

easy

**Die eigene
Website**

**Unser Online-Tipp
für noch mehr Wissen ...**



... aktuelles Fachwissen rund
um die Uhr – zum Probelesen,
Downloaden oder auch auf Papier.

www.InformIT.de

Die eigene Website

Sie können es!

CHRISTIAN WENZ TOBIAS HAUSER



→leicht →klar →sofort

eBook

Die nicht autorisierte Weitergabe dieses eBooks
an Dritte ist eine Verletzung des Urheberrechts!

Bibliografische Information Der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

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.

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 Softwarebezeichnungen und weitere Stichworte und sonstige Angaben, die in diesem Buch verwendet werden, sind als eingetragene Marken geschützt.

Da es nicht möglich ist, in allen Fällen zeitnah zu ermitteln, ob ein Markenschutz besteht, wird das ®Symbol in diesem Buch nicht verwendet.

Umwelthinweis:

Dieses Buch wurde auf chlorfrei gebleichtem Papier gedruckt.

10 9 8 7 6 5 4 3 2 1

08 07 06 05

ISBN 3-8272-6921-0

© 2005 by Markt+Technik Verlag,

ein Imprint der Pearson Education Deutschland GmbH,

Martin-Kollar-Straße 10–12, D-81829 München/Germany

Alle Rechte vorbehalten

Coverkonzept: independent Medien-Design, Widenmayerstraße 16, 80538 München

Coverlayout: adesso21, Thomas Arlt, München

Titelfoto: image 100

Herstellung: Monika Weiher, mweiher@pearson.de

Lektorat: Birgit Ellissen, bellissen@pearson.de

Korrektorat: Marita Böhm, München

Satz: mediaService, Siegen (www.media-service.tv)

Druck und Verarbeitung: Kösel, Krugzell (www.KoeselBuch.de)

Printed in Germany

Inhaltsverzeichnis

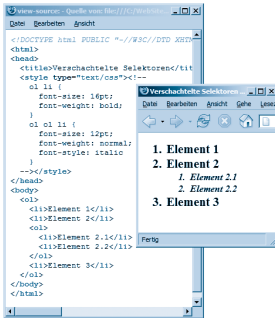
Liebe Leserin, lieber Leser!..... 9

Los geht's 10

Das Web..... 12
 Browser – wofür schreiben Sie?..... 14
 Editor – wo schreiben Sie? 21
 Was Sie sonst noch brauchen 22



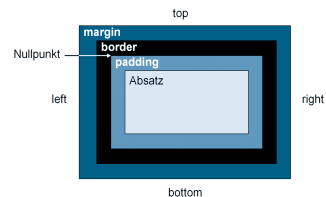
1 Die Grundlagen – HTML und CSS 24



HTML 26
 CSS 40

2 Einfache Webseiten mit Text und Links 54

Formatierung 56
 Überschriften 64
 Absätze aufmöbeln 68
 Links 75



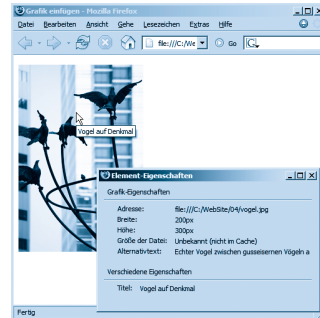
3 Strukturierte Daten – Listen und Tabellen 88



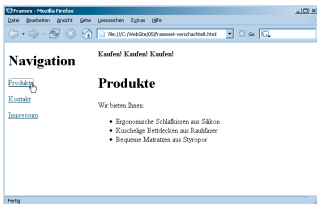
Listen	90
Tabellen.....	95

4 Bunte Bilder 116

Formate und Optimierung.....	118
Einbau	121
Ausrichtung und Abstände	124
Verlinkung.....	127
Hintergründe	132



5 Seitenlayout mit Frames 134



Was sind Frames?	136
Vorbereitungen	137
Frames erstellen	142
Frames verschönern	146
Frames verschachteln.....	150
Auf Frames verlinken	152
Frames mitten auf der Seite: Iframes	155

6 Feedback mit Formularen einholen 158

Wohin mit den Daten?.....160

Formulare versenden 162

Textfelder.....165

Mehrzeilige Textfelder 168

Passwortfelder.....170

Radiobuttons.....173

Checkboxes.....175

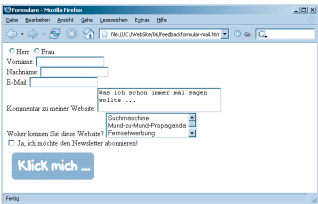
Auswahllisten 178

Mehrfache Auswahllisten.....183

Grafische Versenden-Schaltflächen.....184

Versand per E-Mail 187

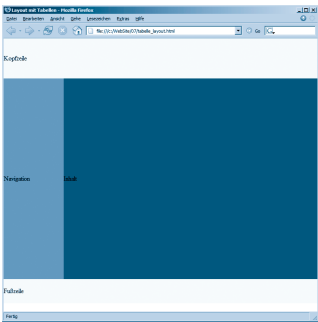
Versand per E-Mail, aber richtig.....190



7 Webseiten aufbauen – Elemente positionieren 194

Tabellen.....196

Positionieren mit CSS.....201

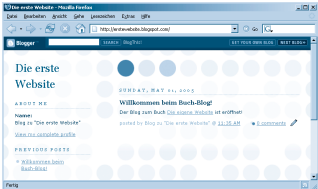


8 Nützliche Erweiterungen für die Website 212

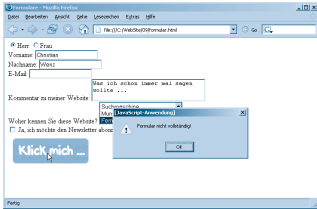
Ein Gästebuch einrichten 214

Ein Diskussionsforum erstellen.....222

Einen Weblog anlegen 232



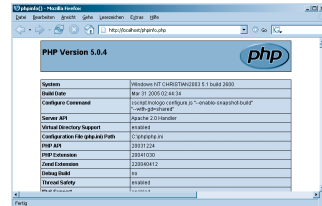
9 Mehr Pep mit JavaScript 238



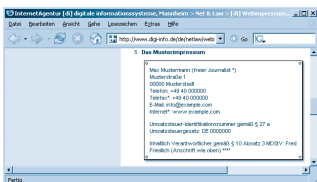
Was ist JavaScript?	240
JavaScript einbauen	241
Den Internet Explorer unter Windows XP Service Pack 2 zur Kooperation bewegen.....	246
Einstieg in die Programmierung.....	249
Beispiel: Das aktuelle Datum ausgeben.....	258
Beispiel: Quicknagation	260
Beispiel: Formularüberprüfung	263
Beispiel: Mehrere Frames auf einmal ändern	267
Beispiel: Frame nachladen	270
Was nicht funktioniert.....	273

10 Dynamische Websites mit PHP 274

Was ist PHP?	276
Installation	278
Ein erster Test	290
Beispiel: Das aktuelle Datum ausgeben	295
Beispiel: Formularversand per E-Mail	298



11 Die Site online stellen 302



Ein Impressum erstellen.....	304
Die Seite online stellen	309

Lexikon 317

Stichwortverzeichnis 329



Liebe Leserin, lieber Leser!

Hätten Sie nicht auch gerne eine eigene Homepage, ein eigenes Zuhause im World Wide Web? Millionen Leute haben bereits diesen Schritt gewagt und sind mit ihrer Firma, ihrem Verein oder ihren Hobbys online gegangen. Eine gut gemachte Webpräsenz zieht Besucher an und kann interessante neue Kontakte zu Gleichgesinnten schaffen.

Der Weg dorthin ist gar nicht so schwer. In diesem Buch finden Sie die grundlegenden Dinge, die Sie bei Ihrer Exkursion in die Gestaltung Ihrer eigenen Website benötigen. Wir beginnen mit dem Aufbau einfacher Websites und zeigen Ihnen nach und nach spannende Erweiterungsmöglichkeiten. Außerdem erfahren Sie, wie Sie Feedback von Ihren Benutzern einholen und ein Gästebuch und ein Diskussionsforum in Ihre Webseite integrieren können.

Doch auch nach der Lektüre des Buchs lassen wir Sie nicht allein. Unter <http://www.hauser-wenz.de/support/> finden Sie alle Übungslektionen dieses Buchs zum Download und direkten Ausprobieren. In jedem Beispiel in diesem Buch finden Sie – in der Form eines Kommentars – den zugehörigen Dateinamen eingefügt. Außerdem können Sie uns auf unserer Website – zu Buchthemen – jederzeit gerne kontaktieren; in der Regel antworten wir in 24 Stunden.

Bleibt uns nur noch, Ihnen gute Reise zu wünschen – und viel Erfolg mit Ihrer eigenen Website!

Die Autoren

P.S.: Jedes Kapitel endet mit einem Satz an »nutzlosem Wissen«, eine Idee, die wir Uwe Baumann verdanken. Nicht alles davon entstammt unseren eigenen Recherchen, sondern wurde von uns auf den sehr amüsanten Webseiten zum Thema wie etwa <http://www.amusingfacts.com/> und <http://www.angelfire.com/ca6/uselessfacts/> gefunden.

Los geht's



Eine erste Website erstellen Sie innerhalb von kurzer Zeit. Damit dabei auch nichts schief geht, sind ein paar Grundlagen notwendig. Damit das Ganze nicht stupide wird, sparen wir in diesem Kapitel auch nicht mit interessanten Geschichten und hilfreichen Informationen. Wenn Sie dieses Kapitel gelesen haben, wissen Sie, was im Web auf Sie zukommt, was Sie für die übrigen Kapitel benötigen und in welchem Umfeld Sie arbeiten.



Das lernen Sie neu:

Der Aufbau des Web	12
Die Wichtigkeit von Browsern	14
Mit welchen Editoren Sie arbeiten können	21

Das Web

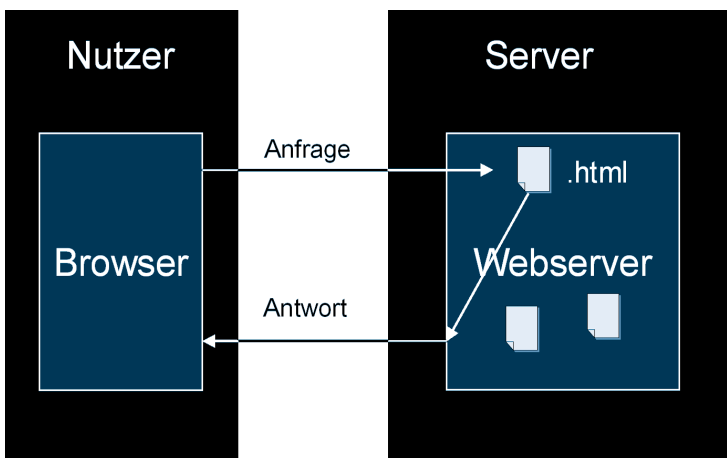
Als Tim Berners-Lee Anfang der 90er Jahre das Web erfand, hatte er nicht mit dem immensen Erfolg gerechnet. Heute hört man schon im Restaurant am Tisch nebenan, dass mal schnell nach einem Thema »gegoogelt« wird oder man ein Buch online kauft. Und auch Sie wollen mit Ihrer ersten Seite ins Web.

Wie kam es dazu? Aus technischer Sicht liegt der große Erfolg wohl in der einfachen Architektur begründet. Man hängt einen Webserver an einer beliebigen Stelle ins Internet und die darauf befindlichen Seiten lassen sich weltweit abrufen. Ein Webserver ist ein leistungsfähiger Rechner, der permanent am Internet hängt und eine spezielle Webserver-Software enthält. Oftmals wird auch diese Software allein als Webserver bezeichnet.

Sehen wir uns etwas genauer an, wie das funktioniert:

- Auf dem Webserver liegen die Webseiten. Ohne Webserver geht es nicht.
- Der Nutzer (auch Client genannt) surft mit einem Browser.
- Der Nutzer fragt beim Server eine Webseite an. Diese Anfrage erfolgt mit einer Internetadresse.
- Der Server findet anhand der Adresse die passende Seite und liefert diese an den Browser.

Dieser Verbindungsvorgang erfolgt über das HTTP-Protokoll. HTTP steht für HyperText Transfer Protocol – das Übertragungsprotokoll für Hypertext, d.h. Text mit Verlinkungen. Anfrage (Request) und Antwort (Response) bilden dabei eine Verbindung. Als Variante gibt es noch eine sichere Verbindung über HTTPS. Dabei werden Anfrage und Antwort verschlüsselt.



Hinweis

*Ein wenig schwierig zu verstehen ist das Adresssystem des Web. Eine Website wird im Allgemeinen per Domain identifiziert. Eine Domain ist z.B. **www.mut.de**. Diese Domain muss einem Webserver oder bei einem Webpace-Anbieter einem Teil des Webserver zugeordnet werden. Die grundlegende Adressierung läuft über IP-Adressen. Eine IP-Adresse besteht aus vier Ziffern (bzw. beim neuen IPv6 aus sechs Ziffern). Diese Ziffern sind die Zahlen von 0 bis 255. Daraus ergibt sich dann z.B. 192.168.0.1. Der Bereich 192.168 ist übrigens für private Netzwerke reserviert, d.h. den können Sie für Ihre Heimnetze verwenden.*

Die Umsetzung von IP-Adressen in Domain-Namen – denn Menschen können sich diese besser merken als die Ziffernfolge – erfolgt mit so genannten Domain-Name-Servern. Mit diesem Webhintergrund kommen Sie allerdings in der Praxis kaum in Berührung.

Als Tim Berners-Lee Aufbau und Architektur des Web ausprobiert hatte, benötigte er eine Beschreibungssprache, also eine Sprache, um die Inhalte – die Webseiten – darzustellen. Die Wahl fiel auf eine Eigenkreation, HTML. HTML steht für HyperText Markup Language, also so viel wie Hypertext-Auszeichnungssprache. Alle Details zu HTML und zu der Ergänzung CSS finden Sie im nächsten Kapitel. Hier wollen wir uns das aus Sicht der Architektur etwas näher ansehen: HTML und auch Technologien wie CSS und die Skriptsprache JavaScript werden vom Browser ausgeführt. Deswegen heißen sie auch **clientseitig**. Das heißt, der Server schickt dem Browser die HTML-Seite und dieser stellt sie dann dar. Dies erklärt auch schon, warum der Browser im Web eine so große Rolle spielt. Sie selbst als Website-Betreiber mit einem Webserver oder einem Website-Platz bei einem Hoster haben keinen Einfluss darauf, welchen Browser Ihre Nutzer verwenden. Sie müssen sich also anpassen und versuchen, möglichst viele Browser zu unterstützen.

Hinweis

Zu den clientseitigen Technologien gehören auch Flash, das Vektoranimationsformat von Macromedia, und Java-Applets, die von einer Java Virtual Machine ausgeführt werden. Flash hat in der Praxis hohe Bedeutung, ist allerdings als komplexes Thema nicht Teil dieses Buchs. Java-Applets werden heute nur noch in speziellen Anwendungen verwendet.

Das Gegenstück zu clientseitigen Technologien sind **serverseitige**. In diese Kategorie gehören PHP, ASP.NET und einige andere. Eine serverseitige Technologie wird immer auf dem Webserver interpretiert oder ausgeführt. Der Browser erhält dann nur das Ergebnis, das wiederum aus HTML und CSS besteht. Sie sehen also, auch wenn Sie PHP einsetzen, müssen Sie am Schluss HTML und CSS liefern. Mehr zu serverseitigen Technologien erfahren Sie in *Kapitel 10*.

Browser – wofür schreiben Sie?

Wenn Sie Ihre erste Webseite entwickeln, sollten Sie sich klar machen, dass Sie nicht nur für den einen Browser schreiben, mit dem Sie surfen. Natürlich verrät ein Blick auf die Marktanteile, dass der Internet Explorer weit in Front liegt. Sie sollten dennoch auf möglichst vielen Browsern Ihre Webseite betrachten und kontrollieren. Hierzu können Sie problemlos mehrere Browser gleichzeitig installieren. Einzig bei älteren Versionen des Internet Explorer funktioniert dies nicht.



Allerdings wankt sein Marktanteil dank des Mozilla Firefox (<http://www.mozilla.org/>). Die Mozilla Foundation ist eine Open Source-Organisation, die nicht nur den Firefox, sondern auch die Mozilla Browser Suite herausgibt. Dort sind dann Mailprogramm und Ähnliches mit integriert. Der Firefox ist ein reiner Browser ohne zusätzliche Programme und kommt auch in diesem Buch für die meisten Bildschirmfotos zum Einsatz. Er konnte seinen Marktanteil je nach Messung innerhalb von kurzer Zeit auf fünf bis zehn Prozent steigern. Microsoft hat bereits auf die Sicherheitsdebatte und die Zugewinne des Firefox reagiert und bringt eine neue Version des Internet Explorers, den Internet Explorer 7, heraus. Die Wirkung dieser Maßnahme bleibt abzuwarten.



Mozilla und Firefox haben ihre Wurzeln im Urbrowser des Web, im Netscape Navigator (<http://www.netscape.de/>). Dieser Browser hat sich ursprünglich aus dem Mosaic, einem Universitätsprojekt, entwickelt. Der Netscape Navigator und Microsoft mit dem Internet Explorer haben sich dann über Jahre einen Browserkrieg um die Marktführerschaft geliefert. Netscape hat schließlich und endlich technisch nicht mehr mitgehalten. In der vierten Generation der Browser ist der Internet Explorer vorbeigezogen. Der Netscape Navigator 4.x hatte zwar auch seine Stärken, war aber in Teilen ein Schnellschuss und hat auch deswegen sehr viele Unterversionen erlebt.



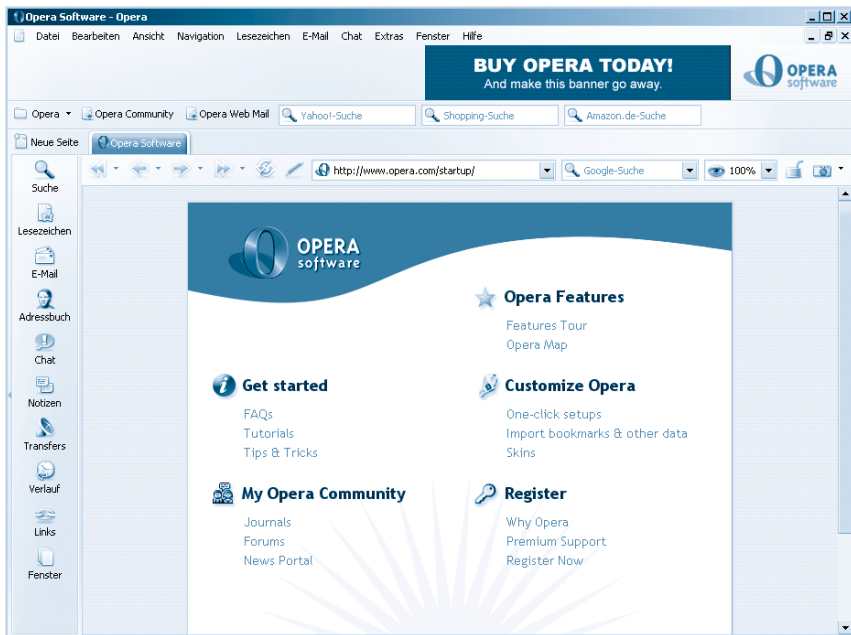
Netscape wurde dann an AOL verkauft. Im Zuge dieser Fusion und der Verbindung mit Time Warner ging Netscape unter. Die lang erwartete neue Version Netscape 6 kam erst nach der Übergabe der Codes an die Mozilla Foundation. Allerdings hatte Netscape auch hier zu schnell geschossen und einen frühen Stand des Mozilla-Codes veröffentlicht. Die Qualität war dementsprechend schlecht und die Darstellungsqualität von Seiten ist beim Netscape Navigator 6 teilweise schlechter als bei Netscape 4.x.



Mittlerweile hat Netscape zwar mit Version 7 eine neue Version auf Basis des Mozilla herausgebracht, allerdings steht diese, sowohl im Hinblick auf Popularität als auch auf Aktualität, im Schatten vom Firefox.

Beim Blick auf die Browser mit den größten Marktanteilen sollten auch nie die kleineren vergessen werden. Schon seit Jahren ist der Opera (<http://www.opera.com/>) der gleichnamigen norwegischen Browserschmiede die Nummer drei auf dem Markt. Ihn gibt es in einer kostenlosen Variante mit Werbung und einer kostenpflichtigen ohne Werbung.

Der Opera ist für alle Betriebssysteme zu haben. Außerdem unterstützt er auch Standards wie XHTML Mobile für mobile Endgeräte. Für normale Webseiten hat der Opera manchmal ein paar Eigenheiten, weswegen Sie auf ihm unbedingt testen sollten.

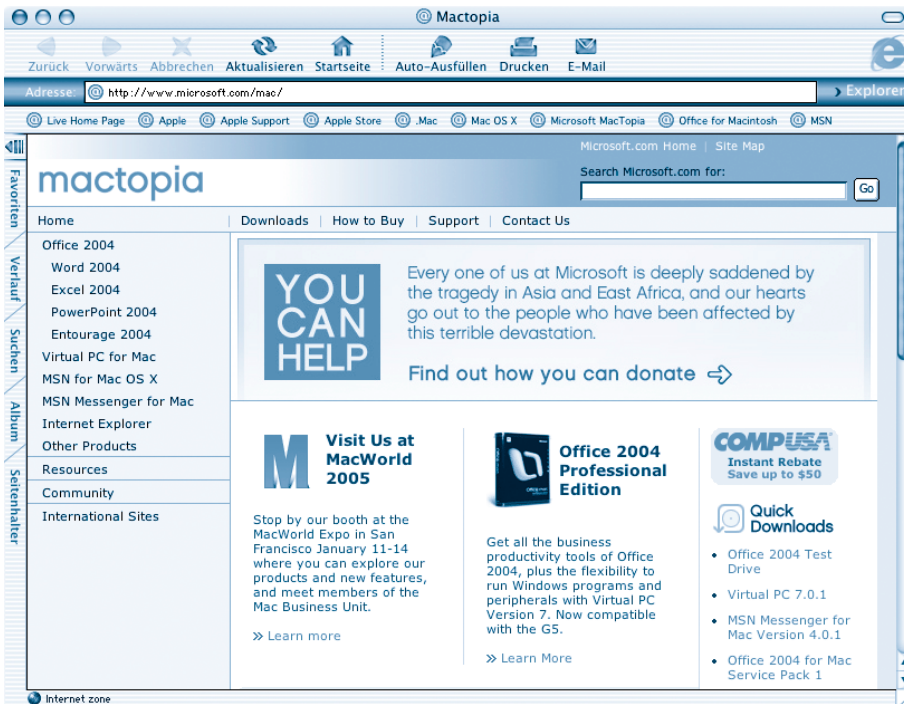


Andere Betriebssysteme sind ein gutes Stichwort. Unter der KDE-Oberfläche des Betriebssystems Linux gibt es den Konqueror als Dateimanager und Browser (<http://www.konqueror.org/>).

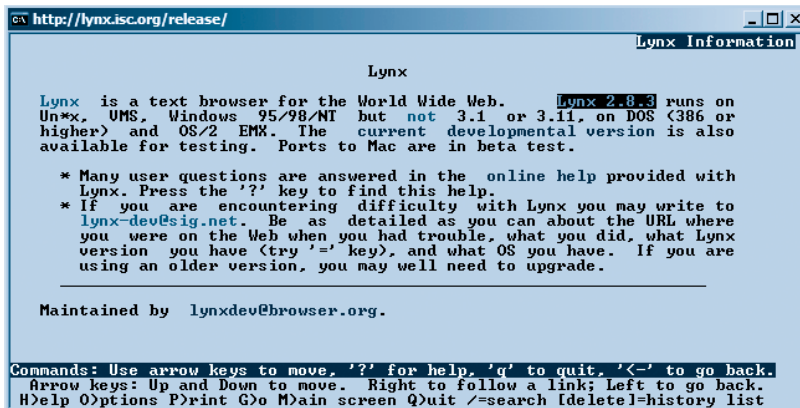
Die Basis des Konquerors ist KHTML. Der Browser allein hätte nicht genug Marktanteil, um ihn zu berücksichtigen. Allerdings wurde die KHTML-Engine auch in den Browser Safari von Apple übernommen. Und dieser hat sich zumindest auf dem Mac sehr schnell einen ordentlichen Marktanteil verschafft. Insgesamt kommen beide zusammen wohl ungefähr auf 1,5 %.



Für den Mac gibt es außerdem noch den Internet Explorer. Allerdings ist diese Version teilweise von einem eigenen Entwicklerteam entwickelt worden. Das heißt, der Internet Explorer 5.x für den Mac ist teilweise standardgetreuer als das Windows-Pendant. Mittlerweile wurde außerdem die Entwicklung eingestellt.



Neben den genannten Browsern gibt es natürlich noch eine ganze Reihe an Exoten. Auf diesen müssen Sie Ihre Webseiten nicht testen, aber Sie sollten zumindest wissen, dass es sie gibt. Reine Texteditoren gehören dazu. Der bekannteste ist sicherlich Lynx. Sich eine Webseite mal in einem Texteditor anzusehen hilft auch, sich einen Eindruck der Struktur zu verschaffen. In diese Kategorie gehören auch Screenreader, die blinden Menschen eine Webseite vorlesen.



Editor – wo schreiben Sie?

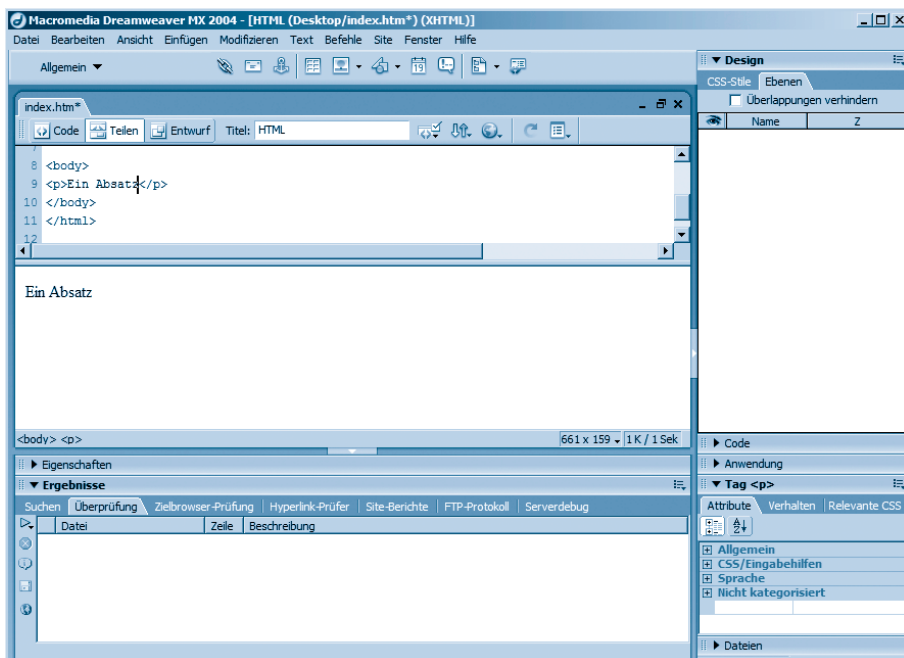
HTML und CSS lassen sich in eine einfache Textdatei mit der Dateierendung *.html* oder *.css* schreiben. Dafür genügt ein normaler Texteditor wie der bei Windows mitgelieferte Editor (*Start/(Alle) Programme/Zubehör/Editor*).



Komfortabler geht es allerdings mit einem besser ausgestatteten Texteditor wie UltraEdit, NoteTab oder Ähnlichen. Spezielle HTML- und CSS-Editoren wie TopStyle bieten dann noch zusätzliche Funktionen wie Code-Vervollständigung, d.h. angefangene Befehle werden vom Editor komplettiert. Und die Luxusvariante ist natürlich ein echter Webeditor wie Macromedia Dreamweaver, Adobe GoLive oder Microsoft FrontPage. Diese letzteren Editoren sind alle WYSIWYG-Lösungen. WYSIWYG steht für »What you see is what you get«. Das bedeutet, dass Sie Elemente direkt auf die Seite ziehen und Seiten ähnlich gestalten können wie beispielsweise in Word. Der zugrunde liegende HTML-Quellcode wird dabei für Sie erzeugt. Allerdings haben Sie bei diesen Editoren auch die Möglichkeit, direkt den HTML-Code zu bearbeiten.

Hinweis

Um von einem Editor unabhängig zu bleiben, verwenden wir in diesem Buch puren Quellcode. Welchen Editor Sie einsetzen, bleibt Ihnen überlassen. Wir raten Ihnen aber auf jeden Fall, sich nicht »nur« auf den Editor zu verlassen, sondern HTML und CSS selbst zu lernen.

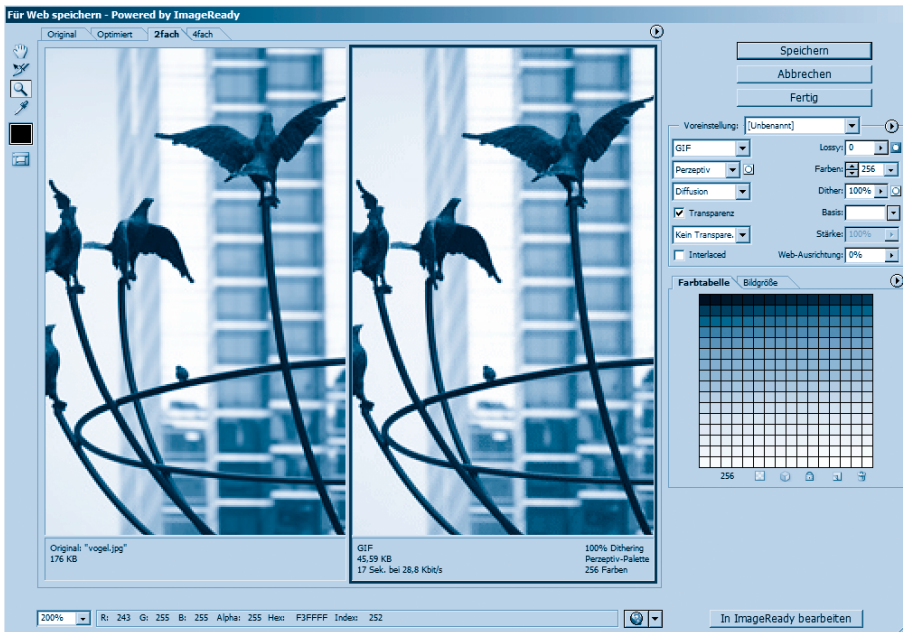


Was Sie sonst noch brauchen ...

Sobald Sie Ihre Webseite fertig haben, wollen Sie diese natürlich auch online stellen. Dafür benötigen Sie Webspace oder einen eigenen Webserver. Wenn Sie sich kein eigenes (Mini-)Rechenzentrum leisten möchten, brauchen Sie dafür einen **Hoster**. Dieser stellt Ihnen dann alle erforderlichen Daten zur Verfügung. Um die Seite online zu stellen, benötigen Sie dann noch einen FTP-Zugriff. Mehr dazu erfahren Sie in *Kapitel 11*.

Für das Webdesign ist außerdem ein **Bildbearbeitungsprogramm** sinnvoll. Damit können Sie Ihre Fotos und Bilder webgerecht aufbereiten. Welche Bildformate im Web benutzt werden, erfahren Sie in *Kapitel 4*.



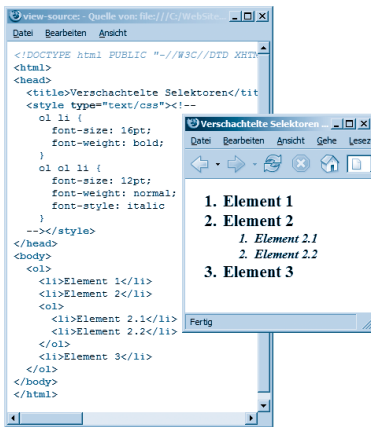


Nutzloses Wissen

Jeder Mensch verbringt eine halbe Stunde seines Lebens als Einzeller.

Kapitel 1

Die Grundlagen – HTML und CSS



Ein klassischer Webeditor nimmt Ihnen beim Erstellen von Webseiten viel Arbeit ab. Sie müssen weite Teile Ihres HTML- oder CSS-Codes nicht mehr selbst tippen, sondern klicken auf mehr oder weniger verständliche Symbole, die oftmals an die Bedienung von Microsoft Word oder anderen Office-Programmen angelehnt sind. Allerdings hat diese Sache einen Haken: Sobald ein Problem auftaucht, z.B. eine Textformatierung nicht mehr stimmt oder eine Tabelle heillos verschachtelt ist, benötigen Sie den Quellcode und müssen ihn verstehen. Deswegen bieten viele Editoren zwar Unterstützung, aber eben auch eine Quellcode-Ansicht.

In diesem Buch wollen wir Sie in die Lage versetzen, Ihre erste »richtige« Webseite zu erstellen. Deswegen erfahren Sie hier die Hintergründe zu den beteiligten Sprachen und Technologien. Sie können das Gelernte in jedem beliebigen Editor einsetzen.

Das können Sie schon:

Der Aufbau des Web	12
Die Wichtigkeit von Browsern	14
Mit welchen Editoren Sie arbeiten können	21



Das lernen Sie neu:

HTML-Grundgerüst einer Website	26
Grundlagen von XHTML	33
Dokumenttypen	36
Einsatz von CSS	40

HTML

Die HyperText Markup Language oder kurz HTML geht auf den Erfinder des Web, Tim Berners-Lee, zurück. Es handelt sich bei HTML um eine einfache Beschreibungssprache, die definiert, welche Elemente eine Webseite enthält und wie diese aussehen. Heute wird HTML vom W3C, dem World Wide Web Consortium (W3C), als Standard betreut. Die Browserhersteller implementieren diesen Standard und machen HTML-Seiten damit darstellbar.

Tipp

Jede Webseite ist ein HTML-Dokument. Das HTML-Dokument wird – wie im Client-Server-Modell aus dem vorangegangenen Kapitel zu sehen – vom Server an den Client geliefert. Unabhängig davon, welche Technologien Sie noch verwenden – CSS, JavaScript, Flash –, HTML bildet immer die Grundlage. Auch wenn Sie mit PHP serverseitig programmieren, liefert das PHP-Skript am Schluss HTML. Deswegen ist es absolut notwendig, sich mit HTML ausführlicher zu beschäftigen!

Eine HTML-Seite ist im Prinzip nur eine Textdatei mit der Dateieindung `.htm` oder `.html`. Vorsicht, bei der Groß- Kleinschreibung wird bei HTML-Dateien unterschieden. Tendenziell sollten Sie alle kleinschreiben und für den Dateinamen auch nicht mehr als acht Zeichen verwenden, damit die Datei auf allen Systemen korrekt verwendet wird. Wenn Sie keinen Webeditor zur Hand haben, können Sie sie mit einem beliebigen Texteditor schreiben.

Hinweis

*Unter Windows reicht schon der eingebaute Editor (**Start/(Alle) Programme/Zubehör/Editor**), unter Linux haben Sie je nach Distribution eine reiche Auswahl, am Mac können Sie auch den vom System mitgelieferten Editor einsetzen.*

HTML besteht aus Elementen; diese heißen auch Tags (engl. für Befehle). Ein Tag erkennen Sie immer an den spitzen Klammern. Ein Tag wird geöffnet und dann mit einem Schrägstrich (/) vor dem Tag-Namen wieder geschlossen. Dieses Beispiel zeigt das `<p>`-Tag, das einen Absatz markiert:

```
<p>Ein Absatz</p>
```



Der Text des Absatzes steht zwischen dem öffnenden und dem schließenden Tag.

Erhält ein Browser eine HTML-Seite – z.B. weil Sie einen URL eintippen oder eine HTML-Seite auf Ihrem lokalen System doppelt anklicken –, geht er die Seite durch und interpretiert die Tags. Sieht er z.B. ein `<p>`-Tag, weiß er, dass es sich um einen Absatz handelt, und stellt den Text darin dar. Dieser Vorgang heißt auch Rendern.

Grundgerüst

Eine HTML-Seite ist immer gleich aufgebaut. Hier starten wir mit dem grundlegenden Aufbau:

1 Öffnen Sie zuerst eine neue Textdatei im Editor.

2 Beginnen Sie mit dem grundlegenden Tag `<html>`, das zu erkennen gibt, dass es sich um eine HTML-Seite handelt.

```
<html>

</html>
```

3 Erstellen Sie dann das `<head>`-Tag für den Kopfbereich der Webseite.

```
<html>
  <head>
    <title>HTML-Seite</title>
  </head>

</html>
```

Es enthält Informationen, die nicht direkt im Browserfenster angezeigt werden. Eine der wichtigsten Informationen ist das `<title>`-Tag mit dem Titel der Webseite. Dieser wird in der Titelleiste des Browsers angezeigt.

4 Erstellen Sie dann den `<body>` für den Körper, sprich den Inhalt der Webseite:

```
<html>
<head>
  <title>HTML-Seite</title>
</head>
<body>
  <p>Inhalt der Seite</p>
</body>
</html>
```

Er enthält alles, was im Browserfenster sichtbar ist. Hier ist z.B. ein `<p>`-Tag für einen Absatz eingefügt.

5 Speichern Sie die Datei mit der Dateiendung `.htm` oder `.html`.

Vorsicht, ein Texteditor legt oft standardmäßig eine Datei mit der Endung `.txt` an.

6 Testen Sie das Dokument im Browser.

Ein Doppelklick öffnet es auf den meisten Systemen im Standardbrowser. In andere Browser können Sie es hereinziehen oder mit *Datei/Öffnen* anzeigen.

Hinweis

Wenn Sie eine Änderung in Ihrem Dokument vornehmen, müssen Sie die Datei speichern und das Dokument im Browser neu laden (aktualisieren). Nur dann sehen Sie auch die Änderung.

Hier das komplette Grundgerüst:

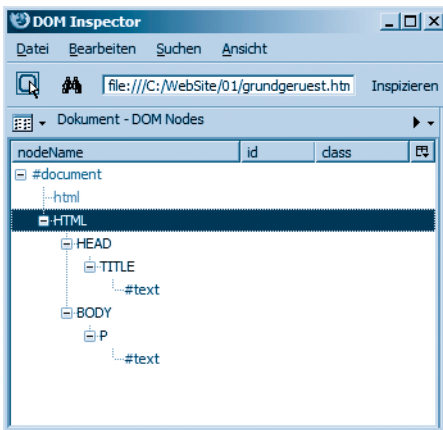
```
<html>
<head>
  <title>HTML-Seite</title>
</head>
<body>
  <p>Inhalt der Seite</p>
</body>
</html>
```

Tipp

Die Einrückungen und Umbrüche sind nur für den Quelltext relevant, nicht aber für die Darstellung im Browser. Sie sollten allerdings trotzdem immer sauber einrücken, da Sie so am besten die Übersicht bewahren. Da Zeilenumbrüche im Quelltext nicht im Browserfenster angezeigt werden, müssen Sie Zeilenumbrüche mit einem eigenen HTML-Tag namens `<br />` ergänzen.



Eingeschlossen wird die HTML-Seite vom `<html>`-Tag. Dieses Tag heißt auch Wurzelement, da es der Ausgangspunkt für alle untergeordneten Tags ist. Sie können sich das so vorstellen, dass eine HTML-Seite immer eine baumartige Struktur ergibt. Der Browser Firefox zeigt diese Struktur im so genannten DOM Inspektor (DOM steht für Document Object Model). Sie finden ihn unter *Extras/DOM Inspektor*, wenn er mit installiert wurde. Aber auch viele HTML-Editoren besitzen eine Ansicht der Hierarchie eines HTML-Dokuments.

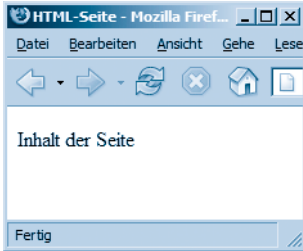


Innerhalb des `<html>`-Tags teilt sich die Webseite immer in zwei Bereiche: den `<head>` oder auch Kopf der HTML-Seite und den `<body>` respektive Körper. Der `<head>` enthält allgemeine Informationen z.B. für Suchmaschinen. Einzig sichtbares Element ist das `<title>`-Tag mit dem Titel der Webseite. Der Inhalt dieses Tags wird vom Browser in der Titelleiste (die obere Leiste) angezeigt.

Hinweis

Das `<title>`-Tag kann keine anderen Tags aufnehmen, sondern nur den Titel. Auf die Formatierung des Titels haben Sie keinen Einfluss. Manche Webseiten spielen deshalb mit Worten nur in Großbuchstaben, Punkten, Bindestrichen oder Ähnlichem herum. Das ist aber meist für den Nutzer eher verwirrend.

Aussehen und Inhalt der Webseite selbst werden im `<body>`-Bereich der HTML-Seite definiert. Im oben gezeigten Beispiel besteht der Inhalt nur aus einem Absatz.



Hinweis

Brower sind sehr tolerant. Sie können beispielsweise auch das `<head>`-Tag weglassen, ohne eine Fehlermeldung zu provozieren. Allerdings ist das nicht empfehlenswert, da sonst wichtige Informationen wie der Titel der Seite fehlen.

Tags und Attribute

Die HTML-Befehle heißen, wie schon erwähnt, Tags. Sie können nun viele Aufgaben in HTML schon mit Tags übernehmen. Allerdings erfordern viele Tags noch weitere Einstellungen. Dafür sind Attribute verantwortlich. Sie folgen immer im öffnenden Tag nach dem Tagnamen und werden mit Leerzeichen voneinander getrennt.

```
<a href="http://www.mut.de/">Markt + Technik</a>
```

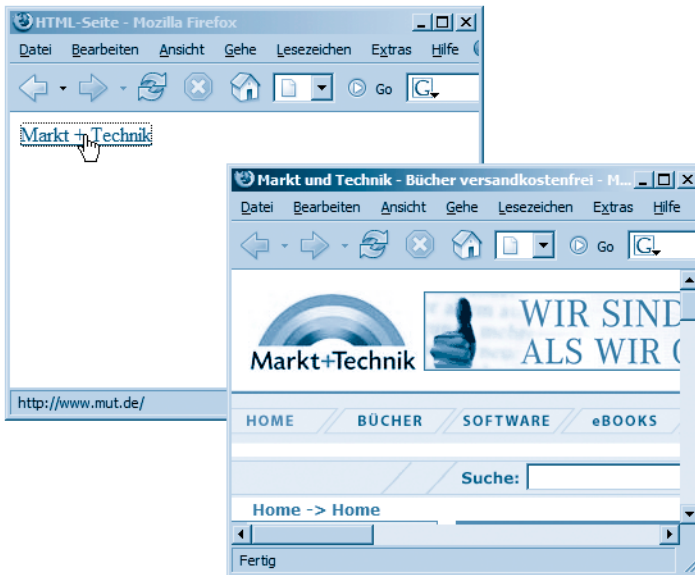
Hier als Beispiel ein Link (`<a>`-Tag): Das Attribut `href` legt das Linkziel fest, das heißt, wohin der Nutzer gelangt, wenn er auf den Link klickt. Der Wert – hier die Adresse (URL) – ist `http://www.mut.de/`, die Website von Markt+Technik. Der Wert wird einem Attribut immer mit dem Ist-Gleich (=) zugewiesen und steht selbst in Anführungszeichen.

Mehrere Attribute stehen mit Leerzeichen getrennt hintereinander:

```
<a href="http://www.mut.de/" target="_blank">Markt + Technik  
</a>
```

Hier gibt das Attribut `target` an, in welchem Fenster der Link geöffnet wird. `_blank` bedeutet, dass ein neues Browserfenster verwendet wird. Mehr zu Links und den Möglichkeiten erfahren Sie in *Kapitel 2*.





Tags können selbst Inhalt in Form von Text oder anderen Tags enthalten. Das Tag wird am Ende dieses Inhalts wieder geschlossen. Manche Tags besitzen aber selbst keinen Inhalt, sondern eventuell nur Attribute. Ein Beispiel ist das Tag `<hr>`, das dazu dient, eine horizontale Linie in eine Seite einzubinden. Wenn ein Tag keinen Inhalt besitzt, müssen Sie es nicht umständlich mit einem eigenen End-Tag schließen:

```
<hr></hr>
```

Stattdessen können Sie die Kurzform wählen und nach einem Leerzeichen einen Schrägstrich an das Ende des Tags stellen:

```
<hr />
```

Hinweis

Browser und HTML-Spezifikation sind sogar so tolerant, dass Sie den Schrägstrich am Ende weglassen können. Da der neuere Standard XHTML (siehe Abschnitt »XHTML«) dieses Ende aber erfordert, sollten Sie es sich angewöhnen.

Tag ist nicht gleich Tag – eine Weisheit, die für die Tage des Lebens sicherlich gilt, für HTML aber nicht sofort verständlich wird. HTML unterscheidet zwischen Blockelementen und Inline-Elementen. Ein Blockelement definiert einen abgegrenzten Bereich, beispielsweise einen Absatz. Ein Inline-Element ist dagegen nur Teil eines Blockelements und definiert selbst keinen Bereich. Ein Beispiel hierfür ist ein Link; ein weiteres Beispiel ist ein Bild.

Inline-Elemente sind immer innerhalb von Blockelementen zu finden. Das heißt, ein Link befindet sich beispielsweise in einem Absatz. Hier ein Beispiel, das Sie direkt ausprobieren können, wenn Sie eine Internetverbindung haben:

```
<p>Der Verlag: <a href="http://www.mut.de/">Markt + Technik  
  </a></p>
```

Eine Sonderrolle nehmen Kommentare ein. Ein Kommentar ist ein Text oder Bereich, den der Browser ignoriert. Das heißt, Sie können dort Dinge hineinschreiben, die ausschließlich im Quellcode zu sehen sind. Ein HTML-Kommentar beginnt mit `<!--` und endet mit `-->`. Testen Sie dies und sehen Sie sich an, dass der Inhalt des Kommentars vom Browser nicht ausgegeben wird:

```
<!-- Dies ist ein Kommentar, der vom Browser ignoriert wird  
  -->
```

Kommentare helfen Ihnen, Bereiche in komplexeren Webseiten mit Erklärungen zu versehen. So finden Sie sich später schneller in den Quellcode hinein. In diesem Buch verwenden wir Kommentare dazu, die Dateinamen der Listings mit anzugeben.

Allerdings sollten Sie nie vergessen, dass Kommentare im Quellcode an den Nutzer mitgeschickt werden. Er kann die Kommentare also sehen, wenn er sich im Browser mit dem jeweiligen Befehl, z.B. *Ansicht/Seitenquelltext anzeigen* (Firefox), *Ansicht/Quelltext* (Internet Explorer), den Quellcode auf den Bildschirm holt.



```
view-source: - Quelle von: file:///C:/WebSite/01/kommentare.html - Mozilla Firefox  
Datei Bearbeiten Ansicht  
<head>  
  <title>HTML-Seite</title>  
</head>  
<body>  
  <p>Der Verlag: <a href="http://www.mut.de/" target="_blank">Markt + Technik</a></p>  
  <!-- Dies ist ein Kommentar, der vom Browser ignoriert wird -->  
</body>  
</html>
```



Hinweis

Webeditoren, Content Management Systeme und viele andere Webtechnologien verwenden Kommentare, um bestimmte Stellen zu kennzeichnen. Das heißt also, dass Kommentare für ein solches System durchaus einmal eine Funktion haben können. Hier gilt also erhöhte Vorsicht, wenn Sie aus einer bestehenden Seite Kommentare löschen, die »verdächtig« funktional aussehen.

XHTML

XHTML gehört zu den Schlagworten, die in den letzten Jahren durch die Webszene geistern. Dabei steckt hinter dem Begriff keine spektakuläre Innovation, sondern nur die Bestrebung, Standards zu vereinheitlichen. Der Hintergrund des Ganzen ist XML (eXtensible Markup Language). Diese Metasprache erlaubt, beliebige Elemente zu definieren und daraus andere Sprachen zu machen. Das wurde in der Praxis auch auf vielfältige Weise genutzt und hat vor allem im Bereich Datenspeicherung und -austausch zur Revolution geführt. Natürlich hat das technisch keine direkten Auswirkungen auf »einfache« Webseiten. Aber um HTML als Untermenge des großen XML-Universums zu etablieren, musste man es in strengere Regeln packen. Diese XML-Regeln machen aus HTML dann XHTML.

Tipp

Sie müssen nicht sofort mit XHTML einsteigen. Die hier vorgestellten Regeln erfordern schon ein wenig Erfahrung mit HTML. Deswegen können Sie auch zu jeder Zeit wieder zurück an diese Stelle blättern und die Regeln nachlesen.

Hier ein Überblick über die wichtigsten Regeln:

- **Tags**, die Inhalt enthalten, müssen immer **geschlossen** werden. Ein Beispiel: Browser kommen mit den folgenden Absätzen zurecht:

```
<p>Absatz 1
<p>Absatz 2
```

XHTML-konform ist dies aber nur, wenn die einzelnen <p>-Tags auch beendet werden:

```
<p>Absatz 1</p>
<p>Absatz 2</p>
```

- Auch **allein stehende Tags** müssen **geschlossen** werden. Beispiele für allein stehende Tags sind `<br>` für einen Zeilenumbruch, `<hr>` für eine horizontale Linie und `<img>` für ein Bild. Nicht erlaubt ist demzufolge:

`<br>`

Korrekt, aber umständlich wäre hingegen:

`<br></br>`

Einfacher geht es mit:

`<br />`

- XML unterscheidet zwischen Groß- und Kleinschreibung. Dementsprechend hat das W3C für XHTML festgelegt, dass Tags und Attribute nur noch in **Kleinschreibung** geschrieben werden dürfen:

`<P>Absatz</P>`

ist also ebenso verboten wie:

`<p>Absatz</P>`

Erlaubt ist nur:

`<p>Absatz</p>`

- **Attributwerte** müssen in **Anführungszeichen** stehen.

`<p align=center>Absatz</a>`

ist also nicht erlaubt. Richtig ist:

`<p align="center">Absatz</a>`

`align="center"` zentriert übrigens den Absatz im Browserfenster.

- **Attribute** müssen einen **Wert besitzen**. Dies gilt vor allem für einige Attribute, die an sich nur einen Wert haben. Statt:

`<select multiple>...`

müssen Sie also Folgendes schreiben:

`<select multiple="multiple">...`

Hierbei handelt es sich um einen Ausschnitt aus einem HTML-Formular. Mehr dazu in *Kapitel 6*.

- Sie müssen immer vor dem öffnenden `<html>`-Tag einen Dokumenttyp (**DocType**) **angeben**. Dies ist notwendig, damit der Browser weiß, welche Version und Variante von XHTML Sie verwenden. Mehr dazu im nächsten Abschnitt.

`<!DOCTYPE html`

`PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"`

`"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`



Diese Regeln sollten Sie einhalten, da es im Prinzip auch keine weiteren Probleme bereitet. Und für die Zukunft bleiben Sie flexibel. Allerdings ist es nicht unbedingt notwendig, allen Regeln zu folgen, die das World Wide Web Consortium (W3C) vorgibt. Wenn Sie hingegen standardkonforme Webseiten erstellen möchten, sollten Sie einen Blick auf den W3C-Validator unter <http://validator.w3.org/> werfen. Dort können Sie einzelne Seiten validieren lassen. Validieren heißt, dass sie auf Standardkonformität überprüft werden. Das Ergebnis kann zwar Aufschlüsse darüber geben, woran Sie arbeiten müssen, gerade bei kleineren Websites spielt es allerdings keine allzu große Rolle. Praktisch ist die Validierung allerdings zum Finden kleinerer Fehler. Im folgenden Bild sehen Sie das Ergebnis einer Validierung.

The screenshot shows the W3C Markup Validation Service interface in a Mozilla Firefox browser window. The address bar shows <http://validator.w3.org/check>. The page header includes the W3C logo, "QUALITY Assurance", and "MARKUP VALIDATION SERVICE v0.6.7". Navigation links include HOME, ABOUT..., NEWS, DOCS, HELP & FAQ, FEEDBACK, and LINK CHECKER. A "Jump To:" section has a "Results" link.

The main content area displays the message: **THIS PAGE IS VALID XHTML 1.0 TRANSITIONAL!**

Below this, a "TIP OF THE DAY:" section suggests: **Use the alt attribute**.

A "NOTE:" section states: "The Validator XML support has [some limitations](#)."

The main text explains: "The uploaded file was checked and found to be valid [XHTML 1.0 Transitional](#). This means that the resource in question identified itself as "XHTML 1.0 Transitional" and that we successfully performed a formal validation using an SGML or XML Parser (depending on the markup language used)."

A W3C XHTML 1.0 icon is shown with the text: "To show your readers that you have taken the care to create an interoperable Web page, you may display this icon on any page that validates. Here is the HTML you should use to add this icon to your Web page:"

```
<p>
  <a href="http://validator.w3.org/check?uri=referer"></a>
</p>
```

Below the code, it says: "Thanks to this code, you will be able to re-validate your Web page by following the link (click on the image), and we encourage you to do so every time you modify your document."

The footer includes "Feedback: [The W3C Validator Team](#)" and "Date: 2004/07/21 10:24:06". A W3C XHTML 1.0 icon is also present in the footer. The status bar at the bottom says "Fertig".

Dokumenttypdefinition (DocType)

Wie bereits erwähnt, erfordert XHTML die Angabe des DocType. Dieser Header, der vor dem eigentlichen HTML-Grundgerüst eingefügt wird, legt fest, welche Version von HTML oder XHTML verwendet wird.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
    <title>HTML-Seite</title>
</head>
<body>
    <p>Inhalt der Seite</p>
</body>
</html>
```

Je nach Version sind mehr oder weniger Tags und Attribute erlaubt. Vor allem die strikte Variante (Strict) untersagt alte Tags, die in der Spezifikation schon als deprecated (nicht mehr empfohlen) gekennzeichnet sind.

In diesem Buch kommt, wie bereits erwähnt, XHTML zum Einsatz. Für XHTML sind die folgenden DocTypes verfügbar:

- Transitional ist die gnädigere Variante, die auch in diesem Buch verwendet wird. Bei HTML hieß diese Variante noch Loose. Sie erlaubt auch noch viele als deprecated gekennzeichnete Tags.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

- Strict ist die strengere Alternative. XHTML 1.0 Strict ist in der Praxis das Strengste, an das man sich halten sollte. Die neuere Version XHTML 1.1 geht mit seiner Modul-Sicht in eine andere Richtung und ist noch nicht ausreichend in den Browsern angekommen. Auch das W3C selbst orientiert seine Seiten aktuell an XHTML 1.0 Strict.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- Frameset ist wie bei HTML von Transitional abgeleitet. Mehr zu Frames in *Kapitel 5*.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```



Hinweis

Es gibt für den DocType auch noch Abkürzungen. z.B. kann der URL zur Dokumenttypangabe – oben beispielsweise "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd" – in verkürzter Form angegeben werden. Welche Variante Sie verwenden, ist nicht von praktischem Belang.

Ein Trugschluss ist, dass die Angabe des DocType nur beim Validieren Auswirkungen hätte. Sie hat vielmehr auch für das Rendering vor allem von CSS Bedeutung. Ist kein DocType oder ein HTML-DocType angegeben, arbeiten moderne Browser im QuirksMode, bei dem teilweise unsauber respektive unterschiedlich gerendert wird. Der saubere Modus wird mit einer der XHTML-DocTypes mit Angabe des URLs erreicht.

Zeichensatz und Sonderzeichen

Deutschsprachige Webseiten kämpfen mit einem unangenehmen Problem: deutschen Umlauten. Deutschsprachige Browser erkennen sie im Allgemeinen problemlos, englischsprachige dagegen nicht unbedingt. Ähnlich problematisch sind andere Sonderzeichen wie z.B. das Eurosymbol.

Um dieses Problem zu lösen, gibt es zwei Möglichkeiten:

- Sie verwenden so genannte Entitäten. Das sind feststehende Zeichenfolgen, die das gewünschte Zeichen repräsentieren. Eine Entität beginnt immer mit kaufmännischem Und (&) und endet mit einem Strichpunkt, z.B.

`ä`

steht für das kleine ä. (auml für a-Umlaut). Probieren Sie es aus, ergänzen Sie in Ihrer ersten Webseite einmal innerhalb von `<p>` und `</p>` die Entität `ä`. Wenn Sie alles richtig geschrieben haben, sehen Sie im Browser stattdessen nur ein ä.

Hinweis

Manche Webeditoren wandeln Umlaute automatisch in Entitäten um. Hier können Sie sich also Arbeit abnehmen lassen.

- Sie geben einen Zeichensatz an, in dem Ihre Webseite verfasst ist. Zeichen, die dieser Zeichensatz kennt, werden dann vom Browser korrekt dargestellt. Es gibt zwei Orte, an denen Sie den Zeichensatz einsetzen können: in der so genannten Processing-Instruction oberhalb des Doc-Type, die wir bisher noch nicht erwähnt hatten:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Diese hat allerdings den großen Nachteil, dass sie die DocType-Erkennung im Internet Explorer aushebelt. Deswegen kommt sie in der Praxis nicht zum Einsatz. Stattdessen verwenden Sie am besten ein so genanntes `<meta>`-Tag innerhalb des `<head>`-Elements der Seite zur Angabe des Zeichensatzes:

```
<meta http-equiv="Content-type" content="text/html;  
charset=iso-8859-1" />
```

Die Frage ist nun, welche Variante Sie wählen sollten. Der Vorteil beim Zeichensatz ist, dass Sie die Angabe einmal vornehmen und sich danach nicht mehr darum kümmern müssen. Bleibt noch die Wahl des Zeichensatzes. Hier hat sich als Standard in deutschen Landen ISO-8859-1 und mit Eurozeichen ISO-8859-15 durchgesetzt. Beide Zeichensätze decken 256 Zeichen ab, darunter die deutschen Umlaute. Die Alternative ist der Einsatz von UTF-8. Dieser Zeichensatz erlaubt wesentlich mehr Zeichen, verwendet aber eine besondere Kodierung für diese Zeichen. Der Haken hierbei ist, dass der Editor, den Sie verwenden, mit UTF-8 zurechtkommen sollte. Ein auf ASCII-basierender Texteditor kann oft kein UTF-8-Dokument speichern.

Unsere Empfehlung für den Hausgebrauch ist dementsprechend ISO-8859-1 oder ISO-8859-15. Bei Letzterem haben Sie zwar das Eurozeichen, dafür fehlt aber z.B. der Gedankenstrich. Für alle Sonderzeichen, die im gewählten Zeichensatz nicht enthalten sind, müssen Sie so oder so eine Entität einsetzen. Dazu zählen z.B. die spitzen Klammern, die von HTML reserviert sind. Die folgende Tabelle 1.1 gibt einen Überblick über die Entitäten, die Sie brauchen können.



Entität	Zeichen	Beschreibung
Ä	Ä	
ä	ä	
Ö	Ö	
ö	ö	
Ü	Ü	
ü	ü	
ß	ß	scharfes s
©	©	Copyright-Zeichen
"	"	gerade Anführungsstriche
€	–	Euro-Währungssymbol (nicht in sehr alten Browsern)
<	<	Kleiner als (spitze Klammer auf) ¹
>	>	Größer als (spitze Klammer zu)
&	&	kaufmännisches UND
 		geschütztes Leerzeichen

Tabelle 1.1: Einige Sonderzeichen

Hinweis

Auf der Support-Website zum Buch finden Sie eine umfangreiche Liste mit Sonderzeichen zum Download. Sie ist über **<http://www.hauser-wenz.de/support/>** zu erreichen. Dieses Buch setzt Sonderzeichen in Entitäten um, damit Sie den Code immer und überall problemlos verwenden können. Die Zeichensatz-Angabe sparen wir uns aus Platzgründen, würden Sie Ihnen allerdings empfehlen.

1. Größer als und Kleiner als wären an sich keine Sonderzeichen, da sich ihr Code innerhalb der 128 ASCII-Zeichen befindet. Sie sind allerdings in HTML spezifisch für Tags eingesetzt und werden deswegen extra als Sonderzeichen benötigt.

CSS

HTML ist die Basis für Webseiten; allerdings war das ursprünglich schon Anfang der 90er Jahre entstandene HTML nicht mehr allen Anforderungen gewachsen. HTML bietet zwar Formatierungsmöglichkeiten für Text und Ähnliches, allerdings lassen sich diese nur spezifisch für jedes Element vornehmen. Es gab kaum Möglichkeiten, den Inhalt vom Layout zu trennen und Formatierungen zentral vorzunehmen. Der zweite Nachteil von HTML wirkt genauso schwer: Die Formatierungsmöglichkeiten sind ausgesprochen begrenzt. Beispielsweise gibt es in HTML Schriftgrößen nur in sieben festgelegten Schritten. Einem Designer, der im Printbereich punktgenaue Angaben macht, ist das zu wenig.

Die Lösung für diese Probleme heißt Cascading Style Sheets oder kurz CSS. Diese Technologie ist wie HTML vom W3C standardisiert und besteht aus vielen Befehlen. Die Syntax ist immer gleich: Sie schreiben zuerst den Befehl und dann nach einem Doppelpunkt den Wert. Das Ganze schließen Sie mit einem Strichpunkt ab:

Befehl: Wert;

Hinweis

Bei nur einem Befehl oder beim letzten Befehl ist der Strichpunkt nicht erforderlich, kann aber gesetzt werden.

Ein Beispiel für einen CSS-Befehl ist z.B. `font-size` für die Schriftgröße. Dort können Sie nicht nur relative Angaben vergeben, sondern auch absolute Schriftgrößen mit verschiedenen Maßeinheiten wie Punkt (pt) oder Zentimeter (cm).

`font-size: 12pt;`

Hinweis

Bei der Schriftgröße ist sich die Webdesigner-Riege uneins, welche Maßeinheit zu bevorzugen wäre. Pixel und Punkt haben Nachteile, da der Internet Explorer dann nicht mehr auf Nutzereingabe die Schriftgröße verändert, Punkt ist bildschirm- respektive auflösungsabhängig. Und relative Schriftgrößen wie `em` sind aus Designersicht wenig genau. In diesem Buch verwenden wir `pt`.



Nun ist nur noch die Frage, wie CSS in HTML eingebunden wird? Dafür gibt es drei Möglichkeiten:

- Als Inline-Stil in einem HTML-Tag.
- Im Kopf der HTML-Seite. Die CSS-Befehle gelten dann für ein oder mehrere Tags.
- In einer externen Datei. So ist zentrales Layout für mehrere HTML-Seiten möglich.

Alle drei Optionen finden Sie in den nächsten Abschnitten erläutert. Wenn Sie ein CSS im Kopf der HTML-Seite oder in einer externen Datei platzieren, müssen Sie außerdem noch die Befehle bestimmten Tags zuweisen. Dafür sind die so genannten Selektoren verantwortlich, denen wir ebenfalls einen eigenen Abschnitt widmen.

Inline-Stile

Ein Inline-Stil ist die einfachste Form, einen oder mehrere CSS-Befehle zuzuweisen. Sie fügen diese einfach im `style`-Attribut für das Tag ein, das Sie formatieren möchten. Hier ein Beispiel – Ausgangspunkt ist ein HTML-Grundgerüst:

1 Erstellen Sie einen Absatz in Ihrer Seite.

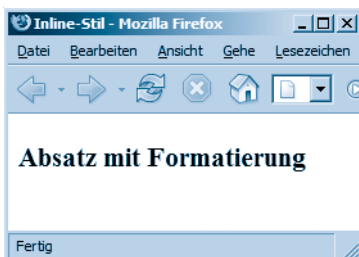
```
<p>Absatz mit Formatierung</p>
```

2 Fügen Sie dann im öffnenden `<p>`-Tag das Attribut `style` hinzu.

3 Vergeben Sie als Wert für das Attribut die gewünschten CSS-Befehle durch Strichpunkte getrennt:

```
<p style="font-size: 10pt; font-weight: bold;">
  Absatz mit Formatierung</p>
```

Den Befehl `font-weight` kennen Sie noch nicht. Er bestimmt die Dicke der Schrift. Der Wert `bold` besagt, dass die Schrift im Fettdruck dargestellt wird.



Tipp

Inline-Stile kommen immer dann zum Einsatz, wenn Sie die Formatierung wirklich nur lokal auf das eine HTML-Element begrenzen möchten. Soll eine Formatierung für mehrere Elemente gelten, sind CSS-Befehle im Kopf der HTML-Seite oder in einer externen Datei vorzuziehen.

Im Kopf der Seite

CSS-Befehle im Kopf der Seite sind sehr praktisch: Sie legen die Befehle einmal im `<head>`-Abschnitt der Seite an und weisen sie dann mit Selektoren den Tags zu.

Auch hier ein Beispiel – Ausgangspunkt ist wieder ein HTML-Grundgerüst.

- 1 Fügen Sie zuerst einen Block mit `<style>`- und `</style>`-Tags innerhalb der `<head>`-Tags ein. Also so:

```
<head>
  <title>CSS im Kopf der Seite</title>
  <style>    </style>
```

- 2 Schreiben Sie in das öffnende `<style>`-Tag, dass es sich um CSS handelt:

```
<style type="text/css">
```

- 3 Fügen Sie HTML-Kommentare innerhalb des `<style>`-Blocks ein.

