorer
- ▶ Einem anderen Browser, wie etwa Mosaic, MacWeb, WinWeb, Cello usw.
- ▶ Einem textbasierten Browser, wie etwa Lynx

Damit erhalten Sie einen guten Eindruck, wie Ihre Seiten in anderen Browsern aussehen. Wenn Sie für Ihre Seiten die Netscape- oder Internet-Explorer-Erweiterungen nutzen, sollten Sie sie sowohl in Netscape als auch im Microsoft Internet Explorer testen, um sicherzugehen, dass diese in beiden Browsern richtig angezeigt werden.

Überprüfen Sie Ihre Links

Der dritte und letzte Test betrifft die Korrektheit Ihrer Links. Am besten können Sie diese natürlich prüfen, indem Sie sich vor einen Browser setzen und sie selbst nachvollziehen. Das ist für kleine Präsentationen ganz einfach, aber es kann doch sehr mühsam sein. Außerdem könnte es sein, dass Sie nach der ersten Überprüfung einige Namen verändern. Da sich das Web ständig irgendwie verändert, können Ihre Links fehlerhaft werden, auch wenn sich an Ihren Seiten nichts geändert hat.

Hinweise auf fehlerhafte Links, die sich auf Ihren eigenen Seiten befinden, die Sie möglicherweise beim Verschieben von irgendwelchen Dingen nicht berücksichtigt haben, finden Sie in den Fehlerprotokollen auf Ihrem Server. Diese Protokolle bezeichnen die Seiten, die nicht gefunden werden konnten: sowohl die fehlende Seite als auch die Seite, die den Link darauf enthielt. Damit dieser Eintrag in der Protokolldatei erscheint, muss natürlich jemand vergeblich versucht haben, dem Link zu folgen. Besser ist es in jedem Fall, den fehlerhaften Link festzustellen, bevor Ihre Leser das tun.

Ein automatischer Link-Checker durchsucht Ihre Seiten und überprüft, ob sämtliche Links der Seiten auch auf wirklich existierende Seiten und Sites im Web verweist. Mit einem automatischen Link-Checker zu arbeiten, ist die beste Methode gebrochene Links zu überprüfen. Einige Link-Checker enthalten Web-Spiders. Dies sind Programme, die das Web durchsuchen und dabei von Link zu Link gehen. So können Sie auch Ihre eigenen lokalen Dokumente überprüfen. Passen Sie aber auf, dass Ihr Link-Checker nicht verrückt spielt und anfängt, zusätzlich zu Ihrer Site, auch noch Sites anderer Leute zu überprüfen. Ein gutes Beispiel eines automatischen Link-Checkers ist MOM-spider, unter www.ics.uci.edu/WebSoft/MOMspider/ zu finden.

Sollten sie eine sehr große Site entwerfen, steht es wohl außer Frage, all Ihre Links manuell zu überprüfen. Es gibt hierfür viele Web-Entwicklungs-Programme auf dem Markt, die die Wartung und die Überprüfung sämtlicher Links Ihrer Seiten übernehmen. Microsoft FrontPage ist ein solches Programm. Mit diesem Programm können Sie interne und externe Links Ihrer Seite validieren und überprüfen.

Eigenständige Link-Checker, wie etwa LinkBot von Tetranet Software, stehen Ihnen ebenfalls zur Verfügung. Dieses Programm sucht nach gebrochenen Ankern, fehlenden IMAGE-Attributen und vielem mehr. Zusätzliche Informationen über dieses Programm erhalten Sie unter <http://tetranetsoftware.com/>.

Verwendungstests

Verwendungstests stellen sicher, dass Ihre Dokumente verwendbar sind, selbst nachdem sie auf einfache technische Korrektheit getestet wurden. Sie können ganz einfach ein paar Webseiten zusammenstellen, aber finden Ihre Leser auch das, was sie suchen? Ist die Organisation so in Ordnung, wie Sie ursprünglich geplant haben? Werden Ihre Leser leicht verwirrt, oder ist die Navigation durch Ihre Seiten sehr schwierig?

Verwendungstests werden schon seit Jahren in vielen Firmen eingesetzt. Die Theorie dahinter ist, dass die Entwickler, welche ein Produkt erzeugen (sei es eine Software-Applikation, ein Videorecorder oder ein Auto), nicht feststellen können, wie gut oder schlecht es in der Handhabung ist, weil sie zu viel darüber wissen. Sie wissen, wie es entwickelt wurde, dann wissen sie natürlich auch, wie man es anwendet. Die einzige Methode, wie Sie feststellen können, wie einfach die Verwendung eines Produkts ist, ist Leute zu beobachten, die das Produkt noch nie gesehen haben. Dabei sollten Sie auf die Stellen achten, an denen diese Anwender Probleme haben. Nach diesen Erfahrungswerten können Sie das Produkt ändern, erneut testen, wieder ändern usw.