```
<style type="text/css"><!--
--></style>
```

Diese Maßnahme verhindert bei Browsern, die mit CSS nicht klarkommen, dass die CSS-Befehle einfach ausgegeben oder interpretiert werden.

- 4 Schreiben Sie anschließend den Namen des Tags, für das Sie einen Stil anlegen möchten, z.B.:

```
<style type="text/css"><!--
  p {
  }
--></style>
```

Dies ist bereits ein Selektor, und zwar ein Selektor für alle `<p>`-Tags, d.h. alle Absätze.



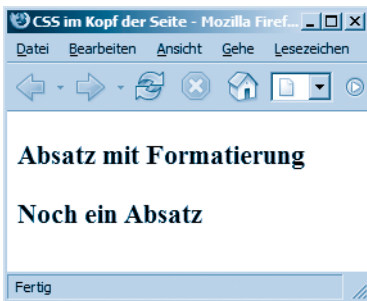
5 Danach fügen Sie in geschweiften Klammern alle CSS-Befehle für diesen Selektor ein.

```
<style type="text/css"><!--
  p {
    font-size: 16pt;
    font-weight: bold;
  }
--></style>
```

6 Zum Schluss fügen Sie im Körper der HTML-Seite noch einen oder mehrere Absätze ohne eigene Formatierungen ein.

Hier der komplette Code:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- css_kopf.html -->
<html>
<head>
  <title>CSS im Kopf der Seite</title>
  <style type="text/css"><!--
    p {
      font-size: 16pt;
      font-weight: bold;
    }
  --></style>
</head>
<body>
  <p>Absatz mit Formatierung</p>
  <p>Noch ein Absatz</p>
</body>
</html>
```



Externe Datei

CSS-Befehle in einer externen Datei bedeuten das Maximum an Zentralisierung. Die externe CSS-Datei kann von beliebig vielen HTML-Dateien verwendet werden. Es steht Ihnen außerdem völlig frei, die CSS-Datei in ein eigenes Verzeichnis zu packen oder sie im gleichen Verzeichnis wie Ihre HTML-Dateien zu belassen. Für die vorliegenden Beispiele legen Sie die CSS-Datei in den gleichen Ordner wie die HTML-Datei.

Eine CSS-Datei hat im Allgemeinen die Dateierendung `.css`. Absolut notwendig ist das zwar nicht, empfehlenswert aber auf jeden Fall. Eine `.css`-Datei ist per se eine einfache Textdatei. Sie enthält ausschließlich die CSS-Selektoren mit den zugehörigen Befehlen, sonst nichts, also keine `<style>`-Tags oder Ähnliches.

Und so erstellen Sie eine externe CSS-Datei:

1 Erstellen Sie in Ihrem Editor eine neue Datei und speichern Sie diese mit der Dateierendung `.css`.

Als Name kommt hier `style.css` zum Einsatz – nicht sehr einfallsreich, aber häufig gebraucht. Sie sollten hier nur dieselben Regeln beachten wie bei HTML-Dateinamen, also idealerweise acht Zeichen und Unterscheidung zwischen Groß- und Kleinschreibung.

2 Schreiben Sie nun einen Selektor in die externe Datei:

```
p {
```

`p` enthält, wie oben schon gesehen, CSS-Befehle für alle `<p>`-Tags.

3 Fügen Sie dann noch CSS-Befehle in den geschweiften Klammern hinzu.

```
p {  
    font-size: 16pt;  
    font-weight: bold;  
}
```

Nun müssen Sie die externe CSS-Datei in jeder HTML-Seite einbinden, in der Sie sie verwenden möchten.



- 1** Fügen Sie dazu das `<link>`-Tag in den `<head>`-Bereich der Seite ein:

```
<head>
  <title>Externes CSS</title>
  <link />
</head>
```

Dieses Tag gehört zu den Tags ohne Inhalt und wird deswegen mit einem Schrägstrich am Ende XHTML-konform geschlossen.

- 2** Ergänzen Sie im `<link>`-Tag das `rel`-Attribut mit der Angabe `stylesheet`:

```
<link rel="stylesheet" />
```

- 3** Anschließend geben Sie das `href`-Attribut mit dem Link auf die externe CSS-Datei an:

```
<link rel="stylesheet" href="style.css" />
```

Der oben gezeigte Link verweist auf die Datei `style.css` im gleichen Ordner. Wollten Sie z.B. auf eine CSS-Datei im Unterordner `css` verweisen, ginge das mit `css/style.css`. Die Verlinkung folgt hier den Regeln für Links (siehe Kapitel 2).

- 4** Zum Schluss geben Sie noch den Typ des verlinkten CSS an. Dieser ist immer `text/css`.

```
<link rel="stylesheet" href="style.css" type="text/css" />
```

Hinweis

Die Reihenfolge der hier erwähnten Parameter ist nicht von Bedeutung. HTML erlaubt alle Attribute eines Tags in beliebiger Reihenfolge. Der Entwickler versucht natürlich wie auch wir hier eine logische Reihenfolge einzuhalten.

Es gibt noch eine zweite Möglichkeit, externe CSS-Dateien einzubinden, nämlich den Befehl `@import`. Er hat den Nachteil, dass er nicht im Netscape Navigator 4.x und älteren Browsern funktioniert. Wenn Sie diesen allerdings ausschließen möchten, ist auch dieser Befehl einsetzbar. Hier ein Beispiel:

```
<style type="text/css">
  <!--
  @import url("style.css")
--></style>
```

Selektoren

Wenn Sie CSS-Befehle nicht direkt als Inline-Stil in das `style`-Attribut für ein Tag schreiben, müssen Sie deutlich machen, für welche Tags die Befehle gelten sollen. Genau diese Aufgabe übernehmen die Selektoren. CSS kennt eine Menge an Selektoren, wobei wir uns hier auf die wichtigsten beschränken, die auch in allen relevanten Browsern funktionieren.

Die erste Gruppe, die **Tag-Selektoren**, haben Sie bereits kennen gelernt. Sie schreiben den Namen des Tags als Selektor. Dann werden alle Tags dieser Art formatiert:

```
p {  
    font-size: 16pt;  
    font-weight: bold;  
}
```

Mehrere Tags lassen sich mit einer Komma-separierten Liste von Tag-Selektoren gemeinsam formatieren. Das heißt, die CSS-Befehle gelten für alle Tags in der Liste. Im folgenden Beispiel gelten die Befehle für Absätze (`<p>`-Tags) und Elemente von Listen (`<li>`, siehe *Kapitel 3*).

```
p, li {  
    font-size: 16pt;  
    font-weight: bold;  
}
```

Selektoren für verschachtelte Tags erlauben Ihnen, ineinander verschachtelte Tags zielgenau zu formatieren. Die verschachtelten Tags werden dabei einfach durch Leerzeichen getrennt eingegeben. Der folgende Selektor formatiert alle Listenelemente, die in einer nummerierten Liste (`<ol>`-Tag) liegen, die wiederum innerhalb einer zweiten nummerierten Liste liegt:

```
ol ol li {  
    font-size: 12pt;  
    font-style: italic;  
}
```

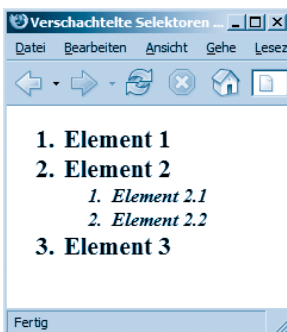
Hinweis

Das Beispiel mit Listen greift zwar Kapitel 3 ein wenig voraus. Wir haben es hier allerdings dennoch gewählt, da es in der Praxis bei weitem das häufigste ist.



Hier ein vollständiges Beispiel, bei dem zuerst die erste nummerierte Liste formatiert wird, dann die darunter verschachtelte:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- css_verschachtelt.html -->
<html>
<head>
  <title>Verschachtelte Selektoren</title>
  <style type="text/css"><!--
    ol li {
      font-size: 16pt;
      font-weight: bold;
    }
    ol ol li {
      font-size: 12pt;
      font-weight: normal;
      font-style: italic;
    }
  --></style>
</head>
<body>
  <ol>
    <li>Element 1</li>
    <li>Element 2</li>
    <li>
      <ol>
        <li>Element 2.1</li>
        <li>Element 2.2</li>
      </ol>
    </li>
    <li>Element 3</li>
  </ol>
</body>
</html>
```



Hinweis

Wichtig ist in diesem Beispiel, dass in der verschachtelten Liste die Schriftstärke mit `font-weight` wieder auf `normal` gesetzt wird. Ansonsten würden die Listenelemente der verschachtelten Liste die Eigenschaft `bold` der übergeordneten Liste erben. Siehe hierzu auch den Abschnitt »Präferenzreihenfolge«.

Die Selektoren, die Sie bisher kennen gelernt haben, sind speziell auf ein oder mehrere Tags beschränkt. Wenn Sie aber ganz bestimmte Elemente Ihrer HTML-Seite mit CSS-Befehlen versehen wollen, sollten Sie zu **Klassen** greifen. Eine Klasse definieren Sie einmal und können Sie dann beliebig vielen Elementen zuweisen.

Und so geht es:

- 1 Zuerst definieren Sie die Klasse mit einem beliebig von Ihnen wählbaren Namen ohne Sonderzeichen.

```
.hervorgehoben {
```

Eine Klasse erkennen Sie immer am vorangestellten Punkt vor dem Namen.

- 2 Anschließend legen Sie die CSS-Befehle fest.

```
font-size: 16pt;
font-weight: bold;
color: red;
}
```

`color` färbt den entsprechenden Text. Die anderen Befehle kennen Sie schon.

- 3 Nun fügen Sie jedem Tag, das die Klasse verwenden soll, das Attribut `class` hinzu.

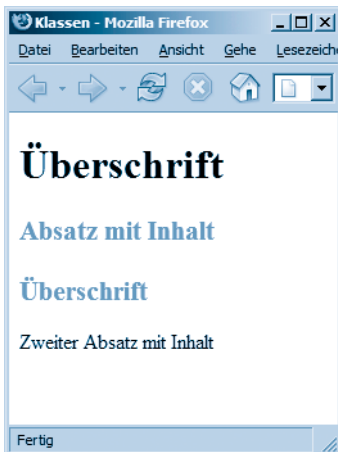
- 4 Als Wert des Attributs geben Sie den Klassennamen ohne vorangehenden Punkt an:

```
<p class="hervorgehoben">Absatz</p>
```



Hier ein vollständiges Beispiel, bei dem zwei HTML-Elemente mit der Klasse formatiert werden und zwei nicht. Zu den formatierten Elementen gehören auch Überschriften (<h1> und <h2>), die im nächsten Kapitel vorgestellt werden.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
  <title>Klassen</title>
  <style type="text/css"><!--
    .hervorgehoben {
      font-size: 16pt;
      font-weight: bold;
      color: red;
    }
  --></style>
</head>
<body>
  <h1>Überschrift</h1>
  <p class="hervorgehoben">Absatz mit Inhalt</p>
  <h2 class="hervorgehoben">Überschrift</h2>
  <p>Zweiter Absatz mit Inhalt</p>
</body>
</html>
```



Klassen können Sie auch mit **Tag-Selektoren kombinieren**. Dazu schreiben Sie einfach vor den Punkt das entsprechende Tag, für das die Klasse gelten soll. Wird die Klasse für ein anderes Tag verwendet, ist sie unwirksam. Im folgenden Beispiel ließen sich mit der Klasse `hervorgehoben` also nur Absätze, nicht aber z.B. Überschriften formatieren:

```
p.hervorgehoben {  
    font-size: 16pt;  
    font-weight: bold;  
    color: red;  
}
```

Hinweis

*Es gibt neben den normalen Klassen noch so genannte **Pseudoklassen**. Sie stehen z.B. bei Links für einen bestimmten Zustand. Sie lernen einige Pseudoklassen in Kapitel 2 kennen.*

Die letzte Selektorenart, die hier zur Sprache kommen soll, sind **IDs**. ID steht dabei für Identifier. IDs funktionieren im Prinzip wie Klassen. Man definiert im Kopf der Seite oder in einem externen CSS eine ID. Statt bei Klassen der Punkt kommt hier das Doppelkreuz (`#`, auch Hash) zum Einsatz.

```
#navigation {  
    color: blue;  
}
```

Beim Tag wird statt dem `class`-Attribut das `id`-Attribut verwendet.

```
<p id="navigation">  
    Navigationselemente  
</p>
```

Hinweis

*Einen gewaltigen Unterschied gibt es zwischen Klassen und IDs. Eine ID kennzeichnet **genau ein** Element innerhalb eines Dokuments, ist also eindeutig. Klassen können dagegen auf beliebig viele Elemente angewendet werden. Zwar kommen Browser auch mit einer mehrfach vergebenen ID zurecht, da dies aber unter Umständen Probleme bei der JavaScript- oder serverseitigen Programmierung hervorrufen kann, sollten Sie eine ID immer pro Dokument nur einmal vergeben.*



In der Praxis werden IDs hauptsächlich für eindeutig auf einer Seite nur einmal vorhandene Elemente wie Navigationsleiste, Kopfzeile, Copyright-Vermerk etc. verwendet. Wie auch Klassen, so lassen sich IDs an bestimmte Tags koppeln. Da die ID allerdings sowieso Tag-übergreifend eindeutig sein sollte, ist das eher eine Spielerei.

Präferenzreihenfolge

Sie haben mehrere Arten kennen gelernt, CSS-Befehle auf HTML-Tags anzuwenden. Diese lassen sich beliebig mischen. Daraus ergibt sich die Frage, was passiert, wenn zwei sich widersprechende CSS-Befehle einem Tag zugewiesen werden. CSS kennt hierfür eine eindeutige Reihenfolge. Diese ließe sich auch berechnen, allerdings reichen in der Praxis vier Regeln:

- Einige CSS-Befehle beispielsweise zu Schriftart und Schriftfarbe werden **vererbt**. Ein Beispiel: Wenn Sie einem `<p>`-Tag eine Formatierung zuweisen, gilt die Formatierung auch für darin eingeschlossene `<b>`-Tags.
 - Vererbte CSS-Befehle werden immer von speziell dem einzelnen Tag zugewiesenen überschrieben. Bei einem Tag selbst gilt die folgende Präferenzreihenfolge:
 - Klasse vor Tag-Selektor
 - ID vor Klasse
 - Inline-Stil vor ID
 - Bei gleichwertigen Stilanweisungen, die z.B. im Kopf der Seite und in einer externen CSS-Datei jeweils als Klasse festgelegt werden, entscheidet die Reihenfolge des Auftauchens im Quellcode. Immer das zuunterst im `<head>`-Bereich eingebundene CSS – egal ob intern oder extern – setzt sich durch.
 - Mit dem Nachsatz `!important` gekennzeichnete CSS-Befehle sind allen nicht damit gekennzeichneten vorzuziehen. Untereinander gelten bei `!important`-Befehlen die oben genannten drei Regeln.
- ```
color: blue !important;
```

## <div> und <span>

Im CSS-Kontext gibt es zwei HTML-Tags, die auf jeden Fall noch vorgestellt werden sollten: <div> und <span>. <div> ist ein Blockelement, ähnlich wie ein Absatz (<p>). Allerdings besitzt <div> keinerlei Eigenleben, also keine eigenen Formate, wohingegen ein Absatz vor- und nachgelagert einen Abstand besitzt. <div>-Elemente kommen z.B. zum Positionieren mit CSS zum Einsatz (siehe *Kapitel 7*).

```
<div>Text im div-Block mit nicht erkennbarem
span-Bereich.</div>
```

Das zweite besondere Tag ist <span>. Hierbei handelt es sich um ein Inline-Tag, es definiert also keinen Block, sondern nur einen Bereich. Dieser Bereich ist ebenfalls nicht vordefiniert. In der Praxis wird <span> hauptsächlich dazu verwendet, CSS-Formatierungen für einen bestimmten Textbereich vorzunehmen.

## Medienspezifische CSS

Der CSS-Standard bietet einige Möglichkeiten, die mit Fug und Recht als fortschrittlich zu bezeichnen sind. Dazu gehört, Stile nur bei bestimmten Medien anzuwenden. Das heißt, Sie definieren CSS-Befehle, die dann z.B. ausschließlich zum Ausdrucken benutzt werden. Das ist z.B. praktisch, wenn Sie dem Nutzer eine Druckversion Ihrer Seite angeben möchten.

Es gibt drei Möglichkeiten, einen Medientyp für Stylesheets oder einzelne CSS-Befehle anzugeben:

- Für externe CSS-Dateien im <link>-Tag mit dem Attribut media.

```
<link href="style_print.css" rel="stylesheet"
type="text/css" media="print" />
```

- Für externe Stylesheets mit @import. Hier schreiben Sie die Medientypen einfach hinter dem URL des CSS. Mehrere Medientypen werden per Kommata getrennt:

```
@import url("style_print.css") print, braille;
```

- Mittels des @media-Befehls direkt im Stylesheet. Dies klappt allerdings erst mit Internet Explorer 5.5 und nicht mit 5.0. Mehrere Medientypen werden auch hier per Kommata aneinander gehängt.

```
@media print {
 CSS-Befehle
}
```





Tabelle 1.2 zeigt einige der wichtigsten Medientypen. Heute am meisten in der Praxis im Einsatz ist `print` für die Druckerausgabe.

| Medientyp               | Beschreibung                                                                                             |
|-------------------------|----------------------------------------------------------------------------------------------------------|
| <code>all</code>        | Für alle Medien. Das ist die Standardeinstellung, wenn kein Medientyp festgelegt ist.                    |
| <code>aural</code>      | Stylesheet für die Sprachausgabe mit Sprachbrowsern.                                                     |
| <code>braille</code>    | Für die Ausgabe mit Braille-Geräten (Blindenschrift).                                                    |
| <code>handheld</code>   | Für mobile Endgeräte.                                                                                    |
| <code>print</code>      | Für die Druckerausgabe.                                                                                  |
| <code>projection</code> | Für Präsentationen auf Beamern.                                                                          |
| <code>screen</code>     | Für die Ausgabe auf dem Bildschirm.                                                                      |
| <code>speech</code>     | Für die Ausgabe mit Sprachsoftware. Dieser Medientyp ist neu und ergänzt ab CSS 2.1 <code>aural</code> . |

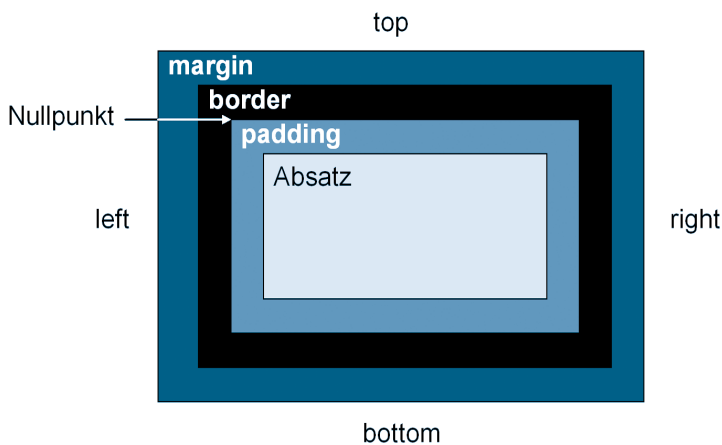
*Tabelle 1.2: Wichtige Medientypen für medienspezifische Style Sheets*

## Nutzloses Wissen

Leonardo da Vinci konnte gleichzeitig mit einer Hand schreiben und mit der anderen malen.

## Kapitel 2

# Einfache Webseiten mit Text und Links



Wer an seiner ersten Webseite bastelt, beginnt meist mit ein wenig Inhalt. Das mag ein Text über sich selbst, die eigene Firma oder das Hobby sein. Wie Sie diesen Text in HTML realisieren und mit CSS garnieren, sprich verschönern, erfahren Sie in diesem Kapitel.

Sobald der Text fertig ist, wird es sehr schnell notwendig, auf andere Seiten zu verweisen. Sei es, dass die eigene Website aus mehreren Seiten bestehen soll oder dass eine befreundete Website nach einem Link verlangt. Vorsicht bei Links nach außen. Hier wird der Einsatz von Haftungsausschlüssen kontrovers diskutiert (Kapitel 11).

**Das können Sie schon:**

|                                |    |
|--------------------------------|----|
| Der Aufbau des Web             | 12 |
| HTML-Grundgerüst einer Website | 26 |
| Einsatz von CSS                | 40 |

**Das lernen Sie neu:**

|                                       |    |
|---------------------------------------|----|
| Formatierung von Text                 | 56 |
| Schriften und Schriftgrößen           | 56 |
| Einsatz von Links                     | 75 |
| Links gestalten                       | 76 |
| E-Mail-Links und Links auf FTP-Server | 80 |
| Rollover-Effekte                      | 84 |

## Formatierung

Text landet in HTML, wie Sie schon in *Kapitel 1* lesen konnten, immer in einem Blockelement wie z.B. einem `<div>`-Block. Das gebräuchlichste Blockelement sind Absätze, also das `<p>`-Tag, wobei p für Paragraph (engl. für Absatz) steht.

Sie können entweder den gesamten Absatz oder auch nur Bereiche eines Textes formatieren. Für Teilformatierungen bieten sich eigene Formatiertags wie `<b>` für fetten Text an. Wollen Sie per CSS formatieren, greifen Sie zum `<span>`-Tag, um einen Textbereich zu markieren (siehe *Kapitel 1*).

## Schrift

Für die Schriftformatierung gab es früher, also vor der Einführung von CSS, das so genannte `<font>`-Tag. Dies sollten Sie aus zwei Gründen nicht mehr verwenden:

- Zum einen ist es vom HTML-Gralshüter W3C als deprecated, also nicht mehr empfohlen gekennzeichnet.
- Zum anderen kommt wirklich jeder relevante Browser mit den viel besseren CSS-Befehlen zur Schriftformatierung zurecht.

Die wichtigste Schriftformatierung ist sicherlich die **Schriftart**, denn sie hat vielleicht die größte Auswirkung auf das optische Erscheinungsbild. Bei der Schriftart gibt es im Web eine große Einschränkung: Der Nutzer muss die Schrift besitzen, die Sie verwenden. Sie haben also nicht die freie Wahl aus allen auf Ihrem System installierten Schriften, sondern Sie müssen eine der auf (fast) allen Systemen vorhandenen Schriften wie Times New Roman, Arial, Helvetica, Verdana, Tahoma, Courier etc. verwenden.

### Hinweis

*Downloadbare Schriften sollten dieses Problem beseitigen. So gab und gibt es sowohl von Bitstream mit TrueDoc eine Lösung für Netscape 4.x als auch von Microsoft die Web Embedded Fonts für den Internet Explorer. Beide Lösungen sind aber jeweils auf die genannten Browser beschränkt und völlig inkompatibel. Insofern ist das Thema downloadbare Schriften heute tot. Wenn Sie eine spezielle Schrift einsetzen möchten, können Sie das in der Praxis nur, indem Sie sie in eine Grafik umwandeln.*



Die Schriftart geben Sie in CSS mit dem Befehl `font-family` an. Sie können mehrere Schriftarten durch Kommata getrennt angeben. Der Browser sucht dann nach der ersten; findet er sie nicht, nimmt er die zweite und so weiter. Außer einer Schriftart können Sie auch eine Schriftfamilie nehmen. Das ist eine bestimmte Art von Schriften. Die genaue Schrift legt dann der Browser bzw. das System fest. Tabelle 2.1 zeigt die möglichen Schriftfamilien.

| Schriftfamilie | Schriften                          | Beschreibung                                                                           |
|----------------|------------------------------------|----------------------------------------------------------------------------------------|
| sans-serif     | Helvetica, Arial, Verdana, Geneva  | Schriften ohne Serifen, das heißt keine Verästelungen an den Enden der Buchstaben      |
| serif          | Times New Roman, Garamond, Georgia | Schriftart mit Serifen                                                                 |
| monospace      | Courier, Everson Mono              | diktengleiche Schriftarten; alle Buchstaben sind gleich breit                          |
| cursive        | Snell Roundhand, Adobe Poetica     | handschriftenähnlich, mit verbundenen Buchstaben oder anderen kursiven Charakteristika |
| fantasy        | Cottonwood, Alpha Geometrique      | außergewöhnliche Schriften z.B. mit Symbolen oder für spezielle Einsatzzwecke          |

*Tabelle 2.1: Schriftfamilien und zugehörige Schriften (geordnet nach der Häufigkeit des Einsatzes)*

Hier ein Beispiel für das Festlegen von Schriftarten:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- font_family.html -->
<html>
<head>
 <title>Schriftart</title>
</head>
<body>
 <p style="font-family: Verdana, Arial, Helvetica,
 sans-serif;">Text mit serifenloser Schrift.</p>
 <p style="font-family: Courier, monospace;">Diktengleiche
 Schrift z.B. für Quellcode-Darstellung.</p>
</body>
</html>
```

### Tipp

*Im Web sind hauptsächlich serifenlose Schriften wie Arial und Verdana gebräuchlich, da sie gut lesbar sind. Vor allem Verdana ist zurzeit in.*

### Hinweis

*Zusammengesetzte Schriftnamen wie Times New Roman werden in Anführungszeichen geschrieben. Wenn Sie die CSS-Befehle im `style`-Attribut in doppelte Anführungszeichen setzen, müssen Sie natürlich einfache Anführungszeichen verwenden.*

## Schriftgröße und -farbe

Wenn Sie die Schriftart festgelegt haben, können Sie noch Schriftgröße und -farbe festlegen. Für die Schriftgröße ist `font-size` zuständig. Wichtig ist die Einheit, die direkt nach der Zahl (ohne Leerzeichen) folgt. Sie haben hier wie bei jeder Maßangabe alle CSS-Maßeinheiten zur Auswahl. Diese unterscheiden sich in absolute Größen und relative Größen. Dies sind die absoluten Größen:

- `cm` für Zentimeter.
- `mm` für Millimeter.
- `in` ist ein Zoll oder Inch. Ein Inch entspricht 2,54 cm.
- `pt` ist ein Punkt und entspricht 1/72 Inch. Diese Einheit kommt in diesem Buch für Schriftgrößen zum Einsatz.
- `pc` ist ein Pica und entspricht 12 Punkt.

Und hier die relativen Größen:

- `px` entspricht einem Pixel, also einem Bildschirmpunkt. Die Größe eines Pixels ist abhängig von der Auflösung des Bildschirms (sie reicht im Allgemeinen von 72 bis 96 Pixel pro Inch). Diese Maßeinheit eignet sich hervorragend für Positionierungs- und Größenangaben. Bei Schriftarten ist sie nicht sinnvoll, da der Internet Explorer Schrift mit Pixelangaben nicht skalieren kann.
- `em` bestimmt die Schriftgröße relativ zur aktuellen Größe. Messgrundlage ist die Schrifthöhe einer bestimmten Schriftart. Ist diese Schrift 14 pt hoch, entspricht dies 1em.
- `ex` bestimmt die Schriftgröße relativ zur Größe des Buchstabens x. 1 `ex` entspricht meist einem halben `em`.
- `%` für Prozentangaben abhängig von der Fenstergröße oder bei Schriftarten von der normalen Schriftgröße. Wird für Schrift kaum verwendet.



Die Schriftfarbe steuern Sie mit dem `color`-Befehl. Hier ist wichtig zu wissen, wie im Web Farben angegeben werden. Die Grundlage der Farben im Web beziehungsweise auf dem Monitor ist das **RGB-Farbmodell**. RGB steht für Rot, Grün und Blau. Diese drei Grundfarben bilden zusammen alle darstellbaren Farben. Eine Farbe setzt sich dabei aus Helligkeitsanteilen von Rot, Grün und Blau zusammen – deswegen nennt man das RGB-Modell auch additiv.

Von jeder der drei Grundfarben gibt es insgesamt 256 Helligkeitsstufen, die von den Zahlen 0 bis 255 dargestellt werden. 255 Rot, 0 Grün und 0 Blau wäre dementsprechend reines Rot. 255 von jeder Grundfarbe ergibt zusammen Weiß, 0 von jeder Grundfarbe ergibt Schwarz.

#### Hinweis

*Früher sprach man von 216 websicheren Farben. Websicher deswegen, weil diese Farben sowohl unter Windows als auch am Mac vorhanden waren, wenn der Monitor oder die Grafikkarte nur 256 Farben darstellen konnte. Hätte man andere Farben verwendet, wären diese zur jeweils nächstähnlichen Farbe verschoben worden, was oft recht psychedelische Effekte hatte. Heutzutage unterstützen fast alle Monitore mehr Farben und die websicheren Farben sind in der Versenkung verschwunden.*

Die Farbkombinationen lassen sich in CSS auf folgende Arten schreiben:

- Als RGB-Farbe mit dem Farbwert in dezimalen Zahlen:

```
color: rgb(255, 0, 0);
```

- Als RGB-Farbe in hexadezimaler Schreibweise. Diese Schreibweise wird auch von HTML verwendet:

```
color: #FF0000;
```

oder bei jeweils zwei gleichen Ziffern auch:

```
color: #F00;
```

Die hexadezimale Schreibweise basiert auf einem Zahlensystem mit der Basis 16 statt mit der Basis 10. Für die Zahlen 10 bis 15 kommen die Buchstaben von A bis F hinzu. Dadurch lässt sich jede Helligkeitsstufe in zwei Ziffern schreiben. Die Umrechnung erfolgt so: Sie nehmen den RGB-Wert und teilen ihn durch 16. Das ganzzahlige Ergebnis ist die erste Ziffer. Bei 207 ergibt dies 12, also C. Der ganzzahlige Rest der Division ist die zweite Ziffer. Zum Ausrechnen multiplizieren Sie 12 mit 16, was 192 ergibt, und ziehen das von 207 ab. Die zweite Ziffer ist also

15 und damit F. Daraus ergibt sich, dass 207 in hexadezimaler Schreibweise CF entspricht. Hierzu können Sie natürlich auch den Taschenrechner des Betriebssystems verwenden.

- Als Farbname. CSS sieht sechzehn standardisierte englischsprachige Farbnamen vor wie z.B. red für Rot, blue für Blau und green für Grün. Dazu gibt es noch viele andere Farbnamen, die viele Browser unterstützen.

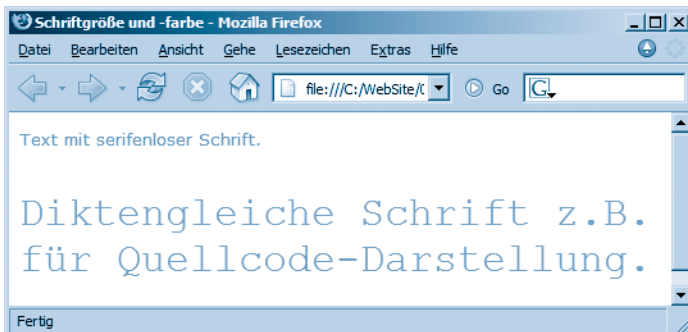
color: red;

### Hinweis

*color ist der einzige Befehl direkt zur Schriftformatierung, der ohne vorangestelltes font auskommt.*

Hier nun ein Beispiel, das Schriftgröße und Farbangabe miteinander verbindet:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- font_size_color.html -->
<html>
<head>
 <title>Schriftart</title>
</head>
<body>
 <p style="font-family: Verdana, Arial, Helvetica,
 sans-serif; font-size: 10pt; color: red;">Text mit
 serifenloser Schrift.</p>
 <p style="font-family: Courier, monospace; font-size: 2em;
 color: rgb(255, 51, 76);">Diktengleiche Schrift
 z.B. für Quellcode-Darstellung.</p>
</body>
</html>
```



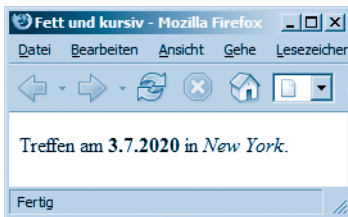


## Fett und kursiv

Die Schriftdicke und die Schrägstellung von Textpassagen gehören zu den wichtigsten Auszeichnungsmerkmalen für einzelne Textbereiche. Fett und kursiv gibt es dementsprechend nicht nur in Word, sondern auch in HTML und CSS. Hier handelt es sich um einen der Fälle, wo die HTML-Befehle noch genauso gebräuchlich sind wie die CSS-Pendants. Wir zeigen Ihnen deswegen beide:

In HTML verwenden Sie das `<b>`-Tag, um einen Bereich fett hervorzuheben, und das `<i>`-Tag, um ihn kursiv zu stellen. Beide lassen sich natürlich auch beliebig kombinieren. Allerdings sollten Sie sie XHTML-konform nur in Block-elementen wie einem Absatz, einem `<div>`-Tag oder einer Tabelle verwenden.

```
<p>Treffen am 3.7.2020 in <i>New York</i>.</p>
```



In CSS ist für die Schriftdicke `font-weight` verantwortlich. Dieser Befehl verwendet entweder die absolute Angabe `bold` für fett, relative Angaben wie `bolder` (dicker) und `lighter` (dünner) oder die Zahlen von 100 bis 700, wobei 400 eine normale Schrift darstellt. In der Praxis ist zu 99 % `bold` im Einsatz, da die anderen je nach Schriftschnitt keine Auswirkungen haben.

`font-style` sorgt mit dem Wert `italic` für kursive Schrift. Hier ein zur vorherigen HTML-Variante funktionsgleiches Beispiel:

```
<p>Treffen am 3.7.2020
in New York.</p>
```

### Hinweis

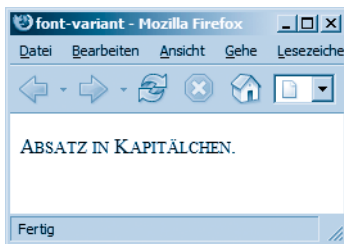
*Es gibt für `font-style` noch den Wert `oblique`, der eine Schrift ebenfalls kursiv stellt. Der Teufel steckt hier im Detail: `italic` verwendet nach Möglichkeit eine kursive Version der Schrift auf dem System (einen so genannten Schriftschnitt). Nur wenn keiner vorhanden ist, wird die Schrift per Berechnung kursiv gestellt. `oblique` berechnet automatisch und ignoriert kursive Schriftschnitte.*

## Typographische Befehle

Typographie ist die Lehre von Schrift und Schriftgestaltung. HTML war ursprünglich aus typographischer Sicht ein Trauerspiel, da es kaum Möglichkeiten gab, auf das Aussehen des Textes Einfluss zu nehmen. Mit CSS hat sich nicht nur einiges bei den gebräuchlichen Befehlen aus den letzten beiden Abschnitten geändert, sondern auch im Speziellen sind neue Möglichkeiten hinzugekommen:

- `font-variant` wird hauptsächlich dazu eingesetzt, Text mit dem Wert `small-caps` in Kapitälchen umzuwandeln. Kapitälchen heißt, dass alle Kleinbuchstaben als verkleinerte Großbuchstaben dargestellt werden. Je nach Schriftart kann es dafür sogar einen eigenen Schriftschnitt geben. Das Gegenstück zu `small-caps` ist der Standardwert `normal`.

```
<p style="font-variant: small-caps;">Absatz in
Kapitälchen.</p>
```



- `text-transform` wandelt einen Text in Kleinbuchstaben (`lowercase`), Großbuchstaben (`uppercase`) oder große Anfangsbuchstaben und dann Kleinschreibung (`capitalize`) um.

```
<p style="text-transform: uppercase;">Text wird
automatisch in Großbuchstaben umgewandelt.</p>
```

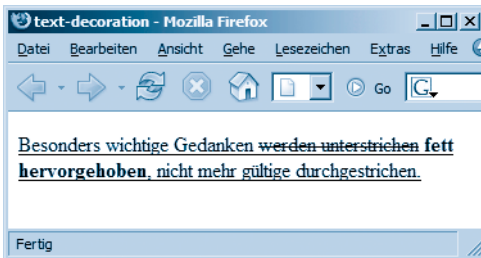


- `text-decoration` dient dazu, Text auszuzeichnen. Die Möglichkeiten sind dabei recht weit reichend: `underline` unterstreicht, `line-through` streicht durch und wird z.B. dazu verwendet, ältere, nicht mehr gültige



Gedanken in Texten festzuhalten. `overline` überstreicht, fügt also eine Linie über dem Text ein. `none` entfernt jede Art von Strich. Mit dem Wert `blink` können Sie außerdem einen Text zum Blinken bringen. Das klappt allerdings nicht im Internet Explorer.

```
<p style="text-decoration: underline;">Besonders wichtige Gedanken werden unterstrichen fett hervorgehoben, nicht mehr gültige durchgestrichen.</p>
```

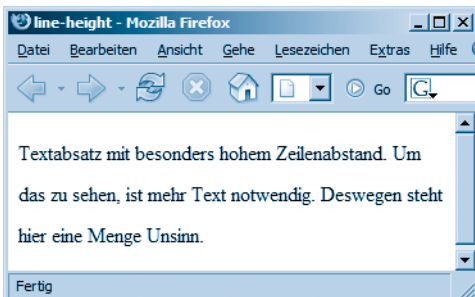


### Tipp

`text-decoration` kommt oftmals bei Links zum Einsatz. Mit `text-decoration: none;` lässt sich nämlich die Unterstreichung des Links entfernen. Allerdings sollten Sie dann darauf achten, dass Links nach wie vor gut genug zu erkennen sind. Unterstreichungen im normalen Text sollten Sie weitgehend vermeiden, da sie zu sehr an Links erinnern.

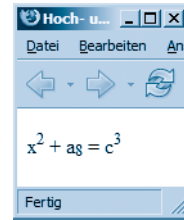
- `line-height` bestimmt die Zeilenhöhe. Die Angabe erfolgt oft in Prozent oder Punkt.

```
<p style="line-height: 200%;">Textabsatz mit besonders hohem Zeilenabstand.</p>
```



Nur ein Bereich, nämlich hochgestellte und tiefgestellte Buchstaben, ist eher eine Sache von HTML als von CSS. HTML sieht hier nämlich die Tags `<sup>` (hochgestellt) und `<sub>` (tiefgestellt) vor, für die es keine direkte CSS-Entsprechung gibt.

```
<p>x² + a₈ = c³
</sup></p>
```



Komplexere Formeln lassen sich in HTML nicht mehr ausdrücken. Diese stellt man entweder als Grafiken ins Netz oder verwendet eine Technologie wie MathML, eine auf XML basierende Sprache für mathematische Formeln. Letztere benötigt allerdings ein eigenes Browser-Plug-In.

### Hinweis

*Der Befehl `font` fasst `font-style`, `font-variant`, `font-weight`, `font-size/line-height` und `font-family` zusammen. Unbedingt vorhanden sein müssen `font-size` und `font-family`, und zwar immer und in genau in ihrer Position am Ende der Aufzählung, die anderen drei sind optional und können in ihrer Reihenfolge vertauscht werden. Separat festlegen müssen Sie `color`.*

## Überschriften

HTML kennt für Text noch weitere Blockelemente außer dem normalen Absatz: die Überschriften. Sie reichen von `<h1>` bis `<h6>`. `<h1>` ist dabei die Überschrift der obersten Ebene. Überschriften werden vom Browser selbstständig mit anderer Schriftgröße und Hervorhebung formatiert.

### Tipp

*In der Praxis macht es allerdings mehr Sinn, die Formatierung von Überschriften per CSS selbst zu übernehmen und die Überschriften-Tags nur als logische Auszeichnung zu verwenden. Logisch heißt, dass Sie für eine oberste Überschriftenebene auch nur `<h1>` verwenden und darunter direkt `<h2>` und nicht wegen kleinerer Schriftgröße `<h3>` oder `<h4>`.*



Ein strukturierter Text zeigt, wie die Überschriften funktionieren. Statt realen Inhalten finden Sie hier kurze Platzhalter:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- ueberschriften.html -->
<html>
<head>
 <title>Überschriften</title>
</head>
<body>
 <h1>Thema des Kapitels</h1>
 <p>Vortext.</p>
 <h2>Inhalt 1</h2>
 <p>Beschreibung.</p>
 <h3>Unterpunkt 1.1</h3>
 <p>Erläuterung.</p>
 <h2>Inhalt 2</h2>
 <p>Beschreibung.</p>
</body>
</html>
```



Mit Hilfe von CSS können Sie dies nun hübscher formatieren. Sehr praktisch sind dazu die Tag-Selektoren (siehe *Kapitel 1*). So geht es Schritt für Schritt:

**1** Erstellen Sie einen `<style>`-Block im Kopf der HTML-Seite.

**2** Zuerst legen Sie eine Schriftart für die gesamte Seite fest. Dies geschieht am besten für das `<body>`-Tag, da die Schriftart an alle untergeordneten Elemente vererbt wird.

```
body {
 font-family: Arial, Geneva, Helvetica, sans-serif;
}
```

### **Tipp**

*Der Netscape Navigator 4.x vererbt nicht korrekt. Wenn Sie eine Schriftart auch in Tabellen verwenden möchten, müssen Sie die Tags für Tabellenzellen (siehe Kapitel 3) mit einbeziehen:*

```
body, td, th {
 font-family: Arial, Geneva, Helvetica, sans-serif;
}
```

**3** Anschließend formatieren Sie die normalen Absätze. Diese Reihenfolge ist zwar nicht nötig, eine logische Anordnung hilft aber, den Code auch im Nachhinein noch verstehen und warten zu können.

```
p {
 font-size: 10pt;
}
```

**4** Als Nächstes ist Überschrift 1 dran. Sie wird vom Browser automatisch sehr groß und fett formatiert. Die fette Formatierung ist in Ordnung, aber die Größe schrauben wir herunter.

```
h1 {
 font-size: 14pt;
 font-weight: bold;
}
```

Eigentlich müssten Sie `font-weight` hier nicht angeben. Da einige Nutzer aber auch eigene Stilangaben treffen, wenn Ihre Stilangaben fehlen, haben wir uns dies angewöhnt.



**5** Überschrift 2 formatieren Sie ähnlich wie Überschrift 1, nur ein wenig kleiner:

```
h2 {
 font-size: 12pt;
 font-weight: bold;
}
```

**6** Bei Überschrift 3 entfernen Sie die fette Formatierung und verwenden stattdessen kursiv:

```
h3 {
 font-size: 10pt;
 font-weight: normal;
 font-style: italic;
}
```

Achten Sie immer auf die Standardwerte. Überschrift 3 ist standardmäßig fett formatiert. Das heißt, Sie müssen in CSS `font-weight` auf `normal` setzen, um den Fettdruck auszuschalten.

Hier das komplette Beispiel in der Übersicht:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- ueberschriften_format.html -->
<html>
<head>
 <title>Überschriften</title>
 <style type="text/css"><!--
 body {
 font-family: Arial, Geneva, Helvetica, sans-serif;
 }
 p {
 font-size: 10pt;
 }
 h1 {
 font-size: 14pt;
 font-weight: bold;
 }
 h2 {
 font-size: 12pt;
 font-weight: bold;
 }
 h3 {
 font-size: 10pt;
 font-weight: normal;
 font-style: italic;
 }
 -->
</style>
</head>
<body>
 <h1>Überschriften</h1>
 <h2>Überschrift 2</h2>
 <h3>Überschrift 3</h3>
</body>
</html>
```

```

--></style>
</head>
<body>
 <h1>Thema des Kapitels</h1>
 <p>Vortext.</p>
 <h2>Inhalt 1</h2>
 <p>Beschreibung.</p>
 <h3>Unterpunkt 1.1</h3>
 <p>Erläuterung.</p>
 <h2>Inhalt 2</h2>
 <p>Beschreibung.</p>
</body>
</html>

```



## Absätze aufmöbeln

Bisher bezogen sich alle Formatierungen direkt auf den Text. Sie können aber auch die Blockelemente selbst formatieren und den Text in den Blockelementen ausrichten. Dies zeigen wir Ihnen hier hauptsächlich am Beispiel von Absätzen. Das Gesagte zum Ausrichten und zu den Rahmen gilt allerdings auch für andere Blockelemente wie Überschriften.





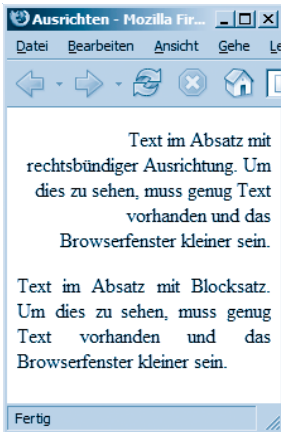
## Ausrichten

Text im Absatz lässt sich recht einfach ausrichten. Dazu dient das Attribut `align` im `<p>`-Tag. Dieses Attribut kennen allerdings auch andere Blockelemente wie Überschriften und `<div>`-Elemente. `align` kann folgende Werte annehmen:

- `left` für linksbündig. Dies ist der Standardwert.
- `center` für zentriert
- `right` für rechtsbündig
- `justify` für Blocksatz. Dieser klappt allerdings in älteren Browsern nicht und führt oftmals zu komischen Ergebnissen, da Browser nicht die Darstellungsqualitäten von Satzprogrammen besitzen und so unschöne Lücken entstehen, da HTML keine Silbentrennung beherrscht.

Hier ein Beispiel für den Einsatz:

```
<p align="right">Text im Absatz mit rechtsbündiger
 Ausrichtung. Um dies zu sehen, muss genug Text vorhanden
 und das Browserfenster kleiner sein.</p>
<p align="justify">Text im Absatz mit Blocksatz. Um dies zu
 sehen, muss genug Text vorhanden und das Browserfenster
 kleiner sein.</p>
```



In CSS gibt es als Entsprechung zum HTML-Attribut `align` den Befehl `text-align`. Er akzeptiert dieselben Werte wie `align`.

```
<p style="text-align: right">Text im Absatz mit rechtsb
 ündiger Ausrichtung. Um dies zu sehen, muss genug
 Text vorhanden und das Browserfenster kleiner sein.</p>
```

```
<p style="text-align: justify">Text im Absatz mit Blocksatz.
Um dies zu sehen, muss genug Text vorhanden und das
Browserfenster kleiner sein.</p>
```

Die CSS-Version ist vorzuziehen, da das HTML-Attribut vom W3C als deprecated gekennzeichnet ist. Praktische Unterschiede gibt es allerdings ansonsten nicht.

### Hinweis

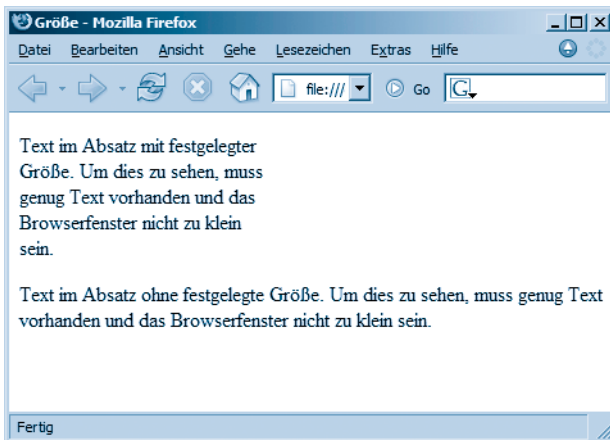
*Achtung, im Internet Explorer richtet text-align auch verschachtelte Blockelemente aus. Dies ist aber ein fehlerhaftes Verhalten und funktioniert z.B. im Mozilla Firefox nicht.*

## Größe, Abstände, Rahmen und das Boxmodell

Die Größe eines Blockelements mit Text richtet sich normalerweise nach der Menge an Text. Sie können allerdings die Breite exakt angeben. Dazu dient der CSS-Befehl `width`, der eine beliebige CSS-Maßeinheit (siehe oben bei der Schriftgröße) akzeptiert:

```
<p style="width: 200px">Text im Absatz mit festgelegter
Größe. Um dies zu sehen, muss genug Text
vorhanden und das Browserfenster nicht zu klein sein.</p>
```

```
<p>Text im Absatz ohne festgelegte Größe. Um dies zu sehen, muss genug Text
zu sehen, muss genug Text vorhanden und das Browserfenster
nicht zu klein sein.</p>
```

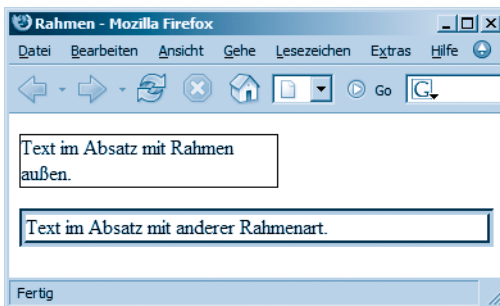


Neben der Größe kann ein Blockelement auch einen Rahmen haben. Hierfür dienen die border-Befehle:

- border-width gibt die Rahmenstärke an.
- border-color gibt die Rahmenfarbe an. transparent ist ein durchsichtiger Rahmen.
- border-style steuert, wie der Rahmen aussieht. dashed ist gestrichelt, dotted gepunktet, solid durchgezogen, double eine doppelte Linie, ridge ist gewölbt und groove wirkt reliefartig. none bedeutet ohne Rahmen, hidden, dass der Rahmen unsichtbar ist.

Hier ein einfaches Beispiel:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- rahmen.html -->
<html>
<head>
 <title>Rahmen</title>
</head>
<body>
 <p style="width: 200px; border-width: 1px; border-style:
 solid; border-color: black;">Text im Absatz mit Rahmen
 außen.</p>
 <p style="border-width: 5px; border-style: groove; border-
 color: blue;">Text im Absatz mit anderer Rahmenart.</p>
</body>
</html>
```



## Hinweis

*Sie können für den Rahmen auch den Sammelbefehl `border` verwenden. Er enthält Stil, Breite und Farbe und zwar in beliebiger Reihenfolge, z.B.*

`border: 5px groove blue;`

*Rahmen lassen sich auch für jede der vier Seiten einzeln festlegen. Dazu verwenden Sie die Befehle `border-left`, `border-top`, `border-right` und `border-bottom`. Diese vier Seitenangaben gibt es wiederum auch in Verbindung mit den drei Einzelbefehlen. Um also z.B. die Dicke der rechten Seite festzulegen, verwenden Sie `border-right-width`. Sie können die vier Seiten allerdings auch in den normalen `border`-Befehlen angeben, indem Sie zwei bis vier Werte hintereinander schreiben:*

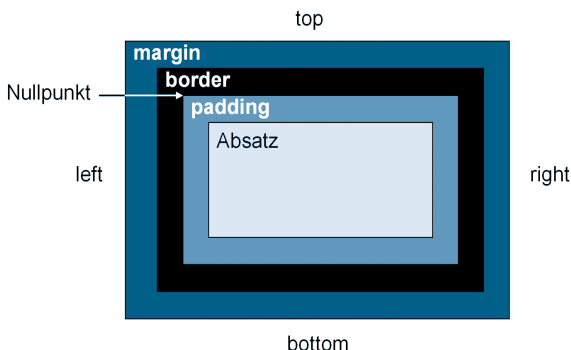
- Bei zwei Werten gilt der erste für oben und unten, der zweite für links und rechts.
- Bei dreien gilt der erste für oben, der zweite für links und rechts und der dritte für unten.
- Bei vieren geht die Reihenfolge oben, rechts, unten und links.

*Hier ein Beispiel:*

`border-color: red green blue red;`

Bis jetzt »pappt« der Text direkt am Rahmen. Dies können Sie allerdings mit dem Innenabstand des Blockelements steuern. Für den Innenabstand ist der CSS-Befehl `padding` zuständig. Den Abstand nach außen, das heißt vom Rahmen zu umliegenden Elementen, regelt `margin`. Wie beim Rahmen können Sie auch hier jeweils alle vier Seiten einzeln steuern.

Diese zwei bilden zusammen mit dem Rahmen das so genannte Boxmodell, das für jedes Blockelement gilt:



Der Nullpunkt ist die linke obere Ecke des Blockelements, an der es positioniert wird (siehe *Kapitel 7*).

### Hinweis

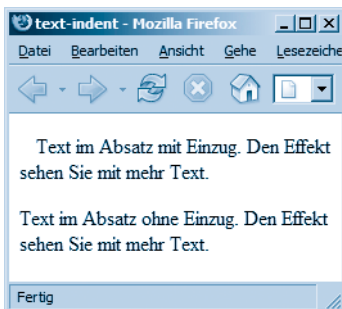
*Das Boxmodell wird nicht in jedem Browser korrekt interpretiert. Der Internet Explorer 5.0 und 5.5 zählen zur Breite eines Elements Innenabstand, also padding, und Rahmen, also border, dazu. Das heißt, dass diese Browser eine Box schmaler interpretieren als standardkonforme Browser. Der Internet Explorer 6 macht alles richtig, wenn ein XHTML-DocType angegeben ist.*

*Wenn Sie Ihr Design völlig gleich gestalten wollen, können Sie einen so genannten Hack, also einen CSS-technischen Trick, verwenden, um diesen Effekt zu vermeiden. Allerdings ist das in der Praxis nur bedingt empfehlenswert und sprengt den Rahmen dieses Buches. Für Details sei hier auf das umfassende Premium-Codebook **HTML/CSS** verwiesen. Es ist erschienen bei Addison-Wesley und wurde von den Autoren dieses Buches verfasst.*

## Einzüge und spezielle Formate

Für Absätze gibt es noch einige Besonderheiten, um Einzüge, die erste Zeile und den ersten Buchstaben zu formatieren. Für einen Einzug der ersten Zeile ist der CSS-Befehl `text-indent` zuständig:

```
<p style="text-indent: 10pt;">Text im Absatz mit Einzug.
 Den Effekt sehen Sie mit mehr Text.</p>
<p>Text im Absatz ohne Einzug. Den Effekt sehen Sie mit mehr
 Text.</p>
```



Um speziell die erste Zeile oder den ersten Buchstaben zu formatieren, gibt es zwei so genannte Pseudoklassen. Das sind CSS-Klassen mit vorgegebener Funktionalität. Sie erkennen sie an dem Doppelpunkt statt dem Punkt vor dem Klassennamen. Für diese Klassen können Sie dann beliebige CSS-Befehle festlegen. Zur Verfügung stehen:

- `:first-letter` für den ersten Buchstaben eines Absatzes.
- `:first-line` für die erste Zeile. Dies ergibt gerade in Verbindung mit `text-indent` interessante Effekte.

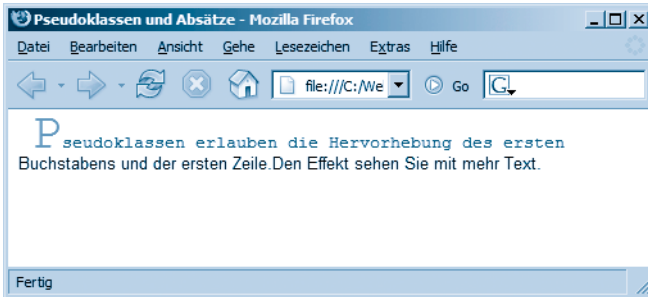
Hier ein Beispiel, das alle drei Befehle kombiniert. Allerdings interpretiert der Internet Explorer dies nicht korrekt:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- first_letter_line.html -->
<html>
<head>
 <title>Pseudoklassen und Absätze</title>
 <style type="text/css"><!--
 p {
 font-family: Arial, Geneva, Helvetica, sans-serif;
 font-size: 10pt;
 text-indent: 10px;
 }
 p:first-line {
 font-family: Courier, monospace;
 color: blue;
 }
 p:first-letter {
 font-size: 30pt;
 line-height: 10pt;
 color: red;
 }
 --></style>
</head>
<body>
 <p>Pseudoklassen erlauben die Hervorhebung des ersten
 Buchstabens und der ersten Zeile. Den Effekt sehen Sie
 mit mehr Text.</p>
</body>
</html>
```



### Tipp

*Für first-letter ist hier die Zeilenhöhe angegeben, da der Zeilenabstand durch den größeren Buchstaben sonst für die ganze Zeile zu hoch wird.*



## Links

Links bzw. Hyperlinks sind vermutlich die wichtigste Erfindung des Web. Nur durch die Möglichkeit, einzelne Dokumente miteinander zu verlinken, wird aus dem Web ein Netzwerk. Sie selbst werden Links an den verschiedensten Stellen benötigen:

- um auf externe Seiten zu verweisen
- um auf verschiedene HTML-Seiten innerhalb Ihrer eigenen Website zu verweisen
- um mittels Anker (bzw. Textmarken) auf Stellen innerhalb einer längeren Seite zu verweisen

Zuständig für Links ist das `<a>`-Tag. `a` steht hier für `anchor`. Das wichtigste Attribut ist `href`. Der Wert dieses Attributs ist das Linkziel. Der folgende Link verweist auf die Homepage vom Verlag Markt+Technik:

```
Markt+Technik
```

Im Browser sehen Sie das Linkziel in der Statusleiste. Wenn der Nutzer mit der Maus über einen Link fährt, ändert sich der Mauszeiger und wird zum Händchen. Standardmäßig ist ein Link blau und unterstrichen, dies können Sie aber selbst bestimmen. Mehr über die Arten von Links erfahren Sie in den nächsten Abschnitten.



## Absolut und relativ

Wenn Sie auf eine andere Website wie z.B. die von Markt+Technik verweisen, handelt es sich dabei um einen absoluten Link auf eine externe Seite. Die Domain identifiziert dabei die Website. Dort können Sie natürlich auch in untergeordnete Verzeichnisse verweisen:

`http://www.domain.de/verzeichnis/`

Oder Sie verweisen direkt auf eine Datei:

`http://www.domain.de/verzeichnis/datei.html`

Die Art der Datei ist dabei natürlich beliebig. Allerdings werden nur *.htm*- und *.html*-Dateien vom Browser direkt interpretiert. Bilder werden vom Browser dargestellt, wenn der Nutzer auf Links klickt.

Wenn Sie nicht auf eine externe Seite verweisen möchten, sondern Dateien innerhalb Ihrer eigenen Site verlinken möchten, gibt es hingegen die Möglichkeit, absolut oder relativ zu verlinken. Bei einem absoluten Link schreiben Sie an den Anfang einen Schrägstrich (/). Dies gibt an, dass es sich um das Wurzelverzeichnis der Website handelt.

Hier ein Beispiel:





Die Struktur besteht aus mehreren Ordnern, wovon *produkte* noch einen Unterordner hat. Nehmen Sie an, Sie wollten auf eine Datei *kontakt.html* innerhalb des Ordners *Kontakt* verlinken. Dies ginge absolut so:

```
/kontakt/kontakt.html
```

Auf ein Bild *bild.jpg* im Ordner *bilder*, der sich wiederum im Ordner *produkte* befindet, kommen Sie so:

```
/produkte/bilder/bild.jpg
```

Das Angenehme bei absoluten Links ist, dass Sie sich keine Gedanken machen müssen, wo die Datei liegt, aus der Sie verlinken. Dies ist bei relativen Links anders – hier ist entscheidend, woher gelinkt wird.

Sehen wir uns jetzt an, wie man relativ verlinkt. Als Beispiel dient wiederum die vorangegangene Struktur. Nehmen wir an, Sie wollten aus der Datei *kontakt.html* im Ordner *kontakt* auf *unternehmen.html* im Verzeichnis *unternehmen* verlinken, dann müssen Sie zuerst einen Ordner hoch in das Hauptverzeichnis. Dies geht mit `../`. Von dort aus müssen Sie in den Ordner *unternehmen* wechseln. Dann sieht der Pfad folgendermaßen aus:

```
../unternehmen/unternehmen.html
```

Wenn Sie relativ auf eine Datei im gleichen Verzeichnis linken möchten, schreiben Sie einfach den Dateinamen. Von *kontakt.html* zu *kontakt2.html*, die sich beide im selben Ordner befinden, kommen Sie also so:

```
kontakt2.html
```

Sie können aber auch mit `./` angeben, dass Sie vom aktuellen Verzeichnis ausgehen:

```
./kontakt2.html
```

Diese beiden Varianten funktionieren völlig gleichberechtigt.

### Tipp

*In der Praxis existieren absolute und relative Links gleichberechtigt. Was Sie für Ihre Website verwenden, ist eine strategische Entscheidung. Sie sollten auf absolute Links mit Domain-Name verzichten, falls sich die Domain ändert. So bleiben Sie flexibel. Ansonsten reagieren absolute Links ausgehend vom Wurzelverzeichnis etwas weniger sensibel auf Veränderungen. Relative Links sind dagegen oft gerade bei tieferen Verzeichnisstrukturen wesentlich weniger Tippaufwand.*

## Linkziel – Öffnen in anderem Fenster

Normalerweise öffnet sich die verlinkte Seite im aktuellen Fenster. Sie können aber schon in HTML ein anderes Fenster als Linkziel angeben. Dafür ist in HTML das `target`-Attribut zuständig. Es bietet folgende Möglichkeiten:

- `_self` öffnet den Link im aktuellen Fenster. Dies ist die Standardeinstellung und muss nicht angegeben werden.
- `_blank` öffnet den Link im neuen Fenster. Bei jedem Klick wird ein neues Fenster geöffnet.

```

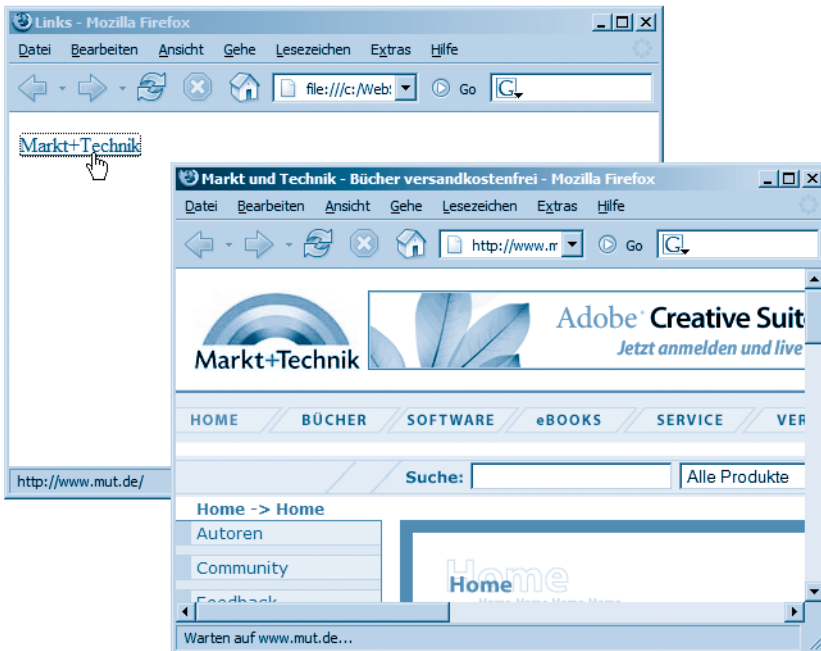
Markt+Technik
```

- `Name` öffnet den Link in einem Fenster, das Name heißt. Gibt es das Fenster nicht, wird ein neues mit diesem Namen geöffnet.

```

Markt+Technik
```

- `_top` und `_parent` sind nur im Umgang mit Frames relevant. Siehe hierzu *Kapitel 5*.



Wenn Sie eine Site per HTML in einem neuen Fenster öffnen, haben Sie keine Kontrolle über das Aussehen der Seite. Sie hat standardmäßig je nach Browser dieselbe Größe wie die Ausgangsseite und besitzt alle Merkmale wie Menüleiste etc., die die öffnende Seite auch besitzt. Wenn Sie hier mehr Kontrolle möchten, müssen Sie das neue Fenster per JavaScript öffnen. Dies wird z.B. bei Bildergalerien gemacht.

## Anker – Links innerhalb einer Seite

Wenn Sie innerhalb einer Seite verlinken, gibt es eine wichtige Besonderheit. Das Linkziel ist kein Dokument, sondern ein Anker – auch Textmarke genannt. Dieser Anker existiert nicht einfach, sondern Sie müssen ihn anlegen. Dies geht ebenfalls mit dem `<a>`-Tag.

Im folgenden Beispiel soll innerhalb einer HTML-Seite nach unten gesprungen werden. Hier die notwendigen Schritte:

**1** Zuerst benötigen Sie eine Datei mit längerem Text. Fehlt Ihnen diese, erzeugen Sie einfach per Copy&Paste eine Menge an Zeilenumbrüchen, damit das Dokument länger wird.

**2** Fügen Sie irgendwo in der Mitte des Dokuments eine Textmarke ein. Dazu vergeben Sie die `id` und das `name`-Attribut:

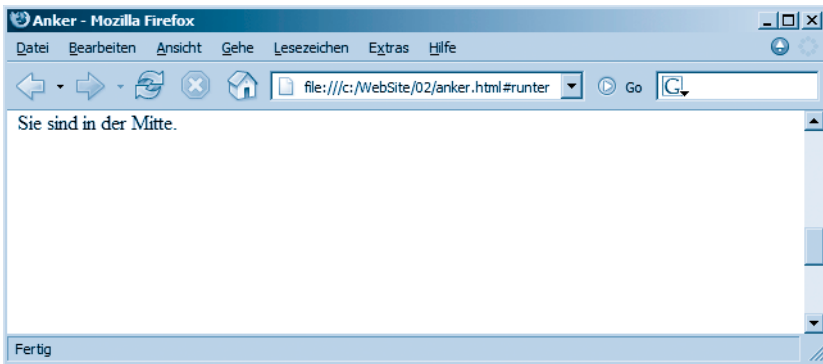
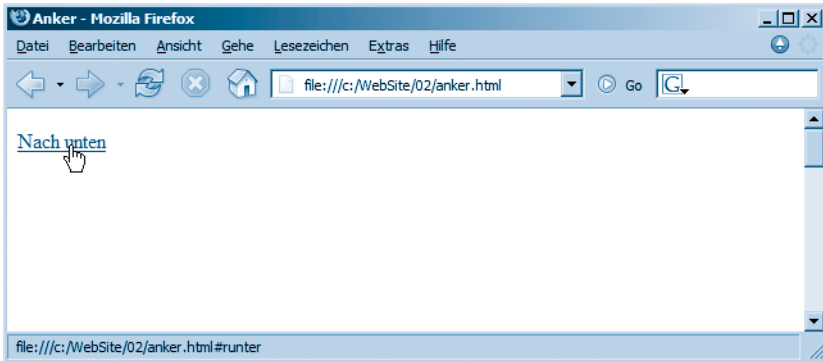
```
<p>Sie sind in der Mitte.
</p>
```

Früher kam nur das `name`-Attribut für Anker zum Einsatz. Im Zuge der XHTML-Kompatibilität sollte es nun eigentlich `id` sein. Um abwärtskompatibel zu bleiben, verwendet man aber mittlerweile beides.

**3** Nun fügen Sie oben in der Datei den Link nach unten auf den Anker ein. Tragen Sie dazu im `href`-Attribut den Namen des Ankers nach einem Doppelkreuz (Hash, `#`) ein:

```
<p>Nach unten</p>
```

Im Browser können Sie nun das Verhalten beobachten. Achten Sie dabei auf die Position des Scrollbalkens und auf den Ankernamen, der beim Klick auf den Link hinter den Dateinamen eingefügt wird. Der Anker selbst ist optisch nicht zu erkennen.



Sie können Anker auch mit relativen und absoluten Links auf andere Dateien verbinden. So haben Sie die Möglichkeit, direkt an eine Stelle in einem Dokument zu springen. Der folgende Link springt z.B. in die XHTML 1.0-Spezifikation an die Stelle, an der die Unterschiede zu HTML erklärt werden:

```

 XHTML-Unterschiede zu HTML
```

## E-Mail- und andere Links

Nur auf Webseiten zu linken ist auf Dauer nicht genug. Ab und an sind andere Linkarten notwendig. Die wichtigsten sind:

- E-Mail-Links auf eine Mailadresse
- FTP-Links auf einen FTP-Dateiserver
- Newsgroup-Links in eine Newsgroup
- Download-Links auf Dateien

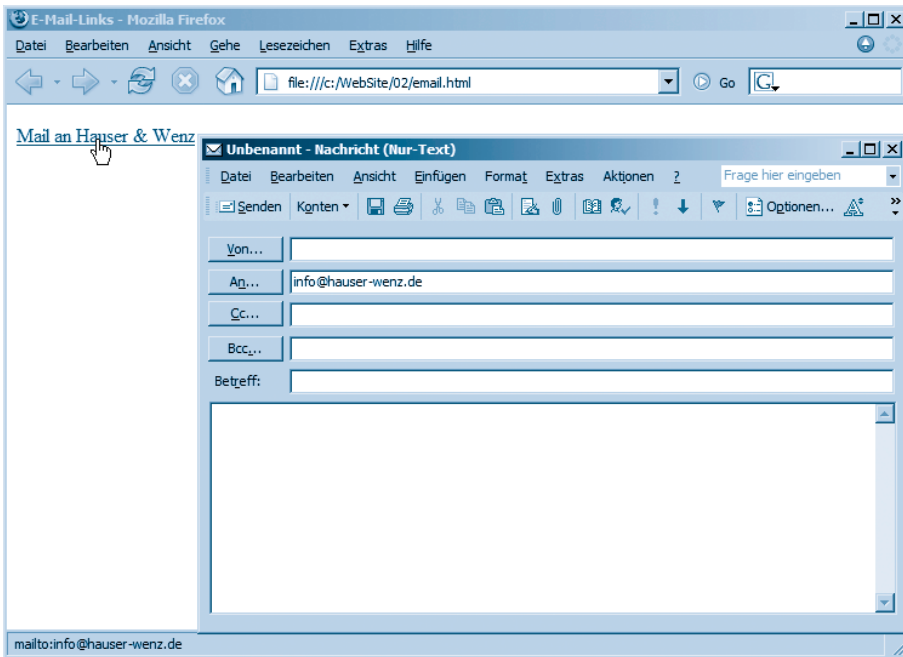


Wir zeigen Ihnen, wie die verschiedenen Arten funktionieren:

**E-Mail-Links** sind ein Herzstück zur Kontaktaufnahme mit Ihren Nutzern und sollten auf kaum einer Website fehlen. E-Mail-Links sind nicht mit Kontaktformularen zu verwechseln, die auch eventuell per E-Mail verschickt werden können, aber eher den Zweck haben, bestimmte Informationen vom Benutzer abzufragen. Zu Kontaktformularen lesen Sie *Kapitel 6*.

Wenn der Nutzer auf einen E-Mail-Link klickt, öffnet sich eine neue E-Mail-Nachricht im Standardmailprogramm, das der Nutzer im Browser eingestellt hat. Dazu müssen Sie nur vor die E-Mail-Adresse `mailto:` in das `href`-Attribut schreiben:

```
Mail an Hauser & Wenz
```



Wenn Sie die E-Mail an mehrere Empfänger schicken möchten, schreiben Sie diese durch Kommata getrennt hintereinander:

```
Mail an Hauser & Wenz
```

### Hinweis

*Ältere Mailprogramme von Microsoft haben eventuell Probleme mit den Kommata und wollen Strichpunkte, was aber wiederum andere Mailprogramme nicht goutieren. Hier ist eine vollständige Abdeckung Utopie.*

Neben der oder den Empfängern können Sie außerdem noch einige Optionen angeben. Sie folgen nach dem Empfänger durch ein Fragezeichen getrennt. Die einzelnen Optionen werden durch kaufmännische Und von-einander abgeteilt.

Dies ist zwar nicht spezifiziert, funktioniert aber mit so gut wie allen Mailprogrammen. Folgende Optionen stehen zur Verfügung:

- cc ergänzt einen Kopie-Empfänger.
- bcc fügt einen Blindkopie-Empfänger ein.
- subject fügt einen Betreff hinzu.
- body enthält einen Textvorschlag für die Mail.

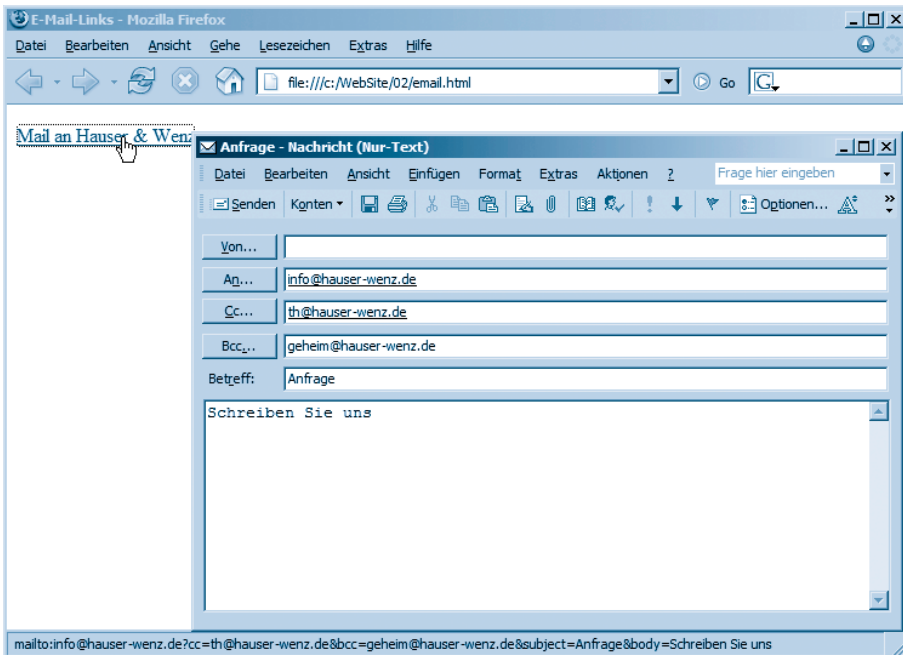
Hier ein Beispiel:

```
<p><a href="mailto:info@hauser-wenz.de?cc=th@hauser-wenz.de
&bcc=geheim@hauser-wenz.de&subject=Anfrage&body=Schreiben
Sie uns">Mail an Hauser & Wenz</p>
```

### Hinweis

*Sie fragen sich vielleicht, wieso Sie überhaupt noch ein Kontaktformular benötigen, wenn Sie E-Mail-Links so genau definieren können. Der Grund ist einfach: Sehr viele Nutzer haben eben kein Standardmailprogramm eingerichtet, verwenden Webmail oder surfen gerade nicht von ihrem eigenen Rechner. Sie haben beim Schicken von Mails ganz einfach Pech gehabt und benötigen deswegen unbedingt ein E-Mail-Formular.*





**FTP-Links** verwenden das FTP-Protokoll (File Transfer Protocol). Wenn Sie Ihre Webseiten bei Ihrem Hoster ins Web stellen, verwenden Sie beispielsweise FTP, um die Dateien zu übertragen. Sie können FTP aber z.B. auch dazu einsetzen, Dateien herunterzuladen. Ein FTP-Link beginnt statt `http://` mit `ftp://`.

```
FTP-Server
```

Sie können aber auch direkt auf eine Datei verlinken:

```
FTP-Server
```

Benötigt der FTP-Server Nutzernamen und Passwort, können Sie diese im Link mit angeben:

```
FTP-Server
```

Allerdings sollten Sie daran denken, dass diese Angaben im Klartext übertragen werden. Übrigens, dieselbe Art von Passwortangabe können Sie auch bei HTTP verwenden, wenn eine Seite auf HTTP-Authentifizierung setzt. Allerdings handelt es sich auch hier um einen Unsicherheitsfaktor. Für Logins und geschützte Bereiche ist HTTPS, die verschlüsselte Variante von HTTP, vorzuziehen.

**Newsgroup-Links** verweisen auf eine Newsgroup, ein Diskussionsforum. Hinter Newsgroups steckt das Usenet, bestehend aus verschiedenen News-Servern. Diese Technologie ist bereits sehr alt, deutlich älter als das Web selbst. Newsreader wie z.B. Outlook Express stellen die Newsgroups textlich dar.

Heutzutage werden Newsgroups vielerorts durch webbasierte Foren ersetzt. Oder aber Sie als Surfer verwenden beispielsweise Google mit dem Oberpunkt *Groups*, um auf Newsgroups zuzugreifen. Dennoch kann es mal vorkommen, dass Sie direkt auf eine Newsgroup verlinken möchten. Dies funktioniert mit `news:` und dann dem Namen der Newsgroup, der sich hierarchisch aus durch Punkte getrennten Einzelteilen aufbaut:

```

 Eine HTML-Newsgroup
```

**Download-Links** verweisen auf beliebige Dateien. Geben Sie dazu einfach den Dateinamen inklusive Endung an. Bei den meisten Dateitypen wird der Browser fragen, ob er die Datei speichern oder öffnen soll, so z.B. beim häufig eingesetzten PDF-Format. Allerdings können Sie z.B. nicht verhindern, dass er bei bekannten Grafikformaten das Bild darstellt. Sie haben hier also keinerlei Einfluss auf die Konfiguration des Nutzers, außer Sie bewegen sich in einem sehr restriktiven Intranet.

```
Ein PDF-Dokument
```

## Formatieren mit Pseudoklassen

Per se sieht ein Link nicht ausgesprochen hübsch aus: blau und unterstrichen. Wenn Sie die Seite dann besucht haben, wird der Link lila dargestellt. Daraus lässt sich schon ablesen, dass ein Link verschiedene Zustände kennt. Diese können Sie am komfortabelsten per CSS mit einigen Pseudoklassen formatieren.

An einem Beispiel erfahren Sie Schritt für Schritt, wie das funktioniert:

- 1 Legen Sie einen Link an.
- 2 Fügen Sie einen `<style>`-Block im Kopf der Seite hinzu.





**3** Ergänzen Sie dann einen Tag-Selektor für den normalen Linkzustand:

```
a:link {
}
```

**4** Fügen Sie hier alle Stilbefehle hinzu, die für alle Linkzustände gelten sollen. Entfernen Sie z.B. die Unterstreichung:

```
a:link {
 font-family: Arial, Geneva, Helvetica, sans-serif;
 font-size: 12pt;
 text-decoration: none;
 color: yellow;
}
```

**5** Wählen Sie für den besuchten Link eine andere Farbe:

```
a:visited {
 color: yellow;
}
```

Dies kann die ursprüngliche Linkfarbe sein, da es in der Praxis nicht mehr sehr üblich ist, besuchte Links andersfarbig darzustellen. Dadurch würde der Nutzer unter Umständen verwirrt.

**6** Als Letztes ergänzen Sie eine andere Formatierung für die `:hover`-Pseudoklasse:

```
a:hover {
 color: red;
 font-weight: bold;
}
```

`:hover` legt die Formatierung fest, in der der Link erscheint, wenn der Nutzer mit der Maus darüberfährt.

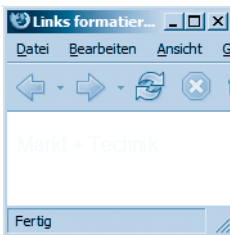
**7** Vergeben Sie für den aktiven (also angeklickten) Link eine andere Farbe:

```
a:active {
 color: red;
}
```

Die Pseudoklasse `:active` klappt nur für das `<a>`-Tag und muss nach `:hover` vergeben werden.

Hier noch einmal das Beispiel im Überblick:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- link_formatieren.html -->
<html>
<head>
 <title>Links formatieren</title>
 <style type="text/css"><!--
 a {
 font-family: Arial, Geneva, Helvetica, sans-serif;
 font-size: 12pt;
 text-decoration: none;
 color: yellow;
 }
 a:visited {
 color: yellow;
 }
 a:hover {
 color: red;
 font-weight: bold;
 }
 a:active {
 color: red;
 }
 --></style>
</head>
<body>
 <p>Markt+Technik</p>
</body>
</html>
```



**Hinweis**

*Ohne die `:hover`-Pseudoklasse klappt ein Rollover-Effekt nur mit Hilfe von JavaScript. Dazu wird in JavaScript eine Grafik mit einer anderen ausgetauscht. Hier sind die optischen Möglichkeiten noch größer, da z.B. ein grafischer Button abgebildet werden kann.*

## Nutzloses Wissen

Äpfel, Zwiebeln und Kartoffeln schmecken weitestgehend gleich; der Unterschied kommt hauptsächlich vom Geruch.

## Kapitel 3

# Strukturierte Daten – Listen und Tabellen



The screenshot shows a Mozilla Firefox browser window titled 'Tabelle - Mozilla Firefox'. The address bar is empty, and the page content displays a table with two columns: 'Obst' and 'Preis pro Kilo'. The table has two rows of data: 'Äpfel' with a price of '2 €' and 'Kirschen' with a price of '4 €'. The browser's menu bar includes 'Datei', 'Bearbeiten', 'Ansicht', 'Gehe', 'Lesezeichen', and 'Extras'. The status bar at the bottom shows 'Fertig'.

| Obst     | Preis pro Kilo |
|----------|----------------|
| Äpfel    | 2 €            |
| Kirschen | 4 €            |

*Daten spielen auf einer Website eine wichtige Rolle. Egal ob Terminliste, ein Kochrezept oder herunterladbare Dateien: Eine Fülle an Informationen will formatiert und benutzerfreundlich aufbereitet werden.*

*HTML bietet dafür zwei Möglichkeiten, die Sie auch aus der Textverarbeitung kennen: Listen für Aufzählungen jeder Art und Tabellen für komplexere Daten.*

**Das können Sie schon:**

|                                |    |
|--------------------------------|----|
| HTML-Grundgerüst einer Website | 26 |
| Einsatz von CSS                | 40 |
| Formatierung von Text          | 56 |
| Schriften und Schriftgrößen    | 56 |
| Einsatz von Links              | 75 |
| Links gestalten                | 76 |

**Das lernen Sie neu:**

|                                          |     |
|------------------------------------------|-----|
| Aufzählungslisten                        | 90  |
| Listen mit Nummerierung                  | 92  |
| Tabellen zur Datenstrukturierung         | 95  |
| Tabellen formatieren                     | 99  |
| Zellen verbinden und Tabellen gruppieren | 109 |

## Listen

Eine Liste ist eine Sammlung mehrerer Elemente. HTML unterscheidet ungeordnete und geordnete Listen. Ungeordnete Listen arbeiten mit Aufzählungszeichen, geordnete mit Nummerierung.

### Tipp

*Mit CSS können Sie zwar eigene grafische Aufzählungszeichen verwenden. Allerdings sind diese Möglichkeiten noch nicht in allen Browsern integriert. Deswegen werden hier oft noch unsichtbare Tabellen verwendet (siehe Kapitel 7).*

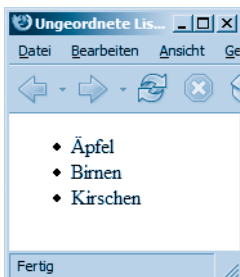
## Aufzählung

Eine ungeordnete Liste wird in HTML mit den `<ul>`-Tags erstellt. `ul` steht für ungeordnete Liste (engl. »unordered list«). Nun müssen Sie noch jedes Listenelement kennzeichnen. Dies geschieht über das `<li>`-Tag:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- liste.html -->
<html>
<head>
 <title>Ungeordnete Liste</title>
</head>
<body>

 Äpfel
 Birnen
 Kirschen

</body>
</html>
```



Eine Liste kann beliebig viele Listenelemente enthalten. Sie können allerdings auch beliebig viele Listen ineinander verschachteln. Dazu schreiben Sie eine neue Liste einfach in eine bestehende. Die verschachtelte Liste steht dabei in `<li>`-Tags.

Hier ein Beispiel:

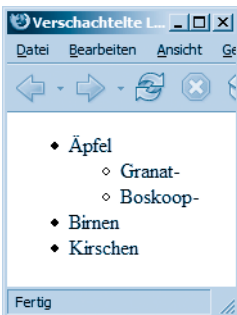
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- liste_verschachtelt.html -->
<html>
<head>
 <title>Verschachtelte Liste</title>
</head>
<body>

 Äpfel

 Granat-
 Boskoop-

 Birnen
 Kirschen

</body>
</html>
```



Bei verschachtelten Listen verwendet der Browser schon automatisch verschiedene **Aufzählungszeichen**. Mit dem `type`-Attribut haben Sie hier allerdings auch selbst die Wahl. Einzige Einschränkung sind ältere Browser, die nicht immer mitspielen.

Folgende Aufzählungszeichen bietet HTML:

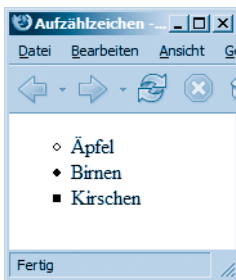
- `circle` ist der standardmäßig verwendete Kreis bzw. in manchen Browsern eine Raute.
- `disc` ist der Kreis bzw. eine Raute ohne Füllung.
- `square` ist ein gefülltes Rechteck.

Hier ein Beispiel:

```

 <li type="circle">Äl;pfel
 <li type="disc">Birnen
 <li type="square">Kirschen

```



## Nummerierte Liste

Eine nummerierte Liste verwendet genau wie die Aufzählung `<li>` für Listenelemente. Einzig das umgebende Tag heißt `<ol>` (für »ordered list«) statt `<ul>`.

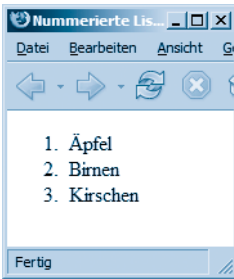
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- liste_nummeriert.html -->
<html>
<head>
 <title>Nummerierte Liste</title>
</head>
<body>

 Äl;pfel
 Birnen
 Kirschen

</body>
</html>
```







Auch nummerierte Listen können Sie verschachteln oder z.B. auch mit ungeordneten Listen mischen. Hier ein Beispiel:

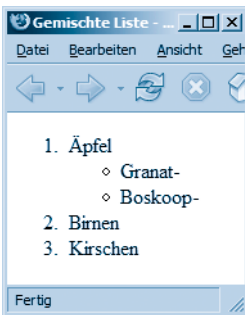
```

 Äl;pfe'l

 Granat-
 Boskoop-

 Birnen
 Kirschen

```



Bei einer nummerierten Liste können Sie die Art der Nummerierung ändern. Hierzu dient wie bei Aufzählungslisten das `type`-Attribut. Folgende Werte stehen zur Auswahl:

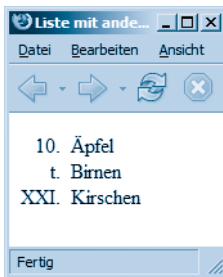
- 1 nummeriert mit arabischen Ziffern. Dies ist der Standardwert.
- a nummeriert mit Kleinbuchstaben.
- A nummeriert mit Großbuchstaben.
- i nummeriert in kleinen römischen Buchstaben.
- I nummeriert in großen römischen Buchstaben.

Mit dem Attribut `start` vergeben Sie im `<ol>`-Tag, bei welcher Nummer die Liste beginnt. Wenn Sie einen neuen Startpunkt für ein Listenelement definieren möchten, verwenden Sie das `value`-Attribut im `<li>`-Tag.

Hier ein Beispiel, das alle drei Möglichkeiten wild mischt:

```
<ol start="10">
 Äpfel
 <li type="a" value="20">Birnen
 <li type="I">Kirschen

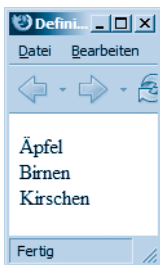
```



## Definitionsliste

Eine Definitionsliste besitzt Einrückungen, aber keine Aufzählungszeichen oder Nummern. Sie ist in ein `<dl>`-Tag (für definition list) eingeschlossen. Jedes einzelne Listenelement ist ein `<dt>`-Tag (für definition term):

```
<dl>
 <dt>Äpfel</dt>
 <dt>Birnen</dt>
 <dt>Kirschen</dt>
</dl>
```



## Formatieren

Die Listenelemente erben die Schriftformatierungen. Das heißt z.B., die Schrift, die Sie für die ganze Seite definieren, wird auch von der Liste übernommen. Ansonsten können Sie den `<li>`-Tags oder in verschachtelter Form (siehe *Kapitel 1*) auch einzelnen Listen eigene Stile zuweisen. Natürlich können Sie auch CSS-Klassen und alle CSS-Befehle zur Textformatierung mit Listen verwenden.

### Hinweis

*CSS bietet außerdem einige Formate für Listenelemente. Allerdings klappen diese noch nicht in allzu vielen Browsern, manche sogar nur im Opera. Deswegen werden sie in der Praxis nicht allzu häufig eingesetzt. Ab und an zum Einsatz kommt `list-style-image` für ein grafisches Aufzählungszeichen:*

```
<ul style="list-style-image: url(bild.jpg);">
 Element 1
 Element 2
 Element 3

```

## Tabellen

Tabellen spielen auf Webseiten in zweifacher Hinsicht eine Rolle: zum einen um Daten zu präsentieren. Zum anderen werden »unsichtbare« Tabellen auch zweckentfremdet, um Inhalte auf einer Website zu positionieren. Letzteres ist Teil von *Kapitel 7*. Allerdings benötigen Sie zum Verständnis die Grundlagen von Tabellen aus diesem Kapitel.

### Aufbau

Stellen Sie sich eine Tabelle in Gedanken vor. Sie besteht aus zwei Dimensionen: Zeilen und Spalten. Wo sich eine Zeile und eine Spalte treffen, liegt jeweils eine Zelle.

HTML muss diese Tabelle in einer Beschreibungssprache abbilden. Dies erfolgt komplett zeilenorientiert. Das heißt, Sie legen eine Zeile an und definieren dann darin die Zellen. Die Spalten entstehen so automatisch.

Für diese Technik sind folgende Tags notwendig:

- `<table>` für die Tabelle selbst.
- `<tr>` für eine Reihe bzw. Zeile. `tr` steht für `table row`.
- `<td>` für eine Zelle. `td` steht für `table data cell`.
- `<th>` für eine Zelle, die eine Überschrift enthält. `th` steht für `table header cell`.

Reihen landen immer in der Tabelle, Zellen dann in Reihen. Ein Beispiel macht das Theoretische verständlicher. Erstellt werden soll eine Tabelle mit drei Reihen und zwei Spalten:

**1** Geben Sie zuerst mit dem `<table>`-Tag an, dass es sich um eine Tabelle handelt:

```
<table>
</table>
```

**2** Standardmäßig haben Tabellen keine Rahmenlinien und sind deswegen unsichtbar. Machen Sie die Tabelle mit dem Attribut `border` und einer Rahmendicke von 1 Pixel sichtbar:

```
<table border="1">
</table>
```

**3** Erzeugen Sie dann mit dem `<tr>`-Tag die erste Zeile:

```
<table border="1">
 <tr>
</tr>
</table>
```

**4** Fügen Sie dann in die Zeile die zwei Zellen ein. Da die Zellen eine Überschrift darstellen sollen, verwenden Sie `<th>`-Tags:

```
<table border="1">
 <tr>
 <th>Obst</th>
 <th>Preis pro Kilo</th>
 </tr>
</table>
```

Im Gegensatz zu normalen Zellen mit `<td>` erzeugt `<th>` Zellen mit automatisch fett gedrucktem Text und zentriertem Inhalt.

**5** Ergänzen Sie nun noch zwei Zeilen mit jeweils zwei normalen `<td>`-Zellen.



Hier sehen Sie den fertigen Code in der Übersicht:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- tabelle.html -->
<html>
<head>
 <title>Tabelle</title>
</head>
<body>
 <table border="1">
 <tr>
 <th>Obst</th>
 <th>Preis pro Kilo</th>
 </tr>
 <tr>
 <td>Äpfel</td>
 <td>2 €</td>
 </tr>
 <tr>
 <td>Kirschen</td>
 <td>4 €</td>
 </tr>
 </table>
</body>
</html>
```



## Tabellen verschachteln

Tabellen lassen sich beliebig verschachteln. Dazu schreiben Sie einfach eine Tabelle in eine Zelle. Der Code bleibt einfach, wird aber schnell sehr lang und unübersichtlich. Hier ist die verschachtelte Tabelle hervorgehoben:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- tabelle_verschachtelt.html -->
<html>
<head>
 <title>Tabelle</title>
</head>
<body>
 <table border="1">
 <tr>
 <th>Obst</th>
 <th>Preis pro Kilo</th>
 </tr>
 <tr>
 <td>Äpfel</td>
 <td>
 <table border="1">
 <tr>
 <td>Granat-</td>
 <td>2 €</td>
 </tr>
 <tr>
 <td>Boskoop-</td>
 <td>4 €</td>
 </tr>
 </table>
 </td>
 </tr>
 <tr>
 <td>Kirschen</td>
 <td>4 €</td>
 </tr>
 </table>
</body>
</html>

```



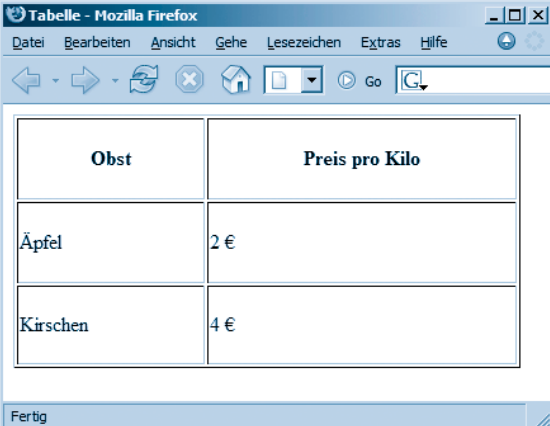
**Tipp**

*Bei komplexeren Tabellen ist es sinnvoll, einen WYSIWYG-Editor wie Macromedia Dreamweaver oder Microsoft FrontPage zu verwenden. Denn damit fällt es doch leichter, den Überblick zu wahren.*

## Tabellenmaße

Normalerweise richtet sich die Größe einer Tabelle nach der Größe des Inhalts und der Größe des Fensters. Sie können allerdings selbst mit den Attributen `width` und `height` Größenmaße angeben. Dies ist zwar nicht vollständig standardkonform, funktioniert aber im Browser sehr gut. Als Werte sind absolute Angaben in Pixel und Prozent in Abhängigkeit von der Fenstergröße möglich.

```
<table border="1" width="400" height="200">
 <tr>
 <th>Obst</th>
 <th>Preis pro Kilo</th>
 </tr>
 <tr>
 <td>Äpfel</td>
 <td>2 €</td>
 </tr>
 <tr>
 <td>Kirschen</td>
 <td>4 €</td>
 </tr>
</table>
```



| Obst     | Preis pro Kilo |
|----------|----------------|
| Äpfel    | 2 €            |
| Kirschen | 4 €            |

### Hinweis

*Die Größenangaben sind eine Minimalgröße für die Zellen. Sollte mehr Inhalt in den Zellen stehen, werden sie entsprechend größer, wenn sowohl Breite als auch Höhe gegeben sind.*

Sie können die Größe aber nicht nur für die Tabelle festlegen, sondern auch für die Zeile oder für einzelne Zellen. Dabei gilt es allerdings, einige Regeln zu beachten:

- Die Breite einer Zelle ist die Breite einer Spalte. Deswegen gibt man selten in der Zeile Größenmaße an und wenn, dann nur die Höhe.
- Eine Zeile ist immer so hoch wie die höchste Zelle.
- Die Breite einer Zelle entspricht der Breite der zugehörigen Spalte. Die Spalte ist immer so breit wie ihre breiteste Zelle.
- Alle Zellen zusammengenommen sollten nicht breiter sein als die Tabelle.

### Tipp

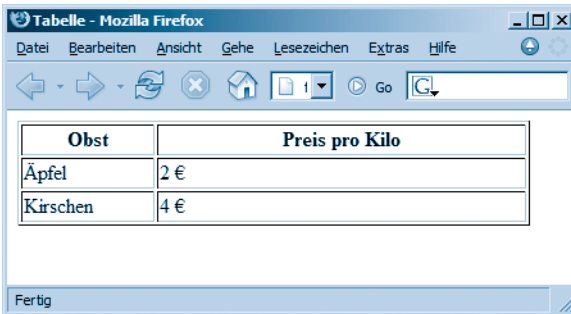
*Da diese Regeln in der Praxis gar nicht so einfach zu beherzigen sind, sollten Sie Ihre Tabelle eventuell erst auf einem Blatt Papier vorzeichnen und sie erst dann in HTML umsetzen.*

Hier eine Tabelle mit einigen Detailangaben:

```
<table border="1" width="400">
 <tr>
 <th width="100">Obst</th>
 <th>Preis pro Kilo</th>
 </tr>
 <tr>
 <td>Äpfel</td>
 <td>2 €</td>
 </tr>
 <tr>
 <td>Kirschen</td>
 <td>4 €</td>
 </tr>
</table>
```







The screenshot shows a Mozilla Firefox browser window titled 'Tabelle - Mozilla Firefox'. The address bar shows a local file path. The table displayed is as follows:

| Obst     | Preis pro Kilo |
|----------|----------------|
| Äpfel    | 2 €            |
| Kirschen | 4 €            |

At the bottom of the browser window, the status bar shows 'Fertig'.

### Hinweis

Die Maßangaben für Tabellen lassen sich auch mit den CSS-Befehlen *height* und *width* treffen. Allerdings sind hier die HTML-Angaben aus Gründen der Abwärtskompatibilität noch deutlich üblicher. Abgesehen davon gibt es im CSS-Standard z.B. auch Befehle für die *minimale Breite und Höhe* (*min-width* und *min-height*). Allerdings ist die Browserunterstützung hier vor allem im Internet Explorer noch nicht gegeben.

## Abstände und Rahmen

Wenn Sie eine Tabelle erstellen, kleben die Inhalte automatisch am Zellenrand und die Zellen sind direkt aneinander gepappt. Hier können Sie mit zwei HTML-Attributen für Abstand sorgen:

- `cellspacing` sorgt für Abstand zwischen Zellen.
- `cellpadding` entfernt den Inhalt vom Zellenrand.

Beide Attribute erwarten als Werte eine Angabe in Pixeln. Die Wirkung sehen Sie, wenn Sie mit `border` einen Rahmen angegeben haben:

```
<table border="1" width="300" cellpadding="10"
 cellspacing="10">
 <tr>
 <th width="100">Obst</th>
 <th>Preis pro Kilo</th>
 </tr>
 <tr>
 <td>Äpfel</td>
 <td>2 €</td>
 </tr>
```

```

<tr>
 <td>Kirschen</td>
 <td>4 €</td>
</tr>
</table>

```



### Hinweis

*Die Abstände einer Tabelle nach außen können Sie am besten über den CSS-Befehl `margin` steuern. Die HTML-Entsprechung wären die weniger flexiblen Befehle `hspace` und `vspace` für den horizontalen und vertikalen Abstand. Sie sind, wenn überhaupt, aber eher bei Bildern üblich (siehe Kapitel 4).*

Den Tabellenrahmen selbst können Sie in HTML nur wenig steuern. Zwar erlaubt das Attribut `bordercolor` die Angabe einer Rahmenfarbe, es gehört aber nicht zur offiziellen Spezifikation und wird im Opera nicht unterstützt.

### Tipp

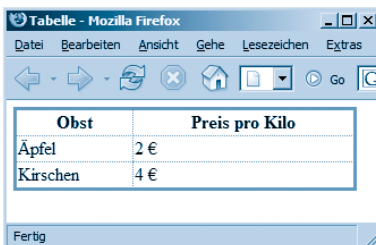
*Leere Zellen werden standardmäßig als Rahmen ohne eingepprägten Inhaltsbereich dargestellt. Wenn Sie dies verhindern möchten, fügen Sie ein geschütztes Leerzeichen (`&nbsp;`) ein.*

Statt des HTML-Rahmens können Sie auch einen CSS-Rahmen für Tabellen verwenden. Der wichtigste Unterschied ist, dass Sie in CSS den Rahmen für die Tabelle (`<table>`-Tag) und die Zellen (`<td>` und `<th>`) getrennt vergeben können. Sie haben sogar die Möglichkeit, mit `border-collapse` zu steuern, ob beide zusammenfallen (`collapse`) oder nicht (`separate`, Standardwert).



Hier ein Beispiel:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- tabelle_rahmen.html -->
<html>
<head>
 <title>Tabelle</title>
 <style type="text/css"><!--
 table, th, td {
 border-width: 3px;
 border-color: red;
 border-style: solid;
 border-collapse: collapse;
 }
 th, td {
 border-width: 1px;
 border-style: dotted;
 }
 --></style>
</head>
<body>
 <table width="300">
 <tr>
 <th width="100">Obst</th>
 <th>Preis pro Kilo</th>
 </tr>
 <tr>
 <td>Äpfel</td>
 <td>2 €</td>
 </tr>
 <tr>
 <td>Kirschen</td>
 <td>4 €</td>
 </tr>
 </table>
</body>
</html>
```



The screenshot shows a Mozilla Firefox browser window titled 'Tabelle - Mozilla Firefox'. The address bar is empty. The table displayed in the browser window is as follows:

| Obst     | Preis pro Kilo |
|----------|----------------|
| Äpfel    | 2 €            |
| Kirschen | 4 €            |

At the bottom of the browser window, the status bar shows 'Fertig'.

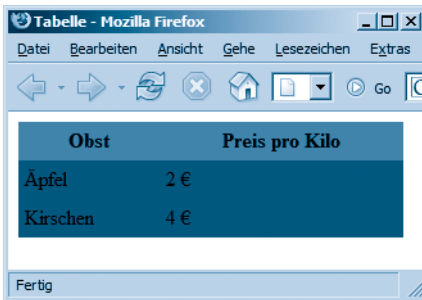
## Hintergründe

Per CSS können Sie Tabellen und Tabellenzellen auch sehr einfach mit Hintergrundfarben und Hintergrundbildern versehen. Der wichtigste Befehl ist `background-color`, mit dem Sie eine Hintergrundfarbe festlegen. Die weiteren Befehle für Hintergrundbilder lernen Sie in *Kapitel 4* näher kennen.

Hier ein Beispiel, das für Kopfzellen eine andere Hintergrundfarbe festlegt als für normale Zellen. Dadurch entsteht schon eine optische Trennung in der Tabelle:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- tabelle_hintergrund.html -->
<html>
<head>
 <title>Tabelle</title>
 <style type="text/css"><!--
 th {
 background-color: green;
 }
 td {
 background-color: blue;
 }
 --></style>
</head>
<body>
 <table width="300" border="0" cellspacing="0"
 cellpadding="5">
 <tr>
 <th width="100">Obst</th>
 <th>Preis pro Kilo</th>
 </tr>
 <tr>
 <td>Äpfel</td>
 <td>2 €</td>
 </tr>
 <tr>
 <td>Kirschen</td>
 <td>4 €</td>
 </tr>
 </table>
</body>
</html>
```





The screenshot shows a Mozilla Firefox browser window titled 'Tabelle - Mozilla Firefox'. The address bar is empty. The menu bar includes 'Datei', 'Bearbeiten', 'Ansicht', 'Gehe', 'Lesezeichen', and 'Extras'. The toolbar contains navigation buttons (back, forward, home, stop, reload) and a search bar. The main content area displays a table with two columns: 'Obst' and 'Preis pro Kilo'. The table has two data rows: 'Äpfel' with a price of '2 €' and 'Kirschen' with a price of '4 €'. The status bar at the bottom shows 'Fertig'.

| Obst     | Preis pro Kilo |
|----------|----------------|
| Äpfel    | 2 €            |
| Kirschen | 4 €            |

## Inhalte ausrichten

Bisher haben wir bei den Zellen die Standardausrichtung verwendet. Bei Kopfzellen sind die Inhalte horizontal und vertikal zentriert, bei normalen Zellen linksbündig und vertikal zentriert. Beides lässt sich für jede Zelle oder die Zeile ändern. Dazu verwenden Sie die Attribute `align` und `valign` (für vertical align) oder alternativ die CSS-Befehle `text-align` und `vertical-align`.

### Hinweis

*Wenn Sie `align` im `<table>`-Tag verwenden, richten Sie damit nicht den Tabelleninhalt, sondern die Tabelle auf der Seite aus. Mit `align="center"` können Sie eine Tabelle z.B. zentrieren.*

`align` beziehungsweise `text-align` kann die folgenden Werte annehmen:

- `left` für linksbündig. Dies ist der Standardwert bei normalen Zellen.
- `center` für zentriert. Das ist der Standardwert für Kopfzellen.
- `right` für rechtsbündig.

`valign` bzw. `vertical-align` kennt folgende wichtige Werte:

- `top` richtet an der Oberkante der Zelle aus.
- `middle` zentriert vertikal.
- `bottom` richtet an der Unterkante der Zelle aus.
- `baseline` richtet die oberste Zeile des Texts auf der niedrigstmöglichen Linie aus.

Hier ein Beispiel, das horizontale und vertikale Ausrichtung kombiniert:

```
<table width="300" height="200" border="1">
 <tr valign="bottom">
 <th>Obst</th>
 <th>Preis pro Kilo</th>
 </tr>
 <tr>
 <td align="center" valign="top">Äpfel</td>
 <td align="center" valign="top">2 €</td>
 </tr>
 <tr style="text-align: center;
 vertical-align: top;">
 <td>Kirschen</td>
 <td>4 €</td>
 </tr>
</table>
```



| Obst     | Preis pro Kilo |
|----------|----------------|
| Äpfel    | 2 €            |
| Kirschen | 4 €            |

## Formatieren

In diesem Abschnitt soll ein aufwändigeres Beispiel Einzelschritte aus den letzten Abschnitten zusammenfassen und das Ganze mit dem Formatieren von textlichen Inhalten verbinden.

Um die Texte per CSS in Tabellen zu formatieren, gibt es mehrere Möglichkeiten:

- Sie greifen mit einem Tag-Selektor auf die Zellen zu.
- Sie verwenden Klassen oder IDs für die Tabelle oder einzelne Zellen.



**Hinweis**

*Der Netscape Navigator 4.x versteht nur CSS-Formatierungen, die direkt einer Zelle zugewiesen werden.*

Gehen wir nun Schritt für Schritt ein Beispiel durch. Ausgangspunkt ist dabei eine einfache Tabelle, bestehend aus zwei Spalten und drei Zeilen:

```
<table width="300">
 <tr>
 <th width="100">Obst</th>
 <th>Preis pro Kilo</th>
 </tr>
 <tr>
 <td>Äpfel</td>
 <td>2 €</td>
 </tr>
 <tr>
 <td>Kirschen</td>
 <td>4 €</td>
 </tr>
</table>
```

Zuerst müssen Sie sich entscheiden, wie die Tabelle am Schluss ungefähr aussehen soll. Hier wollen wir ohne Rahmen aber dafür mit Hintergrundfarbe arbeiten. Als Schriftart soll eine serifenlose Schrift zum Einsatz kommen. Hier die benötigten Schritte:

**1** Entfernen Sie den Rahmen (`border`) um die Tabelle und die Abstände zwischen den Zellen (`cellspacing`). Fügen Sie 5 Pixel Abstand des Inhalts vom Zellenrand ein (`cellpadding`).

```
<table width="300" border="0" cellspacing="0"
 cellpadding="5">
```

**2** Fügen Sie im Kopf der Seite einen `<style>`-Bereich hinzu.

**3** Vergeben Sie für Kopfzellen eine Hintergrundfarbe:

```
th {
 background-color: yellow;
}
```

**4** Vergeben Sie für normale Zellen eine andere Hintergrundfarbe:

```
td {
 background-color: red;
}
```

**5** Legen Sie dann für Kopf- und normale Zellen Schriftfarbe und Schriftgröße fest:

```
th, td {
 font-family: Arial, Geneva, Helvetica, sans-serif;
 font-size: 12pt;
}
```

**6** Nun ändern Sie noch die Ausrichtung der normalen Zellen. Fügen Sie einen Abstand von links für die erste Spalte ein und zentrieren Sie die Inhalte in der zweiten Spalte. Diese Änderung müssen Sie in allen Inhaltszeilen vornehmen:

```
<tr>
 <td style="padding-left: 20px;">Äl;pfel</td>
 <td style="text-align: center;">2 €</td>
</tr>
<tr>
 <td style="padding-left: 20px;">Kirschen</td>
 <td style="text-align: center;">4 €</td>
</tr>
```

Hier der fertige Code in der Übersicht:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- tabelle_formatieren.html -->
<html>
<head>
 <title>Tabelle</title>
 <style type="text/css"><!--
 th {
 background-color: yellow;
 }
 td {
 background-color: red;
 }
 th, td {
 font-family: Arial, Geneva, Helvetica, sans-serif;
 font-size: 12pt;
 }
 --></style>
</head>
```





```

<body>
<table width="300" border="0" cellspacing="0"
 cellpadding="5">
 <tr>
 <th width="100">Obst</th>
 <th>Preis pro Kilo</th>
 </tr>
 <tr>
 <td style="padding-left: 20px;">Äpfel</td>
 <td style="text-align: center;">2 €</td>
 </tr>
 <tr>
 <td style="padding-left: 20px;">Kirschen</td>
 <td style="text-align: center;">4 €</td>
 </tr>
</table>
</body>
</html>

```



The screenshot shows a Mozilla Firefox browser window titled 'Tabelle - Mozilla Firefox'. The address bar is empty. The menu bar includes 'Datei', 'Bearbeiten', 'Ansicht', 'Gehe', 'Lesezeichen', and 'Extras'. The toolbar contains navigation buttons (back, forward, home, stop, reload) and a search bar. The main content area displays a table with two columns: 'Obst' and 'Preis pro Kilo'. The table has two rows: 'Äpfel' with a price of '2 €' and 'Kirschen' with a price of '4 €'. The status bar at the bottom shows 'Fertig'.

| Obst     | Preis pro Kilo |
|----------|----------------|
| Äpfel    | 2 €            |
| Kirschen | 4 €            |

## Zellen verbinden

Alle Zeilen einer Tabelle müssen gleich viele Zellen haben. Ansonsten stimmt die Tabelle nicht mehr. Allerdings können Sie in einer HTML-Tabelle auch mehrere Zellen zu einer zusammenfassen. Horizontal, also spaltenübergreifend, klappt das mit `colspan`, vertikal, dementsprechend reihenübergreifend, mit `rowspan`. Angegeben wird jeweils die Zahl der zusammengefassten Zellen **inklusive** der Ursprungszelle.

### Hinweis

*Achten Sie immer darauf, dass die Zahl der Zellen auch nach dem Verbinden wirklich stimmt. Sonst haben Sie sehr schnell Chaos in Ihrer Tabelle. Dies gilt vor allem, wenn Sie nachträglich neue Zellen hinzufügen.*

Wenn Sie eine Zelle mit einer anderen verbinden, verschwindet eine der beiden Zellen aus der Tabelle. Wir zeigen Ihnen dies anhand eines Beispiels. Gegeben ist folgende Ausgangstabelle:

```
<table width="300" border="1">
 <tr>
 <th width="100">Obst</th>
 <th>Preis pro Kilo</th>
 </tr>
 <tr>
 <td>Äpfel</td>
 <td>2 €</td>
 </tr>
 <tr>
 <td>Kirschen</td>
 <td>4 €</td>
 </tr>
</table>
```

Sie soll nun eine neue Zeile erhalten, die über zwei Spalten geht und einen Copyright-Vermerk enthält. Und so geht es:

**1** Sie ergänzen zuerst die neue Zeile.

```
<tr>
</tr>
```

**2** Anschließend folgt die gewünschte Zelle mit dem Copyright-Vermerk:

```
<tr>
 <td>© Obsthandel Biliger Jakob</td>
</tr>
```

**3** Da sie zwei Spalten überspannen soll, erhält Sie das Attribut `colspan="2"`:

```
<tr>
 <td colspan="2">© Obsthandel Billiger Jakob</td>
</tr>
```

Fertig ist der Code. Sie benötigen keine zweite Zelle in dieser Zeile. Oder drastischer formuliert: Eine weitere Zelle in dieser Zeile wäre die dritte und damit eine zu viel.



The screenshot shows a Mozilla Firefox browser window titled 'Tabelle - Mozilla Firefox'. The address bar is empty. The menu bar includes 'Datei', 'Bearbeiten', 'Ansicht', 'Gehe', 'Lesezeichen', and 'Extras'. The toolbar contains navigation buttons (back, forward, home, stop, reload) and a search bar. The main content area displays a table with two columns: 'Obst' and 'Preis pro Kilo'. The table has three data rows: 'Äpfel' with '2 €', 'Kirschen' with '4 €', and a footer row with '© Obsthandel Büliger Jakob'. The status bar at the bottom shows 'Fertig'.

| Obst                       | Preis pro Kilo |
|----------------------------|----------------|
| Äpfel                      | 2 €            |
| Kirschen                   | 4 €            |
| © Obsthandel Büliger Jakob |                |

## Tabellen gliedern

Sie können Tabellen in einen Kopf- (<thead></thead>), einen Fuß- (<tfoot></tfoot>) und einen oder mehrere Mittelbereiche (<tbody></tbody>) unterteilen. Dies ist erst einmal nur eine rein logische Strukturierung und hat keine optischen Konsequenzen. Sie haben allerdings Möglichkeiten, das zu ändern:

- Sie formatieren mit CSS die einzelnen Bereiche. Das Tag für den Bereich dient dabei als Tag-Selektor:

```
tbody {
 background-color: marine;
}
```

- Sie verwenden das Attribut `rules` für die Tabelle. Es steuert, wie der Tabellenrahmen angezeigt wird. Mit dem Wert `groups` orientiert sich der Rahmen an den Tabellenbereichen. Alternative Werte sind `rows` für nur Zeilen und `cols` für nur Spalten.

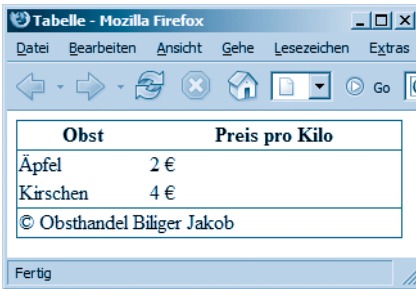
Hier ein Beispiel mit den Bereichen und `rules="groups"`:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- tabelle_gruppieren.html -->
<html>
<head>
 <title>Tabelle</title>
</head>
<body>
 <table width="300" border="1" rules="groups">
 <thead>
 <tr>
 <th width="100">Obst</th>
 <th>Preis pro Kilo</th>
 </tr>
```

```

</thead>
<tfoot>
 <tr>
 <td colspan="2">© Obsthandel Billiger Jakob</td>
 </tr>
</tfoot>
<tbody>
 <tr>
 <td>Äpfel</td>
 <td>2 €</td>
 </tr>
 <tr>
 <td>Kirschen</td>
 <td>4 €</td>
 </tr>
</tbody>
</table>
</body>
</html>

```



### Hinweis

*Laut Spezifikation muss <tfoot> vor <tbody> platziert werden. Allerdings führt dies in Browsern, die die Gruppierung nicht unterstützen, zu einer verkehrten Reihenfolge der Zellen.*



## Tabellen vorstrukturieren

Eigentlich sind HTML-Tabellen vollständig zeilenorientiert aufgebaut. Sie haben allerdings die Möglichkeit, mit dem Tag `<colgroup>` und darin den Tags `<col>` die Spaltenstruktur einer Tabelle vorzudefinieren. Dies geschieht am Anfang der Tabelle und hat vor allem einen Grund: Die Darstellungsperformanz von komplexen Tabellen wird im Browser besser.

Für die Spalten können Sie mit dem `width`-Attribut eine Breite vorgeben. Neben den üblichen Pixel- und Prozentangaben können Sie auch relative Angaben machen: `3*` steht z.B. für 30 %. Hier ein Beispiel, das die Breite der Spalten vorgibt:

```
<table border="1" width="400">
 <colgroup>
 <col width="3*" />
 <col width="7*" />
 </colgroup>
 <tr>
 <th>Obst</th>
 <th>Preis pro Kilo</th>
 </tr>
 <tr>
 <td>Äpfel</td>
 <td>2 €</td>
 </tr>
 <tr>
 <td>Kirschen</td>
 <td>4 €</td>
 </tr>
</table>
```



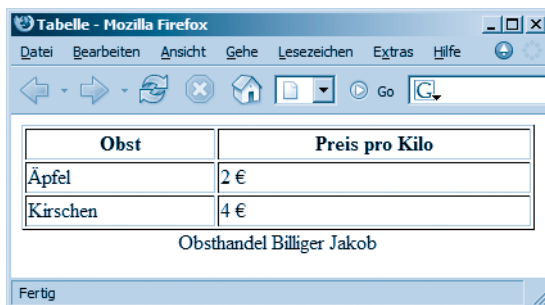
### Hinweis

Die Breitenangaben in Tabellenzellen haben gegenüber den Angaben für die Spalten Präferenz. Das heißt, wenn die Breitenangabe in `<col>` nicht mit der in `<td>` übereinstimmt, wird die von `<td>` verwendet.

## Tabellen beschriften

In der Praxis beschriftet man eine Tabelle meist noch in der Tabelle oder irgendwo auf der HTML-Seite. Es soll aber nicht verschwiegen werden, dass HTML ein eigenes Tag für die Tabellenbeschriftung kennt: `<caption>`. Dieses Tag steht meist oben in der Tabelle und kann mit dem Attribut `align` an allen vier Seiten ausgerichtet werden. Alternativ zum mittlerweile nicht mehr empfohlenen `align` können Sie auch den CSS-Befehl `caption-side` verwenden, der aber im Internet Explorer und Netscape Navigator 4.x nicht funktioniert. Hier ein Beispiel mit `align`:

```
<table border="1" width="400">
 <caption align="bottom">Obsthandel Billiger Jakob
</caption>
 <tr>
 <th>Obst</th>
 <th>Preis pro Kilo</th>
 </tr>
 <tr>
 <td>Äpfel</td>
 <td>2 €</td>
 </tr>
 <tr>
 <td>Kirschen</td>
 <td>4 €</td>
 </tr>
</table>
```



| Obst     | Preis pro Kilo |
|----------|----------------|
| Äpfel    | 2 €            |
| Kirschen | 4 €            |

Obsthandel Billiger Jakob

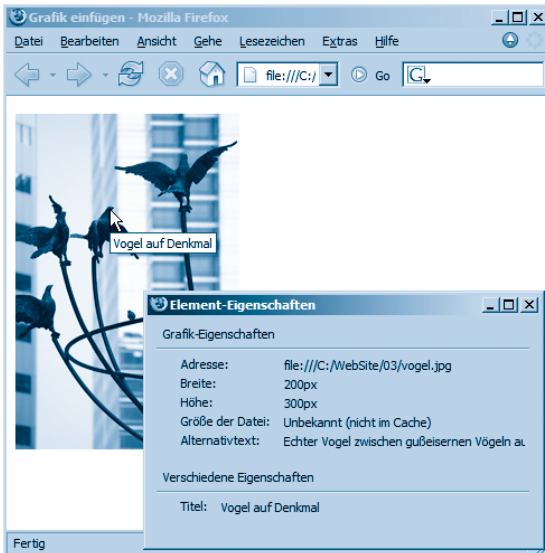


## Nutzloses Wissen

Die Antibabypille funktioniert auch bei Gorillas.

## Kapitel 4

# Bunte Bilder



*Abgesehen von reinen Textbrowsern wie Lynx ist das Web heute multimedial geworden. Fast jede Website verwendet Bilder und Grafiken, um Informationen zu transportieren. Bill Gates, der Microsoft-Gründer, sagte dazu: »Wer die Bilder beherrscht, beherrscht die Köpfe!« Ein Herrschaftsanspruch muss es nicht sein, der Sie zum Einsatz von Bildern bringt, aber das Ziel einer hübscheren Website ist auf jeden Fall sinnvoll.*

*Woher Ihre Bilder stammen oder welches Bildbearbeitungsprogramm Sie verwenden, ist für dieses Kapitel nicht wichtig. Für die wenigen Bildschirmfotos aus der Bildbearbeitung kommt hier Photoshop bzw. Photoshop Elements zum Einsatz.*



**Das können Sie schon:**

|                                  |    |
|----------------------------------|----|
| HTML-Grundgerüst einer Website   | 26 |
| Einsatz von CSS                  | 40 |
| Formatierung von Text            | 56 |
| Einsatz von Links                | 75 |
| Aufzählungslisten                | 90 |
| Listen mit Nummerierung          | 92 |
| Tabellen zur Datenstrukturierung | 95 |

**Das lernen Sie neu:**

|                                 |     |
|---------------------------------|-----|
| Bildformate und Bildoptimierung | 118 |
| Grafiken einbauen               | 121 |
| Bilder verlinken                | 127 |
| Hintergründe                    | 132 |

## Formate und Optimierung

Browser können nur drei Bildformate darstellen. Das heißt, Sie müssen aus diesen dreien wählen, welches Format für den jeweiligen Zweck das richtige ist. Zur Wahl stehen:

- GIF, das CompuServe Graphics Interchange Format
- JPEG von der Joint Picture Experts Group
- PNG, das Portable Network Graphics-Format

Das GIF-Format ist vor allem für Grafiken mit größeren Farbflächen, Buttons und Text geeignet. JPEG hat dagegen seine Stärken im Bereich Fotos. PNG gibt es in zwei Varianten. Eine ist eine Alternative zu GIF, die andere zu JPEG.

### Hinweis

*Dieses Kapitel handelt ausschließlich von Bitmap-Grafiken. Das sind Bilder, die aus einzelnen Pixeln bestehen. Jedes Pixel hat eine Farbe und aus allen Pixeln entsteht das Bild. Im Gegensatz dazu gibt es auch die Vektorgrafik. Sie besteht aus Einzelpunkten, die dann miteinander verbunden werden. Sind die Linien geschlossen, können die Farbflächen gefüllt werden. Vektorgrafiken sind insofern praktisch, als sie Linien und Farbflächen mit wesentlich weniger Speicheraufwand definieren. Im Web kommen sie hauptsächlich im Zusammenhang mit Flash oder SVG (Scalable Vector Graphics) zum Einsatz.*

## GIF

Ursprünglich von CompuServe entwickelt, hat sich GIF schnell zum wichtigsten Webformat hochgearbeitet. Zwar unterstützt eine GIF-Datei nur 256 Farben (= 8 Bit), dafür bietet es sehr interessante Funktionalitäten:

- Transparenz für eine Farbe
- Verlustfreie Komprimierung vor allem für Farbflächen
- GIF-Animation, das heißt Abspielen von mehreren Bildern hintereinander

Leider unterlag der Komprimierungsalgorithmus lange Zeit einem Patent der Firma Unisys, für dessen Lizenzierung die Hersteller von Grafikprogrammen (nicht die Website-Betreiber!) zahlen mussten. Deswegen war GIF schon auf dem Rückmarsch und sollte durch PNG ersetzt werden. Mittlerweile ist das Patent aber ausgelaufen und GIF wieder frei verfügbar.