Web-Sites sind ein gutes Beispiel für ein Produkt, das aus den Verwendungstests profitiert. Selbst wenn Sie nur einen Freund bitten, Ihre Seiten für Sie durchzusehen, kann Ihnen das viel darüber sagen, wie gut die Leute mit der von Ihnen entworfenen Struktur zurechtkommen.

Hier einige Aufgaben, die Sie den Testpersonen beim Überprüfen Ihrer Seiten stellen sollten:

- ▶ Bitten Sie sie, Ihre Seiten ohne ein bestimmtes Ziel zu betrachten, und beobachten Sie, wie sie dabei vorgehen. An welchen Komponenten sind sie als Erstes interessiert? Welchen Pfad wählen sie durch die Präsentation. Auf welchen Seiten halten sie an und lesen ein bisschen, und welche Seiten überblättern sie ganz?
- ▶ Bitten Sie sie, ein bestimmtes Thema oder eine bestimmte Seite zu finden, am besten eine Seite, die irgendwo tief in Ihrer Präsentation verschachtelt liegt. Finden sie die Seite? Welchen Pfad schlagen sie dafür ein? Wie lange dauert es? Wie frustriert sind sie dabei?
- ▶ Bitten Sie sie um Vorschläge. Jeder hat eine Meinung zu den Webseiten eines anderen, aber niemand schickt Ihnen je eine Mail, auch wenn Sie darum bitten. Wenn jemand Ihre Seiten testet, fragen Sie nach Verbesserungsvorschlägen!

Setzen Sie sich mit den Testpersonen zusammen, und machen Sie sich Notizen. Die Ergebnisse sind vielleicht überraschend, und Sie erhalten neue Ideen für die Organisation Ihrer Seiten.

Überprüfen Sie Ihre Protokolle

Eine weitere Methode für den Test Ihrer Dokumente nach der Veröffentlichung im Web ist die Auswertung von Server-Protokolldateien. Ihr Web-Server oder -Anbieter schreibt eine Protokolldatei für jeden Zugriff auf Ihre Seite (immer wenn ein Browser das Dokument anfordert) und woher er kam. Dadurch erhalten Sie mehrere Informationen:

- ▶ Welche Ihrer Seiten die beliebteste ist. Das ist vielleicht eine ganz andere, als Sie erwarten würden. Vielleicht sorgen Sie dann dafür, dass man genau diese Seite noch besser findet.
- ▶ Die Muster, nach denen die Leser auf Ihre Seiten zugreifen, und die Reihenfolge, in der sie sie lesen.
- ▶ Allgemeine Rechtschreibfehler, die die Leser beim Versuch, auf Ihre Seiten zuzugreifen, machen. Dateien, nach denen gesucht wurde, die aber nicht gefunden wurden, erscheinen in Ihren Fehlerprotokollen (die befinden sich in der Regel in demselben Verzeichnis wie die Protokolldateien). Durch die Verwendung symbolischer Links oder Aliase können Sie solche Probleme umgehen, falls sie häufiger auftreten sollten.

Seiten hinzufügen und diese aktualisieren

Selbst nachdem Sie Ihre Seiten veröffentlicht und sie sowohl auf Korrektheit als auch auf Funktionalität getestet haben, ist Ihre Präsentation noch nicht fertig. Man kann sogar sagen, dass sie nie fertig wird. Selbst wenn Sie es schaffen, sie so funktional wie möglich zu machen – es gibt immer wieder neue Informationen, die Sie auf neuen Seiten hinzufügen müssen, Aktualisierungen, Fortschritte im HTML, die Sie nutzen können usw.

Aber wie können Sie Ihre Web-Präsentationen warten? Sie erzeugen neue Seiten und verknüpfen sie durch Links mit den alten. Aber bevor Sie das tun, lesen Sie diesen Abschnitt, damit Sie wissen, wie Sie dabei am besten vorgehen.

Neuen Inhalt hinzufügen

Diesen Abschnitt beginne ich mit einer kleinen Geschichte.

In San Jose, Kalifornien, gibt es eine Touristenattraktion namens Winchester Mystery House. Ursprünglich gehörte dieser Landsitz der Erbin des Winchester-Rifles-Vermögens. Man erzählt sich, dass sie von einem Wahrsager die Botschaft erhalten hatte, dass die Geister aller Menschen, die durch die Winchester-Gewehre gestorben sind,

sie und ihre Familie strafen würden. Von dem Zeitpunkt an glaubte sie, sie könnte die Geister verwirren, wenn Sie an das Winchester-Anwesen ständig neue Räume baute. Das Ergebnis war, dass neue Räume an das bereits existierende Haus angebaut wurden oder wiederum an irgendwelche Anbauten, ohne dass es dafür einen Plan oder irgendein System gab – die Arbeit hörte niemals auf. Es entstanden 160 Räume, Treppenhäuser, die ins Nichts führten, Türen, hinter denen sich nur Wände befanden, geheime Durchgänge sowie ein Grundriss, in dem sich niemand ohne Lageplan zu rechtfindet.

Einige Web-Präsentationen sehen ganz ähnlich aus. Sie haben anfangs eine grundlegende Struktur, sind wohldurchdacht, organisiert und funktional. Je mehr Seiten jedoch in der Präsentation irgendwo hinzugefügt werden, desto schlechter strukturiert wird das Ganze, die ursprünglichen Ziele der Präsentation gehen verloren, und irgendwann hat man nur noch ein Chaos verketteter Seiten, in denen man sich allzu leicht verliert und unmöglich das finden kann, was man eigentlich sucht (siehe Abbildung 21.6).

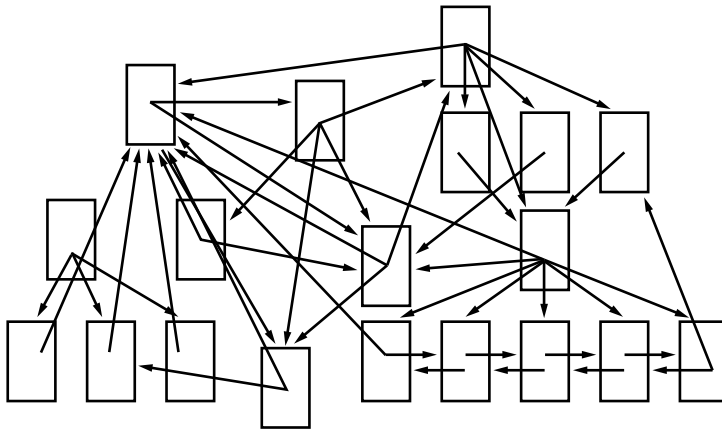


Abbildung 21.7:
Viele Webseiten mit
konfuser
Verknüpfungs-
struktur

Vermeiden Sie, nach dem Modell des Winchester Mystery House vorzugehen! Wenn Sie einer existierenden Präsentation neue Seiten hinzufügen wollen, beachten Sie immer die folgenden Hinweise:

- ▶ **Behalten Sie Ihre Struktur bei.** Wenn Sie den Ratschlägen soweit gefolgt sind, erhält Ihre Präsentation eine grundlegende Struktur, etwa eine Hierarchie oder Linearität. Größtenteils ist es ganz einfach, einer existierenden Struktur neue Informationen hinzuzufügen; es gibt immer einen logisch geeigneten Punkt, an dem sie abgelegt werden können. Versuchen Sie, beim Hinzufügen von Seiten die ursprüngliche Struktur beizubehalten. Wenn Sie dazu zusätzliches Material hinzufügen müssen, damit die neuen Seiten mit den alten übereinstimmen, dann tun Sie das.

- ▶ **Konzentrieren Sie sich auf Ihre Ziele.** Behalten Sie Ihr ursprüngliches Ziel im Auge, wenn Sie einen neuen Inhalt hinzufügen wollen. Wenn der neue Inhalt von Ihren Zielen abweicht oder Sie gar stört, dann verzichten Sie am besten darauf. Wenn sich Ihre Ziele geändert haben, sollten Sie Ihre gesamte Präsentation überarbeiten und nicht einfach nur irgendwelche neuen Informationen einfügen.
- ▶ **Führen Sie gegebenenfalls Verzweigungen ein.** Manchmal fügt man neue Informationen am besten ein, indem man eine völlig neue, untergeordnete Präsentation einfügt, statt zu versuchen, den Inhalt zu ergänzen. Das trifft insbesondere für Hierarchien zu. Wenn der neue Inhalt möglicherweise zu einer eigenen Präsentation gemacht werden kann, dann ziehen Sie diese Lösung in Betracht.

Überarbeiten Sie Ihre Struktur

Manchmal stellt man fest, dass die Präsentation an einem Punkt angelangt ist, wo die ursprüngliche Struktur nicht mehr funktioniert, oder dass sich Ihre Ziele geändert haben und die ursprüngliche Organisation es sehr schwierig macht, zu den neuen Informationen zu gelangen. Vielleicht hatten Sie aber von Anfang an keine Struktur und stellen fest, dass Sie jetzt eine brauchen.

Web-Sites sind lebendige Konstrukte, und wenn Sie sie ändern, dann kann es notwendig werden, Ihren ursprünglichen Plan oder die Struktur zu überdenken. Bleibt zu hoffen, dass Sie dabei nicht ganz von neuem beginnen müssen. Oft gibt es eine Möglichkeit, Teile der Präsentation zu überarbeiten, sodass die neuen Informationen eingefügt werden können und die Gesamtpräsentation erhalten bleibt.