## JPEG

Im Gegensatz zu GIF erlaubt das JPEG-Format alle RGB-Farben. Das sind dann insgesamt 16,78 Millionen (= 24 Bit). Damit lassen sich also auch Fotos farbgetreu darstellen. Um diese Farbmenge verarbeiten zu können, verwendet JPEG eine verlustbehaftete Komprimierung. Im Grafikprogramm stellen Sie meist die Stärke der Komprimierung auf einer Skala von 0 bis 10 oder von 0 bis 100 ein. Vor allem bei Fotos klappt die Komprimierung gut; vor allem bei Schrift und feinen Details versagt sie aber. Ansonsten beherrscht JPEG keine Transparenz und keine Animation.

### Hinweis

*Auch JPEGs mit maximaler Qualität komprimieren leicht verlustbehaftet. Sie sollten also ein JPEG nicht öfter speichern, sondern immer erst am Schluss aus einer Arbeitsdatei in einem verlustfreien Format wie TIFF oder dem Format Ihres Grafikprogramms konvertieren.*

## PNG

PNG steht für Portable Network Graphics. Die dahinter stehende Arbeitsgruppe hat das PNG-Format als Gegenentwurf zum patentierten GIF ins Rennen geworfen, dann aber das Format zu einem Allrounder weiterentwickelt. PNG gibt es in zwei Varianten:

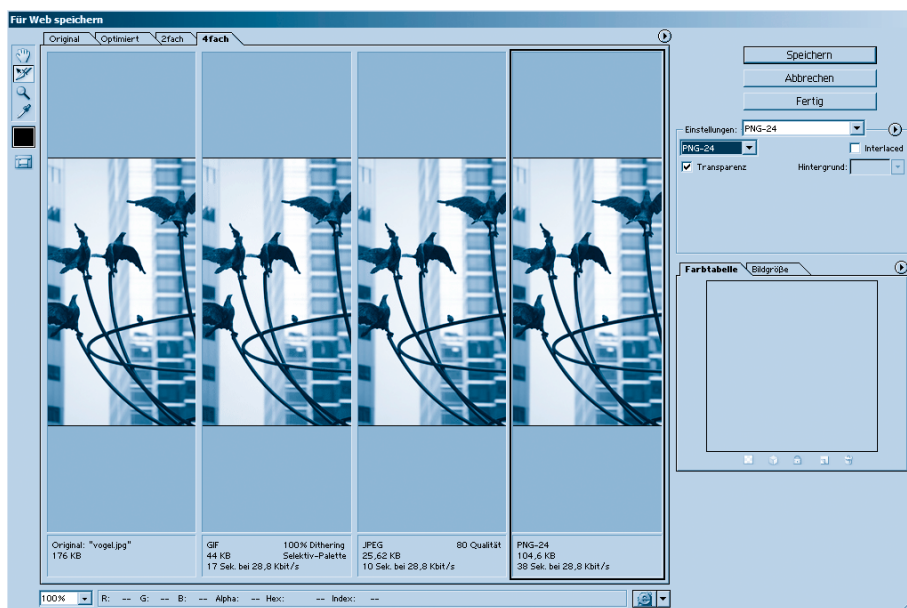
- mit 256 und weniger Farben (oft PNG-8 bzw. bei weniger Farben PNG-6 etc.)
- mit 16,78 Millionen Farben (PNG-24)

Beide Formate speichern verlustfrei. PNG-8 beherrscht wie GIF auch Transparenz, aber keine Animation. Neu ist in PNG-8 die so genannte Alphatransparenz. Damit können manche Pixel auch nur halbtransparent geschaltet werden. Allerdings funktioniert diese nicht im Internet Explorer. Eine umständliche Umgehungsmöglichkeit für diesen Bug gibt es zwar, das hat aber in der Praxis dazu geführt, dass Alphatransparenz noch nicht zum Einsatz kommt. Im Größenvergleich sind GIF und PNG-8 etwa gleichwertig. Oft gewinnt PNG-8 mit leichtem Vorsprung.

PNG-24 kennt keine Transparenz und ist eher für Fotos gedacht. PNG-24 komprimiert im Gegensatz zu JPEG verlustfrei. Allerdings sind die Dateigrößen deutlich höher als bei JPEG, weswegen PNG-24 kaum zum Einsatz kommt.

## Bilder optimieren

Am Anfang steht beim Optimieren die Wahl des richtigen Dateiformats. Hier gilt es, auch Dateigröße und Ergebnis zu vergleichen. Die meisten Grafikprogramme helfen hier mit Übersichten und Direktvergleichen.



Wenn Sie Bilder für das Web einsetzen, gilt es, noch ein paar Regeln zu beachten. Die wichtigste ist, dass Sie keine zu hohe **Auflösung** verwenden sollten. Die Auflösung ist die Zahl der Pixel pro Maßeinheit. Gemessen wird sie meist in Pixel pro Zoll (= Inch = 2,54 cm). Für den Druck mit Druckmaschinen benötigt man Auflösungen bis zu 300. Hier spricht man zwar von dpi, also dots (= Druckpunkte) pro Inch, aber dies entspricht den Pixeln für die Darstellung auf dem Monitor.

Ein Monitor hat bei weitem keine 300 ppi (Pixel pro Inch), sondern je nach System meist zwischen 72 und 96 ppi. Entsprechend reicht es, wenn Sie Bilder für das Web in 72 ppi abspeichern. Alles andere kostet nur zu viel Dateigröße.

Besonders vorsichtig müssen Sie mit **Schrift** in der Grafik sein. Gerade Anti-aliasing, die Glättung von Kanten, kann am Bildschirm zu komischen Ergebnissen führen.



## Einbau

In HTML ist der Einbau einer Grafik oder eines Bildes sehr einfach. Sie verwenden das `<img>`-Tag und geben im `src`-Attribut den Link zur Grafik an. `src` steht für Quelle (engl. source), ist also der Quellpfad des darzustellenden Bildes. Im folgenden Beispiel wird ein Bild mit dem Dateinamen *vogel.jpg* eingebunden, das sich im selben Verzeichnis wie das HTML-Dokument selbst befindet.

```
<p></p>
```



Da ein Bild standardmäßig ein Inline-Element ist, sollte es in einem Absatz oder einem anderen Blockelement stehen. Das ist aber nicht zwingend notwendig, sondern nur, wenn Sie beim Validieren der Website bestehen möchten.

## Größe

Wenn Sie keine Größenangaben vornehmen, wird ein Bild in der Originalgröße angezeigt. Mit den Attributen `width` und `height` können Sie allerdings auch eigene Größenmaße in Pixel oder Prozent von der Originalgröße verwenden. Dies hat auch den Vorteil, dass der Browser bereits die Größe der Grafik für das Rendern kennt.

Hier ein Beispiel, das das Bild um ein Drittel in der Breite und ein Drittel in der Höhe vergrößert:

```
<p></p>
```

Wie Sie auch in der Abbildung sehen können, überzeugt das Ergebnis rein qualitativ nicht. Das Problem ist, dass Browser nicht so gute Berechnungsmethoden zur Bildvergrößerung und -verkleinerung besitzen wie Grafikprogramme. Das heißt, Sie sollten Größenänderungen von Bildern (Skalierungen) fast immer lieber per Grafikprogramm vornehmen. Völlig sinnlos ist es außerdem, ein Bild zu verkleinern, denn dabei wird an den Nutzer nur unnötige Bilddatenmenge geliefert. In der Praxis kann man das manchmal in Bildergalerien sehen, die dann auch für die Miniaturdarstellung (Thumbnail) ewige Ladezeiten benötigen.



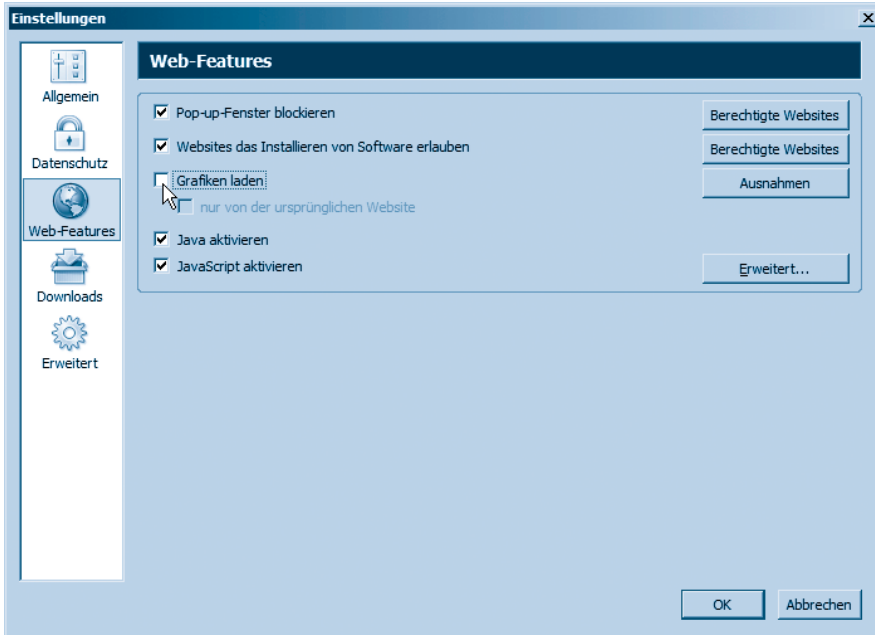
### Tipp

*Das Vergrößern von Bildern kann z.B. bei einfarbigen Flächen als Trick verwendet werden. Sie können ein 1 \* 1 Pixel großes Bild mit einer Farbe erzeugen und dies dann z.B. zu einer Linie machen, indem Sie die Breite skalieren und die Höhe unverändert lassen.*



## Titel und Alternativtext

Sie können nicht davon ausgehen, dass jeder Website-Nutzer Grafiken aktiviert hat. In den meisten Browsern gibt es über die Grundeinstellungen eine Möglichkeit, Grafiken zu unterdrücken. Einige Surfer verwenden beispielsweise auch einen reinen Textbrowser wie Lynx. Und sehbehinderte Surfer verwenden Screenreader, die natürlich keine Grafiken vorlesen können.



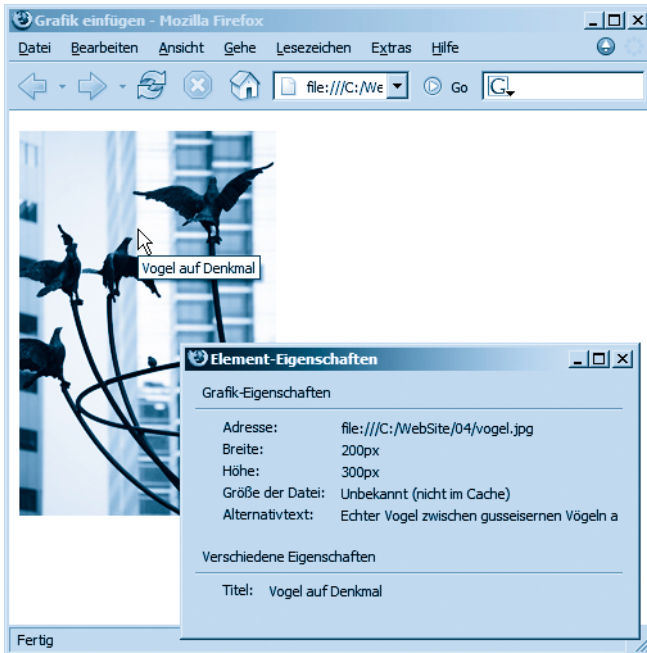
Für all diejenigen sollten Sie vorsorgen und mit dem `alt`-Attribut einen Alternativtext für die Grafik definieren. Dieser Text wird angezeigt, wenn das Bild selbst nicht eingeblendet oder gefunden werden kann, und sollte Inhalt bzw. Informationszweck des Bildes enthalten. In manchen Browsern, vor allem im Internet Explorer, wird das `alt`-Attribut auch dazu verwendet, eine gelb hinterlegte QuickInfo anzuzeigen. Originär dafür vorgesehen ist in der Spezifikation aber das `title`-Attribut. Hier ein Beispiel, wie es korrekt funktioniert:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- grafik.html -->
<html>
<head>
 <title>Grafik einfügen</title>
</head>
```

```

<body>
 <p></p>
</body>
</html>

```



### Hinweis

*Als zusätzliche Angabe können Sie noch das name-Attribut für ein Bild verwenden. Das benötigen Sie beispielsweise, wenn Sie per Skript auf ein Bild zugreifen möchten.*

## Ausrichtung und Abstände

Bilder sind keine Blockelemente, interagieren aber dennoch mit ihrer Umgebung. Den Innen- und Außenabstand können Sie wie auch bei einem Blockelement mit `padding` und `margin` festlegen (siehe *Kapitel 2*). Daneben gibt es einige Attribute, die in diesem Abschnitt Erwähnung finden.





## Textausrichtung

Mit dem Attribut `align` oder dem CSS-Befehl `vertical-align` können Sie Text neben dem Bild ausrichten. Dabei ist egal, ob der Text innerhalb der `<img>`-Tags oder daneben steht. Standardmäßig steht er allerdings daneben. Allerdings und das ist einschränkend: Nur eine Textzeile kann ausgerichtet werden, deren Größe sich nach der Fenstergröße richtet.

Die wichtigsten Werte:

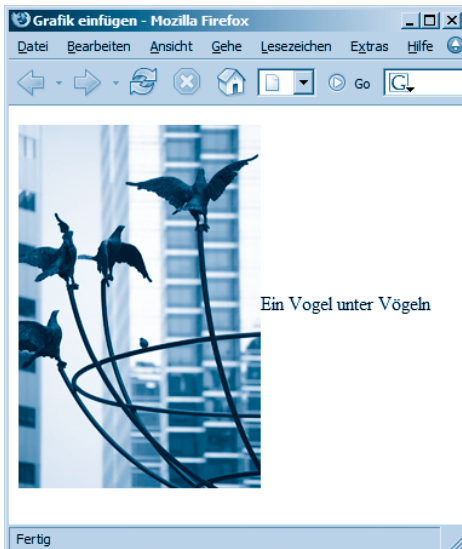
- `top` richtet den Text an der Oberkante des Bildes aus.
- `middle` zentriert den Text am Bild.
- `bottom` richtet an der Unterkante des Bildes aus. Dies ist die Standardeinstellung.

Hier ein Beispiel:

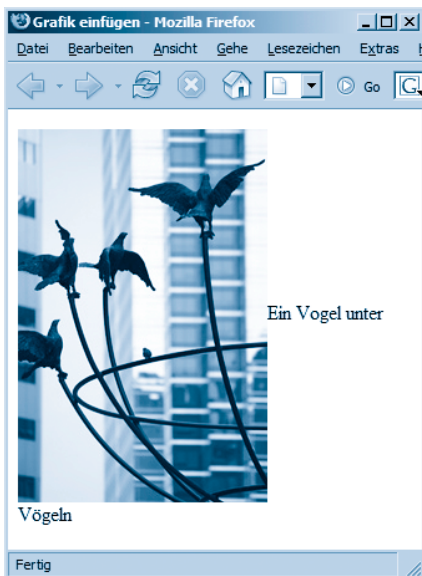
```
<p>
Ein Vogel unter Vögeln</p>
```

Oder mit CSS:

```
<p>
Ein Vogel unter Vögeln</p>
```



Die Textzeile bricht um, wenn das Browserfenster zu klein wird:



### Tipp

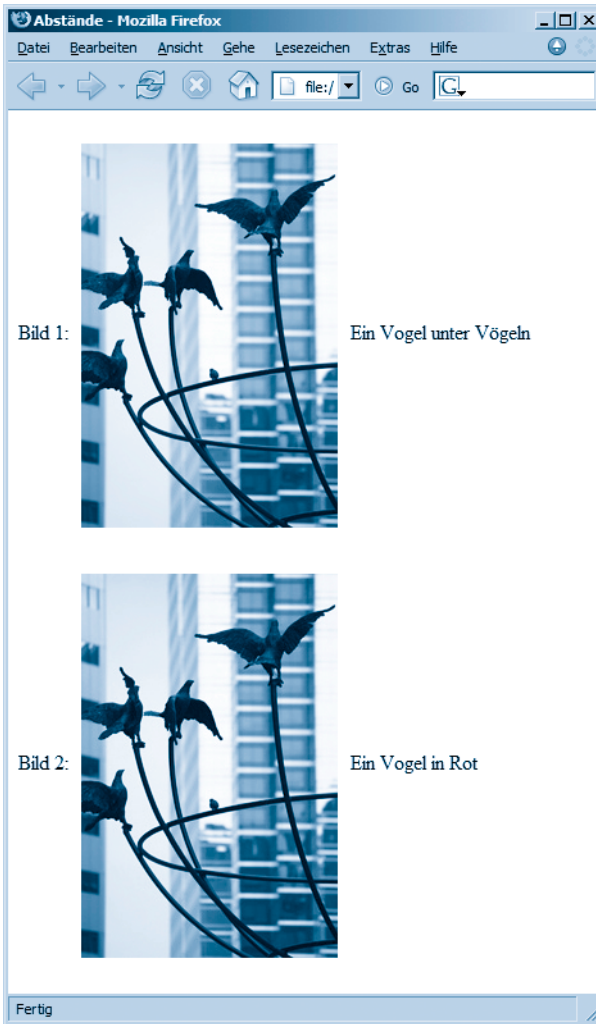
*Für eine exaktere Platzierung von größeren Textmengen arbeitet man im Allgemeinen mit Tabellen oder seit neuestem mit CSS-Positionierung. Siehe hierzu Kapitel 7.*

## Abstände

Den horizontalen und vertikalen Abstand zwischen Bild und Umgebung steuern Sie mit den HTML-Attributen `hspace` und `vspace`. Beide übernehmen einen Wert in Pixeln. Heutzutage ist es allerdings üblicher, dafür die CSS-Alternativen `margin` und `padding` einzusetzen. Im Folgenden finden Sie die HTML-Variante und die CSS-Variante im direkten Vergleich:

```
<p>Bild 1:Ein Vogel unter Vögeln</p>
<p>Bild 2:Ein Vogel in Rot</p>
```





## Verlinkung

Ein Bild als Link zu verwenden, ist ausgesprochen einfach: Sie legen einfach einen `<a>`-Tag um das `<img>`-Tag:

```



```



Allerdings gibt es noch zwei Dinge, die nicht so hervorragend funktionieren: Zum einen erscheint ein blauer Rahmen um das Bild, zum anderen können Sie so nicht nur Bereiche eines Bildes verweissensitiv machen. Für diese Probleme präsentieren die nächsten Abschnitte Lösungen.

## Rahmen

Normalerweise besitzt ein Bild keinen Rahmen, wenn es mit einem Link versehen wird, aber schon. Sie können den Rahmen aber auch selbst steuern, und zwar mit dem HTML-Attribut `border`. Es übernimmt als Parameter die Rahmendicke in Pixel. Wenn Sie sie auf 0 setzen, verschwindet der Rahmen:

```



```



**Tipp**

Alternativ zum HTML-Befehl können Sie zum Einrahmen von Bildern auch die CSS-Befehle für Rahmen verwenden (siehe Kapitel 2). Damit lässt sich ein Bilderrahmen recht individuell gestalten. Für absolute Gestaltungsfreiheit z.B. mit Farbverläufen sollten Sie den Rahmen allerdings mit einem Bildbearbeitungsprogramm direkt in das Bild integrieren.

## Imagemaps

Oftmals ist eine Grafik größer und Sie möchten nur einzelne Bereiche verlinken. Dann gibt es zwei Möglichkeiten:

- Sie schneiden die Grafik auseinander und setzen Sie mit einer HTML-Tabelle wieder zusammen (siehe Kapitel 3 und Kapitel 7).
- Oder Sie verwenden eine Imagemap.

Eine Imagemap definiert Koordinaten auf einem Bild. Und das geht so:

**1** Im `<img>`-Tag geben Sie an, dass Sie eine Imagemap verwenden möchten. Dazu verwenden Sie das Attribut `usemap` und geben als Wert einen Hash und dann den Namen der Imagemap an:

```
<p></p>
```

**2** Definieren Sie dann die Karte an beliebiger Stelle in der Seite. Dazu verwenden Sie `<map>`-Tags.

```
<map name="map1">

</map>
```

XHTML-konform verwenden Sie hier noch zusätzlich ein `id`-Attribut.

**3** Legen Sie dann die einzelnen Linkbereiche an. Dazu dient das `<area>`-Tag.

**4** Vergeben Sie per `shape` eine Form, die der Bereich haben soll, und mit `coords` die zugehörigen Koordinaten, z.B. `rect` für ein Rechteck mit den Koordinaten der linken oberen (x, y) und der rechten unteren Ecke (x, y):

```
<area shape="rect" coords="123,27,184,129" />
```

Als Werte stehen für shape folgende zur Wahl:

- rect für ein Rechteck,
- circle für einen Kreis mit Koordinaten des Kreismittelpunkts gefolgt vom Radius
- polygon für ein Vieleck mit den Koordinaten aller Einzelpunkte
- default für alle Gegenden, die bisher keinem Bereich zuzuordnen sind. Hier sind keine Koordinaten erforderlich.

## 5 Vergeben Sie anschließend weitere Formen.

```
<area shape="poly" coords="28,134, 40,141, 50,146, 68,146,
 73,146, 87,141, 100,132, 108,122, 115,107, 123,107,
 123,129, 170,129, 170,169, 28,169" />
<area shape="circle" />
<area shape="default" />
```

## 6 Vergeben Sie für die Formen per href-Attribut Links.

```
<area shape="poly" alt="Polygon" coords="28,134, 40,141,
 50,146, 68,146, 73,146, 87,141, 100,132, 108,122,
 115,107, 123,107, 123,129, 170,129, 170,169, 28,169"
 href="polygon.html" />
<area shape="rect" alt="Rechteck" coords="123,27,184,129"
 href="rechteck.html" />
<area shape="circle" alt="Kreis" coords="62,94,51"
 href="kreis.html" />
<area shape="default" alt="Daneben!" href="daneben.html" />
```

Sie können außerdem die übrigen von Links und Bildern bekannten Attribute wie alt, title und target einsetzen.

Hier ein komplettes Beispiel in der Übersicht:

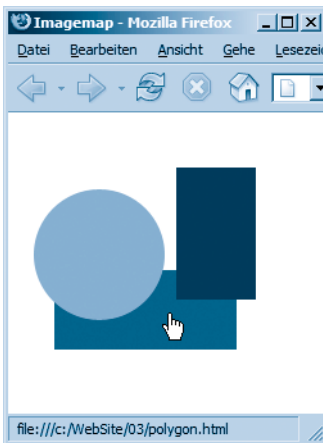
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- imagemap.html -->
<html>
<head>
 <title>Imagemap</title>
</head>
<body>
 <p></p>
 <map name="map1" id="map1">
```



```

<area shape="poly" alt="Polygon" coords="28,134, 40,141,
 50,146, 68,146, 73,146, 87,141, 100,132, 108,122,
 115,107, 123,107, 123,129, 170,129, 170,169, 28,169"
 href="polygon.html" />
<area shape="rect" alt="Rechteck" coords="123,27,184,129"
 href="rechteck.html" />
<area shape="circle" alt="Kreis" coords="62,94,51"
 href="kreis.html" />
<area shape="default" alt="Daneben!"
 href="daneben.html" />
</map>
</body>
</html>

```



### Tipp

*Zugegeben, die Koordinaten in diesem Beispiel haben wir nicht per Versuch und Irrtum herausgefunden, sondern uns helfen lassen. In der Praxis geht es deutlich schneller, wenn Sie sich beim Erstellen einer Imagemap von einem Programm helfen lassen. Dies kann das Grafikprogramm wie Photoshop, Paint Shop oder GIMP sein, oder auch ein HTML-Editor wie z.B. Dreamweaver.*

## Hintergründe

Schon zu reinen HTML-Zeiten war es möglich, Hintergründe für eine Website zu vergeben. Dazu haben Sie einfach das Attribut `background` im `<body>`-Tag verwendet und den Bild-URL angegeben. Mit dem Attribut `bgcolor` ließ sich eine Hintergrundfarbe angeben. Dank CSS gibt es mittlerweile eine Menge mehr Möglichkeiten: Sie können Hintergründe nicht nur für das `<body>`-Tag, sondern für jedes Blockelement vergeben. Außerdem haben Sie die Option, zu steuern, ob ein Hintergrundbild beliebig oft nebeneinander gelegt wird. Dieser Kacheleffekt war noch zu HTML-Zeiten absoluter Standard.

Zuerst erhalten Sie eine Übersicht über die wichtigsten Befehle für Hintergründe:

- `background-image` legt ein Hintergrundbild fest. Mit `url(bild.jpg)` geben Sie dann das Bild an.
- `background-color` definiert eine Hintergrundfarbe. Erlaubt sind die üblichen CSS-Farbangaben (siehe *Kapitel 2*).
- `background-repeat` legt fest, ob eine Hintergrundgrafik wiederholt werden soll. `no-repeat` heißt keine Wiederholung, `repeat` ist der Standardwert und wiederholt die Hintergrundgrafik horizontal und vertikal, `repeat-x` kachelt nur horizontal, `repeat-y` nur vertikal.
- `background-attachment` regelt, ob das Hintergrundbild beim Scrollen mitwandert. Dies geschieht, wenn Sie den Wert `fixed` verwenden. Der Standardwert ist `scroll`.
- `background-position` bestimmt die Position des Hintergrundes. Sie können die x- und y-Koordinate in dieser Reihenfolge angeben.

### Hinweis

*Der Sammelbefehl `background` vereinigt `background-color`, `background-image`, `background-attachment`, `background-repeat` und `background-position`, und zwar in beliebiger Reihenfolge.*

Zum einfacheren Verständnis ein Beispiel. Wir legen in einem Grafikprogramm eine 10 \* 10 Pixel große Grafik an. Sie besteht aus einem nicht gefüllten Quadrat. Alles außer der Linie ist transparent (hier als weiße und graue Kästchen dargestellt).

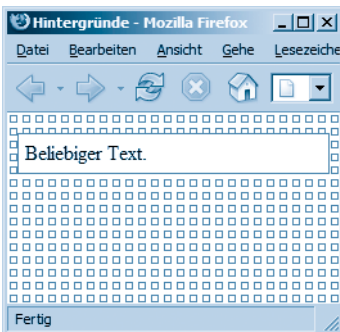






Dieses Bild *hintergrund.gif* verwendet das folgende Skript als Hintergrundbild für die Seite. Der Textabsatz erhält eine Hintergrundfarbe, um sich davon abzusetzen:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- hintergrund.html -->
<html>
<head>
 <title>Hintergrunde</title>
</head>
<body style="background-image: url(hintergrund.gif);">
 <p style="background-color: white; border: 1px solid
 #727171; padding: 5px;">Beliebiger Text.</p>
</body>
</html>
```

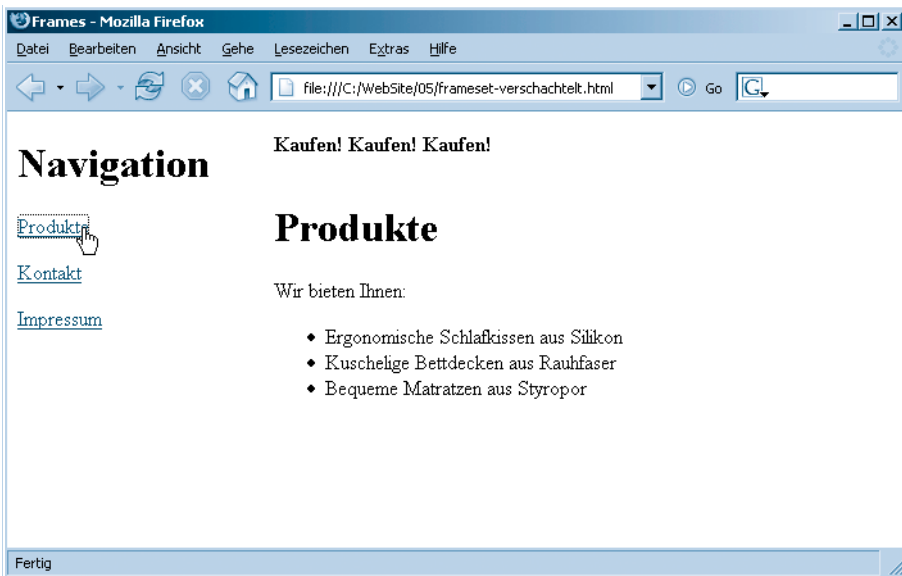


## Nutzloses Wissen

Die Löwin erledigt etwa 90 % der Jagd.

## Kapitel 5

# Seitenlayout mit Frames



*Bis jetzt bestand jede Webseite tatsächlich nur aus einer einzigen Datei. Das ändert sich in diesem Kapitel: Jetzt kann eine Seite aus mehreren Einzeldateien bestehen. Die Technik, die wir dazu verwenden, heißt »Frames«. Eine Seite besteht aus mehreren Frames, die unabhängig voneinander zusammenarbeiten.*

**Das können Sie schon:**

|                                  |     |
|----------------------------------|-----|
| HTML-Grundgerüst einer Website   | 26  |
| Einsatz von CSS                  | 40  |
| Formatierung von Text            | 56  |
| Einsatz von Links                | 75  |
| Aufzählungslisten                | 90  |
| Listen mit Nummerierung          | 92  |
| Tabellen zur Datenstrukturierung | 95  |
| Grafiken einbauen                | 121 |
| Bilder verlinken                 | 127 |

**Das lernen Sie neu:**

|                      |     |
|----------------------|-----|
| Was sind Frames?     | 136 |
| Frames erstellen     | 142 |
| Frames verschachteln | 150 |
| Auf Frames verlinken | 152 |

## Was sind Frames?

Werfen Sie einen Blick auf die Abbildung. Dort sehen Sie eine Webseite, die in Wirklichkeit aus mehreren Dateien besteht. Jeder dieser Teilbereiche lässt sich unabhängig voneinander scrollen. Die Technik, die hier zum Einsatz kommt, nennt sich **Frames**. Ein Frame (auf Deutsch: Rahmen) ist ein Bereich einer Webseite, der – unabhängig vom Rest – ein HTML-Dokument enthält.



Frames haben einige Vorteile. Beispielsweise ist es möglich, die Navigation der Website in einem eigenen Frame unterzubringen. Wenn der Benutzer nun auf eine andere Seite navigiert, bleibt der Navigationsframe unverändert. Das ist natürlich weniger aufwändig, als auf jeder Unterseite erneut die Navigation unterbringen zu müssen. Außerdem sind Änderungen an der Navigation dann nur in einer Datei erforderlich, nicht mehr in allen.

Allerdings sind Frames auch umstritten. Anfänger im Web haben unter Umständen Schwierigkeiten, einzelne Frames auszudrucken oder ein Bookmark (Lesezeichen) abzulegen. Außerdem kann es Probleme geben, wenn ein Dokument, das in einem Frame abgelegt worden ist (etwa eine Inhaltsseite), direkt im Webbrowser aufgerufen wird. Dann fehlen nämlich alle anderen,



unter Umständen wichtigen Frames, wie etwa ein Navigationsframe. Außerdem haben manche Suchmaschinen Frames nicht gerade gerne. Einige (wenige) Browser, beispielsweise solche für mobile Endgeräte, haben Schwierigkeiten, Frames darzustellen.

### Tipp

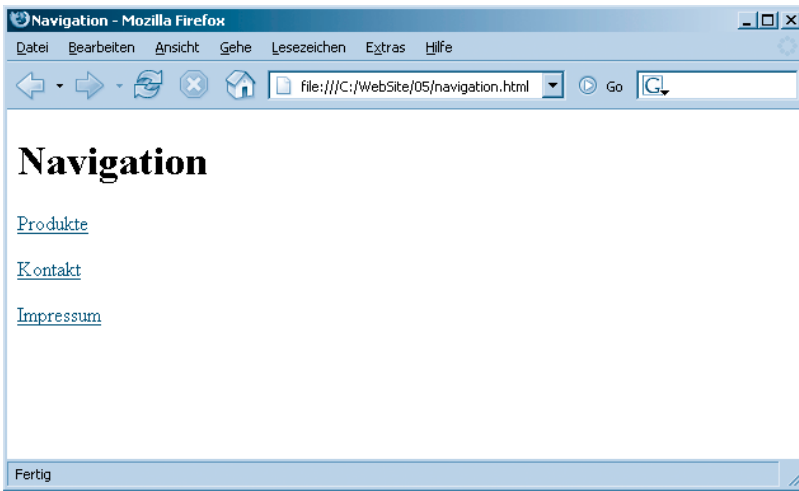
*Zumindest um den Punkt »Frames nachladen« kümmern wir uns in Kapitel 9, in dem es um JavaScript geht.*

Sie müssen es sich also genau überlegen, ob Frames für Ihre Website geeignet sind oder nicht. Im Web scheint es zurzeit so zu sein, dass eine spezielle Form von Frames immer größere Verbreitung findet, so genannte Iframes. Am Ende dieses Kapitels sehen wir uns an, worum es sich hierbei handelt und wie man sie einsetzen kann.

## Vorbereitungen

Um Frames verwenden zu können, brauchen wir zunächst einige Testdateien. Diese werden wir im Verlauf dieses Kapitels immer weiter anpassen. Zunächst einmal benötigen wir ein Dokument, das drei Navigationslinks enthält:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- navigation.html -->
<html>
<head>
 <title>Navigation</title>
</head>
<body>
 <h1>Navigation</h1>
 <p>
 Produkte
 </p>
 <p>
 Kontakt
 </p>
 <p>
 Impressum
 </p>
</body>
</html>
```



Außerdem benötigen wir mehrere Inhaltsseiten: die Start-, Produkt-, Kontakt- und Impressumsseite. Diese sind natürlich hier nur äußerst spärlich mit Inhalt gefüllt, denn es soll primär das Grundprinzip vermittelt werden. Wie Sie diese ansprechender gestalten können, wurde bereits in den vorherigen Kapiteln weitgehend geklärt.

Die erste Inhaltsseite ist die Startseite. Diese enthält – weil es gute Sitte ist – noch einmal die wichtigsten Links zu den entsprechenden Unterseiten.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- start.html -->
<html>
<head>
 <title>Startseite</title>
</head>
<body>
 <h1>Willkommen!</h1>
 <p>
 Herzlich willkommen auf unserer Website! Hier finden Sie
 Informationen über uns und unsere Dienstleistungen.

Nachfolgend einige interessante Links:
 </p>
 <p>
 Produkte
 </p>
 <p>
 Kontakt
 </p>
```



```
<p>
 Impressum
</p>
</body>
</html>
```

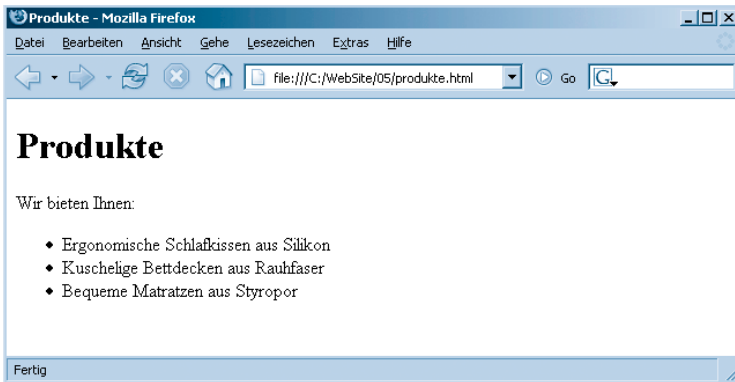


Die nächste Inhaltsseite ist die Produktseite. Sie wird später einmal eine Liste aller Produkte enthalten, möglicherweise sogar auf einzelnen Unterseiten. Doch zunächst müssen ein paar magere Informationen ausreichen:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- produkte.html -->
<html>
<head>
 <title>Produkte</title>
</head>
<body>
 <h1>Produkte</h1>
 <p>
 Wir bieten Ihnen:

 Ergonomische Schlafkissen aus Silikon
 Kuschelige Bettdecken aus Rohfaser
 Bequeme Matratzen aus Styropor

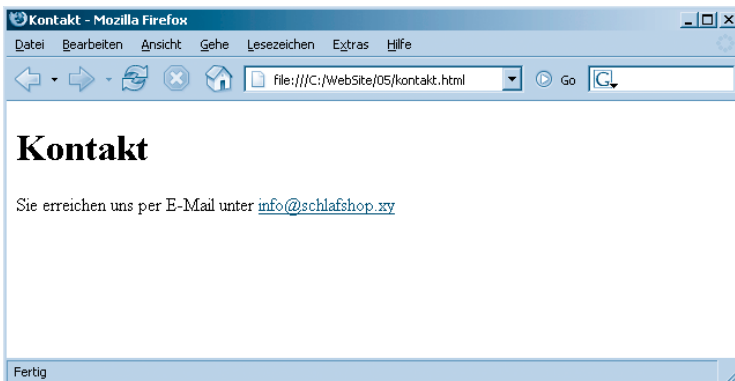
 </p>
</body>
</html>
```



Die Kontaktseite enthält zunächst nur die E-Mail-Adresse – in *Kapitel 6* erfahren Sie, wie Sie zusätzlich noch ein Feedback-Formular unterbringen können:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- kontakt.html -->
<html>
<head>
 <title>Kontakt</title>
</head>
<body>
 <h1>Kontakt</h1>
 <p>
 Sie erreichen uns per E-Mail unter

 info@schlafshop.xy
 </p>
</body>
</html>
```





Abschließend der Pflichteintrag für jede Website: das Impressum. Dort stehen unter anderem der Name des Betreibers und seine Mailadresse; was Sie sonst noch brauchen, erfahren Sie in *Kapitel 11*.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- impressum.html -->
<html>
<head>
 <title>Impressum</title>
</head>
<body>
 <h1>Impressum</h1>
 <p>Betreiber der Site: Dipl.-Faulenzer H. Schlafberg</p>
 <p>
 E-Mail:

 info@schlafshop.xy

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



### Tipp

Natürlich finden Sie die (finalen!) Versionen dieser Dokumente auch in dem Code-Archiv, das Sie bei uns herunterladen können (siehe Vorwort).

## Frames erstellen

So weit, so gut. Die vorherigen Abbildungen haben die erstellten Seiten stets gezeigt, aber immer alleine im Browser. Mit Frames gibt es nun endlich die Möglichkeit, die Navigation mit den Inhaltsseiten zusammenzubringen.

Dazu müssen Sie etwas umdenken. Bis jetzt galt immer: Der Browser lädt (ein!) HTML-Dokument, das wird angezeigt, fertig. Bei Frames ändert sich das ein wenig: Der Browser lädt ein HTML-Dokument, das so genannte Frameset. Dieses Dokument enthält keine Inhalte, sondern Verweise auf mehrere einzelne Unterframes. Diese einzelnen Unterframes sind genau die Dokumente, die wir auf den letzten Seiten zusammengestellt haben.

Dafür sind zwei neue Tags notwendig:

- `<frameset>` enthält eine Definition eines Framesets – also, dass der Browserbereich in mehrere Frames unterteilt werden soll.
- `<frame>` gibt einen einzelnen Frame an.

Dabei ändert sich auch die Dokumenttypangabe am Anfang des Dokuments, in dem das `<frameset>`-Element steht; hier müssen wir Folgendes verwenden:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

### Hinweis

*Bei den einzelnen Frame-Dokumenten dagegen ändert sich in Bezug auf den Dokumenttyp nichts. Erinnern Sie sich: Das sind herkömmliche Dokumente, die Sie auch einzeln und unabhängig voneinander im Browser laden könnten.*

Die Erstellung eines Framesets geht in wenigen Schritten vonstatten. Zu einigen dieser Schritte gibt es allerdings noch eine ganze Reihe an Hintergrundinformationen, die Sie im Folgenden jeweils direkt erwähnt finden.

- 1 Erstellen Sie ein Frameset mit dem `<frameset>`-Element.

```
<frameset>
</frameset>
```

- 2 Geben Sie im `cols`-Attribut die Breite der Frames an.



Es gibt mehrere Möglichkeiten, die Breite anzugeben. Am einfachsten ist es, Pixelwerte zu verwenden. So steht folgende Angabe für zwei Frames, der erste ist 200 Pixel, der zweite 650 Pixel breit:

```
<frameset cols="200,650">
</frameset>
```

Allerdings ist das nicht ganz optimal, denn das setzt die Gesamtbreite beider Frames auf 800 Pixel fest. Hat der Benutzer eine größere Auflösung, wird der Browser den fehlenden Platz irgendwie verwenden, was zu hässlichen Ergebnissen führen kann. Aus diesem Grund können Sie den Browser anweisen, anstelle einer festen Pixelzahl die restliche zur Verfügung stehende Breite zu verwenden. Das geht mit dem Sternchen. Folgende Angabe macht den linken Frame 200 Pixel breit, der rechte Frame nimmt die restliche Breite im Browser in Anspruch:

```
<frameset cols="200,*">
</frameset>
```

Dritte Möglichkeit: Sie verwenden prozentuale Werte. Wenn also der rechte Frame dreimal so breit sein soll wie der linke Frame, können Sie den Wert des `cols`-Attributs wie folgt setzen:

```
<frameset cols="25%,75%">
</frameset>
```

**3** Fügen Sie pro Frame ein `<frame>`-Element innerhalb des `<frameset>`-Elements ein:

```
<frameset cols="200,*">
 <frame />
 <frame />
</frameset>
```

**4** Geben Sie die Adresse des entsprechenden Frames im `src`-Attribut des `<frame>`-Tags an.

Folgende Angabe lädt in den linken Frame die Navigation und in den rechten Frame die Startseite:

```
<frameset cols="200,*">
 <frame src="navigation.html" />
 <frame src="start.html" />
</frameset>
```

**5** Geben Sie jedem Frame noch einen eindeutigen Namen – im `name`-Attribut des `<frame>`-Tags.

```
<frameset cols="200,*">
 <frame src="navigation.html" name="Navi" />
 <frame src="start.html" name="Inhalt" />
</frameset>
```

Am Ende haben Sie folgende HTML-Datei:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<!-- frameset.html -->
<html>
<head>
 <title>Frames</title>
</head>
<frameset cols="200,*">
 <frame src="navigation.html" name="Navi" />
 <frame src="start.html" name="Inhalt" />
</frameset>
</html>
```



Sie sehen, dass das HTML-Dokument keinen `<body>`-Bereich mehr enthält, denn dieser wird in der Frameset-Datei nicht benötigt. Trotzdem ist die Ausgabe im Webbrowser wie gewünscht: zwei Frames, der eine 200 Pixel breit, der andere die restliche Breite. Im linken Frame wurde das Navigationsdokument geladen (*navigation.html*), im rechten Frame die Startseite (*start.html*).

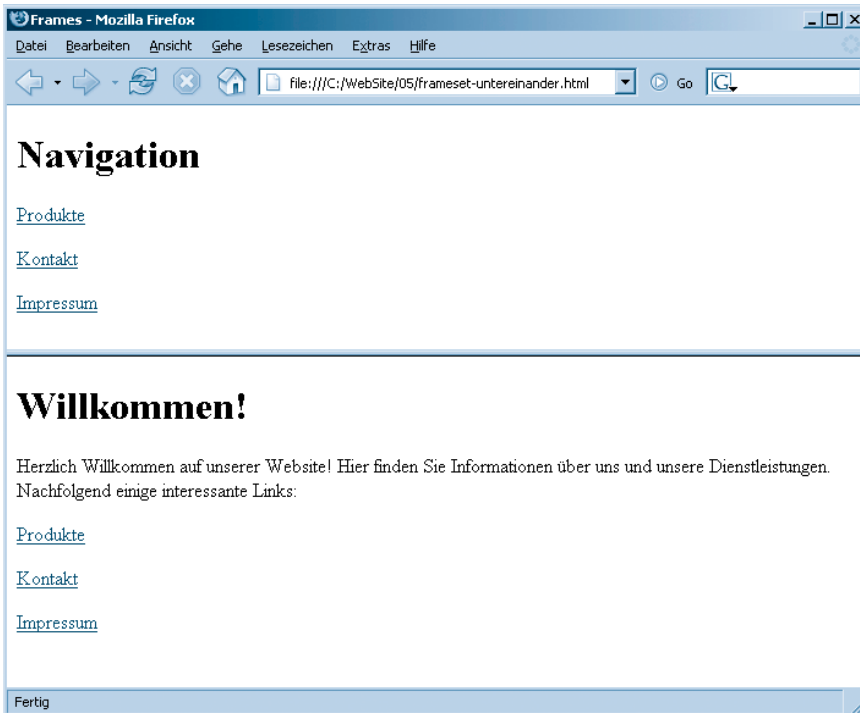


Im Übrigen ist es auch möglich, Frames untereinander anzubringen. Dazu sind die beiden folgenden Modifikationen vonnöten:

- Verwenden Sie das Attribut `rows` statt `cols`.
- Der Wert des Attributs `rows` bezeichnet dann natürlich die Höhe der einzelnen Frames und nicht mehr die Breite.

Nachfolgend noch einmal das Frameset von vorher, allerdings sind die Frames dann untereinander angeordnet:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<!-- frameset-untereinander.html -->
<html>
<head>
 <title>Frames</title>
</head>
<frameset rows="200,*">
 <frame src="navigation.html" name="Navi" />
 <frame src="start.html" name="Inhalt" />
</frameset>
</html>
```



Das Ergebnis sieht zugegebenermaßen etwas ungewohnt aus, aber es funktioniert: Die beiden Frames erscheinen ab sofort untereinander.

## Frames verschönern

Abschließend ist es noch an der Zeit für ein paar kleine Verschönerungsmaßnahmen. Zunächst einmal fällt auf, dass der Balken zwischen den Frames mit der Maus verschoben werden kann. Der Benutzer kann also die Größen der Frames verändern und damit möglicherweise das Layout ruinieren.



Zum Glück kann das verhindert werden. Das geht nicht mit CSS, aber mit HTML: Verwenden Sie das Attribut `noresize`. Allerdings muss noch geklärt werden, wo das hingehört. Die Antwort: in das `<frame>`-Element. Zunächst klingt das etwas merkwürdig, doch mit etwas Nachdenken wird es klar. Stellen Sie sich vor, Sie haben drei Frames innerhalb eines Framesets. Nun möchten Sie aus irgendwelchen Gründen, dass der Balken zwischen dem ersten und dem zweiten Frame verschoben werden darf, der zwischen dem zweiten und dem dritten Balken allerdings nicht. Es muss also pro Framebalken festgelegt werden, wie es sein soll. Dafür gibt es die folgenden Regeln:

- Wenn Sie in einem Frame `noresize` setzen, können beide Framegrenzen (links und rechts bzw. oben und unten) nicht mehr per Maus verschoben werden.
- Wenn Sie `noresize` nicht setzen, können beide Framegrenzen (links und rechts bzw. oben und unten) per Maus verschoben werden.



- Wenn in einem der an den Balken grenzenden Frames `noresize` gesetzt wird, im anderen nicht, »gewinnt« `noresize` und der Balken kann nicht per Maus verschoben werden.

Im nachfolgenden Listing steht `noresize` bei einem der beiden Frames; der Balken dazwischen kann danach nicht mehr verschoben werden:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<!-- frameset-noresize.html -->
<html>
<head>
 <title>Frames</title>
</head>
<frameset cols="200,*">
 <frame src="navigation.html" name="Navi"
 noresize="noresize" />
 <frame src="start.html" name="Inhalt" />
</frameset>
</html>
```

Apropos Verschönerungsmaßnahmen: Die beiden Frames sind ja durch einen etwas unschicken grauen Balken voneinander getrennt. Mit ein klein wenig HTML ist das allerdings kein Problem: Das Attribut `frameborder` aktiviert oder entfernt den Rahmen, je nachdem, welchen Wert Sie angeben:

- Beim Wert `"1"` etwa erscheint der Balken – das ist der Standard.
- Beim Wert `"0"` dagegen verschwindet der hässliche Rahmen.

#### Hinweis

*Zwar sind mit etwas CSS auch andere Effekte möglich (etwa `border-color` für die Farbe des Rahmens), aber auf modernen Websites mit Frames sind praktisch nirgends mehr Rahmen anzutreffen. Also: Weg damit!*

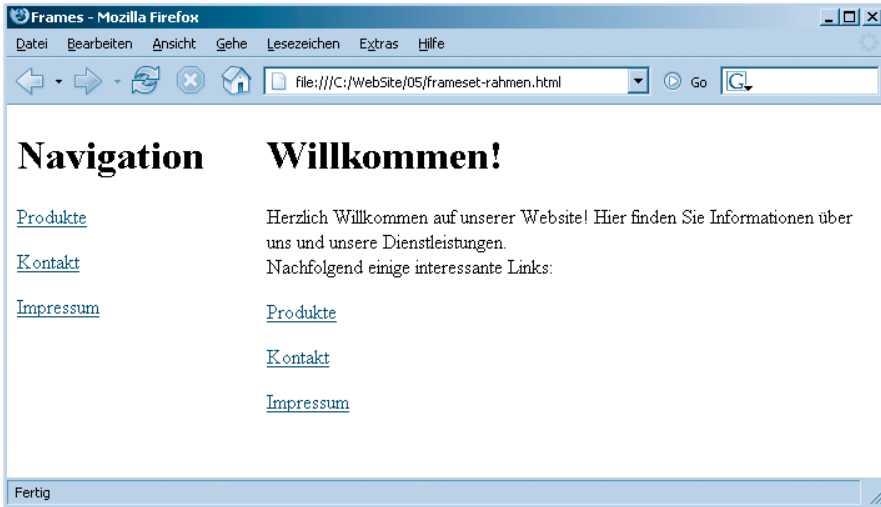
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<!-- frameset-rahmen.html -->
<html>
<head>
 <title>Frames</title>
</head>
<frameset cols="200,*" frameborder="0">
```

```

<frame src="navigation.html" name="Navi"
 noresize="noresize" />
<frame src="start.html" name="Inhalt" />
</frameset>
</html>

```

Jetzt ist der Rahmen endlich verschwunden.



Abschließend ist noch zu klären, was zu tun ist, wenn innerhalb eines Frames so viel Inhalt steht, dass Scrolling notwendig ist. Werfen Sie einen Blick auf die Abbildung, in der wir die Höhe des Browserfensters etwas verringert haben. Die Inhalte haben nicht mehr genug Platz, deswegen erscheint ein Scrollbalken.





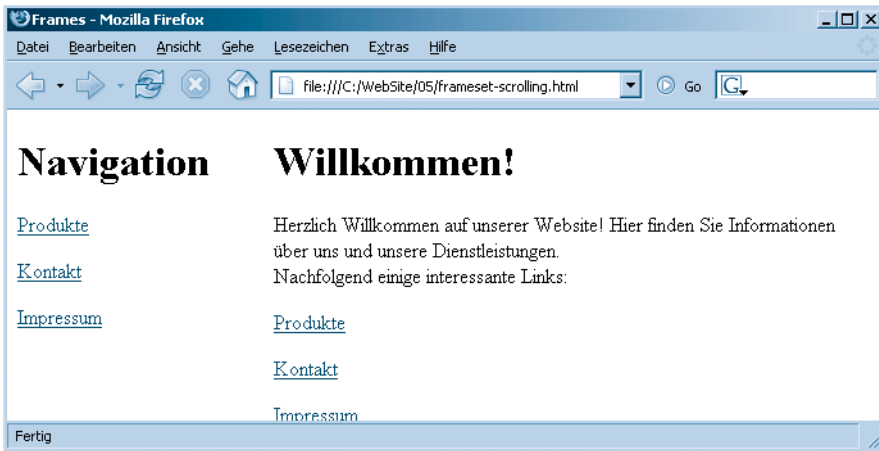
In der Regel ist dieses Verhalten gewünscht. Allerdings nicht immer, beispielsweise bei der Navigation, wo ein Scrollbalken einfach störend wäre. Für diesen Fall können Sie im `scrolling`-Attribut (des `<frame>`-Elements) angeben, wie so etwas gehandhabt werden soll:

- `"auto"`: Der Browser entscheidet automatisch, ob Scrollbalken notwendig sind oder nicht (das ist das Standardverhalten).
- `"no"`: Keine Scrollbalken.
- `"yes"`: Überall Scrollbalken.

So einfach das klingt, so trickreich ist das in der Umsetzung in die Praxis. `"auto"` funktioniert immer, auch `"no"` wird von den Webbrowsern unterstützt. Nur bei `"yes"` scheiden sich die Geister bzw. Webbrowser. Der Internet Explorer zeigt nur senkrechte Scrollbalken an (also keine waagerechten), der Firefox ignoriert `"yes"` komplett. Aber in der Praxis ist das auch nicht gerade relevant, denn wie häufig möchten Sie waagerechte und senkrechte Scrollbalken haben, auch wenn der Inhalt der zugehörigen Webseite breit und hoch genug ist.

Im folgenden Code sehen Sie, dass bei beiden Frames das Scrolling deaktiviert worden ist. Beim Verändern der Fenstergröße wird das im rechten Frame spürbar. Wie bereits erwähnt, macht das hauptsächlich in Frames für Navigation oder Werbung Sinn, aber nicht in Inhaltsframes.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<!-- frameset-scrolling.html -->
<html>
<head>
 <title>Frames</title>
</head>
<frameset cols="200,*" frameborder="0">
 <frame src="navigation.html" name="Navi"
 scrolling="no" />
 <frame src="start.html" name="Inhalt"
 scrolling="no" />
</frameset>
</html>
```



## Frames verschachteln

So weit, so gut. Sie können jetzt also mehrere nebeneinander liegende Frames erstellen und auch mehrere untereinander liegende. Beides zusammen ist ein wenig schwieriger zu verstehen, aber dann auch sehr simpel umzusetzen.

Erinnern wir uns: Mit einem `<frameset>`-Element können Sie den Browserbereich in mehrere Frames unterteilen, entweder nebeneinander (`cols`) oder untereinander (`rows`). Wenn jetzt aber der Bereich sowohl horizontal als auch vertikal geteilt werden muss, müssen Sie beide Methoden kombinieren.

Und das geht so: Zeichnen Sie sich am besten auf, wie die Frame-Struktur aussehen soll. Gehen wir zum Beispiel von vorhin zurück: links Navigation, rechts Inhalt. Angenommen, Sie möchten zur teilweisen Refinanzierung Ihres Webangebots noch an einer Stelle ein Werbebanner unterbringen, beispielsweise über dem Inhalt. Dann muss also der Frame, in dem anfangs die Datei `start.html` geladen wird, in zwei Teile geteilt werden: oben die Werbung, unten der Inhalt.

Das klingt doch genauso wie zuvor das Beispiel mit den untereinander liegenden Frames. Und genau das ist hier der Clou: Um Frames zu verschachteln, ersetzen Sie an einer beliebigen Stelle das `<frame>`-Element durch ein `<frameset>`-Element. Dann sieht das Listing wie folgt aus:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<!-- frameset-verschachtelt.html -->
<html>
```



```

<head>
 <title>Frames</title>
</head>
<frameset cols="200,*" frameborder="0">
 <frame src="navigation.html" name="Navi" />
 <frameset rows="50,*" frameborder="0">
 <frame src="werbung.html" scrolling="no" />
 <frame src="start.html" name="Inhalt" />
 </frameset>
</frameset>
</html>

```

Fehlt abschließend nur noch das Dokument *werbung.html*, das Sie im Folgenden abgedruckt finden:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- werbung.html -->
<html>
<head>
 <title>Werbung</title>
</head>
<body>
 <p style="font-weight: bold;">Kaufen! Kaufen! Kaufen!</p>
</body>
</html>

```



Noch einmal zusammengefasst, was in dem Listing *frameset-verschachtelt.html* passiert:

- Durch `<frameset cols="...">` wird der Browserbereich in zwei Bereiche aufgeteilt, einer links, einer rechts.
- Im linken Bereich wird die Datei *navigation.html* geladen.
- Im rechten Bereich wird mit `<frameset rows="...">` ein weiteres Frameset geladen, das genau diesen Bereich in einen oberen und einen unteren Bereich aufteilt.
- Im rechten oberen Bereich wird die Datei *werbung.html* geladen.
- Im rechten unteren Bereich wird die Datei *start.html* geladen.

Sie sehen also: Gar nicht so schwer, wenn man einmal einen Frame erstellt hat.

## Auf Frames verlinken

Die Frame-Datei sieht im Browser auch ganz gut aus; der positive Ersteindruck schlägt aber in (berechtigten) Ärger um, wenn Sie versuchen, auf einen der Links im linken Frame zu klicken. Die gewünschte Seite wird zwar geladen, aber ... ebenfalls im linken Frame. Werfen Sie einen Blick auf die Abbildung: Nach einem Klick auf den Link »Produkte« in der Navigation wird die Produktseite angezeigt, aber leider am falschen Ort.



Auch das lässt sich beheben. Dem Webbrowser ist kein Vorwurf zu machen, denn der kann ja nicht ahnen, in welchem Frame der Link angezeigt werden soll. Deswegen geht der Webbrowser nach einem ganz einfachen Prinzip vor: Der Link wird im aktuellen Frame (oder Fenster) geöffnet.

Aber erinnern Sie sich noch an *Kapitel 2*? Dort konnten Sie mit dem `target`-Attribut festlegen, ob sich das Linkziel im selben oder in einem neuen Fenster öffnet. Als Attributswert gibt es die folgenden Möglichkeiten:

- `_blank` – in einem neuen Fenster
- `_self` – im aktuellen Fenster
- `_top` – bei Frames im kompletten Anzeigebereich des Webbrowsers (löst also die Frame-Struktur auf)
- `_parent` – bei Frames im übergeordneten Frameset (bei Verschachtelung)

Es gibt noch einige weitere, selten verwendete Möglichkeiten (die auch nicht von allen Browsern unterstützt werden). Die pfiffigste Möglichkeit jedoch ist es, einen Frame-Namen anzugeben. Wenn Sie das tun, öffnet der Browser das Linkziel in dem Frame mit dem entsprechenden Namen. Jetzt wissen Sie auch, wozu das `name`-Attribut im `<frame>`-Tag gut ist.

Also, an die Tat: Passen Sie die Datei *navigation.html* an und setzen Sie bei allen Links das `target`-Attribut. Ein kurzer Blick in die Frameset-Datei klärt auf: Der Inhaltsframe hat den Namen (`name`-Attribut) "Inhalt".

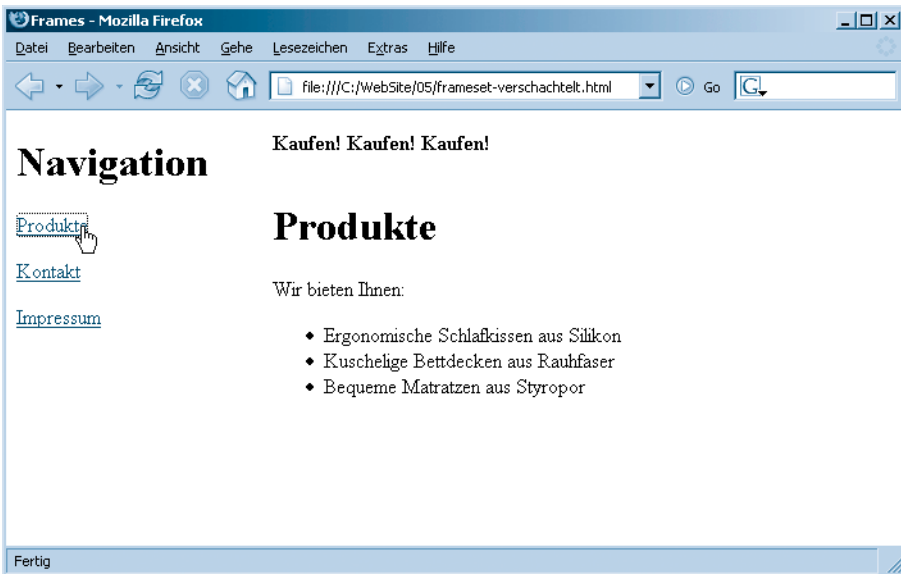
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- navigation.html -->
<html>
<head>
 <title>Navigation</title>
</head>
<body>
 <h1>Navigation</h1>
 <p>
 <a href="produkte.html"
 target="Inhalt">Produkte
 </p>
 <p>
 <a href="kontakt.html"
 target="Inhalt">Kontakt
 </p>
```

```

<p>
 <a href="impressum.html"
 target="Inhalt">Impressum
</p>
</body>
</html>

```

Die Datei *navigation.html* im Download-Archiv ist bereits dementsprechend angepasst. Und wie Sie im Browser sehen, funktioniert das Ganze jetzt auch wunderbar: Das Linkziel öffnet sich im rechten Frame.



### Tipp

Wenn Sie das Linkziel für eine komplette HTML-Seite festlegen möchten, können Sie folgendes Element im `<head>`-Abschnitt des Dokuments verwenden:

```
<base target="Inhalt" />
```

Jetzt öffnen sich automatisch alle Links im Frame namens "Inhalt". Soll sich einer der Links in dem Dokument doch auf der aktuellen Seite öffnen, müssen Sie den zuvor erwähnten Wert `"_self"` verwenden.



## Frames mitten auf der Seite: Iframes

Abschließend noch eine weitere Einsatzmöglichkeit von Frames: mitten auf einer Seite, in ein Dokument eingebettet. Das ermöglicht den gleichen bequemen Zugriff auf den Frame bei Links, zwingt den Webdesigner aber nicht in allzu starke gestalterische Grenzen.

Hierfür benötigen Sie einen so genannten Iframe. Dazu verwenden Sie das `<iframe>`-Element und befolgen diese Schritte:

- 1 Geben Sie im `src`-Attribut den URL des zu ladenden Dokuments an.

```
<iframe src="start.html" />
```

- 2 Geben Sie dem Iframe im `name`-Attribut einen (eindeutigen) Namen.

```
<iframe src="start.html" name="Inhalt" />
```

- 3 Passen Sie die Breite und Höhe des Iframes an – das geht am besten mit CSS, also `width` und `height`.