Manchmal ist es nützlich, zum ursprünglichen Plan für die Präsentation zurückzugehen (sie haben doch einen?) und ihn zu überarbeiten, sodass Sie Ihre Ziele wieder deutlicher sehen. Insbesondere sollten Sie Folgendes ausprobieren:

- ▶ **Listen Sie die Ziele Ihrer Präsentation auf:** Wie sollen die Leser sie verwenden, und wie soll sie ankommen? Vergleichen Sie die neuen mit den alten Zielen. Wenn sie sich unterscheiden, suchen Sie nach Wegen, wie Sie die bereits existierende Präsentation modifizieren können, sodass Ihre Leser die neuen Ziele erreichen können.
- ▶ **Modifizieren Sie Ihre Themenliste.** Das ist in der Regel die schwierigste Aufgabe, weil dabei Teile aus anderen Themen betroffen sind und vieles verschoben werden muss. Versuchen Sie zu verfolgen, welche Themen alt sind, welche neu. Das hilft Ihnen, wenn Sie anfangen, Ihre Seiten zu bearbeiten.
- ▶ **Ändern Sie die Struktur, wenn sie nicht funktioniert.** Wenn Sie eine einfache Web-Struktur hatten, die jetzt zu komplex geworden ist, als dass man sich in ihr bewegen könnte, dann überlegen Sie, ob Sie Ihrer Präsentation nicht eine

strengere Struktur auferlegen können. Wenn Sie eine sehr flache Hierarchie verwenden (mit sehr wenigen Ebenen, aber sehr vielen Optionen auf der ersten Seite), sollten Sie versuchen, diese auszugleichen (mehr Ebenen, weniger Optionen).

Wenn Sie einen neuen Plan haben, sehen Sie in der Regel Bereiche, wo das Verschieben der Seiten oder des Inhalts auf manchen Seiten die Dinge wieder klarer macht. Behalten Sie Ihren neuen Plan während der Änderungen im Auge, und versuchen Sie, schrittweise vorzugehen. Wenn Sie zu viele Veränderungen gleichzeitig ausführen, zerstören Sie möglicherweise Links oder verlieren den Überblick. Wenn Sie Ihre Seiten auf deren Funktionalität geprüft haben, berücksichtigen Sie die Hinweise, die Sie aus dieser Arbeit gewonnen haben.

Zusammenfassung

Planen, schreiben, testen und warten sind die vier Meilensteine beim Entwurf von Webseiten. Planen und schreiben – unter anderem ist das, eine Struktur zu entwickeln, Seiten zu erzeugen, sie zu verknüpfen und dann das Ergebnis zu verfeinern –, das haben Sie bereits durch das ganze Buch hindurch kennen gelernt. In diesem Kapitel haben wir Ihnen die andere Hälfte des Prozesses gezeigt. Diese Hälfte geht jedoch noch weiter, nachdem Sie alles veröffentlicht haben und die Leute Ihre Seiten schon lange lesen.

Beim Testen wird sichergestellt, dass Ihre Seiten funktionieren. Vielleicht haben Sie bisher Ihre Seiten schon mit dem einen oder anderen Browser getestet, Ihre Links überprüft und sichergestellt, dass alle Ihre CGI-Scripts korrekt installiert und von der richtigen Stelle aus aufgerufen werden. Aber hier haben Sie gelernt, richtig zu testen – Integritäts-Tests mit HTML-Validatoren und automatische Programme für die Überprüfung von Links sowie Verwendungstests, mit denen man feststellen kann, wie praktisch die Leser Ihre Seiten finden.

Die Wartung erfolgt, wenn Sie Ihrer Präsentation neue Informationen hinzufügen und sicherstellen, dass alles noch zusammenpasst und funktioniert. Die Wartung verhindert, dass Ihre ursprüngliche Planung durch die Veränderungen zerstört wird. Mit der Wartung wird gewährleistet, dass Ihre grundsätzliche Planung der Site und die darin enthaltenen ursprünglichen Informationen durch neue Informationen nicht unverständlich werden. Für eine angemessene Wartung müssen Sie ab und zu eine neue Site-Struktur und neue Seiten erstellen. Wie man das macht und wie man die Seiten korrekt revidiert, haben Sie ja heute gelernt.

Damit sind Sie fertig. Oder zumindest, bis es wieder an der Zeit ist, etwas zu ändern.

Workshop

Wir sind am Ende des Buches angelangt. Was Präsentation und Veröffentlichen Ihrer Webseiten im World Wide Web anbelangt, sind Sie bestens ausgerüstet. Dieser Workshop, der letzte, befasst sich mit einigen Fragen über HTML-Validierung und enthält außerdem ein Quiz und Übungen, die einige der wichtigsten Punkte dieses Buches betreffen.

Fragen und Antworten

F Ich verstehe immer noch nicht, warum die HTML-Überprüfung notwendig ist. Ich teste meine Seiten in allen möglichen Browsern. Warum soll ich diese zusätzliche Arbeit tun, um sie endgültig HTML-konform zu machen? Spielt das eine Rolle?

A Betrachten Sie es so. Stellen Sie sich vor, nächstes Jahr bringt die Web-Firma Z irgendein wahnsinnig praktisches Autorenwerkzeug für HTML heraus, das es Ihnen ermöglicht, schnell und einfach Webseiten zu erzeugen, sie zu verknüpfen, Hiearchien aufzubauen, die Sie visuell verschieben können, und alles mögliche andere, was bisher nur unter Mühen realisiert werden kann. Und das Werkzeug ist sogar in der Lage, Ihre alten HTML-Dateien einzulesen, so dass Sie nicht alles von Grund auf neu schreiben müssen.