```
<iframe src="start.html" name="Inhalt"
 style="width: 600px; height: 400px;" />
```

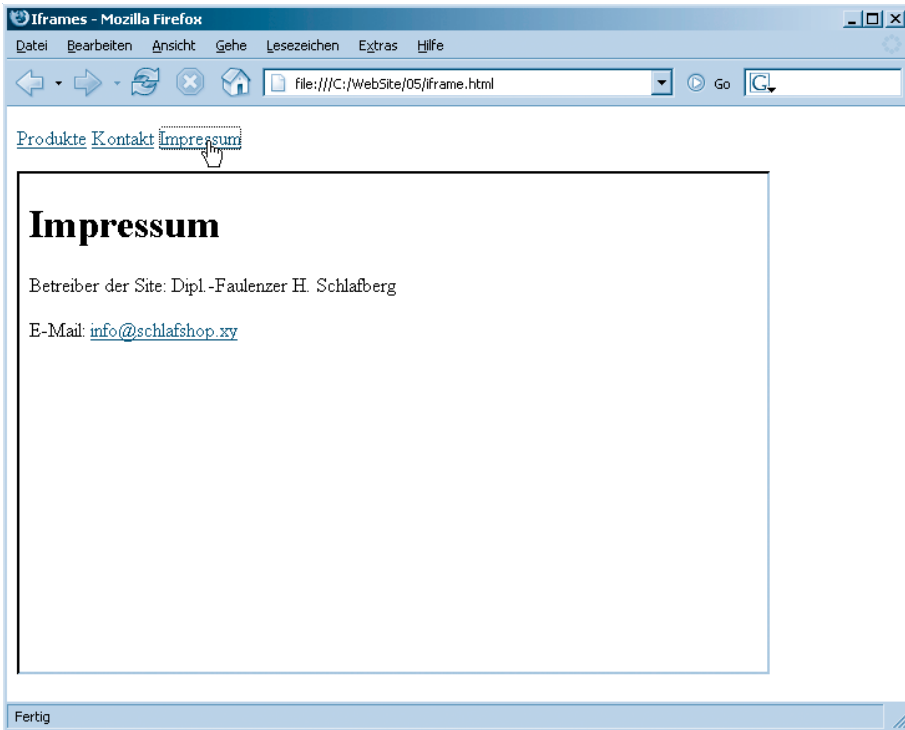
- 4 Sorgen Sie dafür, dass Sie bei allen Links das `target`-Attribut auf den Namen des Iframes setzen.

```
Kontakt
```

Hier der komplette Code:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- iframe.html -->
<html>
<head>
 <title>Iframes</title>
</head>
<body>
 <p>
 <a href="produkte.html"
 target="Inhalt">Produkte
 <a href="kontakt.html"
 target="Inhalt">Kontakt
 <a href="impressum.html"
 target="Inhalt">Impressum
 </p>
```

```
<iframe src="start.html" name="Inhalt"
 style="width: 600px; height: 400px;" />
</body>
</html>
```



### Tipp

*Wenn Ihnen der Rahmen um den Iframe nicht gefällt, verwenden Sie wie bei »normalen« Frames auch das Attribut `frameborder` und setzen Sie es auf "0".*

Sie sehen also – Frames sind gar nicht so schwer, und es lassen sich damit einige pfiffige Effekte realisieren.





## Nutzloses Wissen

In Oklahoma (einem Staat in der Mitte der USA) ist die Waljagd verboten.

## Kapitel 6

# Feedback mit Formularen einholen

The screenshot shows a web browser window with the title 'Formulare - Mozilla Firefox'. The address bar shows the file path 'file:///C:/WebSite/06/feedbackformular-mail.htm'. The form contains the following elements:

- Radio buttons for 'Herr' and 'Frau'.
- Text input fields for 'Vorname:', 'Nachname:', and 'E-Mail:'.
- A text area for 'Kommentar zu meiner Website:' with the placeholder text 'Was ich schon immer mal sagen wollte ...'.
- A dropdown menu for 'Woher kennen Sie diese Website?' with options: 'Suchmaschine', 'Mund-zu-Mund-Propaganda', and 'Fernsehwerbung'.
- A checkbox for 'Ja, ich möchte den Newsletter abonnieren!'.
- A large blue button labeled 'Klick mich ...'.
- A status bar at the bottom labeled 'Fertig'.

*Zugegeben, der Begriff »Formular« klingt relativ erschreckend und weckt Assoziationen zur überfälligen Steuererklärung oder zur Beantragung von Eintrittskarten für ein sportliches Großereignis. Tatsächlich aber sind Formulare auf Websites das A und O, um von den Besuchern der Website Feedback einzuholen.*

*Formulare sind im World Wide Web gang und gäbe; an vielen Stellen fällt es nicht einmal auf, dass hier im Hintergrund ein Formular werkelt. In diesem Kapitel erfahren Sie nicht nur, wie Sie mit Formularen arbeiten können, sondern auch, wo diese sonst noch auftauchen. Doch genug der langen Vorrede, hinein in ein paar Beispiele.*

**Das können Sie schon:**

|                                  |     |
|----------------------------------|-----|
| Formatierung von Text            | 56  |
| Einsatz von Links                | 75  |
| Aufzählungslisten                | 90  |
| Listen mit Nummerierung          | 92  |
| Tabellen zur Datenstrukturierung | 95  |
| Grafiken einbauen                | 116 |
| Frames erstellen                 | 134 |

**Das lernen Sie neu:**

|                                |     |
|--------------------------------|-----|
| Formulardaten übermitteln      | 160 |
| Formular versenden             | 162 |
| Formularelemente               | 165 |
| Formulare per E-Mail versenden | 187 |

## Wohin mit den Daten?

Doch zu früh gefreut. Ein wenig theoretischer Vorbau ist schon notwendig. Die Daten, die ein Surfer in das Formular einträgt, müssen nämlich irgendwie verschickt werden. Dafür gibt es mehrere Möglichkeiten, so genannte Versandmethoden. Webbrowser unterstützen an dieser Stelle die folgenden zwei:

- POST: Die Daten werden als Teil der HTTP-Anforderung an den Webserver geschickt (siehe Einleitungskapitel).
- GET: Die Daten werden an die Adresse (den URL) angehängt.

Die Versandmethode GET wird unter anderem von Google eingesetzt: Die Suchanfrage taucht in der Adresse auf – wie Sie in der Abbildung sehen, taucht hinter *www.google.de* irgendwo der zuvor eingegebene Suchbegriff auf.



Jedoch ist ansonsten GET eher selten im Einsatz. Dass die Daten im Klartext im URL stehen, ist nämlich nicht immer ein gewünschter Effekt. Stellen Sie sich vor, Sie würden sich bei Ihrem Webmail-Anbieter einloggen und Ihr Passwort wäre im Klartext aus der Adresse ersichtlich. Jeder, der Ihnen über die Schulter schaut, kennt dann Ihre Zugangsdaten.

### Hinweis

*Dass sich das noch nicht überall herumgesprochen hat, stellte sich vor einigen Jahren heraus. Das Webmail-Programm eines damals noch recht populären Internetanbieters verwendete tatsächlich GET zum Einloggen. Konsequenz: Das Passwort war Teil der Adresse und wurde beispielsweise auch in der Statuszeile des Browsers eingeblendet – denn beim Laden einer Seite landet der URL genau dort.*



Leider ist POST auch nicht perfekt. POST-Seiten lassen sich nicht als Favorit (Lesezeichen) im Browser ablegen. Deswegen können Sie zwar das Ergebnis einer Google-Suche im Browser abspeichern, nicht aber den Posteingang Ihres Webmail-Anbieters. Trotzdem, in der Praxis wird meistens POST verwendet.

### Tipp

*Bei den meisten Beispielen in diesem Kapitel verwenden wir dennoch GET. Dann nämlich sehen Sie anhand der Adresse, welche Daten übergeben worden sind. Das erleichtert das Testen ungemein.*

So, jetzt ist schon einmal klar, wie die Daten verschickt werden. Fehlt nur noch, wohin damit. Da gibt es mehrere Möglichkeiten. Meist verschickt man die Formulardaten an ein Programm auf einem Webserver, das diese dann in eine Datenbank schreibt oder per E-Mail verschickt. Dazu später mehr. Wichtig ist zunächst nur, wie Sie das in HTML umsetzen können.

Zunächst zum Formularrahmen: Dazu verwenden Sie das HTML-Element `<form>`. Alle Formularelemente – die wir im Folgenden behandeln – müssen zwischen `<form>` und `</form>` stehen. Das sind die einzigen Bedingungen. Beim Rest sind Sie gestalterisch frei, Sie können also die Formularelemente mit Tabellen oder über CSS gestalten und anordnen.

Im `<form>`-Tag bestimmen Sie die wichtigsten Rahmenparameter für das Formular. Dazu gehören:

- Das Versandziel, das Sie im Attribut `action` angeben.
- Die Versandmethode, die Sie im Attribut `method` schreiben.

Hier ein (noch leeres) Formular, das per GET verschickt wird, und zwar an die Datei *formulardaten.html*.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- formular-leer.html -->
<html>
<head>
 <title>Formulare</title>
</head>
<body>
 <form action="formulardaten.html" method="get">
 </form>
</body>
</html>
```

## Tipp

Die Datei **formulardaten.html** befindet sich im Code-Archiv zu diesem Buch, siehe Vorwort. In dieser steht etwas JavaScript-Code, der alle per GET übergebenen Daten ermittelt und dann ausgibt. Das hilft dann ein wenig bei der Fehlersuche. Sie müssen sich diese Datei herunterladen und sie im selben Verzeichnis abspeichern wie Ihre Formulardateien. Beachten Sie aber auch die Hinweise in Kapitel 9 bezüglich JavaScript, das auch deaktiviert werden kann.

Bevor wir fortfahren, noch ein wichtiger Hinweis: Im Folgenden sehen Sie, wie Sie die verschiedensten Formularfelder in eine HTML-Seite einbauen können. Am Ende übergeben Sie die eingetragenen Daten irgendwohin. Für jedes Formularfeld werden zwei Informationen übergeben: der Name des Felds und sein Wert. Letzteres ist meist das, was der Benutzer in das Formular eingibt. Damit Sie später auch noch nachvollziehen können, welche Daten in welches Formularfeld eingegeben worden sind, ist es sehr wichtig, sinnvolle Namen für die einzelnen Formularfelder zu wählen. Doch nun zu den einzelnen Formularfeldern.

## Formulare versenden

Das wohl wichtigste Formularelement ist die Versenden-Schaltfläche: ein Button, mit dem Sie auf Mausklick (oder auch mit der Tastatur) das Formular an seinen Bestimmungsort verwenden können.

The screenshot shows the Deutsche Bahn website interface. At the top, there's a navigation bar with links like 'Fahrpläne', 'Fahrkarten', 'Reisebüro', 'Angebote', 'Service', and 'Int.'. Below this is a search section titled 'Reiseauskunft - Tickets' with input fields for 'von:', 'nach:', 'Datum:' (set to 21.03.2005), and 'Uhrzeit:' (set to 8:59). There are radio buttons for 'Abfahrt' and 'Ankunft', and a 'Suchen' button. To the right, there's a section 'Aktuelle Aktionsangebote' featuring 'Europa-Spezial' with a list of destinations and prices: Amsterdam ab 19 EUR, Brüssel ab 19 EUR, Luxemburg ab 19 EUR, and Prag ab 19 EUR. Below this, it says 'Mit Europa-Spezial in 17 europäische Städte zu besonders attraktiven Preisen. Köln - Brüssel ab 19,- EUR.' There's also a small image of a train and a text box about 'Die BahnCard 25'.



So eine Schaltfläche finden Sie fast überall, unter anderem auch bei der Online-Reiseauskunft der Deutschen Bahn: Der blaue (im Druck leider nur graue) Button mit der Aufschrift »Suchen« ist die Versenden-Schaltfläche.

Dahinter verbirgt sich das HTML-Element `<input type="submit" />`. Das Tag `<input>` wird für viele Formularfeldtypen verwendet; im Attribut `type` geben Sie den spezifischen Typ des Felds an. Bei `type="submit"` liegt eine Versenden-Schaltfläche vor. Ein Mausklick verschickt das Formular. Vor dem Einsatz sollten Sie das Element aber noch ein wenig feinjustieren:

**1** Geben Sie im Attribut `value` die Beschriftung der Schaltfläche an – natürlich ordentlich HTML-kodiert:

```
<input type="submit" value=">Hier klicken<" />
```

**2** Verwenden Sie das Attribut `name` dazu, der Schaltfläche einen (eindeutigen!) Namen zu geben:

```
<input type="submit" name="Versand" />
```

### Tipp

*Das `name`-Attribut ist bei Versenden-Schaltflächen nicht notwendig. Allerdings ist es eine gute Idee, es immer zu setzen, wenn Sie mehrere Versenden-Schaltflächen haben. Ein Beispiel dafür ist bei Google zu sehen: Eine Schaltfläche sucht, die andere (»Auf gut Glück«) springt nach der Suche direkt zum besten Suchergebnis weiter. Wenn Sie eindeutige Namen vergeben haben, können Sie so feststellen, welche der Schaltflächen angeklickt worden ist.*



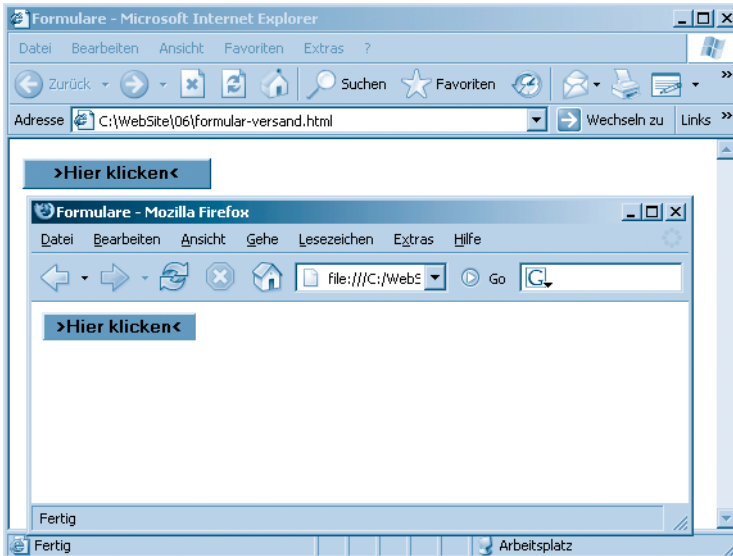
### 3 Passen Sie das Erscheinungsbild der Schaltfläche an – natürlich im style-Attribut.

```
<input type="submit" style="background-color:red;
font-weight:bold;" />
```

Das Ganze noch mal im Zusammenhang:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- formular-versand.html -->
<html>
<head>
<title>Formulare</title>
</head>
<body>
<form action="formulardaten.html" method="get">
 <input type="submit"
 value=">Hier klicken<"
 name="Versand"
 style="background-color:red; font-weight:bold;" />
</form>
</body>
</html>
```

Wie Sie sehen, wurde der Schaltfläche keine Breite gegeben. Das führt dann dazu, dass sie in verschiedenen Browsern etwas unterschiedlich aussieht. Deswegen unser Tipp: Wenn es um pixelgenaues Design geht, müssen Sie auch den width-CSS-Befehl setzen.

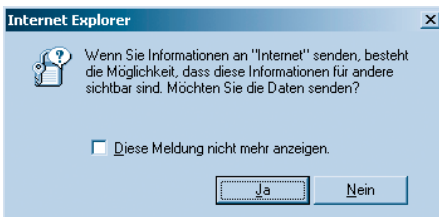




Probieren Sie es aus und verschicken Sie das (sonst leere) Formular. Sie erhalten auf der Ausgabeseite für das einzige Formularfeld – der Schaltfläche – die zwei gewünschten Informationen: den Feldnamen und den Feldwert, hier die Beschriftung der Schaltfläche.



Auf manchen Browserkonfigurationen und -systemen erhalten Sie eine Warnmeldung, wenn ein Formular verschickt werden soll. Dagegen können Sie als HTML-Designer leider nichts unternehmen, das ist eine Sicherheitsmaßnahme der Browserhersteller.



## Textfelder

Ein sehr häufiges Formularelement ist ein Texteingabefeld. Sie finden es beispielsweise bei Google: das Feld, in das der Suchbegriff geschrieben wird.



Dahinter steckt das HTML-Element `<input type="text" />`. Das »kritische« Attribut `type` hat diesmal den Wert `"text"`, das steht für ein einzeliges Texteingabefeld. Das können Sie auf verschiedenste Arten und Weisen anpassen:

**1** Geben Sie im Attribut `name` eine eindeutige Bezeichnung für das Formularfeld an. Der gewählte Name `"Textfeld"` ist dabei nur ein Beispiel und frei wählbar. Wichtig für den Feldtyp ist lediglich das Attribut `type="text"`.

```
<input type="text" name="Textfeld" />
```

**2** Geben Sie mit dem Attribut `size` an, wie breit das Formularfeld ist.

Die Breite ist aber nur ein ungefährender Wert (in Zeichen), der von Browser zu Browser unterschiedlich ist. Besser ist es, Sie setzen auf CSS und geben im CSS-Befehl `width` die gewünschte Feldbreite an:

```
<input type="text" style="width:120px;" />
```

**3** Geben Sie im Attribut `maxlength` an, wie viele Zeichen der Benutzer maximal in das Feld eingeben darf.

Sinnvoll ist das beispielsweise, wenn Sie das Textfeld für einen Benutzernamen verwenden, der nur eine bestimmte Länge haben darf.

```
<input type="text" maxlength="20" />
```

**4** Belegen Sie schließlich im Attribut `value` das Feld mit einem Wert vor. Achtung, die Eingabe in diesem Attribut muss korrekt HTML-kodiert sein:

```
<input type="text" value=">Hier eintragen<" />
```

Nachfolgend ein komplettes Listing:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- formular-textfeld.html -->
<html>
<head>
 <title>Formulare</title>
</head>
<body>
 <form action="formulardaten.html" method="get">
 <input type="text"
 style="size: 100px"
 maxlength="20"
 value="> Hier eintragen <" />


```

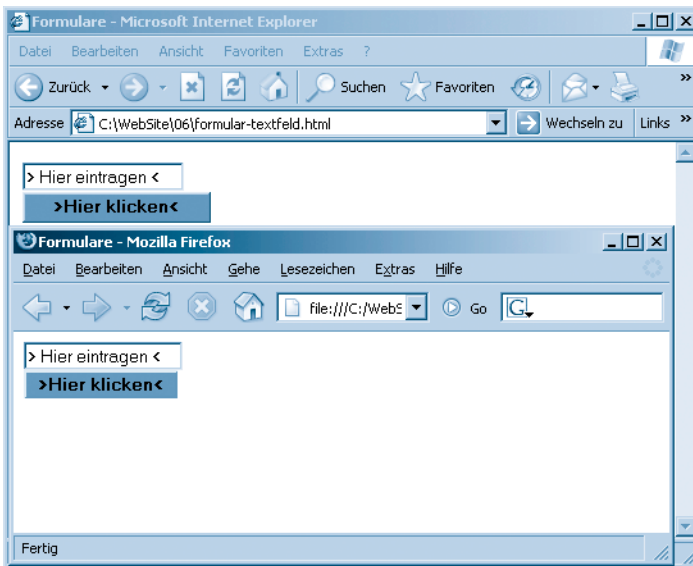


```

 <input type="submit"
 value=">Hier klicken<"
 name="Versand"
 style="background-color:red; font-weight:bold;" />
 </form>
</body>
</html>

```

Im Webbrowser sehen Sie das Textfeld, das in der Tat in verschiedenen Browsern zumindest gleich breit ist.

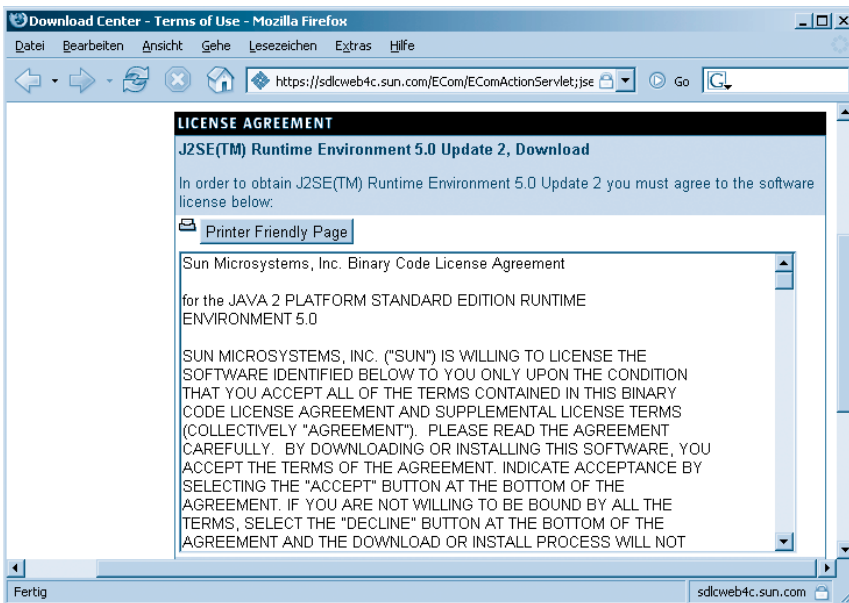


Nach dem Versand des Formulars sehen Sie auch, welche Daten an das Auswertungsskript übergeben werden: Beim Textfeld ist es seine Bezeichnung (Wert des name-Attributs) sowie das, was ins Feld eingegeben worden ist.



## Mehrzeilige Textfelder

Wenn eine Textzeile nicht mehr ausreicht, müssen es mehrere sein. Dazu gibt es ein eigenes HTML-Tag, das ausnahmsweise nicht `<input>` heißt. Die Rede ist von `<textarea>`. Sie finden das Element auf vielen Websites, beispielsweise wenn Sie eine aktuelle Java-Version herunterladen und installieren möchten. Im Textfeld stehen dann die Nutzungsbedingungen. Die meisten mehrzeiligen Textfelder sind allerdings dazu da, dass dort vom Benutzer etwas eingetragen werden kann.



Die Anpassung ist sehr einfach:

- 1 Geben Sie im Attribut `cols` die Breite des Textfelds (in Zeichen) an.
- 2 Geben Sie im Attribut `rows` die Höhe des Textfelds (in Zeilen) an.

Natürlich können Sie auch die CSS-Befehle `width` und `height` verwenden.

```
<textarea cols="50" rows="8"></textarea>
```

- 3 Wenn das Textfeld vorausgefüllt werden soll, geben Sie den entsprechenden Text **zwischen** `<textarea>` und `</textarea>` an:

```
<textarea>Bitte tippen Sie munter drauflos ...
Ich freue mich auf Ihr Feedback</textarea>
```



#### 4 Packen Sie Stilangaben ins style-Attribut:

```
<textarea style="background-color:orange;
font-weight:bold;"></textarea>
```

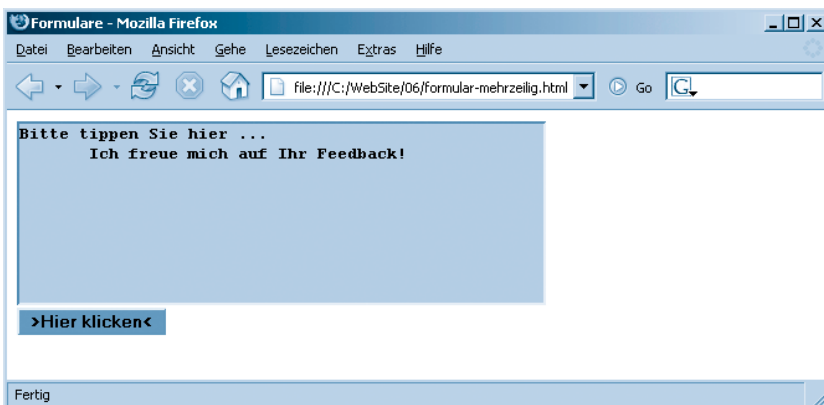
#### 5 Außerdem benötigt das Textfeld noch einen ordentlichen Namen, verwenden Sie dazu das name-Attribut:

```
<textarea name="Mehrzeilig"></textarea>
```

Nachfolgend ein komplettes Formular:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- formular-mehrzeilig.html -->
<html>
<head>
<title>Formulare</title>
</head>
<body>
<form action="formulardaten.html" method="get">
<textarea
cols="50" rows="8"
style="background-color:orange; font-weight:bold;"
name="Mehrzeilig">Bitte tippen Sie hier ...
Ich freue mich auf Ihr Feedback!</textarea>

<input type="submit"
value=">Hier klicken<"
name="Versand"
style="background-color:red; font-weight:bold;" />
</form>
</body>
</html>
```

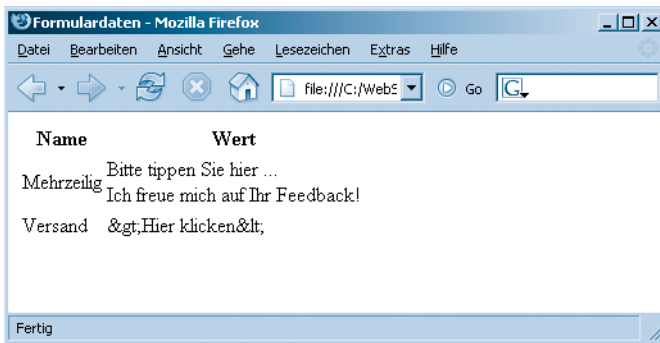


## Hinweis

*Sie sehen hier zwei Unterschiede zu herkömmlichem HTML:*

- *Zeilenumbrüche innerhalb eines mehrzeiligen Textfelds werden tatsächlich als solche dargestellt.*
- *Mehrere Leerzeichen hintereinander werden innerhalb eines mehrzeiligen Textfelds ebenfalls als solche dargestellt.*

Der Versand schließlich bringt zu Tage, was übermittelt wird: der Name des Felds (name-Attribut) sowie das, was eingetragen worden ist – hier der Vorgabetext, weil wir nichts geändert haben. Ansonsten wäre es der Text, den der Surfer eingegeben hat.



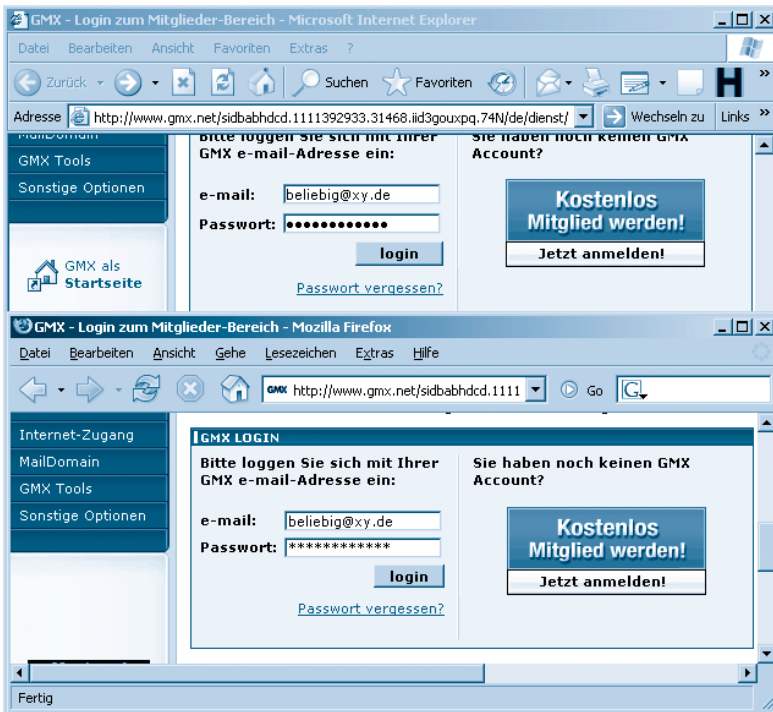
## Passwortfelder

Verwenden Sie Webmail wie etwa Hotmail, Yahoo! Mail oder GMX? Falls ja: Egal, wo Sie sich einloggen, praktisch überall stoßen Sie auf Passwortfelder. Das sind einzeilige Textfelder, bei denen es einen einzigen Unterschied zu herkömmlichen einzeiligen Textfeldern gibt: Die Eingabe ist maskiert. Das bedeutet, dass Sie anstelle des Zeichens, das Sie eingeben, ein Sternchen sehen – oder ein anderes grafisches Symbol, beispielsweise einen runden Punkt.

## Hinweis

*Es handelt sich hierbei lediglich um einen **Sichtschutz** bei der Eingabe, nicht um eine Art von Verschlüsselung von Daten. Das sehen Sie später beim Testen – das Passwort wird im Klartext verschickt.*





Die relevanten Attribute des Passwortfelds sind dieselben wie beim einzeiligen Textfeld `<input type="text">`. Lediglich eine Sache ändert sich: der Formularfeldtyp. Mit `type="password"` im `<input>`-Tag können Sie ein Passwortfeld erzeugen. Mit dem `value`-Attribut ist eine Voraussfüllung möglich, die in der Praxis jedoch so gut wie nie Sinn macht – wir verwenden das hier lediglich aus Beispielszwecken. Und so sieht das in einem kompletten Listing aus:

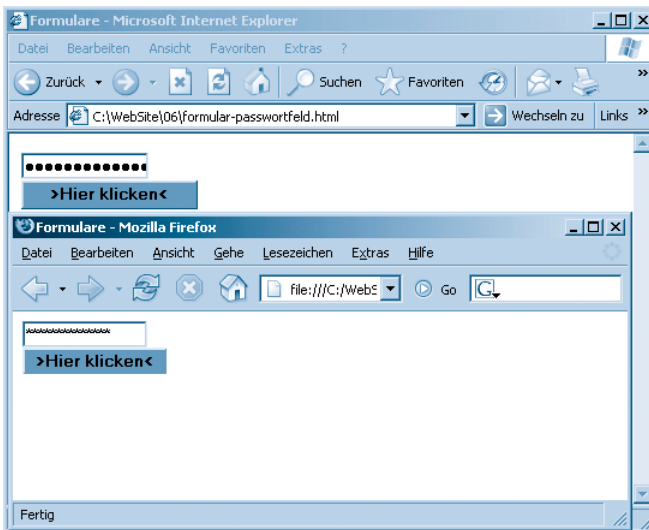
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- formular-passwortfeld.html -->
<html>
<head>
 <title>Formulare</title>
</head>
<body>
 <form action="formulardaten.html" method="get">
 <input type="password"
 name="Passwortfeld"
 style="width:100px"
 maxlength="15"
 value="streng geheim!" />
```

```

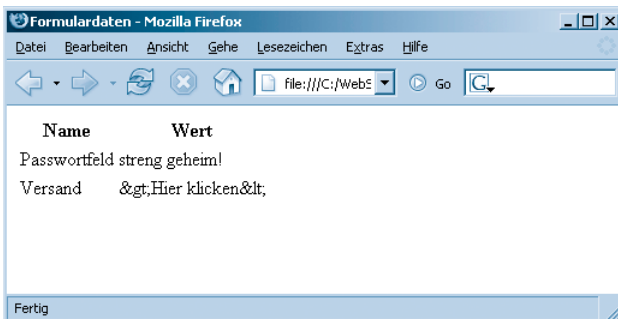

<input type="submit"
 value=">Hier klicken<"
 name="Versand"
 style="background-color:red; font-weight:bold;" />
</form>
</body>
</html>

```

Beim Testen in verschiedenen Browsern sehen Sie die Unterschiede in der Darstellung. Aber eines ist immer dasselbe: Die Eingabe selbst ist maskiert. Und wieder ein Argument dafür, die Feldbreite mit dem CSS-Befehl `width` anzugeben: Ein Kreissymbol ist mitunter breiter als ein Sternchen.



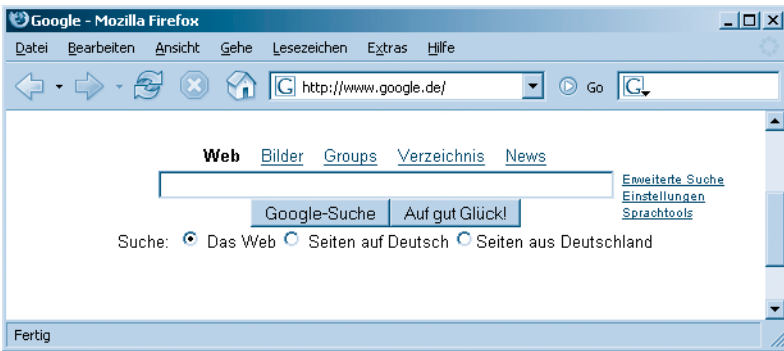
Nach dem Versand sehen Sie auch am URL, dass das Passwort im Klartext vorliegt.





## Radiobuttons

Formulare sind ja meist dazu da, Informationen vom Benutzer abzufragen. Manchmal gibt ein Formular auch schon mögliche Antworten vor. Wenn nur eine davon ausgewählt werden soll, sind Radiobuttons sehr geeignet. Beispielsweise wieder einmal bei Google: Dort wählen Sie aus, welcher Teil durchsucht werden soll – der mit deutschsprachigen Seiten, der mit Seiten aus Deutschland oder der Gesamtkatalog.



Auf Deutsch heißen die »Radioknöpfe« auch **Optionsfelder**, sprich ein Feld, mehrere Optionen. Und genau so ist es bei Radiobuttons: Von allen Elementen einer Felderguppe kann immer nur eines ausgewählt sein. Wenn Sie also hintereinander zwei Radiobuttons auswählen, entfernt der Klick auf den zweiten Button die Markierung des ersten.

Wie der ausgewählte Zustand eines Radiobuttons aussieht, hängt wieder einmal vom Browser und auch dem Betriebssystem ab; meist ist das ein Knödel in einem Kreis.

Realisiert wird das mit dem Element `<input type="radio" />` und mehreren Anpassungsschritten:

**1** Geben Sie zunächst jedem Radiobutton einen eindeutigen Namen im `value`-Attribut.

Richtig gelesen, im `value`-Attribut, **nicht** im `name`-Attribut!

```
<input type="radio" value="s" />
```

**2** Geben Sie jetzt den Gruppennamen im `name`-Attribut an:

```
<input type="radio" name="Gruppe" value="s" />
<input type="radio" name="Gruppe" value="w" />
```

So nämlich entscheidet der Browser, welche Radiobuttons zur selben Gruppe gehören. Es kann ja möglich sein, dass es in ein- und demselben HTML-Formular mehrere Radiobuttongruppen gibt.

### Tipp

*Noch einmal zum Merken: Im name-Attribut geben Sie den **Gruppennamen** an. Den eigentlichen **Wert** (Bezeichner) des Radiobuttons packen Sie in das value-Attribut.*

**3** Wenn ein Radiobutton vorausgewählt werden soll, verwenden Sie das Attribut checked.

Wenn Sie das tun, stellen Sie im Übrigen auch sicher, dass der Benutzer mindestens einen der Radiobuttons auswählt.

```
<input type="radio" checked="checked" />
```

**4** Vergessen Sie nicht, den Radiobutton auch noch zu beschriften, in der Regel direkt links oder rechts neben dem Formularfeld:

```
<input type="radio" /> Beschriftung
```

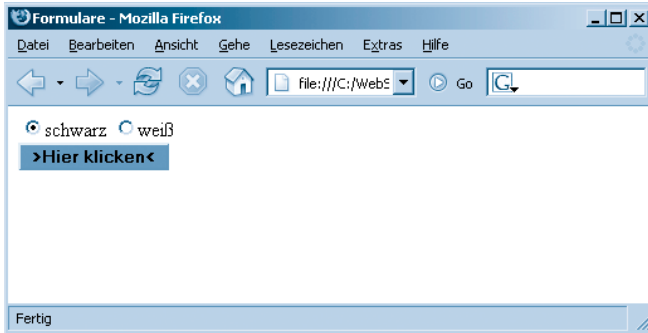
Und hier das Ganze im Zusammenhang:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- formular-radiobutton.html -->
<html>
<head>
 <title>Formulare</title>
</head>
<body>
 <form action="formulardaten.html" method="get">
 <input type="radio"
 name="Gruppe"
 value="s"
 checked="checked" />schwarz
 <input type="radio"
 name="Gruppe"
 value="w" />weiß;

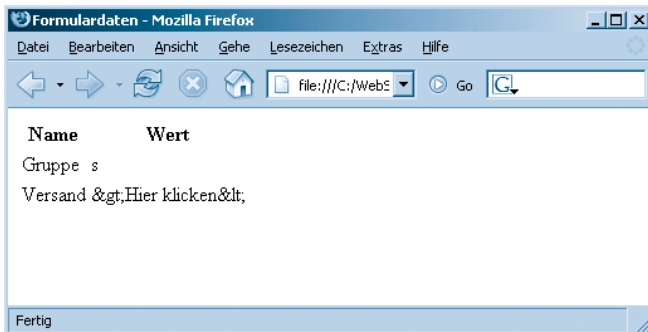
 <input type="submit"
 value=">Hier klicken<";
 name="Versand"
 style="background-color:red; font-weight:bold;" />
```



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

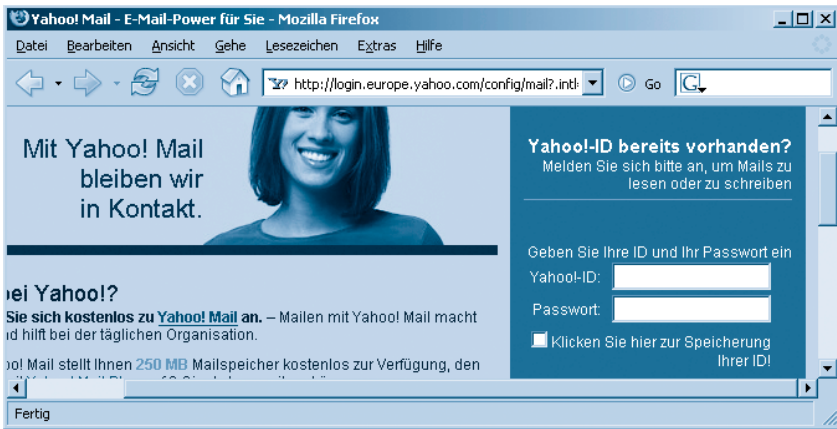


Im Test sehen Sie zwei Dinge. Erstens, es lässt sich tatsächlich nur einer der Radiobuttons auswählen. Und zweitens: Beim Versand wird der Wert (`value`-Attribut) desjenigen Radiobuttons übermittelt, der vorausgewählt ist. Deswegen ist es sehr wichtig, dass Sie ein aussagekräftiges `value`-Attribut verwenden.



## Checkboxes

Eine etwas andere Ausrichtung haben Checkboxes, auf gut Deutsch **Kontrollkästchen**. Diese treten – im Gegensatz zu Radiobuttons – nicht gruppiert auf, sondern jede Checkbox steht für sich alleine und kann unabhängig vom Rest des Formulars an- und ausgekreuzt werden. Die Checkbox finden Sie unter anderem auf einigen Login-Seiten, bei denen Sie gefragt werden, ob das Passwort gespeichert werden soll (damit Sie sich einen erneuten Login ersparen können). So geschieht das unter anderem auch beim Webmail-Anbieter Yahoo! Mail, der Sie fragt, ob der Benutzername gespeichert werden darf (das Passwort aus Sicherheitsgründen natürlich nicht).



Auch hier gibt es wieder Unterschiede in der Darstellung: Wenn eine Check-box (in der Regel ein kleines Quadrat) angekreuzt ist, befindet sich darin entweder ein Kreuz oder ein Häkchen. Das hängt mal wieder vom Betriebssystem und dem Webbrowser ab.

In HTML lässt sich dieser Effekt sehr leicht umsetzen. Sie benötigen wieder einmal das `<input>`-Element und den richtigen Wert für `type`: "checkbox". Der Rest ist schnell erledigt:

**1** Vergeben Sie zunächst einen eindeutigen Namen, den Sie im `name`-Attribut angeben:

```
<input type="checkbox" name="Check" />
```

**2** Außerdem geben Sie jeder Checkbox noch einen Wert – warum, sehen Sie später. Verwenden Sie dazu das `value`-Attribut:

```
<input type="checkbox" value="aktiv" />
```

**3** Soll die Checkbox bereits vorausgewählt sein, verwenden Sie wie bei Radio-buttons auch das Attribut `checked`:

```
<input type="checkbox" checked="checked" />
```

**4** Abschließend weisen Sie noch eine Beschriftung zu:

```
<input type="checkbox" />Kreuz mich an!
```

In lauffähiger Version sehen Sie die Checkbox hier:

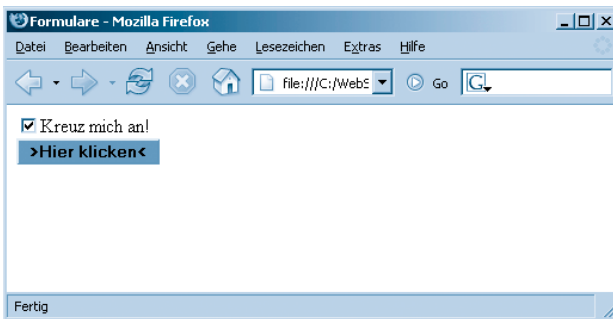
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- formular-checkbox.html -->
```



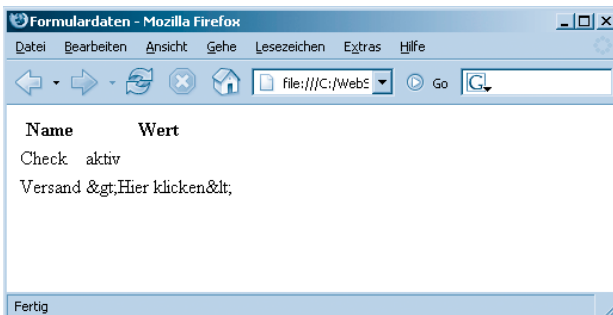
```
<html>
<head>
 <title>Formulare</title>
</head>
<body>
 <form action="formulardaten.html" method="get">
 <input type="checkbox"
 name="Check"
 value="aktiv"
 checked="checked" />Kreuz mich an!

 <input type="submit"
 value=">Hier klicken<"
 name="Versand"
 style="background-color:red; font-weight:bold;" />

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



Beim Versand sehen Sie dann auch, wozu Sie überhaupt einen Wert (`value`) gebraucht haben: Genau der wird nämlich übertragen, wenn die Checkbox ausgewählt worden ist – ansonsten nicht! Geben Sie also einen sinnvollen Wert an (etwa "aktiv" oder "ausgewählt").



## Hinweis

Wenn Sie keinen Wert angeben, versuchen die meisten Browser, aus der Not eine Tugend zu machen, und verschicken als Wert "on", wenn die Checkbox angekreuzt worden ist. Darauf sollten Sie sich aber nicht verlassen. Fazit: Geben Sie einen Wert an!

## Auswahllisten

Wenn Sie den Personen, die Ihr Formular ausfüllen, mehrere Antwortmöglichkeiten schon vorgeben möchten, gibt es noch Alternativen zu Checkboxes und Radiobuttons, die beide bei vielen Antwortmöglichkeiten doch recht viel Platz brauchen. Eine mögliche Alternative besteht in der Verwendung einer Auswahlliste (manchmal auch Pulldown-Menü oder Dropdown-Liste genannt). Sie finden das beispielsweise bei der Homepage der Stadt München (und auch bei zahlreichen anderen Städten). Dort sehen Sie eine Auswahlliste: Auf Mausklick erscheint eine Auswahl möglicher Linkziele; ein Mausklick sorgt dann dafür, dass der Link aufgerufen wird.



**Tipp**

*An dieser Stelle beschäftigen wir uns noch nicht damit, wie Sie die Verlinkung realisieren, sondern lediglich mit der Auswahlliste. In Kapitel 9 jedoch schließen wir das Beispiel ab und bauen den Verlinkungseffekt nach!*

Was steckt dahinter? Ein neues HTML-Element, `<select>`. In mehreren Schritten können Sie dafür sorgen, dass die Liste mit Leben, sprich mit Elementen gefüllt wird.

- 1** Geben Sie zunächst der Auswahlliste einen aussagekräftigen und eindeutigen Namen – natürlich im `name`-Attribut.

```
<select name="Liste">
</select>
```

- 2** Verschönern Sie im `style`-Attribut des `<select>`-Tags die Liste noch etwas.

```
<select style="background-color:yellow;">
</select>
```

- 3** Fügen Sie jetzt die einzelnen Listenelemente zwischen `<select ..>` und `</select>` ein.

Jedes davon besteht aus einem `<option>`-Element. Innerhalb des Elements steht die **Beschriftung** des Eintrags, im `value`-Attribut steht der **Wert**. Alle diese Angaben sind wichtig!

```
<option value="1">Wert 1</option>
```

- 4** Füllen Sie eines der Listenelemente vor.

Verwenden Sie im entsprechenden `<option>`-Element das Attribut `selected`.

```
<option value="2" selected="selected">Wert 2</option>
```

Das war es aber auch schon! Am Ende erhalten Sie ein Formular, das wie folgt aussehen kann:

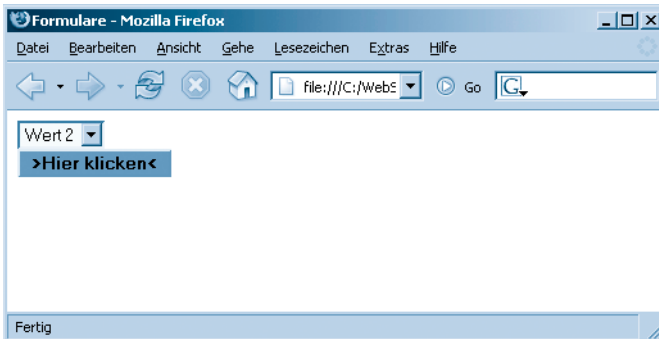
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- formular-liste.html -->
<html>
<head>
<title>Formulare</title>
</head>
```

```

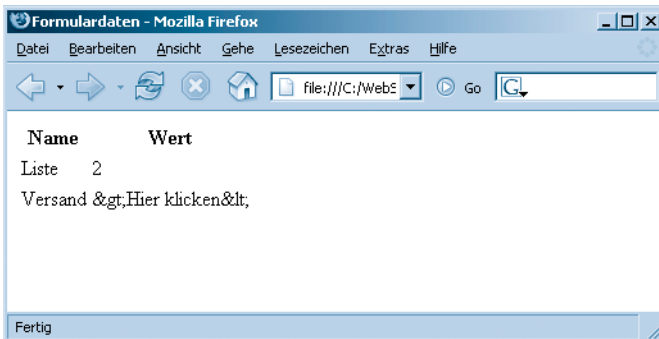
<body>
 <form action="formulardaten.html" method="get">
 <select name="Liste"
 style="background-color:yellow;">
 <option value="1">Wert 1</option>
 <option value="2" selected="selected">Wert 2</option>
 <option value="3">Wert 3</option>
 </select>

 <input type="submit"
 value=">Hier klicken<"
 name="Versand"
 style="background-color:red; font-weight:bold;" />
 </form>
</body>
</html>

```



Wie üblich folgt der Test im Browser samt Versand. Welche Werte werden übergeben? Das Ergebnis: der Name der Auswahlliste – und nicht die Beschriftung, sondern der Wert! Sie sehen jetzt also, warum Sie einen Wert angeben mussten.

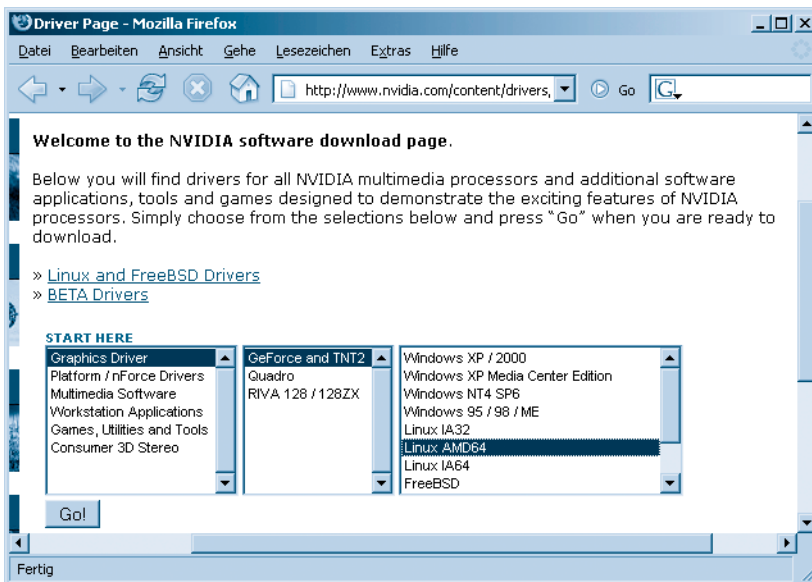




### Hinweis

*Sollten Sie übrigens tatsächlich einmal das `value`-Attribut vergessen, übergeben die meisten Browser stattdessen die Beschriftung des Eintrags. Aber auch hier gilt: Sie sollten sich nicht darauf verlassen – setzen Sie lieber immer das `value`-Attribut.*

Es gibt aber noch eine Abwandlung der Auswahlliste, wie sie beispielsweise im Download-Bereich des Grafikkartenherstellers NVIDIA zu sehen ist: Die Liste zeigt mehrere ihrer Elemente auf einmal an.



Das ist sehr leicht selbst nachzubilden. Setzen Sie dazu einfach im `<select>`-Tag das Attribut `size` auf die Anzahl der Elemente, die gleichzeitig sichtbar sein sollen. Wenn Sie das nicht tun, geht der Webbrowser automatisch von `size="1"` aus, also nur einem sichtbaren Element. Das folgende Listing entspricht weitestgehend dem vorhergehenden, mit einem Unterschied: `size="2"` für die Auswahlliste.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- formular-grosseliste.html -->
<html>
```

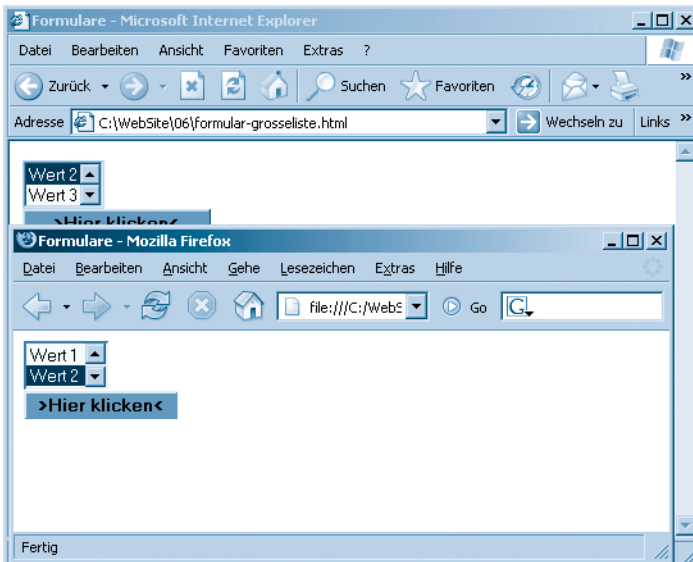
```

<head>
 <title>Formulare</title>
</head>
<body>
 <form action="formulardaten.html" method="get">
 <select name="Liste"
 style="background-color:yellow;"
 size="2">
 <option value="1">Wert 1</option>
 <option value="2" selected="selected">Wert 2</option>
 <option value="3">Wert 3</option>
 </select>

 <input type="submit"
 value=">Hier klicken<"
 name="Versand"
 style="background-color:red; font-weight:bold;" />
 </form>
</body>
</html>

```

Als Ergebnis sehen Sie im Webbrowser, dass nun immer zwei Listenelemente auf einmal angezeigt werden. Auch hier gibt es wieder einmal einen Browserunterschied, den Sie leider nicht umgehen können: Der Internet Explorer scrollt die Liste so herunter, dass das vorausgewählte Element als Erstes angezeigt wird; der Firefox-Browser dagegen beginnt mit dem obersten Element.



## Mehrfache Auswahllisten

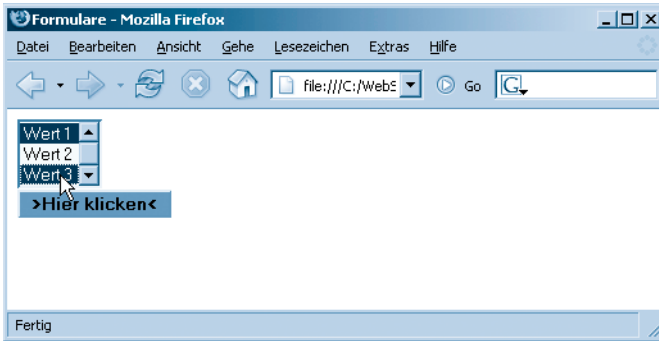
Die bisher vorgestellten Varianten der Auswahllisten hatten eine Einschränkung: Es war nur möglich, eines der Elemente auszuwählen. Mit einer kleinen Erweiterung ist es allerdings auch machbar, dass mehrere Elemente auf einmal selektiert werden können. Einzige dazu notwendige Änderung: Das Attribut `multiple` muss im `<select>`-Tag gesetzt werden. Natürlich müssen mindestens zwei Elemente auf einmal sichtbar sein, also das `size`-Attribut einen entsprechenden Wert haben. Außerdem benennen wir die Auswahlliste leicht um:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- formular-mehrfachliste.html -->
<html>
<head>
 <title>Formulare</title>
</head>
<body>
 <form action="formulardaten.html" method="get">
 <select name="Liste[]"
 style="background-color:yellow;"
 size="3"
 multiple="multiple">
 <option value="1">Wert 1</option>
 <option value="2">Wert 2</option>
 <option value="3">Wert 3</option>
 </select>

 <input type="submit"
 value=" >Hier klicken< "
 name="Versand"
 style="background-color:red; font-weight:bold;" />
 </form>
</body>
</html>
```

Nun gibt es zwei Möglichkeiten, mehrere Elemente auf einmal auszuwählen:

- Halten Sie die `[Strg]`-Taste gedrückt und klicken Sie mit der Maus nacheinander auf alle auszuwählenden Listenelemente.
- Um mehrere **aufeinander folgende** Listenelemente auszuwählen, klicken Sie mit der Maus zunächst das oberste an, halten dann `[⇧]` gedrückt und klicken mit der Maus das unterste Element an.



Bleibt nur noch eine Frage: Was machen die eckigen Klammern am Ende des Listennamens, `Liste[]`? Das hat mit PHP zu tun – einer Skriptsprache, die häufig eingesetzt wird, um Formulardaten zu verarbeiten. PHP verlangt, dass der Name von Mehrfachlisten auf `[]` endet. Browser (und andere Technologien) stören sich nicht daran, also ist es eine gute Idee, sich das so anzugewöhnen.

#### Was ist das?

*PHP, kurz für **PHP: Hypertext Preprocessor**, ist die zurzeit beliebteste serverseitige Skriptsprache und kann viele Dinge, die mit HTML allein nicht möglich wären. In Kapitel 10 erfahren Sie, wie Sie PHP installieren und einsetzen können.*

## Grafische Versenden-Schaltflächen

Zum Abschluss kommen wir noch einmal zu den Versenden-Schaltflächen zurück. Deren Gestaltungsmöglichkeiten sind ja – CSS hin oder her – stark eingeschränkt. Doch es gibt einen möglichen Ausweg: Sie können auch herkömmliche Grafiken in Versenden-Schaltflächen umwandeln. Das wird unter anderem von der Reisebüro-Website Travel Overland so verwendet – der Text *Suchen* ist in Wirklichkeit eine Grafik.





Und wie geht das? Zunächst benötigen Sie eine Grafik, in *Kapitel 4* haben Sie dazu viele Informationen erhalten. Und dann bauen Sie die Grafik wie folgt ein:

**1** Verwenden Sie `<input type="image">`, um eine Grafik als Versenden-Schaltfläche einzubauen.

```
<input type="image" />
```

**2** Geben Sie im Attribut `src` den URL der Grafik an.

```
<input type="image" src="grafik.gif" />
```

**3** Einen Namen brauchen Sie auch noch – wie immer im `name`-Attribut.

```
<input type="image" name="Versendegrafik" />
```

**4** Standardmäßig erscheint ein Rahmen um die Grafik. Den werden Sie mit `border="0"` oder der entsprechenden CSS-Anweisung los.

```
<input type="image" style="border:0px;" />
```

**5** Für Browser, die keine Grafiken anzeigen (ja, diese gibt es), sollten Sie im `alt`-Attribut einen alternativ anzuzeigenden Text angeben.

```
<input type="image" alt="Formular versenden" />
```

Und hier das Ganze zusammenhängend am Stück:

```

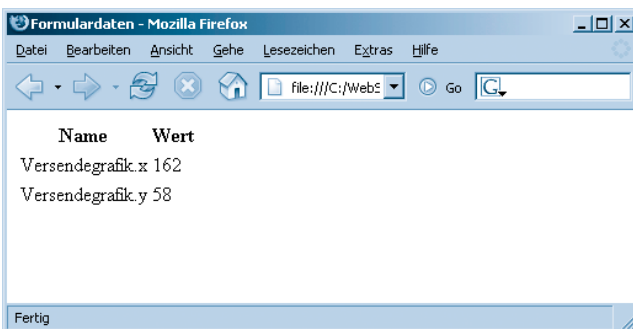
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- formular-grafik.html -->
<html>
<head>
 <title>Formulare</title>
</head>
<body>
 <form action="formulardaten.html" method="get">
 <input type="image"
 src="grafik.gif"
 name="Versendegrafik"
 style="border:0px;"
 alt="Formular versenden" />
 </form>
</body>
</html>

```

Im Webbrowser sehen Sie jetzt die Grafik, und der Mauszeiger wird darüber zur Hand.



Interessant wird es, wenn Sie das Formular verschicken. Dann nämlich übermittelt der Webbrowser zwei überraschende Informationen: Die x- und die y-Koordinate des Mausklicks. Sie erfahren also, wo genau der Benutzer geklickt hat, und könnten so theoretisch auch Imapes realisieren. Doch dass das auch einfacher geht, haben Sie schon in *Kapitel 4* gesehen.



## Versand per E-Mail

Sie kennen nun alle relevanten Formularfelder. Daraus lassen sich alle Arten von Formularen erstellen, kurze und lange. Hier ist ein solches: ein Feedback-Formular für Ihre Website (ohne Formatierungen, um es möglichst kurz und übersichtlich zu halten):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- feedbackformular-mail.html -->
<html>
<head>
 <title>Formulare</title>
</head>
<body>
 <form ...>
 <input type="radio" name="Anrede" value="Herr" />Herr
 <input type="radio" name="Anrede" value="Frau" />Frau

 Vorname: <input type="text" name="Vorname" />

 Nachname: <input type="text" name="Nachname" />

 E-Mail: <input type="text" name="Mail" />

 Kommentar zu meiner Website:
 <textarea name="Kommentar" cols="50" rows="8">
 Was ich schon immer mal sagen wollte ...
 </textarea>

 Woher kennen Sie diese Website?
 <select name="Quelle" size="3">
 <option value="such">Suchmaschine</option>
 <option value="mund">Mund-zu-Mund-Propaganda</option>
 <option value="tv">Fernsehwerbung</option>
 <option value="sonst">Sonstiges</option>
 </select>

 <input type="checkbox" name="Newsletter" value="ja" />
 Ja, ich möchte den Newsletter abonnieren!

 <input type="image"
 src="grafik.gif"
 name="Versendegrafik"
 style="border:0px;"
 alt="Formular versenden" />
 </form>
</body>
</html>
```

### Hinweis

*Eine »Mund-zu-Mund-Propaganda« gibt es eigentlich nicht, der korrekte Begriff ist »Mundpropaganda«. Das Kunstwort in diesem Beispiel haben wir in der Zwiebfisch-Kolumne von Bastian Sick im SPIEGEL gefunden und fanden es sehr gelungen – klingt wie Mund-zu-Mund-Beatmung –, so dass wir es hier einsetzen.*

Fehlt nur noch ein Punkt, im Listing durch drei Punkte markiert: Wohin soll das Formular? In der Regel sollen die Daten gleich per Mail verschickt werden. Da gibt es eine gute und eine schlechte Möglichkeit. Zunächst die schlechte. Die folgenden Angaben müssen alle im `form`-Tag ergänzt werden:

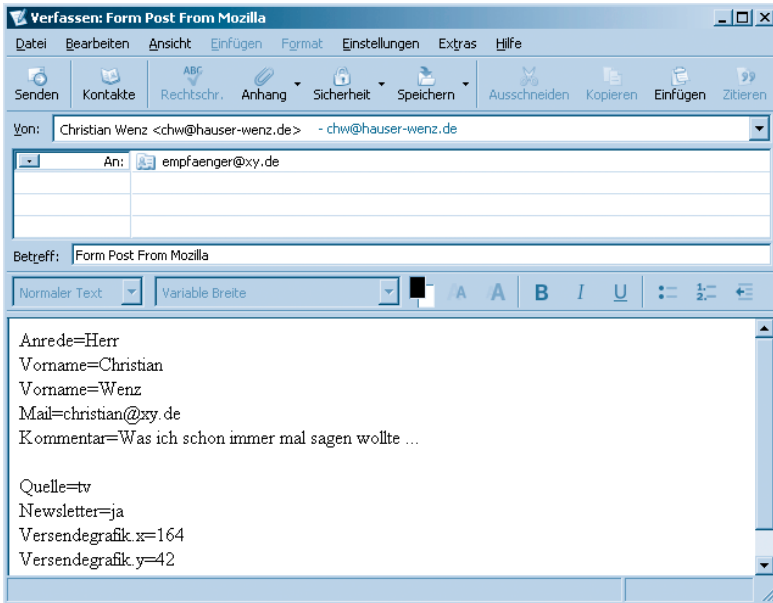
- 1 Setzen Sie die Versandmethode des Formulars auf POST: `method="post"`.
- 2 Geben Sie als Versandziel die E-Mail-Adresse des Empfängers, also meist Ihre E-Mail-Adresse, in folgendem Format an: `action="mailto:empfaenger@xy.de"`.
- 3 Sorgen Sie dafür, dass die Maildaten in einem gut lesbaren Format verschickt werden: `enctype="text/plain"`.

Manchmal scheint das zu funktionieren – die Maildaten kommen beim Empfänger an. Leider ist das jedoch nur die halbe Wahrheit. Das Ganze klappt nämlich nur, wenn der Webbrowser und das E-Mail-Programm zusammenarbeiten. Das ist aber beileibe nicht immer der Fall. Je nach System passiert

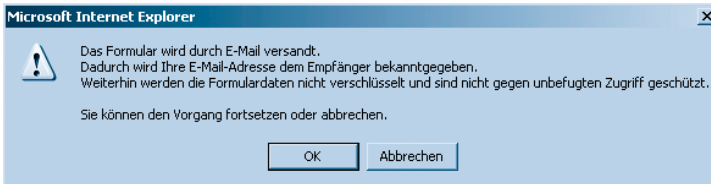




entweder nichts, oder es öffnet sich eine neue Mail, die aber nicht automatisch verschickt wird.



Noch schlimmer: Selbst wenn es funktioniert, erscheint in den meisten Browsern beim Versand des Formulars eine Furcht einflößende Warnmeldung.



## Hinweis

*Warum ist das so? Irgendwoher muss die Website ja wissen, wie vom Client aus eine E-Mail verschickt werden kann. HTML macht es sich hier sehr einfach: Es delegiert die Arbeit an den Webbrowser. Und wenn der kein integriertes Mailprogramm hat (momentan hat praktisch nur noch die Mozilla Suite ein integriertes Mailprogramm), delegiert er diese Arbeit an das Standardmailprogramm im System. Und hier ist es wie im wahren Leben: Je mehr delegiert wird, desto mehr Informationen können verloren gehen.*

## Versand per E-Mail, aber richtig

Doch zum Glück gibt es noch Alternativen. Beispielsweise den Anbieter Formmailer, im Web erreichbar unter <http://www.formmailer.com/>.



Die Idee dahinter ist genauso einfach wie pfiffig. Auf dem Webserver von Formmailer liegt ein Programm, das dazu in der Lage ist, zuverlässig E-Mails zu versenden. Alles, was Sie nun tun müssen, ist, dem Formmailer mitzuteilen, wohin die E-Mails geschickt werden soll. Das geschieht grob in zwei Schritten: Zuerst melden Sie sich bei Formmailer an, dann passen Sie Ihr Formular an:

- 1** Klicken Sie auf <http://www.formmailer.com/> auf der Hauptseite in der linken Navigationsleiste auf **Neues Konto**.
- 2** Akzeptieren Sie die angezeigten Bedingungen für die Nutzung des Formulars.
- 3** Wählen Sie eine Benutzerkennung aus und geben Sie Ihre Anschrift an.



**Kennung wählen**

Um auf Ihr Konto zugreifen zu können, müssen Sie eine Benutzerkennung und ein Passwort spezifizieren. Wählen Sie hierzu jetzt eine beliebige Kennung und ein Passwort aus. Die Benutzerkennung darf bis zu 20 Zeichen lang sein und aus den Buchstaben A-Z sowie den Ziffern 0-9 bestehen. Bitte geben Sie das gewünschte Passwort zweimal ein, um Tippfehler zu vermeiden. In Ihrem Passwort sollten keine Sonderzeichen vorkommen, da sonst das Passwort unter Umständen nicht immer korrekt erkannt wird.

Gewünschte Benutzerkennung:

Gewünschtes Passwort:

Passwort bitte wiederholen:

Fertig

Sie erhalten eine Bestätigungsmeldung (oder eine Fehlermeldung, wenn etwas fehlt oder der von Ihnen gewählte Benutzername bereits vergeben ist).

**4** Klicken Sie jetzt auf *Kontozugang*, wieder in der linken Navigationsleiste.

**5** Geben Sie Ihre Nutzerdaten an, um sich einzuloggen.

**Aufforderung**

Geben Sie Benutzernamen und Passwort für "Formmailer" auf <http://www.formmailer.com> ein

Benutzername:

Passwort:

☐ Den Passwort-Manager benutzen, um dieses Passwort zu speichern.

OK Abbrechen

**6** Klicken Sie auf den Link *Neues Formular einrichten*.

**7** Geben Sie im Formular drei Daten an: den URL des Formulars, den URL der Ergebnisseite sowie die E-Mail-Adresse, an die das Formular verschickt werden soll.

Die Ergebnisseite wird angezeigt, nachdem das Formular versendet wurde. Hier kann auch beispielsweise schlicht »Danke« stehen. Ganz wichtig: Unter *Verwendungszweck* geben Sie an, ob Sie das Formular privat oder geschäft-

lich nutzen. Nur im ersten Fall ist Formmailer gratis; kommerzielle Nutzer müssen einen geringen Obolus entrichten.

Formmailer - Neues Formular einrichten - Mozilla Firefox

Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe

http://www.formmailer.com/addform.php?se

Bitte geben Sie die URL des Formulars sowie der Seite an, die nach dem Absenden des Formulars angezeigt werden soll.

URL des Formulars:

URL der Ergebnisseite:

Bitte geben Sie die Email-Adresse an, an die der Formularinhalt geschickt werden soll.

Email-Adresse:

Fertig

**8** Sie bekommen als Ergebnis ein HTML-Element angezeigt, das Sie in das gewünschte Formular einbauen müssen, nach folgendem Muster:

```
<input type="hidden" name="formmailer" value="XXXXX" />
```

Anstelle von XXXXX erhalten Sie fünf Ziffern. Diese müssen Sie an der entsprechenden Stelle statt XXXX schreiben!

### Hinweis

*Es handelt sich bei `<input type="hidden" />` um ein weiteres Formularfeld: ein verstecktes Feld. Das wird im Browser nicht angezeigt, aber beim Formularversand mit übermittelt. Richtig »versteckt« ist das Feld jedoch nicht, da es im HTML-Quelltext steht.*

**9** Ändern Sie das Formular so, dass es an Formmailer verschickt wird, d.h. ergänzen Sie im `<form>`-Element: `action="http://send.formmailer.com/"` `method="post"`.

Am Ende könnte also Ihr Formular wie folgt aussehen. Denken Sie auf jeden Fall daran, anstelle von XXXXX Ihre persönlichen fünf Ziffern anzugeben:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- feedbackformular-formmailer.html -->
<html>
<head>
 <title>Formulare</title>
</head>
```



```

<body>
 <form action="http://send.formmailer.com/"
 method="post">
 <input type="hidden"
 name="formmailer"
 value="XXXXX" />
 <input type="radio" name="Anrede" value="Herr" />Herr
 <input type="radio" name="Anrede" value="Frau" />Frau

 Vorname: <input type="text" name="Vorname" />

 Nachname: <input type="text" name="Nachname" />

 E-Mail: <input type="text" name="Mail" />

 Kommentar zu meiner Website:
 <textarea name="Kommentar" cols="30" rows="2">
 Was ich schon immer mal sagen wollte ...
 </textarea>

 Woher kennen Sie diese Website?
 <select name="Quelle" size="3">
 <option value="such">Suchmaschine</option>
 <option value="mund">Mund-zu-Mund-Propaganda</option>
 <option value="tv">Fernsehwerbung</option>
 <option value="sonst">Sonstiges</option>
 </select>

 <input type="checkbox" name="Newsletter" value="ja" />
 Ja, ich möchte den Newsletter abonnieren!

 <input type="image"
 src="grafik.gif"
 name="Versendegrafik"
 style="border:0px;"
 alt="Formular versenden" />
 </form>
</body>
</html>

```

### Hinweis

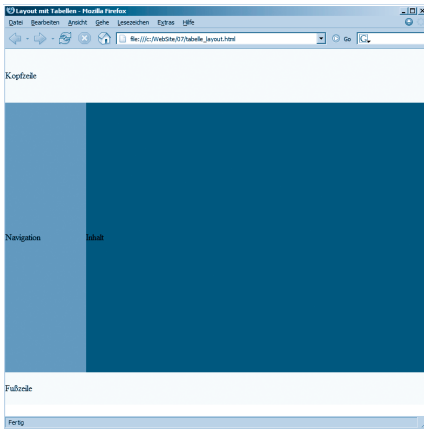
*Wichtig: Jedes Formular müssen Sie im Administrationsmenü von **<http://www.formmailer.com/>** einzeln registrieren (Sie erhalten dann auch eine neue Nummer)!*

## Nutzloses Wissen

Die längste Unterwasserpartie von Monopoly dauerte 45 Tage.

## Kapitel 7

# Webseiten aufbauen – Elemente positionieren



*In einem Layoutprogramm wie Adobe InDesign oder Quark XPress ziehen Sie Absätze und Bilder einfach an die Stellen, an die Sie sie haben möchten. Ähnlich läuft das Zeichnen von Folien in Microsoft PowerPoint oder in Open Office Impress. Im Web, das heißt mit HTML und CSS, ist die Sache nicht ganz so einfach. Zwar wollen viele WYSIWYG-Editoren wie Macromedia Dreamweaver oder Microsoft FrontPage diesen Eindruck erwecken, aber er täuscht.*

*Ursprünglich gab es zum Positionieren in HTML nur einen Trick: Tabellen – eigentlich zur Datenpräsentation gedacht, lassen sie sich ohne Rahmen und Abstände zum Platzieren von Elementen verwenden. Eine echte Positioniermöglichkeit kam erst in CSS hinzu. Allerdings ist sie in älteren Browsern, vor allem dem Netscape Navigator 4.x, zumindest problematisch und nach wie vor eine aufwändige Angelegenheit. Beide Varianten lernen Sie in diesem Kapitel kennen.*

**Das können Sie schon:**

|                                  |     |
|----------------------------------|-----|
| Formatierung von Text            | 56  |
| Einsatz von Links                | 75  |
| Aufzählungslisten                | 90  |
| Listen mit Nummerierung          | 92  |
| Tabellen zur Datenstrukturierung | 95  |
| Grafiken einbauen                | 116 |
| Frames erstellen                 | 134 |
| Formulare verwenden              | 158 |

**Das lernen Sie neu:**

|                                |     |
|--------------------------------|-----|
| Tabellen zum Positionieren     | 196 |
| Komplettes Layout mit Tabellen | 200 |
| CSS zum Positionieren          | 201 |
| Komplettes Layout mit CSS      | 209 |

## Tabellen

Tabellen kennen Sie aus *Kapitel 3*. Wenn Sie Tabellen zum Positionieren verwenden möchten, müssen Sie nur wissen, dass Sie die Breite und Höhe von Tabellenzellen angeben können. Damit lässt sich dann eine Layouttabelle basteln.

### Unsichtbar

Wichtigste Voraussetzung für eine Tabelle zum Positionieren ist, dass der Nutzer sie nicht sehen kann. Und tatsächlich, wenn Sie sich den Quellcode verschiedenster Websites ansehen, finden Sie meist eine Tabelle und dort immer die Angabe `border="0"`.

Eine rahmenlose Tabelle reicht allerdings zum pixelgenauen Platzieren nicht. Sie müssen auch die Zellenabstände entfernen:

```
<table border="0" cellpadding="0" cellspacing="0">
```

Den Rest erledigen die Größenangaben. Wollen Sie z.B. ein Bild um 100 Pixel nach rechts und um 100 Pixel nach unten auf der Seite platzieren, verwenden Sie eine solche Tabelle:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- tabellen_posi.html -->
<html>
<head>
 <title>Positionieren mit Tabellen</title>
</head>
<body>
 <table>
 <tr>
 <td width="100" height="100"></td>
 <td> </td>
 </tr>
 <tr>
 <td> </td>
 <td>

 </td>
 </tr>
 </table>
</body>
</html>
```







## Abstand zum Fensterrand

Wenn Sie im letzten Abschnitt genau hingeschaut haben, konnten Sie sehen, dass das Bild nicht exakt auf 100 Pixel positioniert ist, sondern ein wenig weiter in das Fenster hineinragt. Dies liegt daran, dass Browser automatisch einen Seitenrand in ihrem Fenster vergeben.

Da dieser nicht unbedingt einheitlich festgelegt ist und beim akkuraten Positionieren nur stört, sollten Sie ihn loswerden. Dafür gibt es zwei Methoden:

- Per CSS, wenn Sie für das `<body>`-Tag die Werte 0 für `margin` und `padding` vergeben.
- Per HTML mit den offiziell nicht vorgesehenen Attributen `leftmargin="0"` und `topmargin="0"` für den Internet Explorer und `marginwidth="0"` und `marginheight="0"` für alle anderen Browser.

Die reine CSS-Variante können Sie verwenden, wenn Sie auf den Netscape Navigator 4.x verzichten. Beim Einsatz von stark abwärtskompatiblen Tabellen ist die HTML-Variante zu bevorzugen.

Hier ein Beispiel, in dem ein Bild wirklich exakt im oberen Eck des Browserfensters platziert wird:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- fensterabstand.html -->
<html>
<head>
 <title>Fensterabstand</title>
</head>
<body leftmargin="0" topmargin="0" marginwidth="0" margin-
height="0">
 <div></div>
</body>
</html>
```



## Zentrieren

Wie Sie Inhalte in Tabellen zentrieren, kennen Sie bereits aus *Kapitel 3*. Wenn Sie aber eine komplette Tabelle zentrieren oder rechtsbündig ausrichten möchten, können Sie dazu `align` verwenden. Mit diesem einfachen Trick erreichen Sie ein komplett zentriertes Layout:

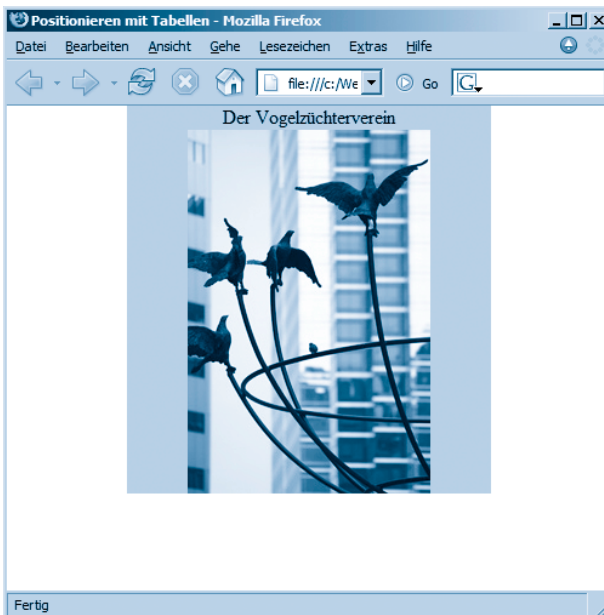


```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- tabellen_zentrieren.html -->
<html>
<head>
 <title>Positionieren mit Tabellen</title>
</head>
<body leftmargin="0" topmargin="0" marginwidth="0"
 marginheight="0">
 <table border="0" cellpadding="0" cellspacing="0" width="300"
 align="center" style="background-color: #BBBBDD">
 <tr>
 <td style="text-align: center">
 Der Vogelzüchterverein</td>
 </tr>
 <tr>
 <td style="text-align: center">

 </td>
 </tr>
 </table>
</body>
</html>

```



## Komplettes Layout

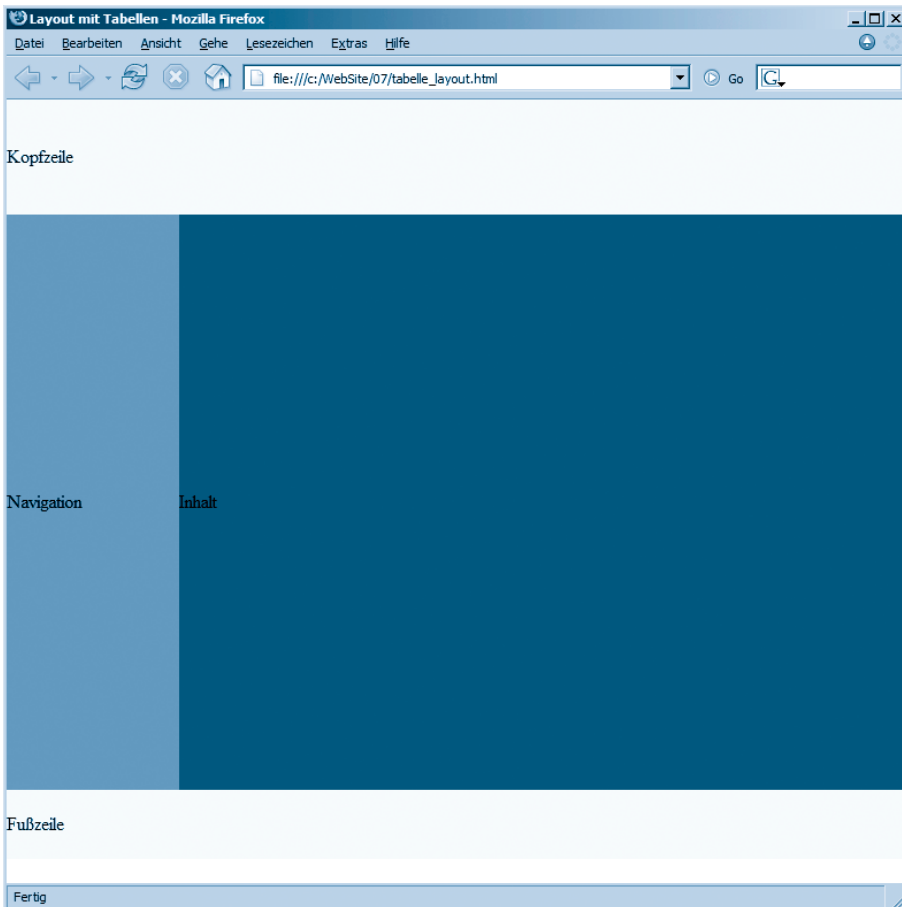
Natürlich ist jedes Seitenlayout individuell. Dennoch haben sich im Web schon einige Gemeinsamkeiten herausgebildet. z.B. ist die Navigationsleiste meist links oder oben und Kopf- bzw. Fußzeile sind sehr üblich geworden. Dementsprechend gibt es einige Grundlayouts. Eines davon finden Sie hier als praktisches Beispiel für eine komplette Lösung:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- tabellen_layout.html -->
<html>
<head>
 <title>Layout mit Tabellen</title>
</head>
<body leftmargin="0" topmargin="0" marginwidth="0"
 marginheight="0">
 <table border="0" cellpadding="0" cellspacing="0"
 width="100%">
 <tr>
 <td colspan="2" height="100" style="background-color:
 yellow;">Kopfzeile</td>
 </tr>
 <tr>
 <td width="150" height="500" style="background-color:
 red;">Navigation</td>
 <td style="background-color: blue;">Inhalt</td>
 </tr>
 <tr>
 <td colspan="2" height="60" style="background-color:
 yellow;">Fußzeile</td>
 </tr>
 </table>
</body>
</html>
```

### Tipp

*Wenn Sie ein eigenes Layout entwickeln, gibt es zwei Möglichkeiten: Entweder zeichnen Sie sich die Tabelle auf und bauen sie dann Schritt für Schritt. Oder Sie nehmen den vorliegenden Code oder einen anderen Code und ändern ihn. Ersteres liefert meist bessere Resultate, Letzteres geht schneller.*





## Positionieren mit CSS

Sie können mit CSS dasselbe erreichen wie mit Tabellen. Der Weg dorthin ist unter Umständen etwas schwieriger, vor allem wenn Sie Tabellen schon kennen. Außerdem sind ältere Browser wie der Netscape Navigator 4.x nur schwer mit einem solchen Layout zum Arbeiten zu bewegen.

Die Vorteile eines CSS-Layouts sind allerdings auch vielfältig: Sie trennen das Layout besser vom Inhalt. Außerdem ist ein CSS-Layout behindertengerecht. Hierfür steht der Begriff Accessibility (dt. Zugänglichkeit). Ein Tabellenlayout ist in dieser Hinsicht nicht so gut, da z.B. Screenreader, die eine Website vorlesen, in Layouttabellen schnell durcheinander kommen.

## Absolut und relativ

Die Positionierung mit CSS kann auf verschiedene Arten erfolgen. Welche Art Sie verwenden möchten, entscheiden Sie mit dem CSS-Befehl `position`. Diese Werte stehen zur Wahl:

- `absolute` positioniert absolut. Sie geben einen Pixelwert für `top` (y-Koordinate) und für `left` (x-Koordinate) an. Die Positionierung erfolgt relativ zum übergeordneten Element, das heißt relativ zu dem Blockelement, in dem ein Element verschachtelt ist. Befindet sich ein Element auf der obersten Ebene, wird absolut zur linken oberen Ecke des Browserfensters positioniert.
- `relative` positioniert relativ zur Position eines Elements im normalen HTML-Fluss. Auch hier geben Sie `top` (y-Koordinate) und `left` (x-Koordinate) an, allerdings sind die Angaben dann relativ zum vorangegangenen Element zu verstehen.
- `fixed` fixiert ein Element an einer Position wie bei der absoluten Positionierung. Allerdings bewegt sich das Element beim Scrollen nicht mit. Dies wird aber nicht von allen Browsern unterstützt.
- `static` ist die normale Positionierung im HTML-Fluss. Dieser Wert muss nicht angegeben werden.

In der Praxis kommen absolute und relative Positionierung zum Einsatz. Der Nachteil der absoluten Positionierung ergibt sich aus ihrem wichtigsten Vorteil: Dadurch, dass Sie die Elemente exakt platzieren können, sind sie auch völlig unabhängig voneinander. Sie können sich also auch bei größeren Inhalten einfach überlappen. Bei der relativen Positionierung kann das nicht passieren. Dafür haben Sie keine exakte Kontrolle über die Koordinaten eines Elements, da es ja von anderen abhängig ist. Sehen Sie sich dazu jeweils ein Beispiel an.

Zuerst Schritt für Schritt die absolute Positionierung:

**1** Erstellen Sie zwei `<div>`-Blöcke.

Sie können auch andere Elemente absolut positionieren. In der Praxis werden allerdings sehr oft `<div>`-Blöcke verwendet, da sie selbst nicht formatieren.

**2** Vergeben Sie für die `<div>`-Blöcke jeweils eine ID, damit Sie sie per Style Sheet ansprechen können.

```
<div id="absatz1">Ein Absatz, der positioniert wird.
</div>
<div id="absatz2">Noch ein Absatz, der positioniert wird.
</div>
```



**3** Erstellen Sie ein Style Sheet im Kopf der Seite.

**4** Greifen Sie auf die zwei IDs zu.

```
#absatz1 {
}
#absatz2 {
}
```

**5** Vergeben Sie für beide Bereiche eine Größe von 200 \* 300 Pixel und jeweils eine Hintergrundfarbe.

**6** Positionieren Sie den ersten Bereich absolut auf 50 und 100 Pixel.

```
#absatz1 {
 width: 200px;
 height: 300px;
 background-color: red;
 position: absolute;
 top: 50px;
 left: 100px;
}
```

**7** Den zweiten Bereich positionieren Sie nach demselben Muster auf 150 und 200 Pixel.

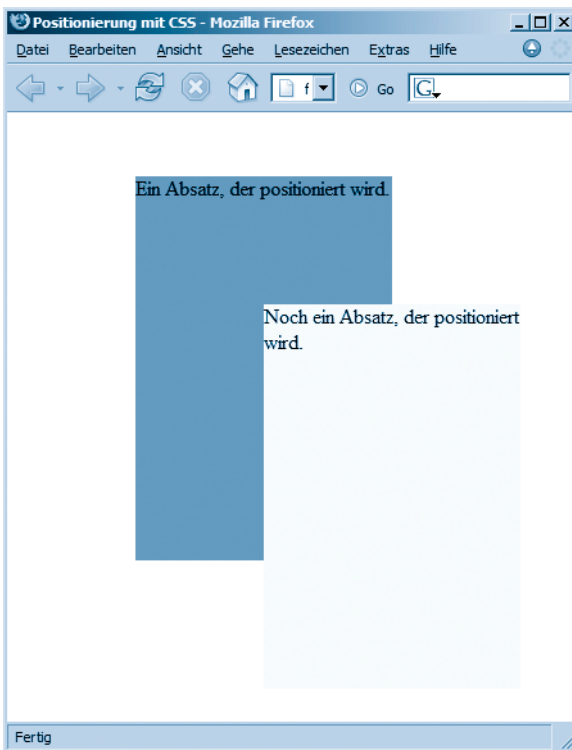
Hier der Code in der Übersicht:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- css_absolute.html -->
<html>
<head>
 <title>Positionierung mit CSS</title>
 <style type="text/css"><!--
 #absatz1 {
 width: 200px;
 height: 300px;
 background-color: red;
 position: absolute;
 top: 50px;
 left: 100px;
 }
 -->
```

```

#absatz2 {
 width: 200px;
 height: 300px;
 background-color: yellow;
 position: absolute;
 top: 150px;
 left: 200px;
}
--></style>
</head>
<body>
 <div id="absatz1">Ein Absatz, der positioniert wird.
 </div>
 <div id="absatz2">Noch ein Absatz, der positioniert wird.
 </div>
</body>
</html>

```

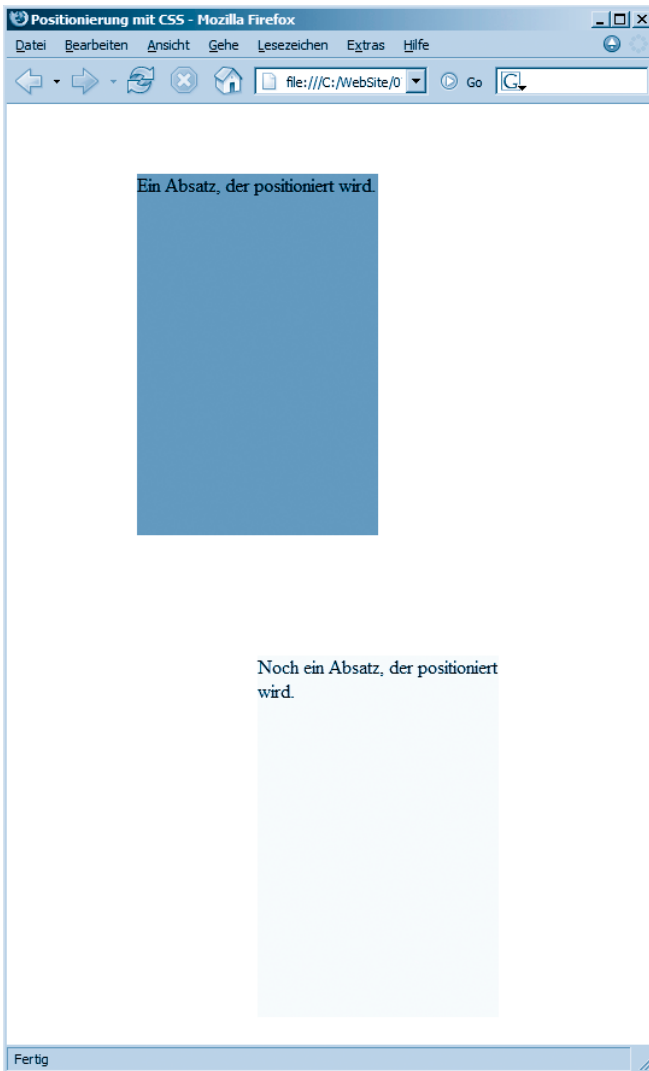




Nun ändern Sie zum Vergleich nur die absolute Positionierung in die relative. Die geänderten Stellen sind hervorgehoben:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- css_relative.html -->
<html>
<head>
 <title>Positionierung mit CSS</title>
 <style type="text/css"><!--
 #absatz1 {
 width: 200px;
 height: 300px;
 background-color: red;
 position: relative;
 top: 50px;
 left: 100px;
 }
 #absatz2 {
 width: 200px;
 height: 300px;
 background-color: yellow;
 position: relative;
 top: 150px;
 left: 200px;
 }
 --></style>
</head>
<body>
 <div id="absatz1">Ein Absatz, der positioniert wird.
 </div>
 <div id="absatz2">Noch ein Absatz, der positioniert wird.
 </div>
</body>
</html>
```

Sie sehen im Ergebnis, dass der zweite Absatz nun am ersten orientiert verschoben wird.



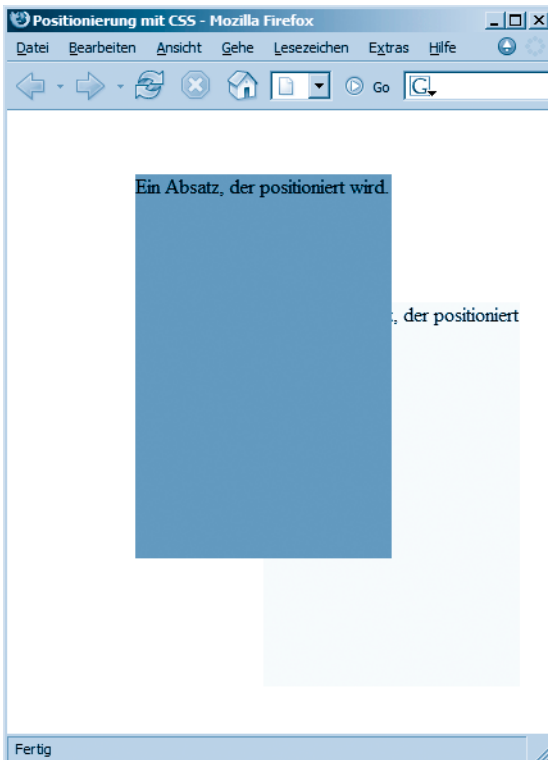
## Überlappung

Wenn sich Elemente überlagern, liegt immer das Element oben, das zuletzt im HTML-Quelltext steht. Im Beispiel aus dem letzten Abschnitt ist das der zweite Absatz. Sie können allerdings auch explizit festlegen, auf welcher Ebene ein Element liegt. Dafür dient der `z-index`. Als Wert geben Sie eine ganze Zahl an. Das Element mit der höheren Zahl liegt oben – bei gleicher Zahl entscheidet wieder die Reihenfolge im Quellcode.



Der folgende Code vertauscht die Rangfolge für die zwei Absätze:

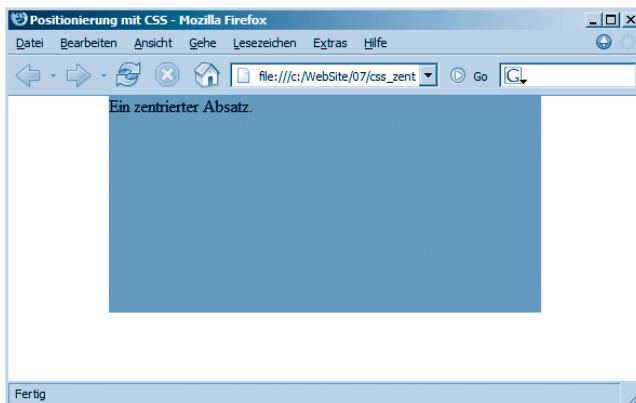
```
#absatz1 {
 width: 200px;
 height: 300px;
 background-color: red;
 position: absolute;
 top: 50px;
 left: 100px;
 z-index: 2;
}
#absatz2 {
 width: 200px;
 height: 300px;
 background-color: yellow;
 position: absolute;
 top: 150px;
 left: 200px;
 z-index: 1;
}
```



## Zentrieren

Elemente mit CSS zu zentrieren ist ein wenig schwieriger als mit einer Tabelle. Wobei schwierig wohl nicht das richtige Wort ist. Die Wahrheit ist, dass der Trick nicht weithin bekannt ist. Zum Zentrieren verwenden Sie für `margin-left` und `margin-right` jeweils den Wert `auto`:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- css_zentrieren.html -->
<html>
<head>
 <title>Positionierung mit CSS</title>
 <style type="text/css"><!--
 body {
 margin: 0px;
 padding: 0px;
 }
 #absatz1 {
 width: 400px;
 height: 200px;
 background-color: red;
 margin-left: auto;
 margin-right: auto;
 z-index: 2;
 }
 --></style>
</head>
<body>
 <div id="absatz1">Ein zentrierter Absatz.
 </div>
</body>
</html>
```



## Komplettes Layout

Für ein komplettes Layout mit CSS müssen Sie alle gewünschten Bereiche der Seite als `<div>`-Blöcke anlegen. Dann ist allerdings nichts mehr völliger Standard. Es gibt verschiedene Techniken, die Elemente anzuordnen. Die vertikale Anordnung erfolgt bei absoluter Positionierung und fester Angabe der Höhe mit festen Abständen.

Für die Positionierung der Navigationsleiste verwendet das folgende Beispiel einen Trick: Der Inhalt in der Mitte der Seite wird mit `margin-left` um die Breite der Navigationsleiste nach rechts gerückt. Insgesamt ergibt sich genau dasselbe Layout, wie Sie es vorher im Abschnitt mit Tabellen gesehen haben:

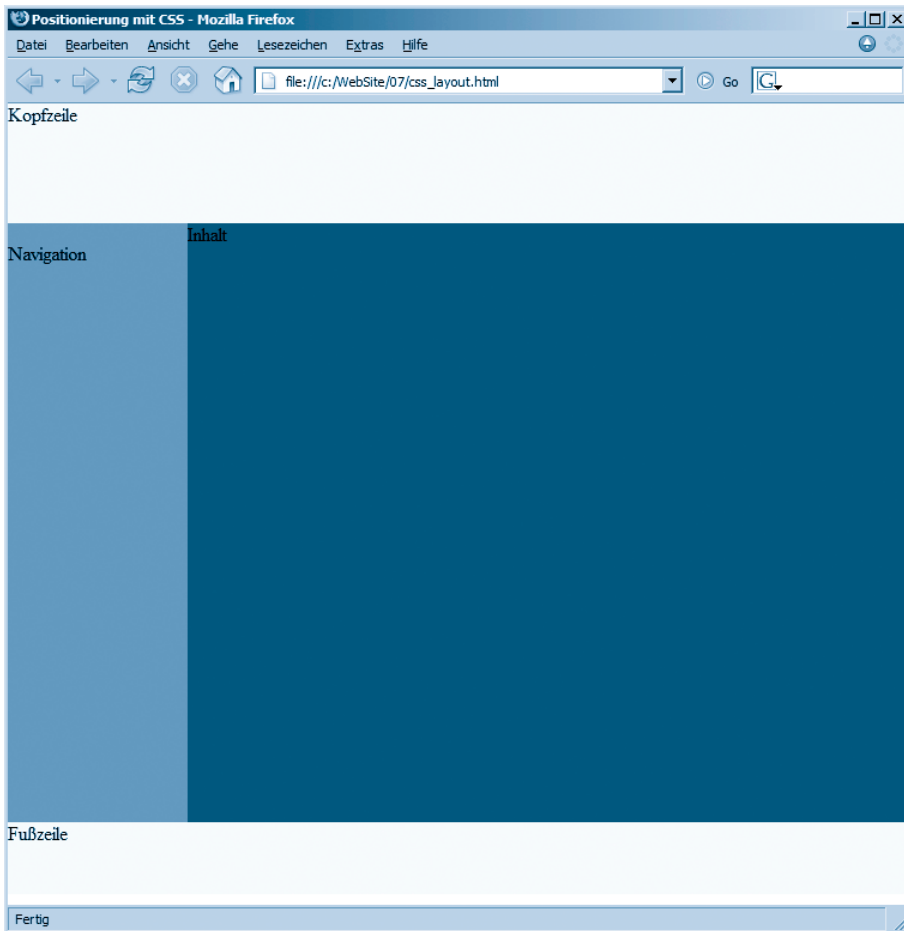
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- css_layout.html -->
<html>
<head>
 <title>Positionierung mit CSS</title>
 <style type="text/css">
 <!--
 body {
 margin: 0px;
 padding: 0px
 }
 #kopf {
 position: absolute;
 left: 0px;
 top: 0px;
 z-index: 3;
 width: 100%;
 height: 100px;
 background-color: yellow;
 border: none
 }
 #mitte {
 position: absolute;
 left: 0px;
 top: 100px;
 z-index: 1;
 width: 100%;
 height: 500px;
 background-color: blue;
 border: none
 }
 </style>
</head>
<body>
```

```

#mitte_inhalt {
 margin-left: 150px
}
#navigation {
 position: absolute;
 left: 0px;
 top: 100px;
 z-index: 2;
 width: 150px;
 height: 500px;
 background-color: red;
 border: none;
}
#fuss {
 position: absolute;
 left: 0px;
 top: 600px;
 z-index: 4;
 width: 100%;
 height: 60px;
 background-color: yellow;
 border: none;
}
-->
</style>
</head>
<body>
 <div id="kopf">
 Kopfzeile
 </div>
 <div id="navigation">
 <p>Navigation</p>
 </div>
 <div id="mitte">
 <div id="mitte_inhalt">Inhalt</div>
 </div>
 <div id="fuss">
 Fußzeile
 </div>
</body>
</html>

```



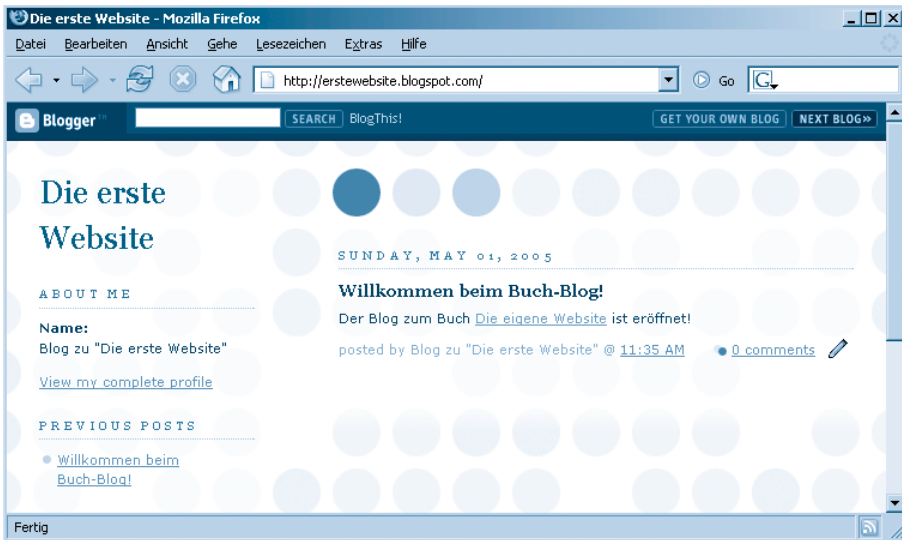


## Nutzloses Wissen

Der Flughafen von Gloucestershire (England) spielte lautstark Lieder von Tina Turner, um Vögel zu verjagen.

## Kapitel 8

# Nützliche Erweiterungen für die Website



*Ein aktueller Musiktitel besagt: »Sometimes you can't make it on your own« – manchmal schaffst du es nicht alleine. So ist es häufig auch im Web. Für Spezialaufgaben wie Gästebücher oder On-line-Foren kann es lohnenswert sein, auf einen externen Anbieter zu setzen. Gerade im privaten Bereich ist das »Outsourcing« von einem solchen Feature meist die schnellere und wartungsfreiere Variante, als wenn Sie selbst in die Tasten greifen.*



**Das können Sie schon:**

|                                    |     |
|------------------------------------|-----|
| Formatierung von Text              | 56  |
| Einsatz von Links                  | 75  |
| Aufzählungslisten                  | 90  |
| Listen mit Nummerierung            | 92  |
| Tabellen zur Datenstrukturierung   | 95  |
| Grafiken einbauen                  | 116 |
| Frames erstellen                   | 134 |
| Formulare verwenden                | 158 |
| Mit Tabellen und CSS positionieren | 194 |

**Das lernen Sie neu:**

|                                |     |
|--------------------------------|-----|
| Ein Gästebuch einrichten       | 214 |
| Ein Diskussionsforum erstellen | 222 |
| Einen Weblog anlegen           | 232 |

## Ein Gästebuch einrichten

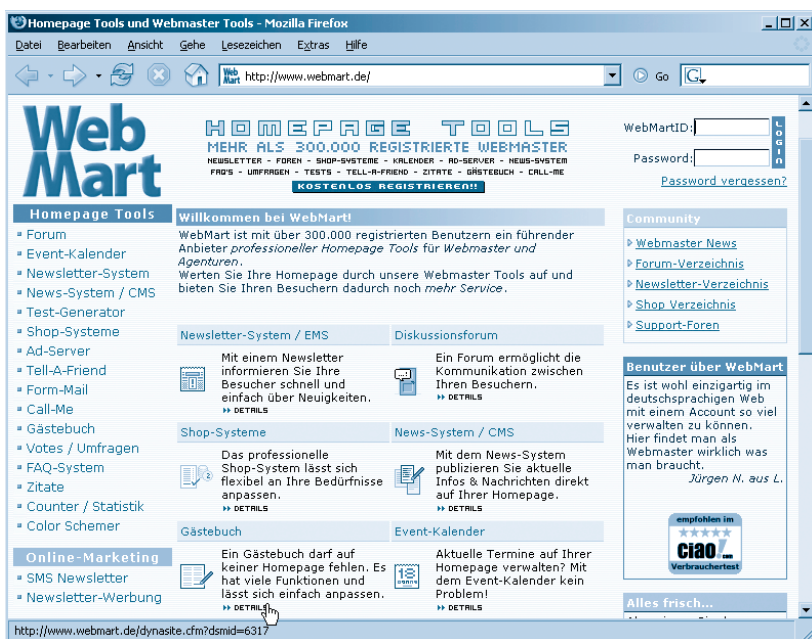
Ein Gästebuch bietet eine willkommene Gelegenheit für die Besucher einer Website, einen kleinen Gruß oder eine Anregung zu hinterlassen. Gerade im privaten Bereich, bei Vereinen etwa, ist so etwas eine gute Möglichkeit, um die Website um eine dynamische Komponente zu erweitern.

Freilich ist das mit HTML alleine nicht möglich, Sie benötigen eine serverseitige Technologie wie etwa PHP (siehe *Kapitel 10*). Oder Sie lassen jemand anderes die Arbeit machen und verlagern das Gästebuch – und nur das Gästebuch – auf den externen Server.

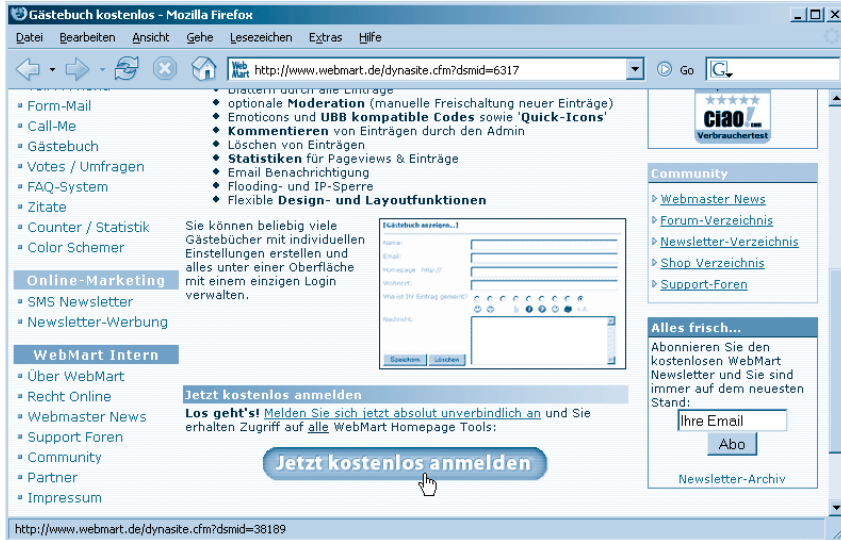
Unter <http://www.webmart.de/> gibt es eine große Menge an solchen möglichen Erweiterungen für Websites. Der Clou: Die meisten davon sind gratis, lediglich mit relativ unaufdringlichen Werbeeinblendungen versehen. So profitieren beide: Sie erhalten ein kostenloses Gästebuch (oder ein anderes Feature), der Anbieter wiederum hofft auf rege Werbeeinnahmen.

Von der statischen Webseite bis hin zum dynamischen Gästebuch sind es im wahrsten Sinne des Wortes nur ein paar Mausklicks. Befolgen Sie die nun aufgeführten Schritte, und schon sind Sie am Ziel:

**1** Klicken Sie auf der Seite <http://www.webmart.de/> auf den *Details*-Link im Bereich *Gästebuch* (im mittleren Teil der Seite).

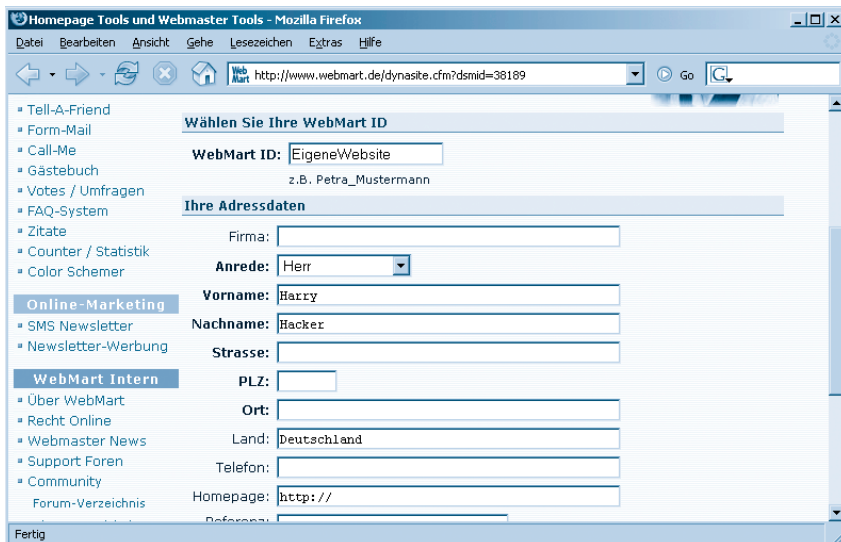


2 Klicken Sie auf der Detailseite auf die Schaltfläche *Jetzt kostenlos anmelden*.

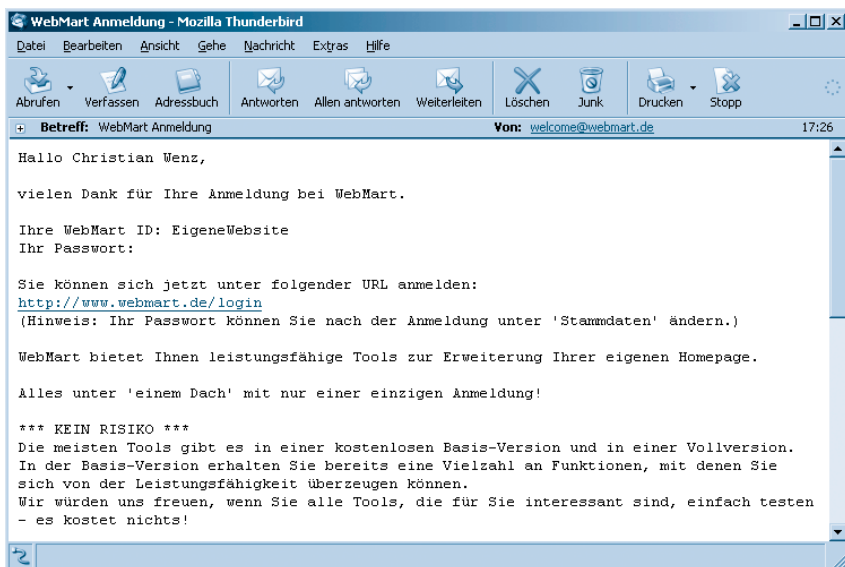


3 Füllen Sie das Formular aus und klicken Sie auf *Anmeldung abschicken*.

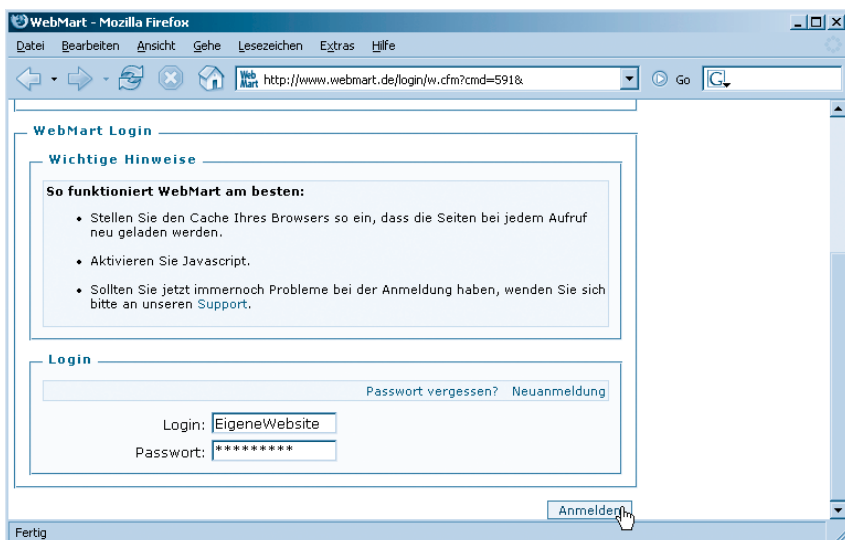
Zwei Punkte sind hierbei besonders wichtig: die *WebMart ID*, das ist Ihr Benutzername bei webmart.de. Und die E-Mail-Adresse, denn dorthin verschickt der Dienst Ihr Passwort. Das ist gleichzeitig eine Sicherheitsmaßnahme, um Ihre E-Mail-Adresse zu überprüfen.



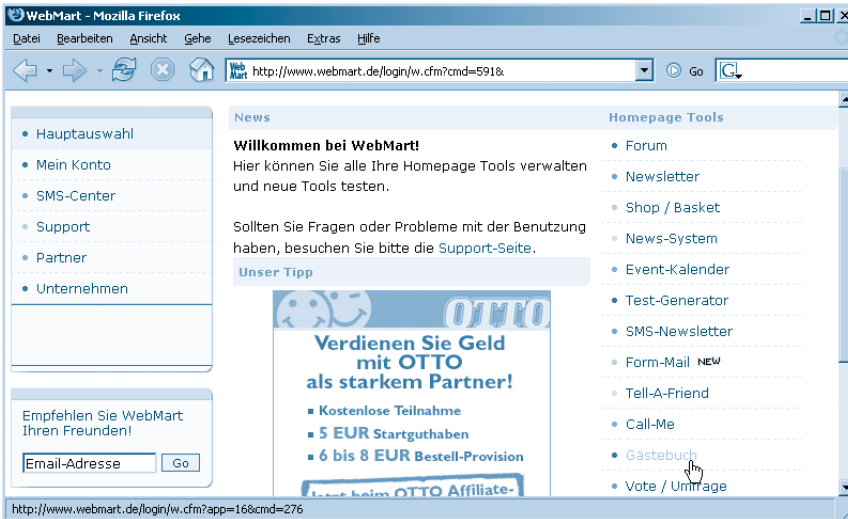
**4** Warten Sie auf eine E-Mail mit Ihrem automatisch erzeugten Passwort.  
In der Abbildung ist das Passwort übrigens aus technischen Gründen entfernt worden. ;-)



**5** Loggen Sie sich unter <http://www.webmart.de/login> ein.  
Es erscheint die Startseite von WebMart.

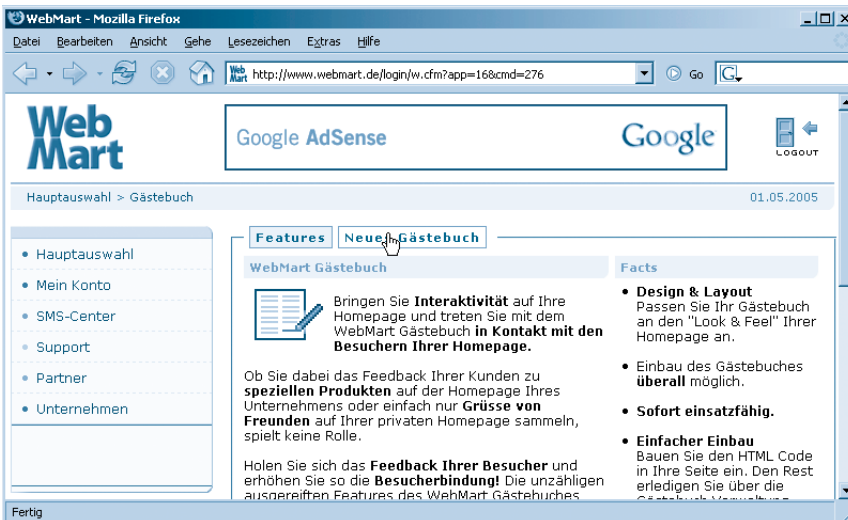


**6** Klicken Sie in der rechten Navigationsspalte auf den Punkt *Gästebuch*.



**7** Klicken Sie auf *Neues Gästebuch*.

Dadurch rufen Sie eine Maske auf, über die Sie ein neues Gästebuch anlegen können.



**8** Geben Sie einen Namen für das Gästebuch an sowie die Website, auf die Sie es einbinden möchten. Klicken Sie dann auf *Gästebuch anlegen*.

Der Name des Gästebuchs ist insofern sehr wichtig, weil dieser dann über dem Gästebuch erscheint. Doch keine Sorge, Sie können diese Angabe später auch noch einmal ändern.

The screenshot shows the 'Neues Gästebuch anlegen' (Create new guestbook) form in the WebMart application. The browser is Mozilla Firefox, and the URL is <http://www.webmart.de/login/w.cfm?app=16&cmd=276>. The page has a header with the WebMart logo, a Google AdSense banner, and a Google logo. A navigation menu on the left includes links to 'Hauptauswahl', 'Mein Konto', 'SMS-Center', 'Support', 'Partner', and 'Unternehmen'. The main content area has tabs for 'Features' and 'Neues Gästebuch'. The 'Neues Gästebuch anlegen' section contains a form with the following fields: 'Name' (filled with 'Gästebuch für "Die eigene Website"'), 'z.B. Mein Gästebuch' (placeholder), and 'Homepage' (filled with 'http://www.hauser-wenz.de/'). A 'Gästebuch anlegen' button is at the bottom right of the form. The date '01.05.2005' is displayed in the top right corner.

Es erscheint eine Ergebnisseite, die besagt, dass das Anlegen funktioniert hat.

9 Klicken Sie jetzt auf den Namen des Gästebuchs.

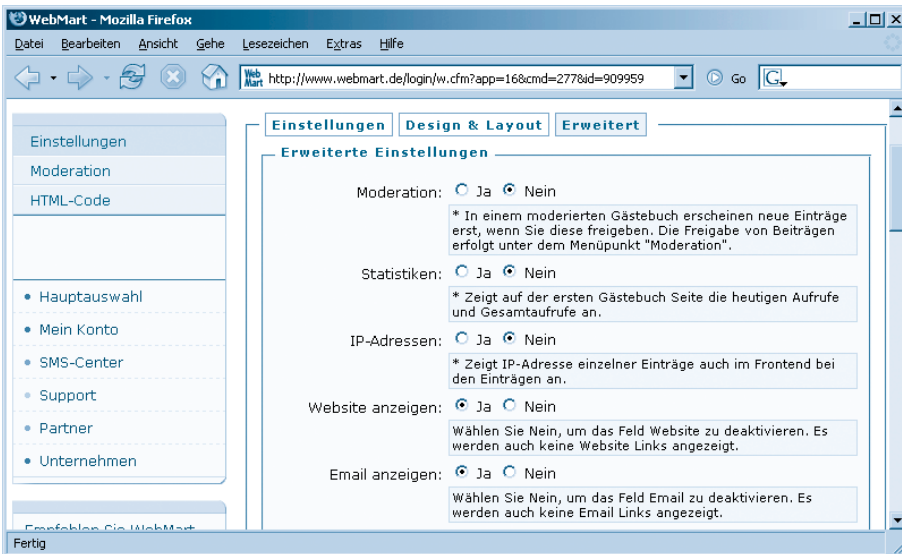
The screenshot shows the 'Übersicht' (Overview) page in the WebMart application. The browser is Mozilla Firefox, and the URL is <http://www.webmart.de/login/w.cfm?app=16&cmd=327>. The page has a header with the WebMart logo, a newsletter banner for 'newsletter.de', and a 'Logout' button. A navigation menu on the left is the same as in the previous screenshot. The main content area has tabs for 'Übersicht', 'Features', and 'Neues Gästebuch'. The 'Übersicht' tab is active, showing a list of guestbooks. The first entry is 'Gästebuch für "Die eigene Website"..' with a 'Löschen' (Delete) button and an 'Upgrade!' button. Below the list, there is a section for 'IP-Sperre' (IP blocking) with a link to 'Klicken Sie hier um IP Adressen zu sperren oder zu bearbeiten.' The date '01.05.2005' is displayed in the top right corner. The footer shows the URL <http://www.webmart.de/login/w.cfm?app=16&cmd=277&id=909959>.



Es erscheint ein Administrationsmenü, in dem Sie das Gästebuch anpassen können. Dort sind unter anderem die folgenden Aktionen möglich:

- Ändern des Namens und des Titels des Gästebuchs
- Sprache des Gästebuchs
- E-Mail-Adresse, an die bei einem neuen Eintrag eine E-Mail geschickt wird
- Begrüßungstext
- Text, der nach dem Eintragen einer Mitteilung ins Gästebuch angezeigt wird (»Danke ...«)

## 10 Tätigen Sie im Register *Erweitert* einige Sicherheitseinstellungen.



Unter anderem sind die folgenden Einstellungen empfehlenswert:

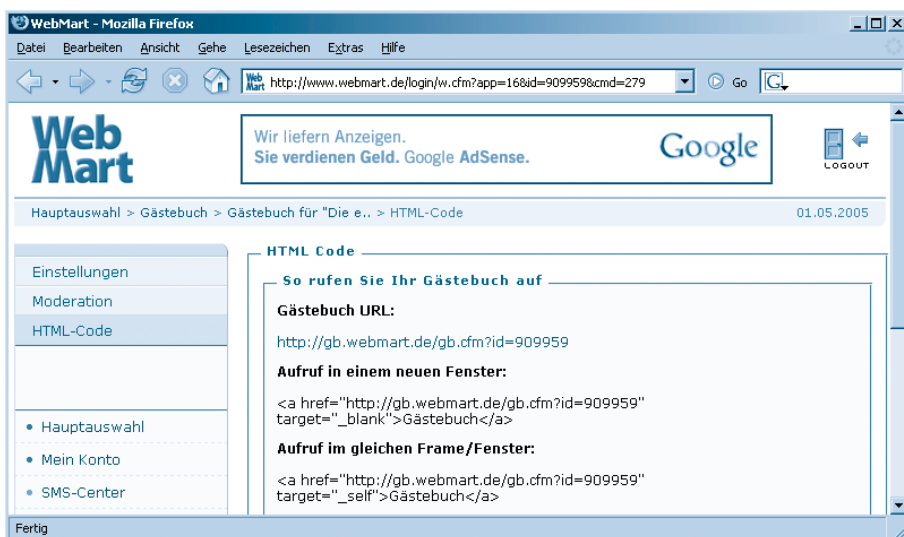
- **Moderation: Nein.** Das bedeutet, dass Einträge immer sofort ins Gästebuch eingetragen werden, ohne dass Sie zustimmen möchten. Aktivieren Sie allerdings unter *Einstellungen* unbedingt die Option, dass Sie bei neuen Einträgen eine E-Mail erhalten. Sie müssen nämlich bei verletzenden Einträgen, Beleidigungen etc. unter Umständen schnell reagieren können und den Eintrag entfernen.

- **Validierung: Ja.** Die Benutzer müssen vor dem Abschicken eines Eintrags einen Code aus einer Grafik abtippen. Das erschwert es automatischen Programmen von Spammern ohne Sozialleben, per Skript Werbung für blaue Pillen und Ähnliches in Ihr Gästebuch einzustellen.
- **Klickbare Links: Nein.** Dann bringt es Spammern ohne Sozialleben und mit zu viel Zeit nichts, Links auf ihre Homepages in Ihr Gästebuch zu schreiben, denn die werden jetzt nicht automatisch mit `<a>`-Tags versehen.

Lesen Sie sich aber auch unbedingt die Beschreibungen der anderen Einstellungen durch, ob da etwas für Sie dabei ist.

**11** Klicken Sie in der linken Navigationsleiste auf den Punkt *HTML-Code*.

Dort erhalten Sie den URL Ihres neuen Gästebuchs sowie den HTML-Code, um darauf zu verlinken – aber Sie wissen ja bereits, wie das geht.



**12** Rufen Sie das Gästebuch im Webbrowser auf und tragen Sie zur Begrüßung einen Eintrag ein.

Das Gästebuch öffnet sich dabei in einem neuen Fenster.

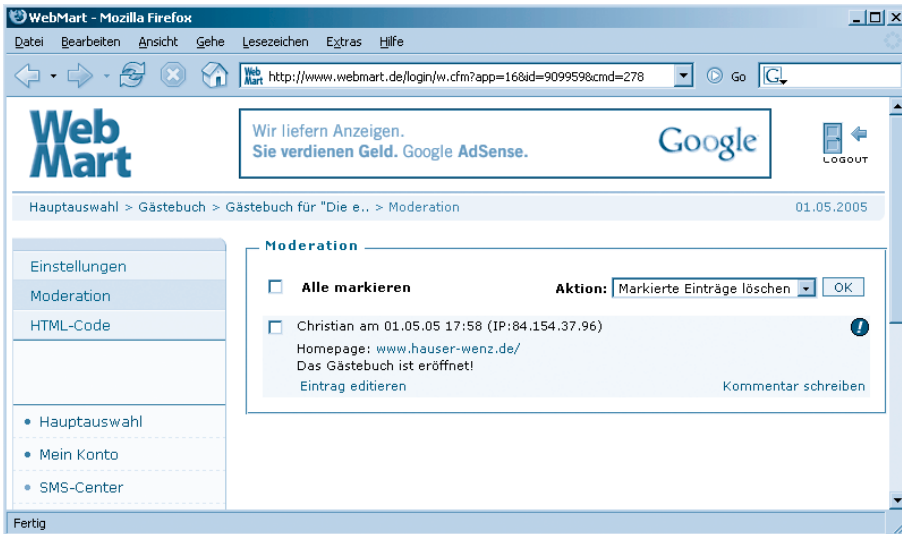






**13** Klicken Sie im Administrationsmenü in der linken Navigationsleiste auf den Punkt *Moderation*.

Dort erhalten Sie eine Übersicht aller Einträge. Sie können so auch einzelne Einträge bearbeiten oder löschen.