Toll, sagen Sie. Sie kaufen das Programm und versuchen, Ihre HTML-Dateien einzulesen. Aber Ihre HTML-Dateien enthalten Fehler. Diese Fehler traten in den Browsern nie zutage, aber nichtsdestotrotz handelt es sich um Fehler. Da das Autorenwerkzeug nicht so nachsichtig ist wie die Browser (und das muss sein bei den Funktionen, die es bietet), können Sie Ihre Originaldateien nicht mehr lesen, bevor Sie sie nicht manuell korrigiert haben. Das kann dauern, wenn Sie in jeder Datei nur ein paar Fehler gemacht haben. Sie sparen sich also viel Zeit, wenn die Dateien von vorneherein korrekt sind.

F Muss ich all meine Dateien mit Weblint und dem HTML-Validator des W3C überprüfen? Das ist unheimlich viel Arbeit.

A Sie müssen nicht, wenn Sie nicht die Zeit oder die Geduld haben. Aber ich kann es Ihnen nur immer wieder empfehlen, weil beide Programme unterschiedliche Funktionen bieten, die gleichermaßen wichtig sind. Weblint zeigt Ihnen die offensichtlicheren Fehler auf Ihren Seiten und bringt noch andere praktische Informationen. Es entdeckt beispielsweise fehlenden ALT-Text. Der HTML-Validator ist vollständiger, aber auch strenger. Er zeigt Strukturfehler in Ihren Dokumenten auf, aber die Fehlermeldungen sind extrem schwierig zu verstehen.

Quiz

1. Richtig oder falsch? Um Webseiten zu erstellen, muss man einzig und allein HTML beherrschen.
2. Wie kann man den Umfang von medienintensiven Webseiten, also solchen Seiten, die viele Grafiken und Multimedia-Elemente enthalten, reduzieren?
3. Auf was müssen Sie achten, wenn Sie Webseiten mit Framesets gestalten?
4. Richtig oder falsch? Wenn alle Links auf dem lokalen Computer funktionieren, muss man sie, wenn man sie in eine entfernte Web-Site herauflädt, nicht mehr überprüfen.
5. Womit können sich Ihre Webseiten besser lesen lassen?

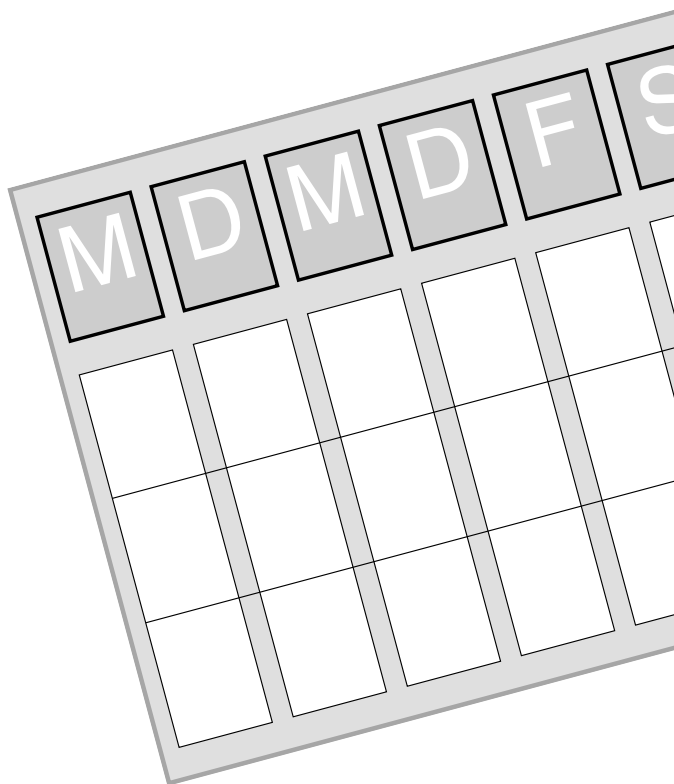
Antworten

1. Das mochte vielleicht im Anfangsstadium des Web richtig gewesen sein, entspricht aber heute nicht mehr der Realität. Wenn Sie beim Erstellen Ihrer Webseiten nur HTML-Tags verwenden, müssen Sie trotzdem noch andere Technologien kennen: etwa Cascading Style Sheets (siehe Tag 10), JavaScript und Dynamic HTML (siehe Tag 15), um nur einige Technologien zu nennen, mit denen Sie dem derzeitigen Stand der Webseiten erreichen können.
2. Reduzieren Sie die Größe der Grafiken und Animationen. Indem Sie Bilder (JPG) komprimieren, können Sie auch die Bildergröße reduzieren. Sollten diese Maßnahmen nichts bringen, können Sie immer noch ein Minibild des Bildes oder einen der Animation-Frames anbieten, die der Leser dann herunterladen oder ansehen kann. Zusätzliche Informationen hierzu können Sie im Kapitel des 7. Tages nachschlagen.
3. Wenn Sie mit Framesets arbeiten, unterteilen Sie den Browser-Bildschirm nicht in zu viele Frames. Das könnte den Leser verwirren. Außerdem hätten Sie nicht genügend Platz, Ihre Seiten angemessen aufzuzeigen, da die Auflösungen zu niedrig wären. Sollte eine der Seiten Links zu anderen Seiten oder Sites enthalten, denken Sie daran `TARGET="TOP"` zu verwenden, welches dem Leser die Möglichkeit gibt, aus Ihrem Frameset herauszugehen und zu anderen Sites zu navigieren. Sollten Ihre geframten Seiten Grafiken enthalten, reduzieren Sie deren Größe, damit diese auch in die geframten Seiten mit niedrigeren Auflösungen passen. Bieten Sie Inhalte innerhalb eines `<noframes>`-Elements an, damit Leser, die keinen Frame-kompatiblen Browser benutzen, ebenfalls Zugang zum Inhalt Ihrer Site haben. Zusätzliche Informationen erhalten Sie im Kapitel des 19. Tages.

4. Falsch. Wenn Sie Ihre Webseiten auf einen entfernten Server übertragen, können Links ungültig werden. Schlagen Sie im Kapitel des 19. Tages nach, um Genaueres darüber zu erfahren.
5. Versuchen Sie nicht zu viele Inline-Hyperlinks zu verwenden. Verwenden Sie stattdessen Listen und Tabellen. Außerdem sollten Ihre Seiten nicht zu lange Paragraphen enthalten, Überschriften als Überschriften verwendet werden und nicht dafür, wichtige Punkte der Seiten hervorzuheben. Übertreiben Sie es nicht mit der Text hervorhebung (kursiv und fett), da dies vom eigentlichen Textinhalt ablenkt. Stattdessen sollten Sie mit animierten Grafiken arbeiten, um die Aufmerksamkeit des Lesers auf besonders wichtige Punkte Ihrer Seite zu lenken. Schlagen Sie noch einmal in den Kapiteln Tag 16 und Tag 17 nach.

Übungen

1. Da wir schon fast am Ende dieses Buches angelangt sind, kann es logischerweise nur eine Übung geben: Erstellen Sie Ihre eigene Web-Site (falls Sie das nicht schon getan haben). Verwenden Sie hierfür Tags, die Sie gut kennen, fügen Sie Grafiken, Links und Tabellen ein. Die etwas Fortgeschritteneren unter Ihnen können Frames, Cascading Style Sheets, Dynamic HTML und andere HTML-Datenerweiterungen verwenden.
2. Veröffentlichen Sie Ihre Webseiten im World Wide Web so, wie es im Kapitel des 19. Tages besprochen wurde. Überprüfen Sie zuerst Ihre Seiten und lassen Sie anschließend jedermann wissen, dass Sie im Web sind. Denken Sie daran: Das Schöne am World Wide Web ist, dass Ihre Seiten nicht in Stein gemeißelt sind und sie jederzeit verändert, gelöscht oder verbessert werden können.



Stichwortverzeichnis



169
" 705
<a> 112
<address> 169
 156
<basefont> 179
<big> 157
<blink> 180
<blockquote> 170
<body> 82, 230, 233

 167, 181

 369
<caption> 353
<center> 177, 369
<cite> 154, 170
<code> 154
<dd> 98
<dfn> 154
<div> 176, 373
<dt> 98
 154
 177, 232
<h1> 84
<h2> 84
<h3> 84
<h4> 84
<h5> 84
<h6> 84
<head> 81
<html> 81
<i> 156
 202
<kbd> 154
 89f.
<menu> 126
<nobr> 181
<p> 87
<s> 157
<samp> 154
 154
<sub> 157
<sup> 157
<table> 348
<td> 349
<th> 349
<title> 83
<tr> 349

<tt> 156
<u> 157
 95, 126
<var> 154
<wbr> 181
»404, Datei nicht gefunden« 703
8.3-Konvention 711

A

a, Attribute 113
Abbildungen 638
–, interne 200
Abbildungsplan 268
Alias 703
align-Attribut 165, 175, 208, 372
alink-Attribut 231
AltaVista 731
Ampersand 172
Anchors 128
Anker 132
Anonymous FTP 140
Apache Web Site 20
Attribute 113, 202, 221, 237
–, ALIGN 372
–, align 165, 175, 208
–, alink 231
–, background 233
–, bgcolor 230
–, bgproperties 234
–, border 217, 227
–, CELLPADDING 359
–, CELLSPACING 360
–, CLEAR 369
–, clear 169
–, color 232
–, face 179
–, HREF 113
–, href 119, 121
–, link 231
–, lowsrc 228
–, NAME 129
–, noshade 166
–, size 178
–, src 202, 228
–, text 231
–, type 92, 96
–, VALIGN 372