### Tipp

*Gehen Sie mit den Features zum Nachbessern von Einträgen sehr vorsichtig um. Denn: Ein wenig Kritik muss jeder vertragen können, wir leben in keinem Zensurstaat, was auch für das Internet gelten sollte. Sie sollten also nur dann Einträge löschen, wenn diese ehrverletzend, stark beleidigend oder rechtlich bedenklich sind.*

Das Gästebuch ist jetzt fertig. Sie müssen sich nur noch überlegen, wie Sie es am besten in Ihre Website integrieren. Dazu gibt es mehrere Möglichkeiten und Optionen:

- Öffnen Sie das Gästebuch in einem neuen Fenster, da es ja auf einem anderen Webserver liegt.
- Öffnen Sie das Gästebuch in einem Frame oder Iframe, damit es so aussieht, als wäre das Gästebuch auf demselben Server wie die eigentliche Homepage.
- Sie können im Administrationsmenü eine CSS-Datei hochladen, die dann im Gästebuch aktiv ist. So passen Sie das Aussehen des Gästebuchs an das Layout Ihrer restlichen Website an.

### Hinweis

*Das im Beispiel angelegte Gästebuch ist im Support-Bereich zu diesem Buch unter **<http://www.hauser-wenz.de/support/>** verlinkt. Wenn Sie mögen, hinterlassen Sie doch einfach einen kurzen Gruß! Support-Anfragen allerdings sollten Sie unter **<http://www.hauser-wenz.de/support/kontakt/>** stellen, dann erhalten Sie auch eine Antwort!*

## Ein Diskussionsforum erstellen

Unter <http://www.webmart.de/> gibt es noch viele weitere Tools, unter anderem ein Diskussionsforum. Der Clou: Mit Ihrem Login für das Gästebuch können Sie ebenfalls ein Diskussionsforum anlegen.

Doch es sollen auch andere Anbieter zu Wort kommen. Unter <http://www.forumfactory.com/> gibt es ein weiteres kostenloses und werbefinanziertes Forum. Lassen Sie sich durch den URL nicht täuschen, es ist in der Tat ein deutschsprachiges Angebot.



Was nun ist der Unterschied zwischen einem Gästebuch und einem Forum? In beiden kann ein Benutzer einen Kommentar oder allgemeiner einen Text hinterlassen. Der große Vorteil eines Forums: Dort geschieht das wesentlich strukturierter, es kann also eine Art von Diskussion aufkommen.

Während in einem Gästebuch alle Einträge im selben Datenspeicher, nacheinander beziehungsweise untereinander landen, kann man in einem Forum mehrere Themen angeben und dann themenspezifisch diskutieren. Zu diesem Buch beispielsweise ist ein Gästebuch hilfreich, wenn man einfach mal hallo sagen will, oder z.B., wie man das Buch findet. Ein Diskussionsforum könnte dazu verwendet werden, sich zum Buch auszutauschen, zu Buchthemen Fragen zu stellen und Gleichgesinnte zu finden.

Also, geben Sie <http://www.forumfactory.com/> eine Chance. Die folgenden Schritte führen Sie schnell zum fertigen Online-Forum:

- 1 Klicken Sie auf der Homepage von Forum Factory links auf den Link *Anmelden*.



- 2 Geben Sie Ihre Benutzerdaten an und klicken Sie auf *Anmelden*.

Zu den benötigten Daten gehören Ihr Name, Ihre E-Mail-Adresse, der URL Ihrer Homepage sowie ein gewünschter Benutzername für das Forum.

Auf der Folgeseite erscheinen nochmals alle Daten zur Bestätigung.

**3** Wenn die Daten in Ordnung sind, klicken Sie auf *Daten speichern/Forum anlegen*. Falls nicht, gehen Sie mit der *Zurück*-Schaltfläche des Webbrowsers auf die vorherige Seite.

**4** Warten Sie auf die Bestätigungsmail in Ihrem (elektronischen) Postfach.

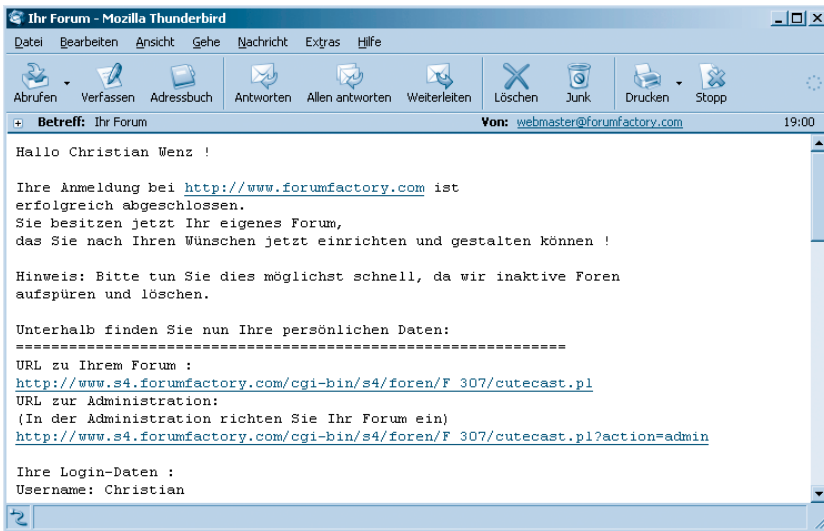
Dort finden Sie nicht nur Ihre Zugangsdaten (aus technischen Gründen ist auch diese Abbildung ohne Passwort, in Ihrer Mail ist es natürlich enthalten), sondern auch wichtige Links, und zwar:

- Wie Sie zu Ihrem Forum kommen.
- Wie Sie in den Administrationsbereich des Forums gelangen.

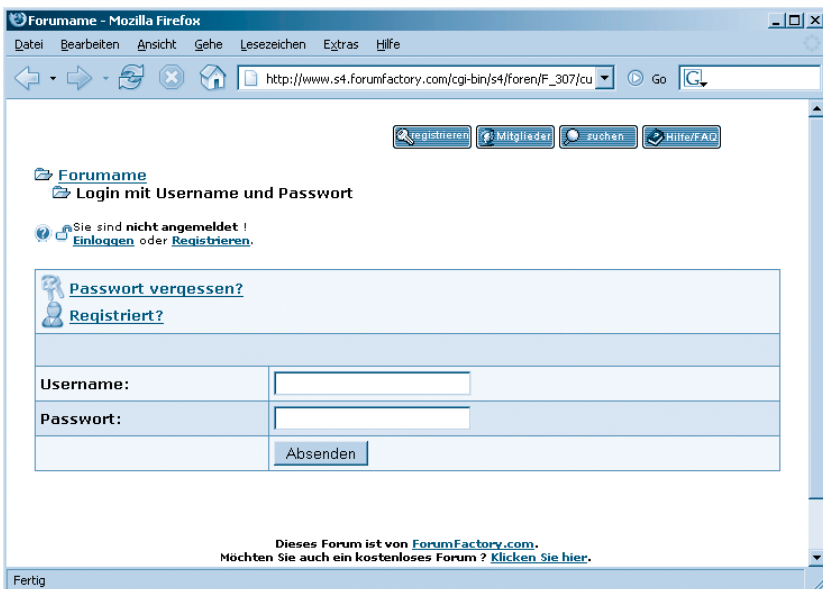
### Tipp

Sollten Sie einmal die Mail verlieren, erhalten Sie diese Informationen auch im Online-Bereich von **forumfactory.com**. Und: Wenn Sie die Adresse des Forums noch finden, gelangen Sie in den Administrationsbereich, indem Sie einfach hinten **?action=admin** anhängen.

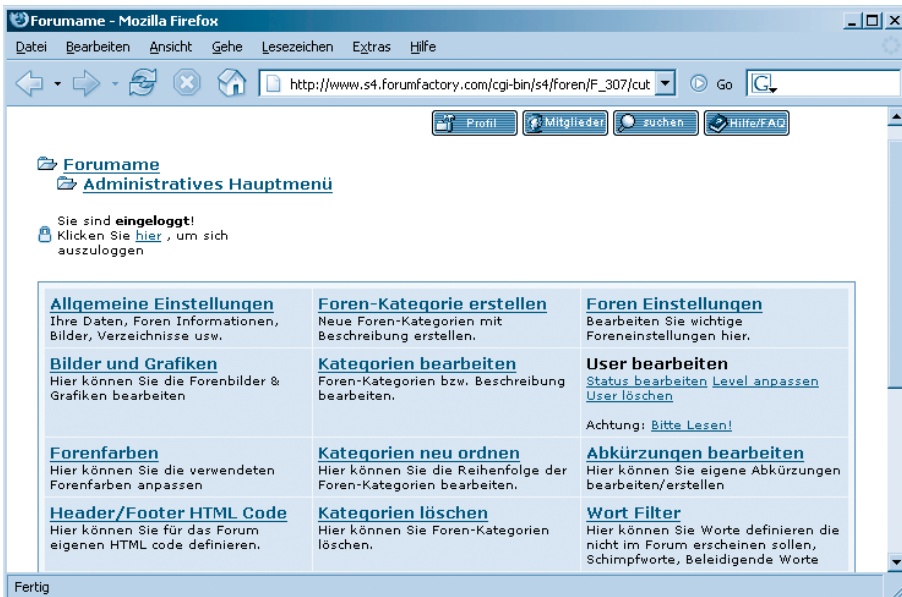




**5** Gehen Sie auf die Startseite des Administrationsbereichs (Link in der Mail) und geben Sie dort die Login-Daten aus der Mail an.

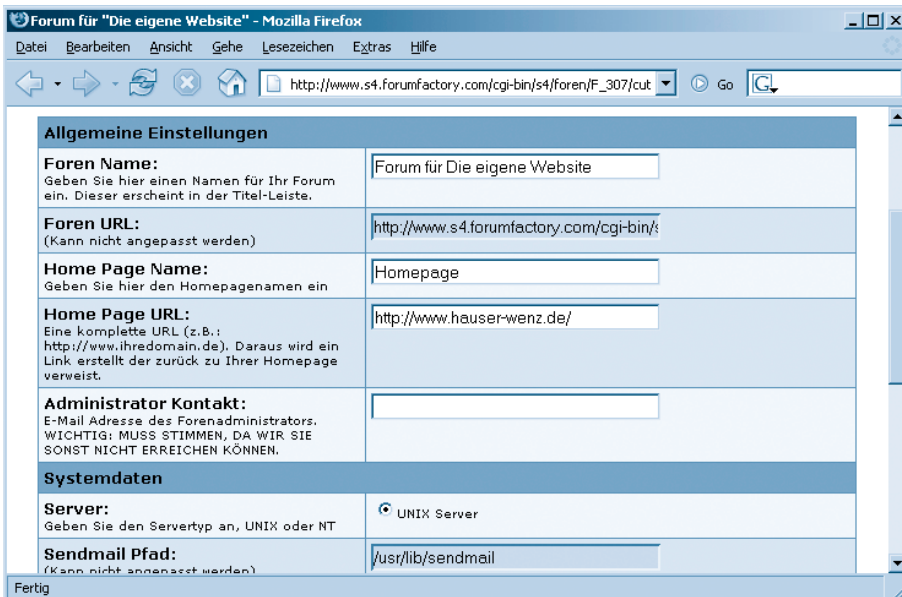


Sie gelangen dann in den Administrationsbereich des Forums. Der sieht auf den ersten Blick ziemlich unübersichtlich aus, doch das liegt unter anderem auch daran, dass es eben so viele Einstellungsmöglichkeiten gibt.



6 Klicken Sie auf *Allgemeine Einstellungen*.

Dort können Sie unter anderem den Namen des Forums einstellen, denn die Vorgabewerte sind nicht so sinnvoll.



**7** Klicken Sie auf *Administratives Hauptmenü* und dann auf *Foren-Kategorie erstellen*.

Forum für Die eigene Website - Mozilla Firefox

Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe

http://www.s4.forumfactory.com/cgi-bin/s4/foren/F\_307/cut

Forum für Die eigene Website

Administratives Hauptmenü

Foren-Kategorie erstellen

Sie sind **eingeloggt!**  
Klicken Sie [hier](#), um sich auszuloggen

**Kategorie:** Rund ums Buch

**Forum Name:** Ankündigungen

**Forum Typ:** ☐ Öffentlich ☐ Privat ☐ Geschlossen ☒ Ankündigung ☐ Unregistriert ☐ Registriert

**Beschreibung:** Neuerungen, Updates, ...

Absenden

Fertig

Dort können Sie Kategorien anlegen und dann innerhalb dieser Kategorie gleich ein Forum anlegen. Interessant ist die Angabe beim Punkt *Forum Typ*. Dort gibt es mehrere Optionen:

- *Öffentlich:* Jeder kann Nachrichten lesen und einstellen.
- *Privat:* Nur spezielle Nutzer haben Zugriff auf das Forum.
- *Geschlossen:* Das Forum ist geschlossen und damit nicht mehr zugänglich.
- *Ankündigung:* Nur der Administrator (also Sie) darf in das Forum posten.
- *Unregistriert:* Das Forum steht auch unregistrierten Nutzern zur Verfügung.
- *Registriert:* Das Forum steht nur registrierten Nutzern zur Verfügung.

Legen Sie also eine Kategorie und ein Forum an.

**8** Legen Sie ein zweites Forum in derselben Kategorie an.

Nach dem Erstellen der Kategorie landen Sie erneut auf der Seite zum Einrichten einer neuen Kategorie. Sie können dort aber auch bestehende Kategorien verwenden (wird bei *Kategorie* zur Auswahl angeboten) und somit neue Foren in bestehenden Kategorien anlegen.

Forum für Die eigene Website - Mozilla Firefox

Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe

http://www.s4.forumfactory.com/cgi-bin/s4/foren/F\_307/cut Go

Foren-Kategorie erstellen

Sie sind **eingelogg!**  
Klicken Sie [hier](#), um sich auszuloggen

|                                         |                                                                                                                                                                                                                       |
|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Kategorie:                              | <input checked="" type="radio"/> Rund ums Buch<br><input type="radio"/>                                                                                                                                               |
| Forum Name:                             | Fragen                                                                                                                                                                                                                |
| Forum Typ:                              | <input checked="" type="radio"/> Öffentlich <input type="radio"/> Privat <input type="radio"/> Geschlossen <input type="radio"/> Ankündigung<br><input type="radio"/> Unregistriert <input type="radio"/> Registriert |
| Beschreibung:                           | Fragen (und hoffentlich auch Antworten) zum Buch. Bitte immer mit Seitenzahl!                                                                                                                                         |
| <input type="button" value="Absenden"/> |                                                                                                                                                                                                                       |

Fertig

## Hinweis

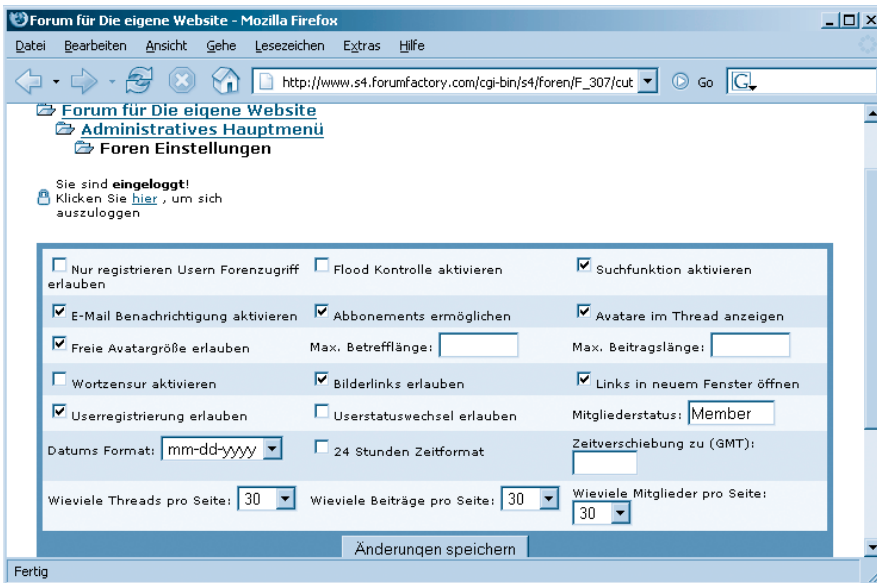
Wenn Sie wieder zurück in die Hauptseite des Administrationsmenüs gehen, finden Sie dort zwei Punkte, die ähnlich klingen, aber nicht das-selbe leisten:

- **Foren-Kategorie erstellen** – Dort können Sie eine neue Kategorie samt Forum anlegen oder ein neues Forum in einer bestehenden Kategorie erstellen.
- **Kategorien bearbeiten** – Dort können Sie alle Kategorien und Foren bearbeiten. Die werden zwar etwas unübersichtlich untereinander dargestellt, aber Sie haben Zugriff auf alle wichtigen Angaben wie etwa Name, Typ und Beschreibung.

**9** Klicken Sie auf den Link *Administratives Hauptmenü* und dann auf *Foren Einstellungen*.



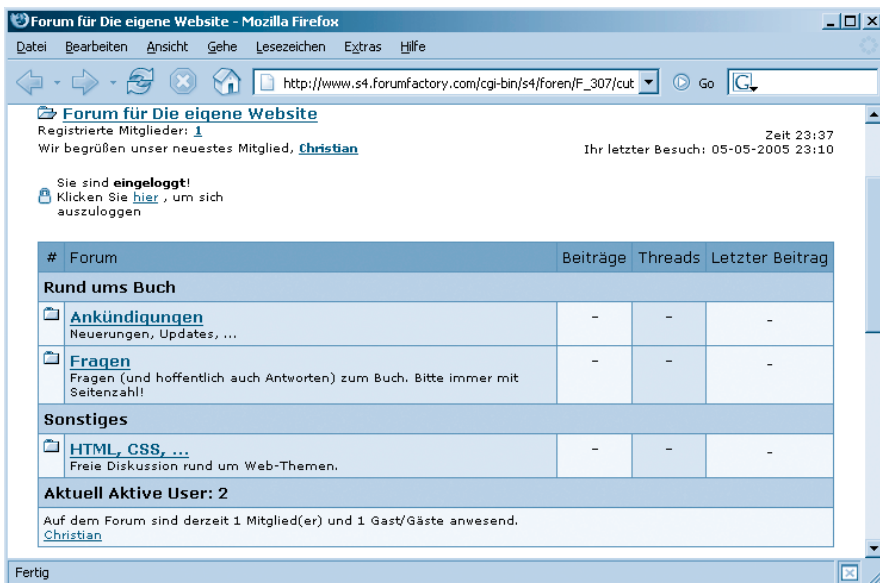




Hier haben Sie Zugriff auf einige, auch sicherheitsrelevante Einstellungen des kompletten Forums. Hier eine Auswahl der wichtigsten Punkte sowie Empfehlungen, wie diese einzustellen sind:

- *Nur registrierten Usern Forenzugriff erlauben* verbietet es anonymen Nutzern, das Forum einzusehen. Das sollte in der Regel nicht eingeschaltet werden, außer Sie haben wirklich ein berechtigtes Interesse daran, alle Nutzer zu kennen. Viele werden das aber als Gängelung interpretieren.
- *Flood Kontrolle aktivieren* beobachtet, ob vom selben Rechner in kurzer Zeit ganz viele Forenbeiträge kommen. Wenn Sie das aktivieren (was empfehlenswert ist), haben Sie so einen begrenzten Schutz vor automatischen Spam-Programmen (Sie erinnern sich, die von Leuten ohne Sozialleben, mit zu viel Zeit und fetttriefenden Haaren geschrieben worden sind).
- *Datums Format* stellt das Format des Datums eines Eintrags ein. Hier ist das US-Format voreingestellt, das mit dem Monat beginnt und dann erst den Tag folgen lässt. Das ist natürlich unschön, also stellen Sie es auf *dd-mm-yyyy* um.
- *24 Stunden Zeitformat* zeigt Uhrzeiten nicht im 12-Stunden-Format an, sondern verwendet das in unseren Gefilden gebräuchliche Uhrzeitformat von 0 bis 24 Uhr.
- *Zeitverschiebung zu (GMT)* sollte leer bleiben, denn Cutecast ist automatisch so konfiguriert, dass die mitteleuropäische Zeitzone verwendet wird.

**10** Klicken Sie auf den Link zum Forum, ganz oben im Administrationsmenü. Sie sehen alle angelegten Kategorien mit den zugehörigen Foren.



**11** Legen Sie in den Foren je eine Begrüßungsmitteilung an, in der Sie zum Beispiel noch erklären können, worum es in dem jeweiligen Forum genau geht. Wechseln Sie dazu zuerst in das jeweilige Forum und klicken Sie dann auf den Link *Schreiben*.



**12** Sehen Sie sich – eingeloggt mit Ihrem Administratorkonto – einen der Beiträge an.

Sie sehen einige administrative Optionen, unter anderem die folgenden:

- Sie können einen Beitrag verändern.
- Ganz unten können Sie einen Thread komplett löschen oder an eine andere Stelle verschieben.

Hier gelten natürlich dieselben Vorsichtsmaßnahmen wie beim Verändern oder Löschen eines Gästebucheintrags: Gehen Sie hier sehr bedacht vor.



### Tipp

Unter <http://www.forumfactory.com/faq/> gibt es eine recht umfangreiche Dokumentation zur Benutzung des Forums.

Auch dieses Forum ist im Support-Bereich zu diesem Buch (unter <http://www.hauser-wenz.de/support/>) verlinkt, so dass Sie sich gerne dort austauschen können.

# Einen Weblog anlegen

Einer der neuesten Trends im World Wide Web sind so genannte Blogs. Das ist die Kurzform von Weblogs, also Webtagebüchern. So können Sie Ihre Freunde und Gleichgesinnte über Ihr Leben auf dem Laufenden halten. Es gibt verschiedene Arten von Weblogs: Manche langweilen durch arg narzisstische Beschreibungen des Alltagstrots, andere enthalten kontroverse Meinungen zu aktuellen Themen oder witzige Links auf interessante Websites.

Auch die Autoren dieses Buchs haben einen Weblog, und zwar unter <http://www.hauser-wenz.de/blog/>. Der enthält eine ganze Reihe englischer Einträge, aber auch immer wieder mal deutsche Inhalte.



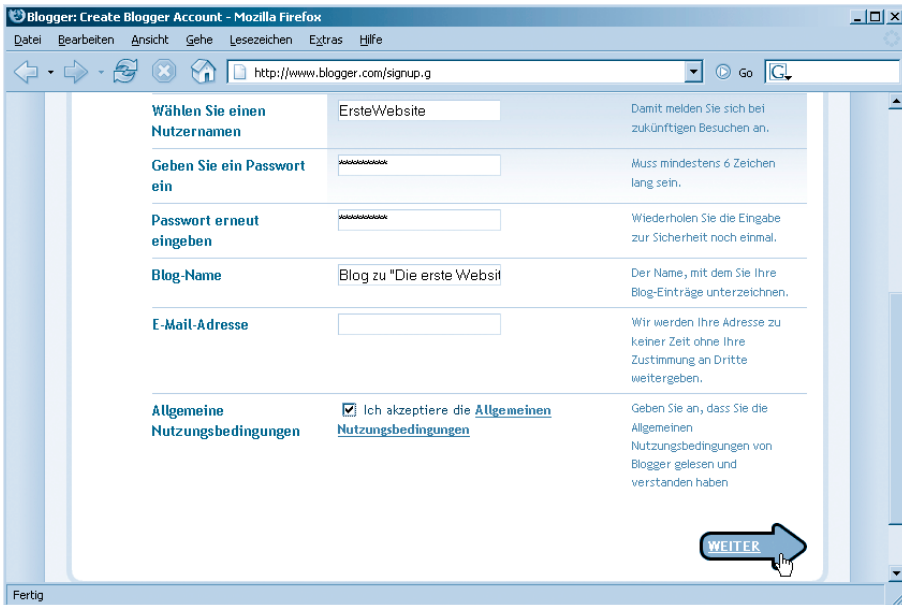
Und auch Sie können sehr schnell einen Weblog erstellen. Eine Reihe kostenloser Dienste steht zu Ihrer Verfügung, wovon wir exemplarisch einen sehr bekannten herauspicken:

- 1 Gehen Sie auf <http://www.blogger.com/> und klicken Sie auf *Blog erstellen*.





## 2 Geben Sie die Grundparameter für Ihren Weblog an.



Dazu gehören der Name des Blogs, ein Nutzernamen samt Passwort sowie Ihre E-Mail-Adresse. Sie müssen auch die allgemeinen Nutzungsbedingungen des Systems akzeptieren – lesen Sie sie also am besten vorher einmal durch.

**3** Suchen Sie sich einen URL für den Weblog aus.

Der Blog hat eine Adresse der Form *http://xxx.blogspot.com/*, wobei Sie sich das xxx recht frei wählen können. Außerdem können Sie einen Titel für den Weblog angeben und müssen mal wieder ein Wort aus einer Grafik abtippen. Sie wissen ja: Spammer sind Leute ohne Sozialleben, mit zu viel Zeit, fettigen Haaren und niedrigem IQ.

**2 Benennen Sie Ihren Blog**

**Blog-Titel** ? Die erste Website Geben Sie einen Titel für Ihren Blog ein.

**Blog-Adresse (URL)** ? http://erstewebsite.blogspot.com Damit gelangen Sie und andere zu Ihrem Blog, um darin zu lesen oder einen Link anzulegen.

**Word Verification**  Geben Sie die Zeichen aus dem unten angezeigten Bild ein.

uodbdsys

**ODER**

**Erweiterte Einrichtung** Sie wollen Ihren Blog woanders hosten? Benutzen Sie die [Erweiterte Blog-Einrichtung](#). Dort können Sie Ihren Blog woanders, als auf Blogspot hosten.

**WEITER**

Fertig

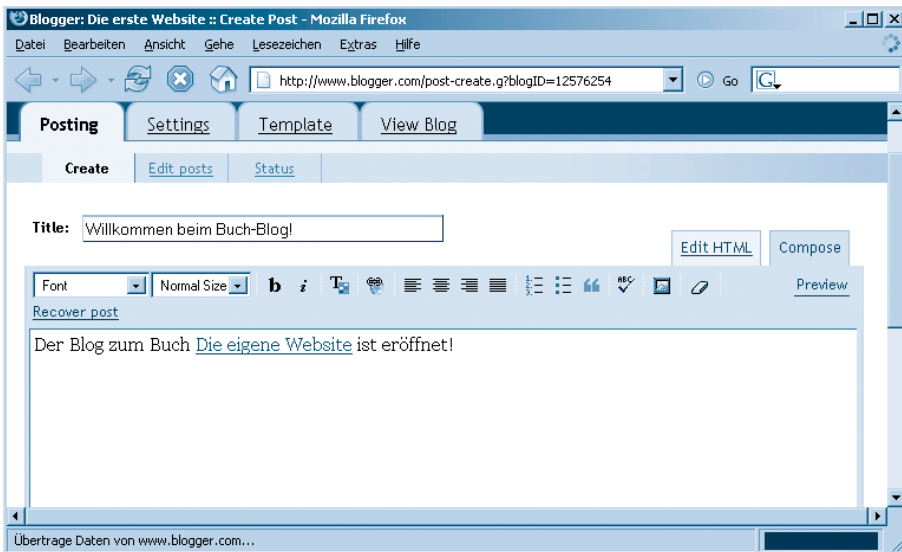
**4** Wählen Sie eine der angebotenen Vorlagen aus.

Dann legt Blogger den Weblog an und bietet Ihnen den Link *Eintrag starten*. Da klicken Sie natürlich drauf.



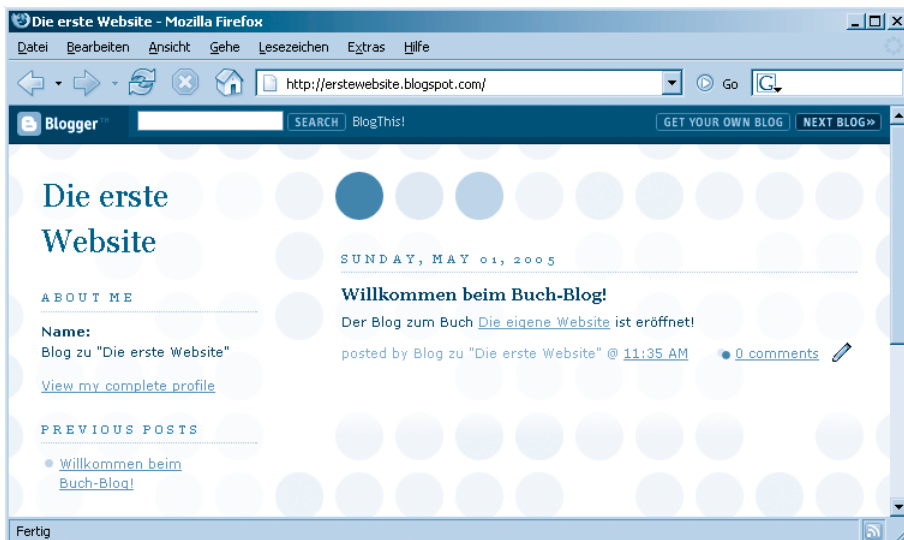


5 Schreiben Sie einen Blogbeitrag und klicken Sie auf *Publish Post*.



Blogger legt den Eintrag an und veröffentlicht ihn. Jetzt können Sie alle daran teilhaben lassen!

## 6 Schauen Sie sich Ihren Weblog im Webbrowser an.



Nun können Sie die Adresse des Weblogs (*http://<irgendetwas>.blogspot.com/*) weitergeben und einen Link darauf in Ihre Website integrieren. Wenn Sie sich allerdings bei *http://www.blogger.com/* einloggen, können Sie sich ins Administrationsmenü begeben und dort unter anderem:

- neue Einträge anlegen
- alte Einträge verändern
- alte Einträge löschen
- das Design des Weblogs ändern
- Kommentare zu Weblog-Einträgen entfernen (etwa bei Spam-Angriffen)
- und vieles mehr

Das ist natürlich nur der Anfang, aber Sie sehen schon: Die Möglichkeiten sind endlos. Ein gut geführter Weblog mit guten Inhalten schafft es dann unter Umständen mal in den SPIEGEL, wie erst unlängst geschehen. Doch Sie müssen natürlich Marketing für den Weblog treiben und die Adresse weitergeben. Irgendwoher müssen die Besucher von Ihrem Blog erfahren, zumindest am Anfang; danach läuft es hoffentlich über Mundpropaganda.





**Hinweis**

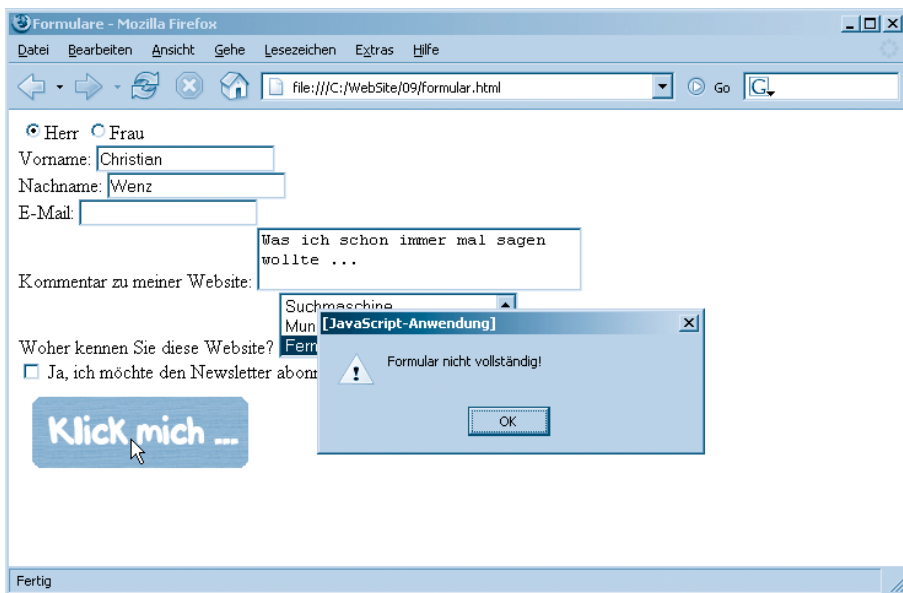
*Für alle Webdienste in diesem Kapitel gilt, dass diese zum Zeitpunkt der Bucherstellung verfügbar waren und davor jahrelang gute Dienste geleistet haben. Dennoch kann es natürlich passieren, dass der eine oder andere Dienst sein Geschäftsmodell ändert (sprich Geld haben will) oder gar eingestellt wird. Aber durch Google und andere Suchmaschinen finden Sie sicherlich schnell Ersatz – oder sehen Sie im Support-Bereich zu diesem Buch nach.*

## Nutzloses Wissen

Alle zwei Wochen stirbt irgendeine Sprache aus.

## Kapitel 9

# Mehr Pep mit JavaScript



Die letzten Kapitel haben gezeigt, wie Sie mit HTML und CSS tolle Websites erstellen können. Allerdings sind diese Seiten alle statisch. Das bedeutet: immer gleich, keine wirkliche Reaktion auf Dinge, die der Benutzer tut.

Mit der Programmiersprache JavaScript lassen sich diese Beschränkungen umgehen – und es ist gar nicht mal so schwer! Dieses Kapitel bietet eine Einführung in die Sprache und vor allem viele Beispiele, die direkt eingesetzt werden können.

**Das können Sie schon:**

|                                    |     |
|------------------------------------|-----|
| Formatierung von Text              | 56  |
| Einsatz von Links                  | 75  |
| Aufzählungslisten                  | 88  |
| Grafiken einbauen                  | 116 |
| Frames erstellen                   | 134 |
| Formulare verwenden                | 158 |
| Mit Tabellen und CSS positionieren | 194 |
| Ein Gästebuch einrichten           | 214 |
| Ein Diskussionsforum erstellen     | 222 |
| Einen Weblog anlegen               | 232 |

**Das lernen Sie neu:**

|                        |     |
|------------------------|-----|
| Einstieg in JavaScript | 240 |
| Datum ausgeben         | 258 |
| Quicknavigation        | 260 |
| Mehrere Frames ändern  | 267 |
| Frames nachladen       | 270 |

## Was ist JavaScript?

JavaScript erblickte im September des Jahres 1995 das Licht der Welt. Netscape stellte damit die lang erwartete Version 2.0 ihres Webbrowsers vor. Eine der zahlreichen Neuerungen war die Integration einer Programmiersprache. Diese hieß zunächst LiveScript, doch drei Monate später hatten die Marketingabteilungen von Netscape und Sun Microsystems vereinbart, das Ganze in JavaScript umzutaufen, damit es so ähnlich klingt wie Java, damals ein weiterer heißer Trend im Internet.

### Hinweis

*Was JavaScript mit Java zu tun hat? Praktisch nichts. Nun gut, JavaScript und Java sehen innerhalb gewisser Grenzen ähnlich aus, aber das war es auch schon. Ansonsten: ein reiner Marketinggag. Das führt aber leider immer noch dazu, dass viele Leute Java und JavaScript nicht auseinander halten können: Merken Sie sich einfach Folgendes: Im Webbrowser hat Java zurzeit kaum mehr eine Bedeutung, JavaScript ist aber präsenter denn je.*

Bei JavaScript handelt es sich um eine Programmiersprache. Das ist schon ein ganzes Stückchen mehr als die Themen, die wir bisher in diesem Buch behandelt haben. HTML und CSS sind »nur« Seitenbeschreibungssprachen: Sie geben an, welche Elemente sich auf einer Seite befinden, in welcher Reihenfolge sie angezeigt werden sollen und wie sie zu formatieren sind. Außerdem können Sie bei modernen Browsern die einzelnen Elemente noch platzieren. Das war es allerdings auch schon.

Bei einer Programmiersprache dagegen können Sie das Ganze noch mit ein wenig Dynamik versehen: Sie können auf Benutzereingaben reagieren, mit dem aktuellen Tagesdatum arbeiten, auf Frames zugreifen, neue Fenster öffnen und vieles mehr.

Um das zu bewerkstelligen, müssen Sie programmieren. Das heißt, Sie schreiben auf, was JavaScript tun soll. Da der Browser, in dem JavaScript-Code abläuft, schlecht im Raten ist, was Sie von ihm erwarten, müssen Sie diese Anweisungen in einem bestimmten Format schreiben. Sie müssen die so genannte Syntax von JavaScript einhalten. Das sind Vorschriften, wie Code aussehen muss.



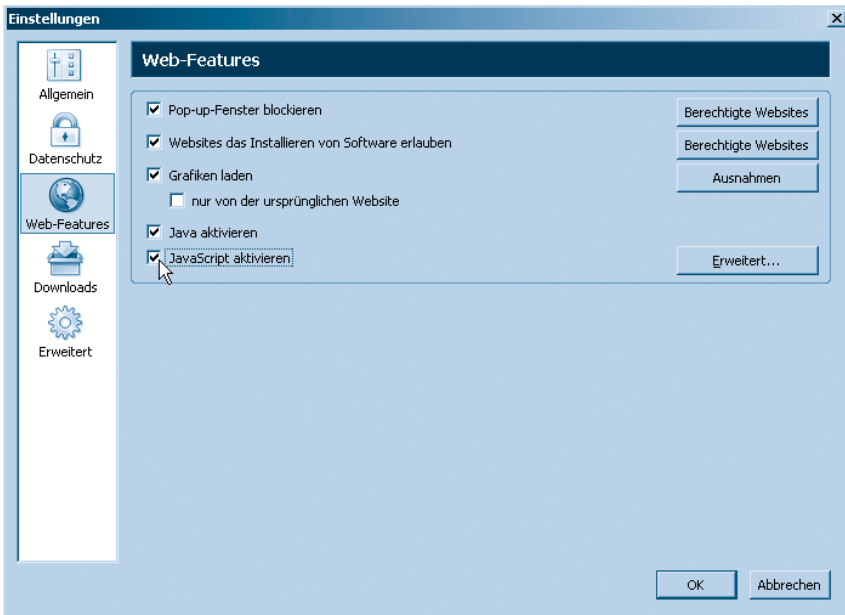
Auch wenn einige der Codeelemente in diesem Kapitel auf den ersten Blick etwas wüst aussehen mögen, lassen Sie sich davon nicht abhalten. Wir beginnen mit einer leichten Einführung in die Sprache JavaScript, um Ihnen ein Gefühl dafür zu geben, wie JavaScript aussieht und wie es funktioniert. Dann aber bringen wir zahlreiche Codebeispiele, die Sie praktisch 1 : 1 in Ihre eigenen Webseiten übernehmen können. Wie immer in diesem Buch erklären wir allerdings auch genau, was in dem Code passiert. Nur dann wissen Sie nämlich, was im Hintergrund passiert, und können unter Umständen auch Anpassungen vornehmen.

## JavaScript einbauen

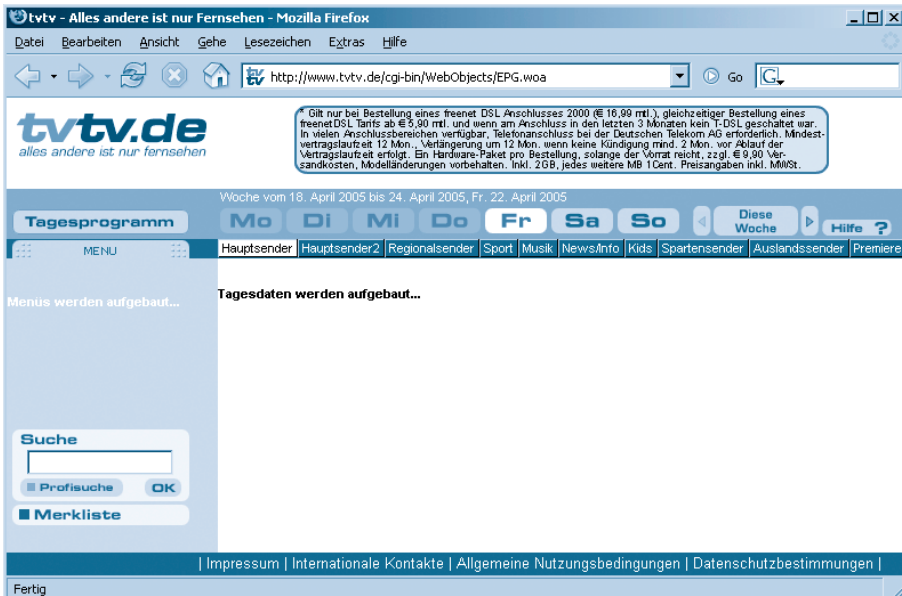
JavaScript ist in den HTML-Code eingebettet, so ähnlich wie CSS in HTML eingebettet werden kann. Sprich, der Browser muss JavaScript verstehen und dann den Code ausführen. JavaScript ist damit eine so genannte **Skriptsprache** oder **Interpretersprache**, denn es wird ein Programm benötigt, das den Code übersetzen – **interpretieren** – kann. Der Browser kann JavaScript interpretieren.

Bleibt die Frage, ob jeder Browser das kann. Die Antwort: Prinzipiell ja. Die meisten Webbrowser unterstützen JavaScript, und von den aktuellen und relevanten Browsern tun das fast alle. Allerdings traten bei sowohl dem Internet Explorer als auch dem Mozilla-Browser in letzter Zeit immer wieder mal Sicherheitslücken auf, die mit der Skriptsprache in Zusammenhang standen. Aus diesem Grund gibt es doch einige Anwender, die JavaScript deaktivieren; Beim Internet Explorer wurde es sogar lange Zeit empfohlen. Beispielsweise finden Sie im Firefox-Browser unter *Extras/Einstellungen/Web-Features* den Punkt *JavaScript aktivieren*. Standardmäßig ist die Checkbox angekreuzt, aber diese Einstellung kann eben auch deaktiviert sein. Daraus leitet sich die folgende wichtige Regel ab: Setzen Sie JavaScript nur als Ergänzung ein, denn die Webseite **muss** stets auch ohne JavaScript funktionieren.

Das ist eigentlich ein relativ trivialer Gedanke, doch gibt es in der Tat zahlreiche Websites, die ohne JavaScript einfach nicht mehr funktionieren. Probieren Sie es doch einmal aus: Deaktivieren Sie JavaScript und sehen Sie dann, ob Ihre Lieblingswebsites noch so funktionieren wie zuvor.



Die beiden folgenden Abbildungen sollen das demonstrieren: eine ansonsten empfehlenswerte Website mit tagesaktuellem Fernsehprogramm einmal ohne und einmal mit aktiviertem JavaScript.



Dummerweise erscheint keine Warnmeldung, wenn JavaScript ausgeschaltet ist. Der Anwender wundert sich also und guckt sprichwörtlich in die Röhre.



Doch nun zu JavaScript selbst. Hier Ihre erste Zeile JavaScript-Code:

```
alert("JavaScript ist aktiviert!");
```

Programmieren ist immer, als würde man schlechtes Englisch schreiben. Obiger Code gibt eine Warnmeldung aus («alert» ist Englisch für Alarm oder alarmieren) mit dem Text »JavaScript ist aktiviert!«. Der Text ist in doppelte Anführungszeichen eingeschlossen; das ist sehr wichtig, damit JavaScript weiß, wo der Text beginnt und wo er aufhört. Am Ende des Programmierbefehls kommt immer ein Strichpunkt. Auch dies ist ein Hinweis an JavaScript, damit bekannt ist, wo der Befehl aufhört (ein Befehl kann sich nämlich über mehrere Zeilen erstrecken).

Eingebaut wird das Ganze mit einem `<script>`-Element. Dort geben Sie zwei Attribute an:

- Das Attribut `language` wird auf `"JavaScript"` gesetzt.
- Das Attribut `type` wird auf `"text/javascript"` gesetzt.

Hier kollidieren ein wenig die Ansprüche von Standardhüter und Pragmatiker. Offiziell »richtig« ist die Verwendung des `type`-Attributs; wenn es allerdings auch in älteren Browsern funktionieren soll, muss das `language`-Attribut eingesetzt werden. Logische Konsequenz: Wie verwenden einfach beides. Hier der Code von vorher noch einmal innerhalb von `<script>`-Elementen.

```
<script language="JavaScript"
 type="text/javascript"><!--
alert("JavaScript ist aktiviert!");
//--></script>
```

### Tipp

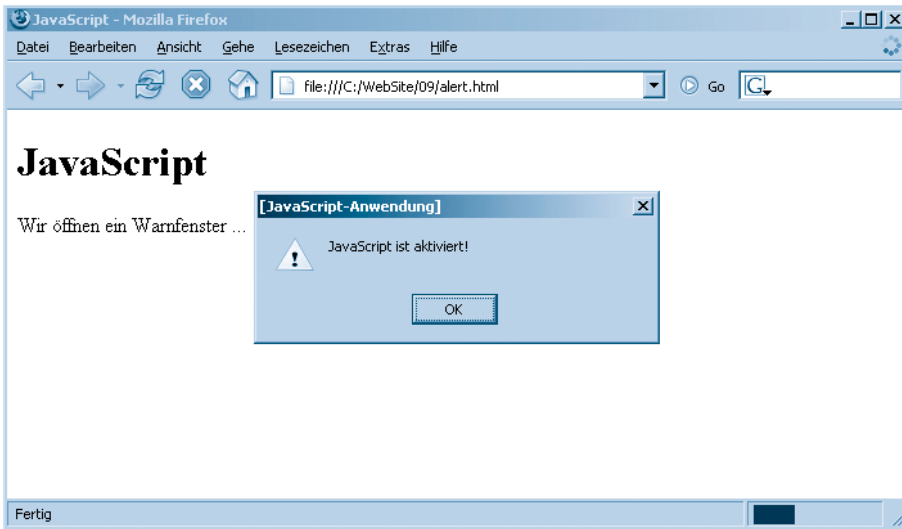
*Die Verwendung von `<!--` und `-->` ist ein kleiner Kniff, um ältere Browser, die gar kein JavaScript unterstützen, nicht vor den Kopf zu stoßen. Die nämlich erkennen `<!--` als Beginn eines HTML-Kommentars, sprich sie ignorieren den JavaScript-Code. Würden Sie das weglassen, würde ein (sehr, sehr) alter Browser den JavaScript-Code einfach im Webbrowser anzeigen.*

Hier noch einmal der komplette JavaScript-Code in ein HTML-Dokument integriert:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- alert.html -->
<html>
<head>
 <title>JavaScript</title>
</head>
<body>
 <h1>JavaScript</h1>
 <p>
 Wir öffnen ein Warnfenster ...
 </p>
 <script language="JavaScript"
 type="text/javascript"><!--
 alert("JavaScript ist aktiviert!");
 //--></script>
</body>
</html>
```







Der JavaScript-Code kann also an beliebiger Stelle im Code platziert werden. Üblicherweise verwendet man ihn im `<head>`-Abschnitt des HTML-Dokuments, aber wie im Listing zu sehen ist das auch überall sonst im Code zulässig.

Es gibt aber noch eine zweite Möglichkeit. Der komplette JavaScript-Code kann auch in einer externen Datei stehen. Die externe Datei sieht dann z.B. so aus – nur Code, kein `<script>`-Element:

```
// alert.js
alert("JavaScript ist aktiviert!");
```

Sobald Sie in einem JavaScript-Listing zwei Schrägstriche sehen (`//`), bedeutet das Folgendes: Alles danach in dieser Zeile wird vom JavaScript-Interpreter ignoriert. Das ist ein so genannter Kommentar. So können Sie der Nachwelt Informationen darüber hinterlassen, was Sie sich bei dem entsprechenden Quellcode gedacht haben – beziehungsweise wie der JavaScript-Code funktioniert.

Die externe Datei (Dateiendung ist immer `.js`) wird jetzt wie folgt in das HTML-Dokument eingebunden:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- alert-extern.html -->
<html>
<head>
 <title>JavaScript</title>
</head>
```

```

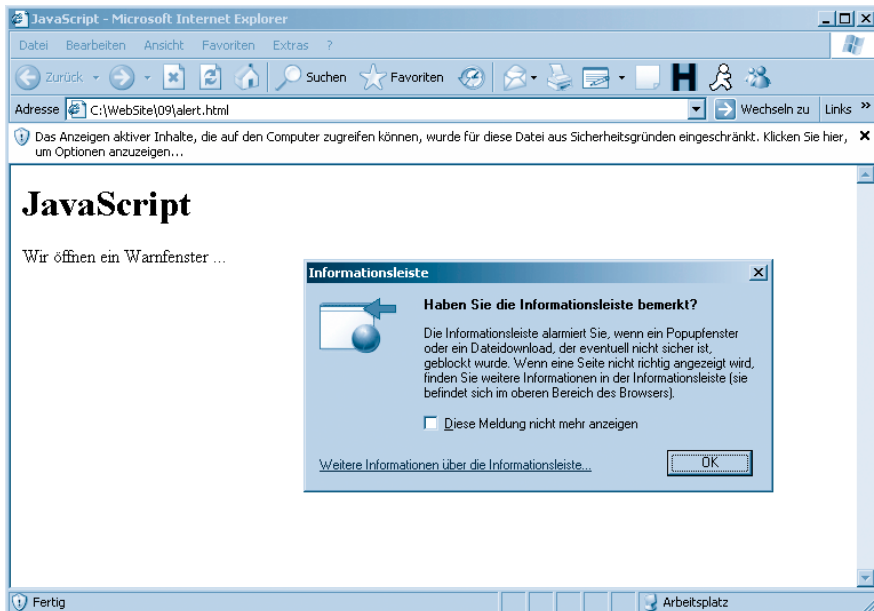
<body>
 <h1>JavaScript</h1>
 <p>
 Wir öffnen ein Warnfenster ...
 </p>
 <script language="JavaScript"
 type="text/javascript"
 src="alert.js"></script>
</body>
</html>

```

Sie müssen also den Dateinamen im src-Attribut angeben. Alles, was dann zwischen <script> und </script> steht, wird ignoriert! Sie müssen sich also bei jedem <script>-Element entscheiden: entweder direkt JavaScript-Code einfügen oder das src-Attribut setzen.

## Den Internet Explorer unter Windows XP Service Pack 2 zur Kooperation bewegen

Wenn Sie die vorherigen Beispiele im Internet Explorer aufrufen und dabei Windows XP verwenden (und natürlich das Service Pack 2 installiert haben), stoßen Sie auf Probleme. Zunächst erscheint ein Warnfenster, das uns mitteilt, dass eine Informationsleiste eingeblendet worden ist.

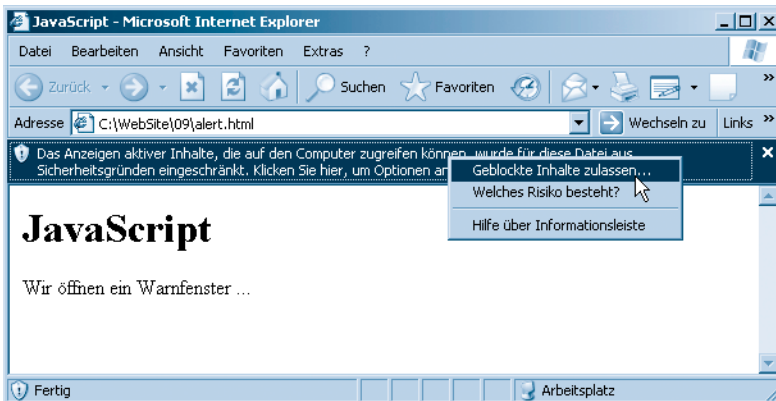


Ein Blick in diese Informationsleiste macht klar, dass »das Anzeigen aktiver Inhalte«, was auch immer das sein mag, »aus Sicherheitsgründen eingeschränkt« worden ist. Auf gut Deutsch: Der Internet Explorer hat JavaScript-Code festgestellt und schlägt Alarm. Doch warum das Ganze? Der Hintergrund: Wenn in einer lokalen Datei auf der Festplatte JavaScript-Code enthalten ist, kann dieser möglicherweise ganz gefährliche Dinge anstellen. Deswegen erscheint die Warnmeldung. Wenn Sie dieselbe Seite aus dem Internet aufrufen (siehe *Kapitel 11*, wie Sie ihre HTML-Dokumente online stellen können), kommen Sie übrigens ohne Warnmeldung aus.

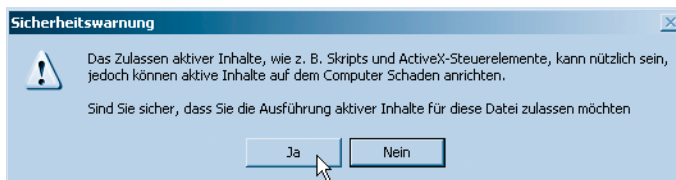
Beim lokalen Testen ist das allerdings mehr als nervig. Denn es sind immer dieselben Schritte nötig, um den JavaScript-Code zur Ausführung zu bringen – bei jedem Seitenaufruf wieder:

**1** Klicken Sie auf die Informationsleiste.

**2** Wählen Sie den Eintrag *Geblockte Inhalte zulassen...*



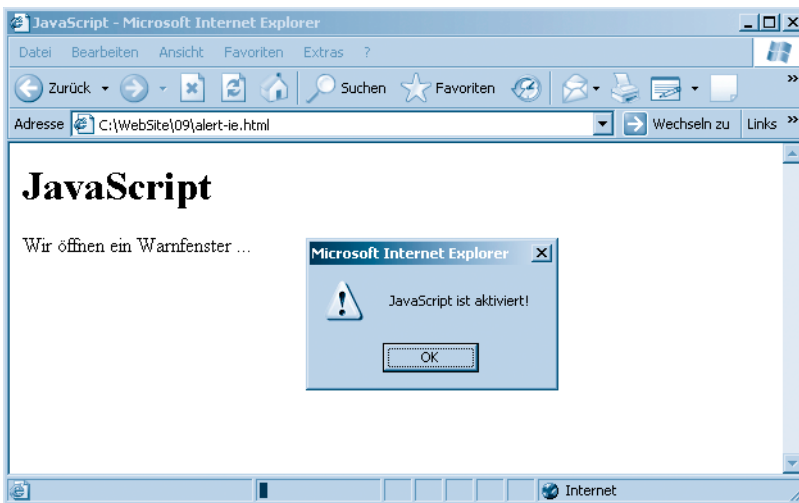
**3** Bestätigen Sie die **erneute** Abfrage mit einem Klick auf *Ja*.



Wie gesagt – jedes Mal müssen Sie das wieder so machen. Das nervt schon nach kurzer Zeit. Zum Glück gibt es einen kleinen Workaround. Dazu müssen Sie an den Anfang der Datei einen Kommentar nach folgender Machart einfügen:

```
<!-- saved from url=(0026)http://www.hauser-wenz.de/ -->
```

Sie können auch jede beliebige andere Adresse angeben, Hauptsache, die Länge des URLs wird in die Klammern davor richtig eingetragen – im Beispiel sind es genau 26 Zeichen. Dann nämlich behandelt der Internet Explorer die Seite wie eine Internetseite – auch wenn sie lokal aufgerufen wird. Das sehen Sie in der Abbildung unter anderem daran, dass rechts unten nicht mehr *Arbeitsplatz* steht, sondern *Internet*. Hier zunächst noch der Code, der im Internet Explorer funktioniert – und der gleichzeitig Muster dafür ist, wie wir alle weiteren JavaScript-Listings in diesem Buch anpassen, so dass sie auch lokal im IE funktionieren:



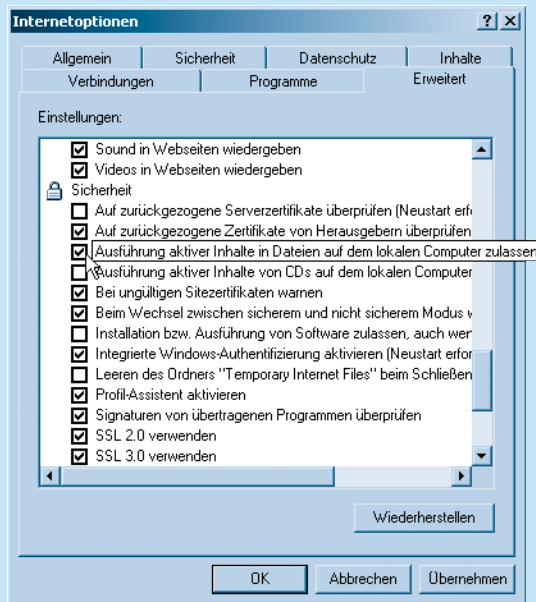
### Hinweis

*Allerdings klappen dann im Internet Explorer lokal unter anderem Links nicht mehr wie gewünscht. Sie sollten den speziellen HTML-Kommentar also nur zum Testen einsetzen.*



### Tipp

Am bequemsten ist es freilich, Sie schalten diese Warnmeldungen im Internet Explorer komplett aus. Dazu benötigen Sie den Menüpunkt **Extras/Internetoptionen/Erweitert**; dort finden Sie unter **Sicherheit** bei **Erweitert** unter **Sicherheit** den Punkt »Ausführung aktiver Inhalte in Dateien auf dem lokalen Computer zulassen«.



## Einstieg in die Programmierung

Steigen wir in die Programmierung von JavaScript ein. Wie Sie bereits gesehen haben, gibt es verschiedene Befehle, die alle mit einem Strichpunkt enden. Wenn Sie in JavaScript programmieren, müssen Sie ganz exakt arbeiten. Bei HTML kann es sein, dass trotzdem etwas angezeigt wird, wenn Sie einen Fehler machen. Bei der Programmierung ist das etwas anderes.

Sehen Sie sich mal den folgenden Code an:

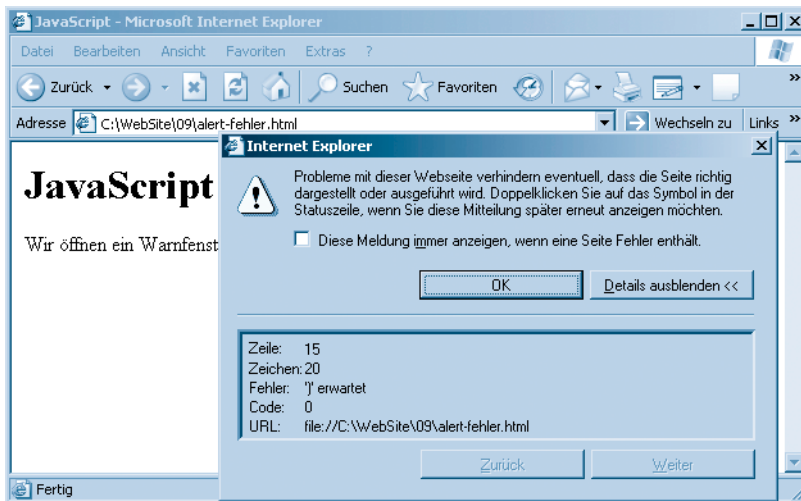
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- saved from url=(0026)http://www.hauser-wenz.de/ -->
<!-- alert-fehler.html -->
```

```

<html>
<head>
 <title>JavaScript</title>
</head>
<body>
 <h1>JavaScript</h1>
 <p>
 Wir öffnen ein Warnfenster ...
 </p>
 <script language="JavaScript"
 type="text/javascript"><!--
 alert(JavaScript ist aktiviert!);
 //--></script>
</body>
</html>

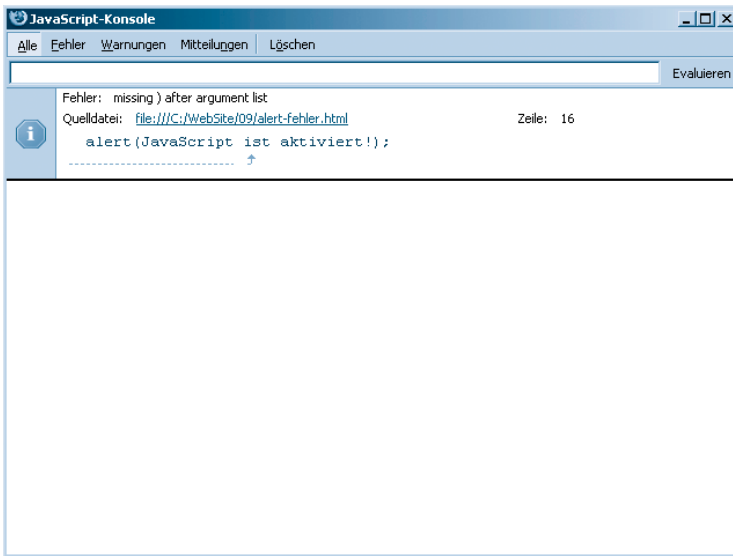
```

Wenn Sie die Seite im Internet Explorer aufrufen, geschieht zunächst gar nichts, es erscheint auch kein Fenster. Eventuell sehen Sie links unten im Fenster ein gelbes Warndreieck, aber auch nicht auf allen Systemen. Aber klicken Sie doch einmal links unten auf das Symbol, sei es ein Dreieck oder auch ein blaues »e«. Sie erhalten eine Fehlermeldung – im Code stimmt also irgendetwas nicht.



Im Firefox-Browser geht das ganz ähnlich. Die Datei *alert-fehler.html* öffnet kein neues Fenster. Zeit, dem auf den Grund zu gehen. Unter *Extras/JavaScript-Konsole* wird das JavaScript-Kontrollfenster aufgerufen. Dort sehen Sie etwaige Fehlermeldungen, so auch in diesem Fall.





### Hinweis

*Aber was war jetzt eigentlich der Fehler? Werfen Sie noch einmal einen Blick auf die entsprechende Zeile:*

```
alert(JavaScript ist aktiviert!);
```

*Es fehlen die Anführungszeichen rund um die Zeichenkette. Richtig ist also Folgendes:*

```
alert("JavaScript ist aktiviert!");
```

Wenn also etwas nicht klappen will, finden Sie hier Rat.

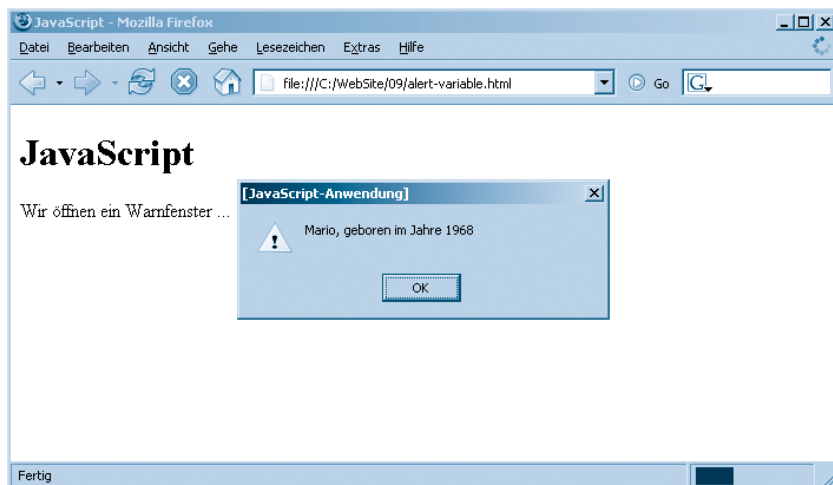
Das wichtigste Element bei der Programmierung sind zweifelsohne Variablen. Das sind Datenspeicher, die verschiedene, variable (sprich veränderliche – daher der Name) Werte enthalten.

Wenn Sie das erste Mal eine Variable verwenden, müssen Sie das Schlüsselwort `var` davor schreiben. Mit dem Gleichheitszeichen weisen Sie der Variablen einen Wert zu. Dabei gilt: Texte (Zeichenketten) in Anführungszeichen, Zahlen nicht. Außerdem braucht die Variable noch einen Namen. Verwenden Sie am besten nur Buchstaben, vor allem keine Leerzeichen. Zusammengefasst sieht das dann beispielsweise so aus:

```
var name = "Mario";
var jahr = 1968;
```

Diese Variablen können ausgegeben werden, beispielsweise wieder mit `alert()`. Dazu geben Sie den Variablennamen einfach innerhalb der runden Klammern an – ohne Anführungszeichen, denn es geht um die Variable, nicht um einen Text:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- saved from url=(0026)http://www.hauser-wenz.de/ -->
<!-- alert-variable.html -->
<html>
<head>
 <title>JavaScript</title>
</head>
<body>
 <h1>JavaScript</h1>
 <p>
 Wir öffnen ein Warnfenster ...
 </p>
 <script language="JavaScript"
 type="text/javascript"><!--
 var name = "Mario";
 var jahr = 1968;
 alert(name + ", geboren im Jahre " + jahr);
 //--></script>
</body>
</html>
```



Werfen wir noch einen Blick auf die Anweisung zum Ausgeben der Variablen:  
`alert(name + ", geboren im Jahre " + jahr);`

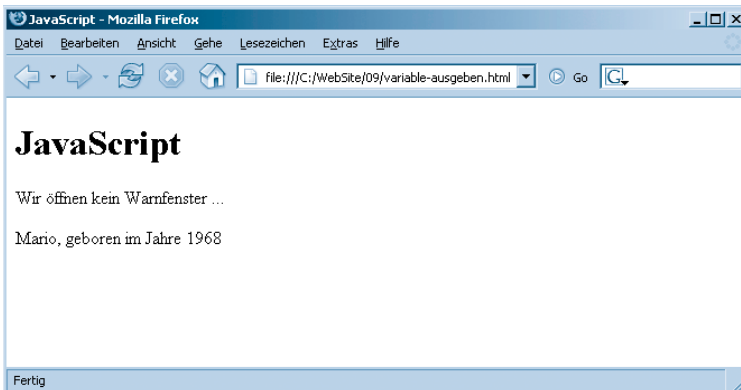




Durch die Pluszeichen werden drei Dinge aneinander gehängt: die Variable `name` (ohne Anführungszeichen), der Text `", geboren im Jahre "` (mit Anführungszeichen) sowie die Variable `jahr` (ohne Anführungszeichen). Der Inhalt der beiden Variablen und der Text werden dann auf einmal ausgegeben, wie auch die Abbildung zeigt.