–, value 95
 –, vlink 231
 –, vspace 215
 –, width 164
 Auszeichnungen 63, 80f., 118
 Authentifizierung 704

B

b (Stil-Auszeichnung) 156
 background-Attribut 233
 Beglaubigung und Sicherheit 704
 Benannte Entitäten 172
 bgcolor-Attribut 230
 bgproperties-Attribut 234
 Bilder als Hintergründe 233
 Bild-Formate 222
 Binär-Dateien 711
 Binär-Modus 711
 blockquote 170
 BMP 222
 body 105f.
 body (Auszeichnung) 82
 border-Attribut 217, 227
 br/> 167
 Browser 17f., 24, 638
 –, Lynx 31
 –, Mosaic 18

C

Carriage Return 712
 Cell Padding 359
 CGI-Scripts 704
 Character Entities 172
 Character Styles 152
 clear-Attribut 169
 code (Stil-Auszeichnung) 154
 color-Attribut 232
 COLS 411

D

Dateiendungen 222, 710
 Dateitypen 710
 Daumennagel 223
 dd (Auszeichnung) 89
 definition term 98
 Definitions-Listen 98
 dfn (Stil-Auszeichnung) 154

Drehbuch-Schreiben 55
 dt (Auszeichnung) 89
 Dungeon 52

E

em (Stil-Auszeichnung) 154
 E-Mail 21, 141
 Enterprise-Server 713
 Ersatzzeichen 174
 Explorer 26
 External Images 201

F

face-Attribut 179
 Farbennennungen 229
 FastTrack-Server 713
 File, David 725
 File-URL 143
 Formular 23
 Frame-Definitionsdocument 409
 FTP 21
 FTP-URL 140

G

Gestaltung 616
 GIF 201, 222
 GIF87 201
 GIF89 201
 Glossar-Listen 98
 Gopher 21
 Gopher-URL 141

H

h1 bis h6 (Überschriften) 84, 105
 head 105f.
 head (Auszeichnung) 81
 Home-Page 37f., 714
 Horizontale Linien 162
 Host-Namen 136
 hr/> (Auszeichnung) 162
 href-Attribut 113, 119, 121
 hspace-Attribut 215
 HTML 14, 25, 62
 –, auswerten 747
 html (Auszeichnung) 81, 105f.
 HTML DTD 747
 HTML-Definition 747



HTML-Dokument
–, formatieren 73
–, Strukturierung 80
HTML-Kennungen 69
HTML-Validator 747
HTTP 21, 139, 702
HTTPD 702
HTTPD-Server 702
HTTP-URL 139
Hypermedia 14
Hypertext 14
Hypertext Markup Language 25, 62
Hypertext Transfer Protocol 21, 139
Hypertext-Auszeichnungssprache 62

I

i (Stil-Auszeichnung) 156
Icons 219
Imagemap 268
–, Imaptool 274
–, LiveImage 274
–, Mapedit 274
–, Poor Person es Image Mapper 274
Infoseek 734
Inline Images 201
Integritätstest 746
Internet Explorer 26
Internet Service Provider 705
ISO-Latin-1-Zeichensatz 173
ISP 705

J

JPEG 201, 222

K

kbd (Stil-Auszeichnung) 154
Kennung 63
Kennungs-Sprache 66
Klienten 24
Kommentare 106
Konverter 74
Konvertieren 712

L

Layout 638
Leere Zellen 352
li (Auszeichnung) 89, 126

Linefeed 712
Link Menus 125
link-Attribut 231
List Item 90
Listen
–, Auszeichnungen 89
–, Glossar-Listen 89, 98
–, nummerierte oder geordnete 90
–, ungeordnete 89, 95
–, verschachtelte 100
Listen-Eintrag 90
Listen-Kennungen 89
List-Tags 89
LiveWire 713
Logische Stil-Auszeichnungen 153
Logische Stile 153
lowsrc-Attribut 228
Lycos 729, 733

M

Mailto 636
Mailto-URL 141
MapThis 274
markup language 66
menu (Auszeichnung) 89
Message-ID 142
Microsoft Internet Explorer 26
monospaced 159
MUD 52
Multi-User Dungeon 52

N

NAME-Attribut 129
Navigation 44
Navigations-Verbindungen
–, explizite 624
–, implizite 625
Netscape 27
Netscape Communications Corporation 27
Newsgroup 142, 737
News-URL 142
no break 181
nobr 181
Non-Anonymous FTP 141
noshade-Attribut 166
Numerische Entitäten 172

O

ol (Auszeichnung) 89
 Online-Hilfe 15
 Options-Felder 23
 Ordered List 90
 Organisation 44, 764

P

p (Auszeichnung) 87, 105
 Passwort 704
 Physische Stil-Auszeichnung 153, 156
 Physische Stile 156
 PICT 222
 Plandatei 273
 Port-Nummer 137
 PostMaster2 735
 PPP-Internet-Verbindung 26
 PPP-Verbindung 714
 Präsentationen überarbeiten 764
 pre (Auszeichnung) 159
 preformatted 159
 Protokoll 136
 Protokolldatei 740

Q

QuickTime-Video 746

R

Rechtschreibung 637
 Remote Pages 122
 Remote Server 712
 Remote-Management-Tool 712
 Reservierte Zeichen 174

S

samp (Stil-Auszeichnung) 154
 Schriftart 177
 Schriftgrad 177f.
 Server-Includes 741
 SGML 62, 747
 Signatur 188, 635
 Site-Indizes 729
 SiteManager 713
 size-Attribut 178
 SLIP 26
 SLIP-Verbindung 714

SRC-Attribut 228

src-Attribut 202

Standard Generalized Markup Language 62

Standard-Indexdatei 709

Stil-Auszeichnung 153

Storyboard 55

Strategie für den Einsatz von HTML-Erweiterungen 607

strong (Stil-Auszeichnung) 154

Submit It! 736

Suchmaschinen 729

–, AltaVista 731

–, Excite 731

Symbole 219

T

Tabellendaten 349

Tabellenzeile 349

Tabellenzellen 349

Table Data 349

Table Header 349

Table Row 349

Tag 63

Tags

–, 169

–, <a> 112

–, 156

–, <basefont> 179

–, <big> 157

–, <blink> 180

–, <blockquote> 170

–, <body> 82, 230, 233

–,
 167, 181

–,
 369

–, <caption> 353

–, <center> 177, 369

–, <cite> 154, 170

–, <code> 154

–, <dd> 98

–, <dfn> 154

–, <div> 176, 373

–, <dt> 98

–, 154

–, 177, 232

–, <h1> 84

–, <h2> 84



- , <h3> 84
- , <h4> 84
- , <h5> 84
- , <h6> 84
- , <head> 81
- , <html> 81
- , <i> 156
- , 202
- , <kbd> 154
- , 89f.
- , <menu> 126
- , <nobr> 181
- , <p> 87
- , <s> 157
- , <samp> 154
- , 154
- , <sub> 157
- , <sup> 157
- , <table> 348
- , <th> 349
- , <title> 83
- , <tr> 349
- , <tt> 156
- , <u> 157
- , 95, 126
- , <var> 154
- , <wbr> 181
- TARGET 402
- Telnet 21
- Testprogramm 747
- text-Attribut 231
- thumbnail 223
- Titelleiste 83
- title (Auszeichnung) 83, 105
- tt (Stil-Auszeichnung) 156
- type-Attribut 92, 96

U

- Überschriften 84
- ul (Stil-Auszeichnung) 126
- Ungeordnete Listen 95
- Uniform Resource Locator 19, 29, 136
- URL (Uniform Resource Locator) 14, 19, 29, 123, 133, 136, 145
- , Ersatzzeichen 138

- , FTP-URL 140
- , Host-Name 136
- , Port-Nummer 137
- , Protokoll 136
- , Spezial-Zeichen 138
- , Verzeichnis 136
- Usenet 142, 737
- Usenet News 21
- Usenet News-URL 142

V

- value-Attribut 95
- var (Stil-Auszeichnung) 154
- Verbindung mit anderen Dokumenten 122
- Verknüpfung 112, 135
- , Auszeichnung 116, 144, 160
- Verschachtelte Listen 100
- vlink-Attribut 231
- Vorformatierter Text 159
- vspace-Attribute 215

W

- W3 14
- Wagenrücklauf 712
- WAIS-Datenbanken 21
- Web Browser 24
- Web Site-Strukturen
- , Hierarchien 44
- , linear 47
- , linear mit Alternativen 48
- , linear und hierarchisch 50
- , verwoben 52
- WebCrawler 729
- Weblint 749
- WebMap 274
- Web-Master 707
- Web-Präsentation 37, 746
- Webseite 38
- Web-Server 28, 702, 707
- WebSite 19
- What You See Is What You Get 62
- white space 159
- width-Attribut 164
- Winchester Mystery House-Modell 763
- word break 181

World Wide Web 14, 30

–, Haupt-Themen 42

–, Server 28

World Wide Web Virtual Library 728

World Wide Web-Dokumente 37

–, Hierarchien 44

–, lineare Struktur 47

–, lineare Struktur mit Alternativen 48

–, verwobene 52

World Wide Web-Präsentation 37

–, Ziele 41

World Wide Web-Server 14

WWW 14

WYSIWYG, Prinzip 62

Y

Yahoo! 725

Yang, Jerry 725

Z

Zeichen-Entitäten 172

Zeichen-Gestaltung 152

Zeilenumbruch 167

Zellenabstände 360

Zellenausrichtung 369

Zork 52

Zugriffsberechtigung 716

Zugriffszähler

–, Jcount 742

–, LiveCounter 742

–, Net-Trak 742

–, WebTracker 742