Auf die Dauer werden die Warnfenster allerdings etwas nervig. Es gibt aber eine Alternative: Mit `document.write()` geben Sie direkt Text im Browser aus, können also so HTML erzeugen. Denken Sie nur an die richtigen Anführungszeichen. Hier derselbe Code wie zuvor, allerdings ohne Warnfenster, sondern mit Direktausgabe:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- saved from url=(0026)http://www.hauser-wenz.de/ -->
<!-- variable-ausgeben.html -->
<html>
<head>
 <title>JavaScript</title>
</head>
<body>
 <h1>JavaScript</h1>
 <p>
 Wir öffnen kein Warnfenster ...
 </p>
 <script language="JavaScript"
 type="text/javascript"><!--
 var name = "Mario";
 var jahr = 1968;
 document.write(name + ", geboren im Jahre " + jahr);
 //--></script>
</body>
</html>
```



An einigen Stellen im Code müssen Entscheidungen gefällt werden. Sehen wir uns einmal als Beispiel die folgende Formulierung im Deutschen an: »Wenn heute Samstag oder Sonntag ist, erhöhe die Benzinpreise, ansonsten senke sie.« Dieser Satz lässt sich fast ohne Änderungen ins »Englische« übertragen, und (fast) fertig ist das JavaScript-Programm. Das könnte also wie folgt aussehen:

```
if (tag == "Samstag") {
 erhoeheBenzinpreise();
} else if (tag == "Sonntag") {
 erhoeheBenzinpreise();
} else {
 senkeBenzinpreise();
}
```

Dieses Pseudolisting enthält mehrere relevante Elemente:

- `if` steht für »wenn«. Das leitet eine Bedingung ein.
- Die Bedingung steht in runden Klammern. Das ist ganz wichtig!
- Was passieren soll, wenn die Bedingung erfüllt ist, befindet sich hinter der Bedingung in geschweiften Klammern. Diese Klammern (verfügbar über `[AltGr]+[7]` und `[AltGr]+[0]`) zeigen dem JavaScript-Interpreter, wo der Code beginnt und wo er aufhört.
- `else` steht für »sonst«. Wenn also weder die Bedingung bei `if` noch die bei `else if` erfüllt ist, wird berücksichtigt, was bei `else` angegeben ist.

Sie sehen also: Es ist, wie bereits erwähnt, wie etwas holprig formuliertes Englisch.

Jetzt müssen wir uns nur noch um das Erhöhen der Benzinpreise kümmern. Dazu legen wir eine Variable an, in der wir den (fiktiven) Benzinpreis speichern. Hier gibt es eine Besonderheit: JavaScript kommt aus den USA, dort gibt es kein Dezimalkomma. Stattdessen wird ein Dezimalpunkt verwendet. Also müssen Sie auch bei Dezimalzahlen in der Programmierung einen Punkt schreiben.

```
var benzinpreis = 1.23;
```

Diesen Preis erhöhen oder erniedrigen wir dann. Dazu verwenden wir erneut das Gleichheitszeichen. Schauen Sie sich mal folgende Anweisung an:

```
benzinpreis = benzinpreis + 0.10;
```

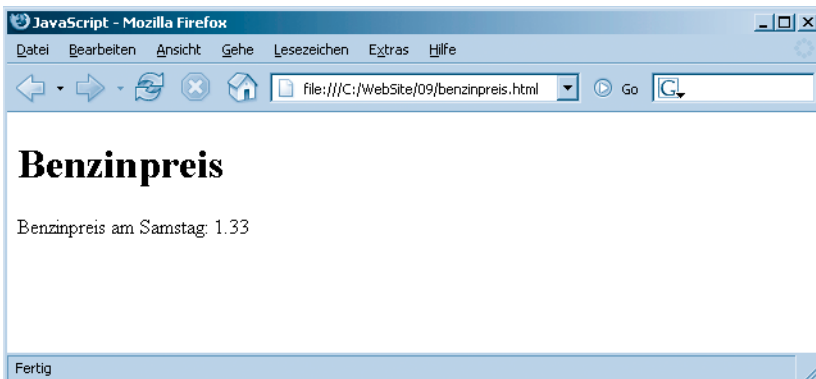


Lesen Sie diese Zeile von links nach rechts: Die Variable `benzinpreis` erhält einen Wert, der `benzinpreis + 0.10` beträgt. Auf gut Deutsch: Der Benzinpreis wird um 10 Cent erhöht.

Daraus lässt sich schon ein komplettes Listing zusammenstellen:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- saved from url=(0026)http://www.hauser-wenz.de/ -->
<!-- benzinpreis.html -->
<html>
<head>
 <title>JavaScript</title>
</head>
<body>
 <h1>Benzinpreis</h1>
 <script language="JavaScript"
 type="text/javascript"><!--
var benzinpreis = 1.23;
var tag = "Samstag";

if (tag == "Samstag") {
 benzinpreis = benzinpreis + 0.10;
} else if (tag == "Sonntag") {
 benzinpreis = benzinpreis + 0.05;
} else {
 benzinpreis = benzinpreis - 0.02;
}
document.write("Benzinpreis am " + tag +
 ": " + benzinpreis);
//--></script>
</body>
</html>
```



Probieren Sie es aus: Geben Sie in Zeile 13 der Variablen `tag` verschiedene Werte und beobachten Sie, wie sich der Preis jeweils verändert.

### Hinweis

*Wichtig: Wenn Sie innerhalb von `if` einen Vergleich anstellen (also überprüfen, ob eine Variable einen bestimmten Wert hat), müssen Sie ein doppeltes Gleichheitszeichen verwenden. Also so:*

- *Richtig:* `if (tag == "Samstag") { ... }`
- *Falsch:* `if (tag = "Samstag") { ... }`

Abschließend noch ein letzter Punkt: Wie Sie gesehen haben, gibt es in JavaScript einige Befehle, die Sie mit runden Klammern hinter dem Befehlsnamen aufrufen können. Sie haben bisher `alert()` und `document.write()` kennen gelernt. Man nennt diese Programmierkonstrukte **Funktionen** oder **Methoden**. Sie können solche Funktionen auch selbst schreiben. Wir wollen uns hier ansehen, wie wir eine Funktion schreiben können, die wir zum Erhöhen des Benzinpreises einsetzen können. Dazu befolgen Sie die folgenden Schritte:

**1** Fangen Sie mit dem Schlüsselwort `function` an.

```
function
```

**2** Geben Sie danach einen Namen ein, wie Sie auch einen Variablennamen formulieren würden.

```
function erhoeheBenzinpreis()
```

**3** Schreiben Sie jetzt innerhalb von geschweiften Klammern den Code, den die Funktion ausführen soll:

```
function erhoeheBenzinpreis () {
 benzinpreis = benzinpreis + 0.01;
}
```

Der nachfolgende Code verwendet erneut das Beispiel mit den Benzinpreisen. Diesmal gibt es drei Funktionen: `erhoeheBenzinpreis()`, `erhoeheBenzinpreisGanzStark()` und `senkeBenzinpreis()`, die dann nacheinander aufgerufen werden:



```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- saved from url=(0026)http://www.hauser-wenz.de/ -->
<!-- benzinpreis-funktionen.html -->
<html>
<head>
 <title>JavaScript</title>
</head>
<body>
 <h1>Benzinpreis</h1>
 <script language="JavaScript"
 type="text/javascript"><!--
function erhoeheBenzinpreis() {
 benzinpreis = benzinpreis + 0.01;
}
function erhoeheBenzinpreisGanzStark() {
 benzinpreis = benzinpreis + 0.05;
}
function senkeBenzinpreis() {
 benzinpreis = benzinpreis - 0.01;
}

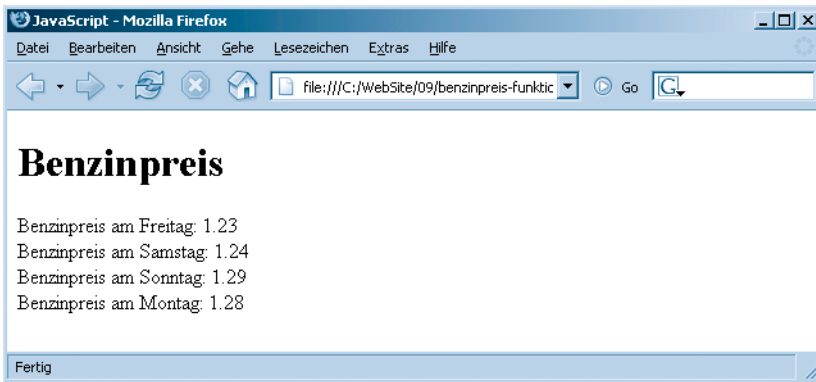
var benzinpreis = 1.23;
var tag = "Freitag";
document.write("Benzinpreis am " + tag +
 ": " + benzinpreis + "
");

var tag = "Samstag";
erhoeheBenzinpreis();
document.write("Benzinpreis am " + tag +
 ": " + benzinpreis + "
");

var tag = "Sonntag";
erhoeheBenzinpreisGanzStark();
document.write("Benzinpreis am " + tag +
 ": " + benzinpreis + "
");

var tag = "Montag";
senkeBenzinpreis();
document.write("Benzinpreis am " + tag +
 ": " + benzinpreis);
 //--></script>
</body>
</html>

```



So, genug selbst programmiert. Jetzt kennen Sie einige wesentliche Grundpfeiler von JavaScript. Im Folgenden finden Sie ein paar praktische Rezepte, die sich auf Ihrer Website möglicherweise ganz gut machen.

## Beispiel: Das aktuelle Datum ausgeben

Ein bereits klassischer Effekt besteht darin, das aktuelle Tagesdatum auf einer Website auszugeben. Mit ein wenig JavaScript kein Problem. Wie in diversen TV-Kochsendungen »haben wir mal etwas vorbereitet«. Im Downloadarchiv finden Sie eine entsprechend vorbereitete Datei *datum.js*. Diese bauen Sie wie folgt in Ihre HTML-Seite ein:

```
<script language="JavaScript"
 type="text/javascript"
 src="datum.js"></script>
```

Die Datei setzt mehrere Variablen:

- `datum_tag` für den aktuellen Tag
- `datum_monat` für den aktuellen Monat
- `datum_jahr` für das aktuelle Jahr
- `datum_stunde` für die aktuelle Stunde
- `datum_minute` für die aktuelle Minute
- `datum_sekunde` für die aktuelle Sekunde

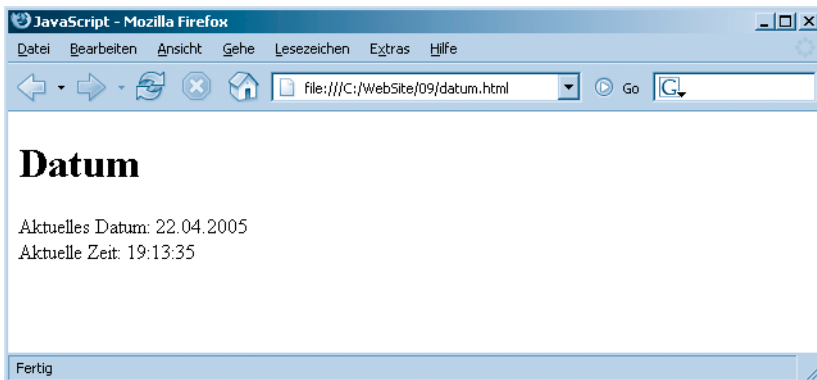


Sie können dann aus diesen Werten eine entsprechende JavaScript-Anweisung mit `document.write()` zusammensetzen, um das aktuelle Datum ausgeben zu können:

```
document.write("Aktuelles Datum: " +
 datum_tag + "." + datum_monat + "." + datum_jahr);
document.write("
Aktuelle Zeit: " +
 datum_stunde + ":" + datum_minute + ":" + datum_sekunde);
```

Hier ein komplettes Listing:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- saved from url=(0026)http://www.hauser-wenz.de/ -->
<!-- datum.html -->
<html>
<head>
 <title>JavaScript</title>
 <script language="JavaScript"
 type="text/javascript"
 src="datum.js"></script>
</head>
<body>
 <h1>Datum</h1>
 <script language="JavaScript"
 type="text/javascript"><!--
 document.write("Aktuelles Datum: " +
 datum_tag + "." + datum_monat + "." + datum_jahr);
 document.write("
Aktuelle Zeit: " +
 datum_stunde + ":" + datum_minute + ":" + datum_sekunde);
 //--></script>
</body>
</html>
```



### Tipp

*So können Sie zumindest so tun, als sei Ihre Website tagesaktuell:  
Verwenden Sie einfach Code nach folgendem Muster:*

```
document.write("Letzte Änderung: " +
 datum_tag + "." + datum_monat + "." + datum_jahr);
```

## Beispiel: Quicknagation

Ein netter Effekt, der sich auf zahlreichen Websites befindet, ist eine so genannte Quicknavigation. Mit HTML-Mitteln wird eine Auswahlliste erzeugt. Wenn ein Element in der Liste ausgewählt wird, lädt der Webbrowser automatisch die damit assoziierte Webadresse. Sehr praktisch – und funktioniert natürlich mit JavaScript.

Den Großteil der Arbeit haben wir bereits erledigt; im Downloadarchiv finden Sie die JavaScript-Datei *quicknavigation.js*. Sie müssen nur noch die folgenden simplen Schritte erledigen:

- 1 Erstellen Sie eine HTML-Aufzählungsliste.

```
<select>
</select>
```

- 2 Fügen Sie einige Elemente ein. Setzen Sie das `value`-Attribut auf den jeweils zu ladenden URL:

```
<select>
 <option value="">---Bitte wählen---</option>
 <option value="http://www.mut.de/">Markt + Technik
 </option>
 <option value="http://www.pearsoned.de/">Pearson
 Education</option>
 <option
 value="http://www.hauser-wenz.de/publikationen.php">
 H & W Publikationsliste</option>
</select>
```





**3** Aktivieren Sie die JavaScript-Datei, die Sie auf unserer Website heruntergeladen haben:

```
<script language="JavaScript"
 type="text/javascript"
 src="quicknavigation.js"></script>
```

**4** Rufen Sie den Code in der JavaScript-Datei auf, indem Sie die `onchange`-Eigenschaft der Auswahlliste wie folgt setzen:

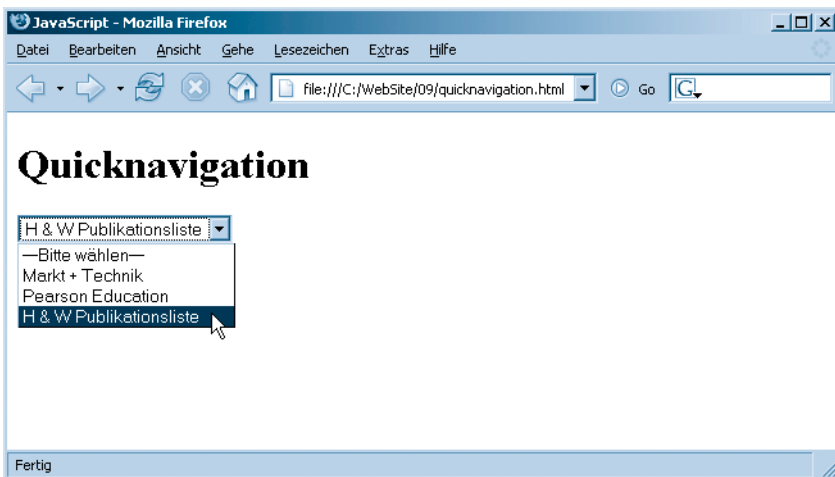
```
<select onchange="ladeSeite(this);">
```

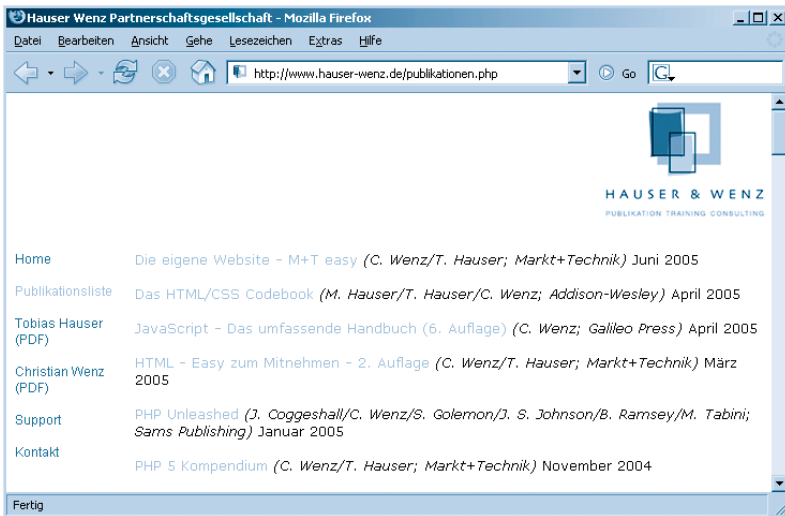
Das war es auch schon. Um den technischen Hintergrund zu erläutern: Beim Attribut `onchange` des `<select>`-Tags wird JavaScript genau dann aktiv, wenn sich die Auswahl der Liste ändert. In der JavaScript-Datei *quicknavigation.js* gibt es die Funktion `ladeSeite()`, die den entsprechenden URL lädt.

#### Hinweis

*Aus diesem Grund war auch der erste Eintrag in der Auswahlliste leer, denn das Laden der neuen Seite funktioniert nur bei Elementen, die extra ausgewählt werden müssen.*

Laden Sie die Seite im Webbrowser und wählen Sie einen Eintrag aus. Nach nur kurzer Zeit öffnet sich die gewünschte Webadresse.





Hier noch einmal der komplette Code:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- saved from url=(0026)http://www.hauser-wenz.de/ -->
<!-- quicknavigation.html -->
<html>
<head>
 <title>JavaScript</title>
 <script language="JavaScript"
 type="text/javascript"
 src="quicknavigation.js"></script>
</head>
<body>
 <h1>Quicknavigation</h1>
 <form>
 <select onchange="ladeSeite(this);">
 <option value="">---Bitte wählen---</option>
 <option value="http://www.mut.de/">Markt + Technik
 </option>
 <option value="http://www.pearsoned.de/">Pearson
 Education</option>
 <option
 value="http://www.hauser-wenz.de/publikationen.php">
 H & W Publikationsliste</option>
 </select>
 </form>
</body>
</html>
```



## Beispiel: Formularüberprüfung

Beim Erstellen eines HTML-Formulars gibt es viele Dinge zu beachten, wie bereits *Kapitel 6* gezeigt hat. Auch beim Versand gibt es einige potenzielle Gefahren. An dieser Stelle wollen wir einen besonders ärgerlichen Punkt behandeln: unvollständig ausgefüllte Formulare. Absichtlich falsch ausgefüllte Formulare sind ein Ärgernis, lassen sich aber kaum vermeiden – wie wollten Sie den Eintrag »Mein Name ist Donald Duck und ich wohne in Entenhäusen« abfangen. Bei unabsichtlichen Fehleingaben ist es etwas anderes: Hier wollen die Besucher die Daten angeben, haben aber etwas übersehen.

Hier kann JavaScript ins Spiel kommen. Damit ist es möglich, ein Formular automatisch überprüfen zu lassen. Der dabei zugrunde liegende Code ist relativ komplex, befindet sich aber vollständig im Downloadarchiv in der Datei *formularvalidierung.js*. Und so bauen Sie das Skript in Ihr HTML-Formular ein:

- 1 Laden Sie die externe JavaScript-Datei folgendermaßen (die JavaScript-Datei muss sich dabei im selben Verzeichnis befinden wie Ihr HTML-Dokument):

```
<script language="JavaScript"
 type="text/javascript"
 src="formularvalidierung.js"></script>
```

- 2 Rufen Sie den Code in der JavaScript-Datei auf, indem Sie die `onsubmit`-Eigenschaft des Formulars auf folgenden (fixen!) Wert setzen:

```
<form onsubmit="return formularvalidierung(this);">
```

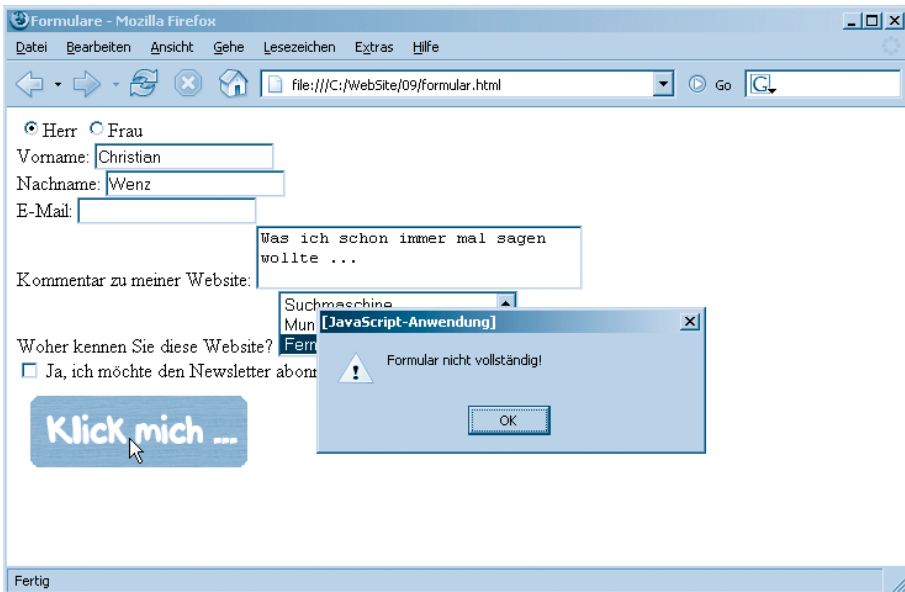
Das war es auch schon! Hier noch einmal eines der Formulare aus *Kapitel 6*, erweitert um die (automatische!) Formularvalidierung:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- saved from url=(0026)http://www.hauser-wenz.de/ -->
<!-- formular.html -->
<html>
<head>
 <title>Formulare</title>
 <script language="JavaScript"
 type="text/javascript"
 src="formularvalidierung.js"></script>
</head>
<body>
 <form action="..." method="post"
 onsubmit="return formularvalidierung(this);">
 <input type="radio" name="Anrede" value="Herr" />Herr
```

```



```



Beim Versuch, das Formular ohne weitere Angaben abzuschicken, erscheint eine Fehlermeldung. Der Benutzer wird hoffentlich dadurch dazu ermuntert, alle Formularfelder mit einem Wert zu bestücken.

### Tipp

*Wenn Sie die Fehlermeldung selbst anpassen möchten, müssen Sie die Datei **formularvalidierung.js** bearbeiten. Dort befindet sich in der sechsten Zeile von unten folgende Anweisung:*

```
alert("Formular nicht vollständig!");
```

*Die alert()-Anweisung kennen Sie ja bereits. Ändern Sie den Text in den runden Klammern und speichern Sie das Ergebnis; schon erscheint eine andere Fehlermeldung.*

Nun ist es allerdings so, dass nicht alle Felder unbedingt Pflichtfelder sein müssen. Nehmen wir beispielsweise das Formular. Die E-Mail-Adresse benötigen wir nicht unbedingt, und es ist auch arg unhöflich, die Benutzer dazu zu nötigen, den Newsletter zu abonnieren. Für diesen Fall ist in der Datei *formularvalidierung.js* bereits eine entsprechende Erweiterung integriert: Alle Formularfelder, deren Name (name-Attribut) auf `_optional` endet, werden von der Überprüfung ausgenommen. Nachfolgend das angepasste Listing für das Formular, bei dem das E-Mail-Feld und die Checkbox für den Newsletter beide optional sind. Die Abbildung zeigt, dass dann der Formularversand ohne Fehlermeldung funktioniert.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- saved from url=(0026)http://www.hauser-wenz.de/ -->
<!-- formular-optional.html -->
<html>
<head>
 <title>Formulare</title>
 <script language="JavaScript"
 type="text/javascript"
 src="formularvalidierung.js"></script>
</head>
<body>
 <form action="..." method="post"
 onsubmit="return formularvalidierung(this);">
 <input type="radio" name="Anrede" value="Herr" />Herr
 <input type="radio" name="Anrede" value="Frau" />Frau

 Vorname: <input type="text" name="Vorname" />

```

```

Nachname: <input type="text" name="Nachname" />

E-Mail: <input type="text" name="Mail_optional" />

Kommentar zu meiner Website:
<textarea name="Kommentar" cols="30" rows="2">
 Was ich schon immer mal sagen wollte ...
</textarea>

Woher kennen Sie diese Website?
<select name="Quelle" size="3">
 <option value="such">Suchmaschine</option>
 <option value="mund">Mund-zu-Mund-Propaganda</option>
 <option value="tv">Fernsehwerbung</option>
 <option value="sonst">Sonstiges</option>
</select>

<input type="checkbox" name="Newsletter_optional"
 value="ja" />
Ja, ich möchte den Newsletter abonnieren!

<input type="image"
 src="grafik.gif"
 name="Versendegrafik"
 style="border:0px;"
 alt="Formular versenden" />

</form>
</body>
</html>

```

The screenshot shows a Mozilla Firefox browser window with the title 'Formulare - Mozilla Firefox'. The address bar shows the file path 'file:///C:/WebSite/09/formular-optional.html'. The form is displayed with the following elements:

- Radio buttons for 'Herr' (selected) and 'Frau'.
- Text input for 'Vorname' with the value 'Christian'.
- Text input for 'Nachname' with the value 'Wenz'.
- Text input for 'E-Mail'.
- Text area for 'Kommentar zu meiner Website' with the text 'Was ich schon immer mal sagen wollte ...'.
- Dropdown menu for 'Woher kennen Sie diese Website?' with options 'Suchmaschine', 'Mund-zu-Mund-Propaganda', and 'Fernsehwerbung' (selected).
- Checkbox for 'Ja, ich möchte den Newsletter abonnieren!'.
- A button labeled 'Klick mich ...' with a mouse cursor pointing at it.

The status bar at the bottom of the browser window shows 'Fertig'.



**Hinweis**

*Um das Formular zu vervollständigen, müssen Sie natürlich das Versandziel des Formulars ändern (action-Attribut). Wie das geht, erfahren Sie in Kapitel 6.*

## Beispiel: Mehrere Frames auf einmal ändern

Eine der am häufigsten gestellten Fragen zu Frames lautet: Wie kann ich den Inhalt mehrerer Frames auf einmal ändern. Die Antwort: Mit HTML alleine ist das leider nicht so einfach möglich. Mit JavaScript allerdings ist es nicht sonderlich schwierig. Die JavaScript-Datei *framesladen.js* im Downloadarchiv enthält die wesentlichen Infos dazu, Sie erledigen nur einige simple Restarbeiten:

**1** Binden Sie die externe JavaScript-Datei für das Laden mehrerer Frames auf einmal ein:

```
<script language="JavaScript"
 type="text/javascript"
 src="framesladen.js"></script>
```

**2** Setzen Sie, immer wenn Sie zwei Frames auf einmal ändern sollen, einen Link der folgenden Machart in Ihr HTML-Dokument:

```
<a href="javascript:framesLaden('Name1', 'url1.html',
 'Name2', 'url2.html');">Hier klicken!
```

Der mit *javascript:* beginnende URL ruft eine JavaScript-Funktion auf, die sich in der Datei *framesladen.js* befindet. Als Parameter übergeben Sie jeweils abwechselnd den Frame-Namen (name-Attribut) und einen URL. Obiges Beispiel würde also das Dokument *url1.html* in den Frame Name1 laden und *url2.html* in den Frame Name2.

**Hinweis**

*Achten Sie dabei unbedingt auf die einfachen Apostrophe rund um die Frame-Namen und -URLs. Außerdem darf der Link kein target-Attribut enthalten.*

Nachfolgend ein kleines Beispiel, das auf dem Frames-Beispiel in *Kapitel 5* basiert. Die Datei *navigation.html* wird ein wenig modifiziert: Wenn ein Navigationspunkt ausgewählt wird, lädt der Webbrowser wie bisher in den rechten Frame die gewünschte Inhaltsseite, aber in den linken Frame zusätzlich die Startseite.

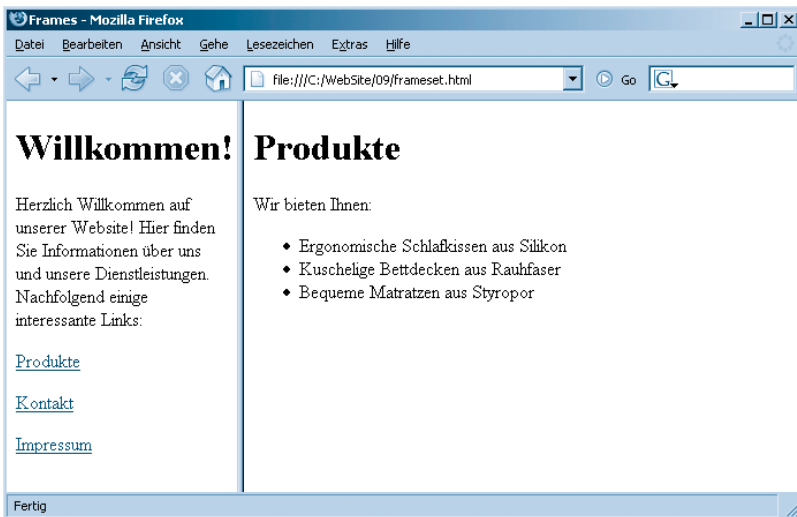
Ob das sinnvoll ist, darüber mag man streiten, aber für das Beispiel ist es eine prima Illustration. Hier die angepasste Datei *navigation.html* – alle anderen Frames-Dateien aus *Kapitel 5* bleiben gleich (und sind auch im Code-Archiv für dieses Kapitel):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- navigation.html -->
<html>
<head>
 <title>Navigation</title>
 <script language="JavaScript"
 type="text/javascript"
 src="framesladen.js"></script>
</head>
<body>
 <h1>Navigation</h1>
 <p>
 <a href="javascript:framesLaden('Navi', 'start.html',
 'Inhalt', 'produkte.html');">Produkte
 </p>
 <p>
 <a href="javascript:framesLaden('Navi', 'start.html',
 'Inhalt', 'kontakt.html');">Kontakt
 </p>
 <p>
 <a href="javascript:framesLaden('Navi', 'start.html',
 'Inhalt', 'impressum.html');">Impressum
 </p>
</body>
</html>
```

Werfen Sie einen Blick auf die beiden Abbildungen: Vor dem Klick auf den Link ist im linken Frame das Navigationsdokument, danach die Startseite.







### Hinweis

Beim Testen im Internet Explorer (Windows XP, Service Pack 2) erhalten Sie die am Anfang des Kapitels gezeigte Warnmeldung, wenn Sie das Beispiel lokal von Ihrer Festplatte starten. Aufgrund der Verlinkung in dem Code lässt sich das leider nur schwer vermeiden. Im Internet tritt diese Fehlermeldung dann jedoch nicht mehr auf.

## Beispiel: Frame nachladen

Ein weiteres Problem mit Frames kann mit etwas JavaScript ein wenig gelindert werden. Frames funktionieren nämlich so lange gut, so lange der Benutzer nicht versucht, ein Dokument, das sich innerhalb eines Frames befinden, alleine (also ohne Frames) in einem Browserfenster zu laden. Dann fehlt nämlich in vielen Fällen die komplette Navigation. Das geschieht häufig in Zusammenhang mit Suchmaschinen.

Doch mit JavaScript lässt sich auch dies beheben. Ein klein wenig Code führt dazu, dass das Frameset nachgeladen wird, wenn ein Dokument alleine aufgerufen worden ist.

Das ganze Vorgehen besteht aus zwei Teilen:

- In jedem Einzeldokument müssen Sie dafür sorgen, dass geprüft wird, ob ein Frameset vorhanden ist (gut!) oder nicht (schlecht!).
- Im Frameset-Dokument müssen Sie dafür sorgen, dass gegebenenfalls ein Frame-Dokument »nachgeladen wird«.

Das technische Vorgehen ist das folgende: Wenn ein HTML-Dokument außerhalb der Frame-Struktur (also ohne Frames) im Browser aufgerufen wird, leitet ein kleines Stückchen JavaScript den Benutzer weiter zur Seite *frameset-auto.html?seite.html*. In dieser Seite (*frameset-auto.html*) wird abgefragt, was nach dem Fragezeichen kommt, in diesem Fall: das Dokument *seite.html*. Genau dieses Dokument wird nun per JavaScript in den Inhaltsframe geladen.

Sie selbst können das in drei einfachen Schritten umsetzen; der Großteil der Arbeit ist bereits erledigt und die Datei befindet sich im Downloadarchiv zu diesem Buch:

**1** Laden Sie in allen Einzeldokumenten die externe Überprüfungsdatei *framescheck.js*:

```
<script language="JavaScript"
 type="text/javascript"
 src="framescheck.js"></script>
```

**2** In das Dokument *frameset-auto.html*, das das Frameset enthält, müssen Sie die JavaScript-Datei *framesnachladen.js* einbinden:

```
<script language="JavaScript"
 type="text/javascript"
 src="framesnachladen.js"></script>
```



Das war es auch schon – wirklich! Im Downloadarchiv für dieses Kapitel sind die entsprechenden Einzeldateien *start.html*, *produkte.html*, *kontakt.html* und *impressum.html* dementsprechend angepasst; hier exemplarisch die Datei *kontakt.html*:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- kontakt.html -->
<html>
<head>
 <title>Kontakt</title>
 <script language="JavaScript"
 type="text/javascript"
 src="framescheck.js"></script>
</head>
<body>
 <h1>Kontakt</h1>
 <p>
 Sie erreichen uns per E-Mail unter

 info@schlafshop.xy

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

Die Datei *frameset-auto.html* ist mit der Datei *frameset.html* identisch, mit einem Hauptunterschied: Die Datei *framesnachladen.js* wird per `<script>`-Element aktiviert:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<!-- frameset-auto.html -->
<html>
<head>
 <title>Frames</title>
 <script language="JavaScript"
 type="text/javascript"
 src="framesnachladen.js"></script>
</head>
<frameset cols="200,*">
 <frame src="navigation.html" name="Navi" />
 <frame src="start.html" name="Inhalt" />
</frameset>
</html>
```

Wenn Sie jetzt beispielsweise die Datei *kontakt.html* laden, leitet Sie der Webbrowser automatisch zur Datei *frameset-auto.html* weiter; dort wird dann die Kontaktseite in den Inhaltsframe geladen.



### Tipp

Auch dieses Beispiel ist anpassbar:

- Wenn Ihr Inhaltsframe nicht "Inhalt" heißt, passen Sie die Datei **framesnachladen.js** an und ersetzen dort in der folgenden Zeile Inhalt durch den gewünschten Frame-Namen.

```
window.frames["Inhalt"].location.href =
```

- Wenn Ihre Frameset-Seite nicht *frameset-auto.html* heißt (sondern beispielsweise **index.html**), passen Sie die Datei **framescheck.js** an und ersetzen dort in der folgenden Zeile **frameset-auto.html** durch den gewünschten Dateinamen (entfernen Sie aber nicht das Fragezeichen):

```
location.href = "frameset-auto.html?" +
```

### Hinweis

Auch bei diesem Beispiel gibt der Microsoft Internet Explorer unter Windows XP mit Service Pack 2 (oder höher) eine Warnmeldung aus, wie eingangs gezeigt. Im Internet ist das dann jedoch nicht mehr der Fall.



## Was nicht funktioniert

Im Zusammenhang mit JavaScript werden häufig immer dieselben Fragen gestellt, beispielsweise:

Kann ich mit JavaScript ...

- ... drucken, ohne dass der Benutzer das bestätigen muss?
- ... Dateien auf dem Rechner des Benutzers abspeichern?
- ... Daten vom Rechner des Benutzers einlesen?
- ... einen sicheren Passwortschutz für eine Website implementieren?
- ... zuverlässig meinen Quellcode vor neugierigen Augen schützen?
- ... auf Datenbanken zugreifen?
- ... Mails versenden?
- ... ein Gästebuch einrichten?
- ... ein Forum aufstellen?

Die Antwort auf all diese Fragen: **nein**.

JavaScript ist sehr mächtig, hat allerdings auch seine Grenzen. Und einer Sache sollten Sie sich stets bewusst sein: JavaScript kann sehr einfach in Webbrowsern ausgeschaltet werden; in manchen Firmen ist das sogar vorgeschrieben. Sie sollten also immer damit rechnen, dass Ihre Besucher kein JavaScript unterstützen. Deswegen sollten JavaScript-Effekte nur eine optionale Ergänzung Ihrer Website sein, nicht jedoch deren Grundlage. Ansonsten sehen sich Besucher schnell woanders um.

### Tipp

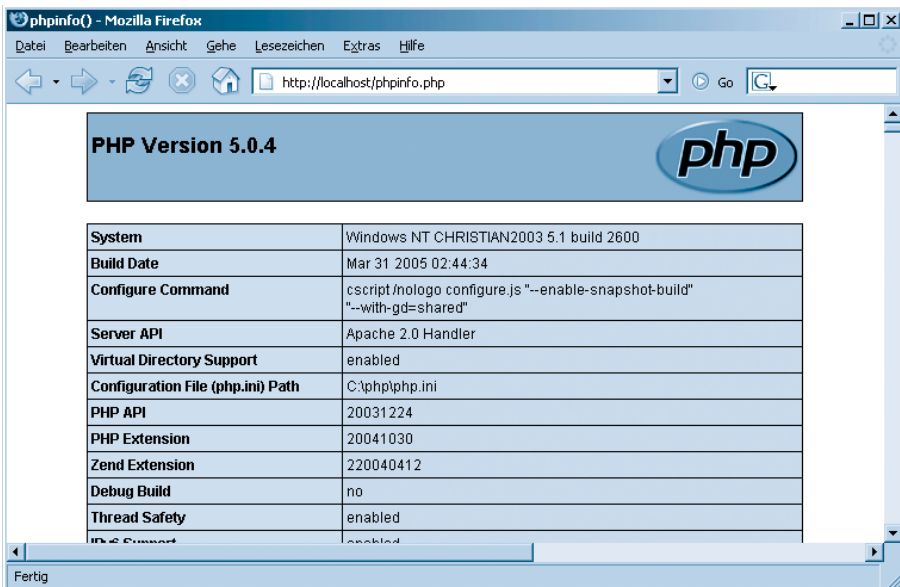
Weitere häufig gestellte (und auch beantwortete) Fragen zu JavaScript finden Sie unter <http://dcljs.de/>. Auch können wir Ihnen JavaScript-Titel aus eigener Feder ans Herz legen: Unter <http://www.hauser-wenz.de/publikationen.php> finden Sie eine Liste.

## Nutzloses Wissen

Die meisten Toilettenspülungen »spielen« den Ton E.

## Kapitel 10

# Dynamische Websites mit PHP



Wie Sie im letzten Kapitel gesehen haben, ermöglicht JavaScript viele Dinge, die mit HTML alleine nie zu schaffen gewesen wären. Jetzt gehen wir noch einen Schritt weiter und zeigen Ihnen einen Shooting Star im Webseitenbereich – PHP. Damit können Sie Ihre eigenen Gästebücher, Weblogs oder Foren erstellen – oder Systeme anderer einfach einsetzen.

**Das können Sie schon:**

|                                                |     |
|------------------------------------------------|-----|
| Formatierung von Text                          | 56  |
| Einsatz von Links                              | 75  |
| Aufzählungslisten                              | 88  |
| Grafiken einbauen                              | 116 |
| Frames erstellen                               | 134 |
| Formulare verwenden                            | 158 |
| Mit Tabellen und CSS positionieren             | 194 |
| Gästebuch, Diskussionsforum und Weblog anlegen | 212 |
| Einstieg und Verwendung von JavaScript         | 238 |

**Das lernen Sie neu:**

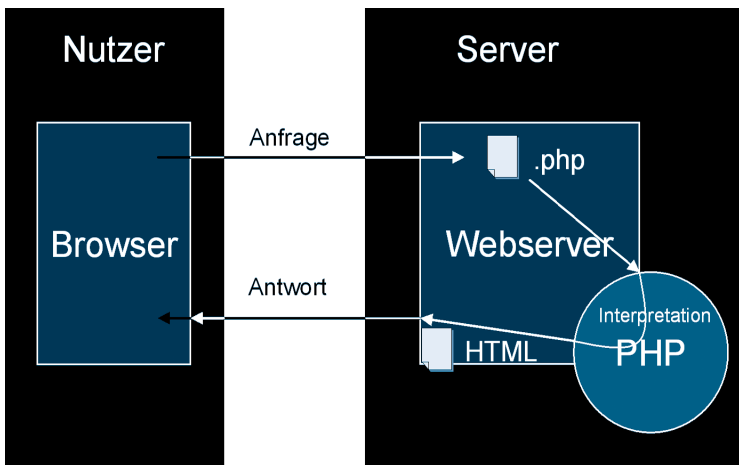
|                                |     |
|--------------------------------|-----|
| PHP kennen lernen              | 276 |
| PHP installieren               | 278 |
| Datum mit PHP                  | 295 |
| Formulare per E-Mail versenden | 298 |

## Was ist PHP?

Während es sich bei HTML, CSS und JavaScript um so genannte clientseitige Technologien handelt, ist PHP etwas ganz anderes, nämlich eine serverseitige Technologie.

Erinnern Sie sich an das Client-Server-Modell aus dem Einleitungskapitel? Noch einmal kurz wiederholt: Es gibt einen Client, das ist meistens der Webbrowser, also das lokale System des Benutzers. Und es gibt den Webserver, das ist ein Rechner, der irgendwo auf der Welt steht und Webseiten ausliefert. HTML, CSS und JavaScript werden vom Server an den Client geschickt und dort **interpretiert**, also ausgeführt. Der Browser muss sich also darum kümmern, dass die Darstellung des Ganzen korrekt ist; damit hat der Server nichts zu tun.

PHP dagegen läuft komplett auf dem Server. Ein PHP-Programm, oder auch ein PHP-Skript, wird auf dem Webserver ausgeführt. Das Ergebnis dieses Programms ist in der Regel HTML-Markup – das dann wiederum an den Client geschickt und dort angezeigt wird. Die Übersichtsgrafik zeigt das Ganze noch einmal im Zusammenhang.



Da am Ende HTML dabei herauskommt, wieso also eine weitere, dieses Mal serverseitige Technologie? Nun, mit PHP können Sie Dinge tun, die mit HTML, CSS und JavaScript nur mit großen Einschränkungen oder gar nicht möglich sind. Dazu gehört unter anderem Folgendes:

- das Versenden einer E-Mail
- die Verwendung von Datenbanken





- das dynamische Erzeugen von Grafiken nach Bedarf
- die Abspeicherung von Nutzerdaten
- der Zugriff auf Dateien auf dem Server
- und vieles mehr

Sie sehen also: Die Verwendung von PHP lohnt sich auf jeden Fall. Allerdings gibt es das nur zu einem bestimmten Preis.

Keine Sorge, PHP ist kostenlos, aber doch ein Stückchen aufwändiger und komplizierter als etwa HTML und CSS und ebenfalls komplexer als JavaScript. Sowohl die Anforderungen an Sie sind etwas höher als auch die Anforderungen an Ihren Webhoster – also den Anbieter, bei dem Sie Ihre Internetsite lagern werden (mehr dazu in *Kapitel 11*).

Doch keine Sorge, das kriegen Sie problemlos hin. Alles, was Sie zunächst brauchen, ist ein lokales Testsystem. Denken Sie noch einmal ans Client-Server-Modell: Wenn PHP eine serverseitige Skriptsprache ist, benötigen Sie einen Webserver, auf dem dann PHP läuft. Nun ist es ein wenig teuer und aufwändig, wenn Sie den ganzen Tag online sein müssten, um auf einem Webserver bei Ihrem Hoster Ihre PHP-Seiten zu testen. Deswegen installieren Sie einfach auf Ihrem »normalen« Rechner einen Webserver und PHP.

PHP gibt es auf der PHP-Homepage unter <http://www.php.net/>, die auch in der Abbildung zu sehen ist. Als Webserver eignet sich unter anderem der Apache, der mit etwa zwei Dritteln Marktanteil der Weltmarktführer ist – und ebenfalls kostenlos zur Verfügung steht, unter <http://httpd.apache.org/>.

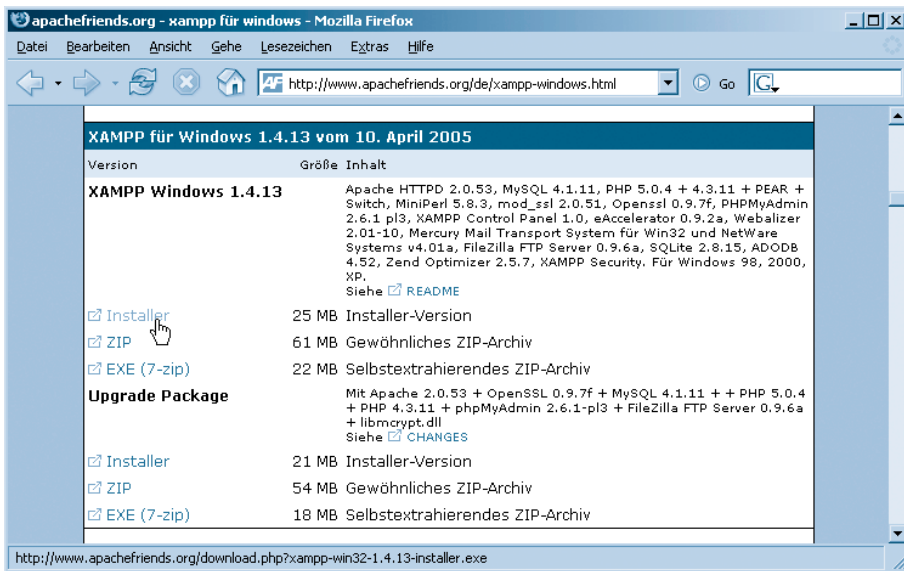


# Installation

Die Installation des Ganzen kann recht kompliziert sein, doch es gibt einige tolle Softwarepakete, die Ihnen die Arbeit fast komplett abnehmen. Das Team von *apachefriends.org* hat es sich zur Aufgabe gemacht, Apache, PHP und zahlreiche weitere nützliche Tools in einem einzelnen, kompakten Paket anzubieten und die Installation damit stark zu erleichtern. Das Paket heißt XAMPP und gibt es für mehrere Betriebssysteme, wobei vor allem die Varianten für Windows und Linux sehr gut sind. Wir werden XAMPP deswegen für diese beiden Betriebssysteme vorstellen. Auch für Mac existiert eine Beta-Version, doch ist auf dem Apple-Betriebssystem (ab Version X) beispielsweise der Apache-Webserver eigentlich schon mit dabei.

Zunächst werfen wir einen Blick auf die Windows-Version. Diese installiert sich weitestgehend automatisch. Wir gehen diese Installation jedoch Schritt für Schritt durch und erklären jeweils, was da im Hintergrund passiert, damit Sie immer im Bilde sind.

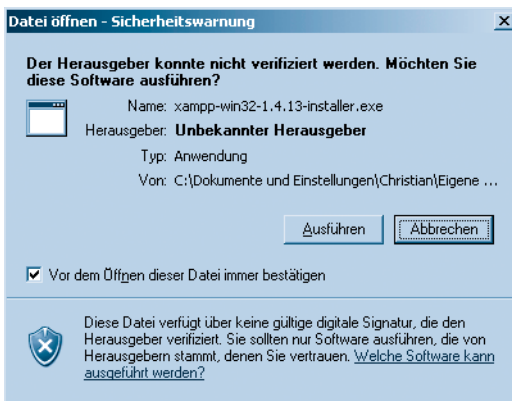
**1** Laden Sie von <http://www.apachefriends.org/de/xampp-windows.html> das XAMPP-Paket herunter.



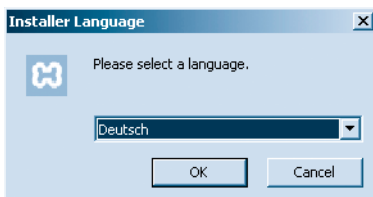
Wählen Sie dazu unter *XAMPP Windows x.y.z* (die zur Drucklegung aktuelle Versionsnummer war 1.4.13, aber möglicherweise steht für Sie bereits eine neuere Version zur Verfügung) den Punkt *Installer*, der etwa 25 Mbyte groß ist. Es gibt auch noch andere Pakete, aber die sind entweder deutlich größer oder nicht so bequem in der Handhabung.

## 2 Starten Sie die heruntergeladene Datei.

Unter Umständen (sprich Windows XP SP2) erscheint eine Warnmeldung, der Herausgeber der Datei könne nicht identifiziert werden. Hintergrund: Solch eine Herausgeber-Signatur kostet recht viel Geld, die das *apachefriends.org*-Projekt nicht aufbringen kann und auch nicht aufbringen will. Wenn Sie die Datei von der offiziellen Downloadseite bezogen haben, können Sie die Warnmeldung ignorieren.

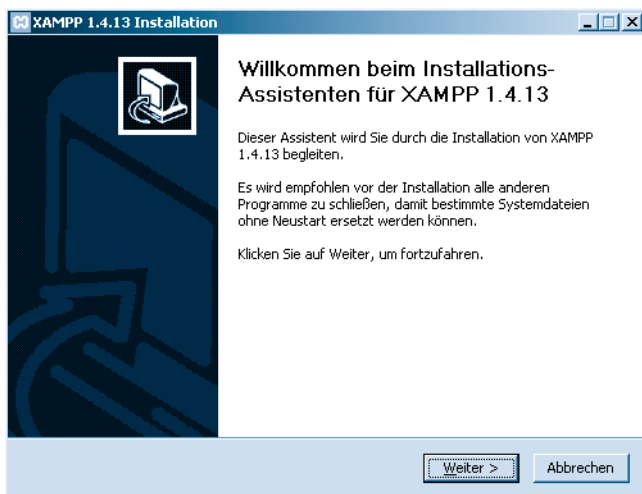


## 3 Wählen Sie eine Sprache für die Installation, am besten *Deutsch*.



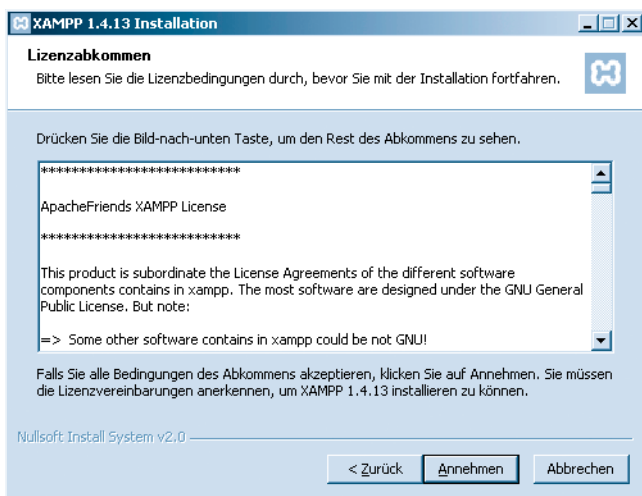
Das Installationsprogramm verwendet dann die gewünschte Sprache.

## 4 Klicken Sie auf *Weiter* – es kann losgehen!



## 5 Lesen und bestätigen Sie die Lizenzvereinbarungen.

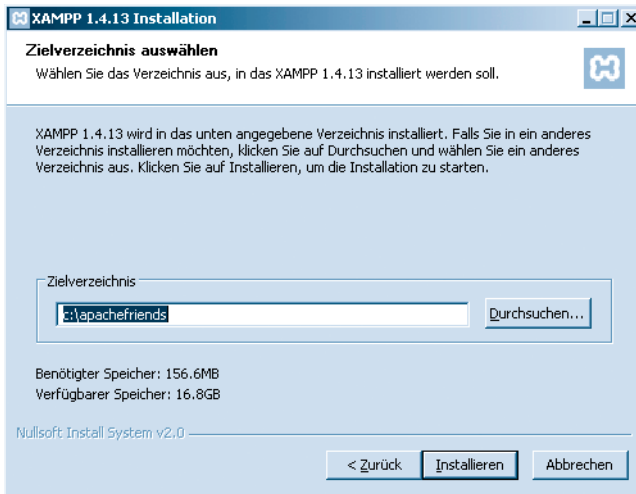
Keine Sorge, da steht nichts Schlimmes drin. Das Lesen lohnt sich auf jeden Fall. Und wenn Sie nicht auf *Annehmen* klicken, lässt sich die Software nicht installieren.



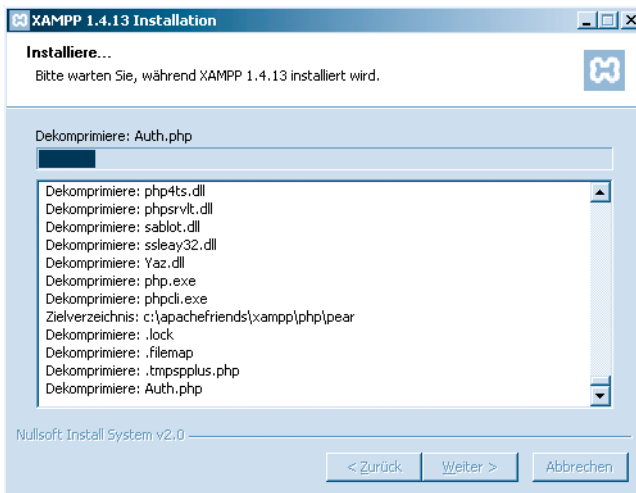
## 6 Wählen Sie ein Installationsverzeichnis und klicken Sie auf *Installieren*.



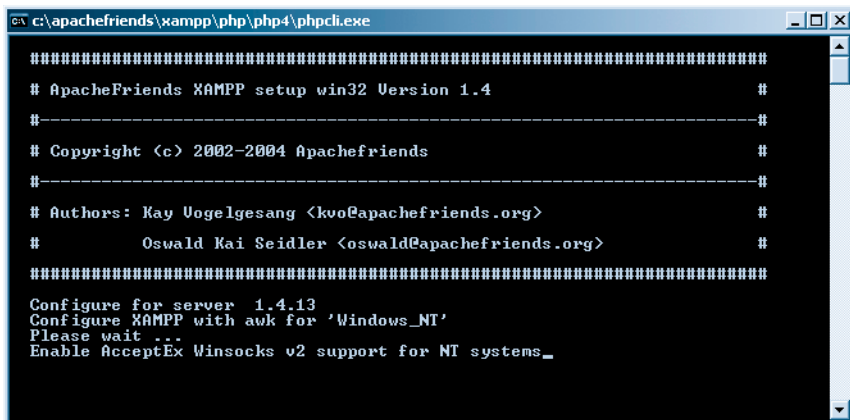
Sie benötigen ungefähr 150 Mbyte freien Speicherplatz. Die Standardvorgabe, *C:\apachefriends*, ist meist eine ganz gute Wahl. Sie können auch ein anderes Verzeichnis angeben. Merken Sie sich aber unbedingt, wo Sie letztendlich das Ganze hinstalliert haben!



XAMPP installiert jetzt alle Dateien. Sie können dabei zuschauen und sehen eine riesige Anzahl von Dateien, die auf die Festplatte kopiert werden. Wenn Sie genau hinsehen, entdecken Sie auch immer wieder mal, dass PHP in den Dateinamen vorkommt – ein gutes Zeichen.



Am Ende des Kopiervorgangs öffnet sich ein schwarzes Fenster, eine DOS-Eingabeaufforderung. Nicht erschrecken, da erledigt XAMPP noch abschließende Konfigurations- und Installationsarbeiten.



```
c:\apache\friends\xampp\php\phpcli.exe

#####
ApacheFriends XAMPP setup win32 Version 1.4
#-----# #
Copyright (c) 2002-2004 ApacheFriends
#-----# #
Authors: Kay Vogels <kvo@apachefriends.org>
Oswald Kai Seidler <oswald@apachefriends.org>
#####
Configure for server 1.4.13
Configure XAMPP with awk for 'Windows_NT'
Please wait ...
Enable AcceptEx Winsocks v2 support for NT systems_
```

**7** Klicken Sie auf *Fertig stellen*.

Herzliche Gratulation, die Installation hat geklappt!



**8** Wählen Sie im Windows-Startmenü den Eintrag *Start/apachefriends/xampp/xampp control panel*.

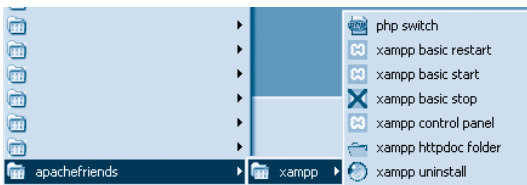


In der Programmgruppe *xampp* befinden sich einige praktische Einträge, unter anderem die folgenden:

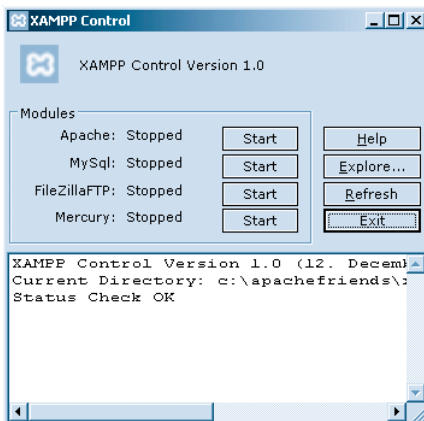
- *xampp basic start*: startet den Webserver.
- *xampp basic restart*: startet den Webserver neu.
- *xampp httpdoc folder*: öffnet den Ordner, in dem HTML-, CSS- und PHP-Dateien des Webserver liegen.
- *xampp control panel*: startet ein spezielles Konfigurationsprogramm für XAMPP.

### Tipp

Das XAMPP Control Panel können Sie auch über eine Verknüpfung auf dem Desktop starten.

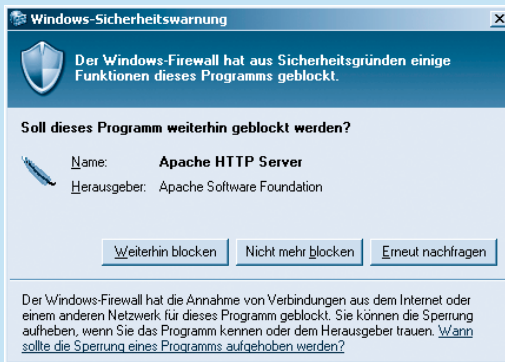


9 Klicken Sie im Fenster *XAMPP Control* in der Zeile, in der *Apache* steht, auf *Start*. Der Apache-Webserver wird gestartet. Ganz analog können Sie unter anderem auch MySQL starten, das ist eine sehr mächtige Datenbank, sowie andere Module von XAMPP. Zunächst aber reicht Apache vollkommen aus, um PHP-Skripte auszuführen.



## Tipp

Wenn Sie Windows XP einsetzen und dort – was sehr zu empfehlen ist – mindestens Service Pack 2 installiert haben, wird unter Umständen die integrierte Firewall nervös. Sie merkt, dass auf einmal versucht wird, einen Webserver zu starten. Das könnte ein Anzeichen für einen Virus sein, deswegen gibt die Firewall eine Warnmeldung aus. Es liegt aber in diesem Fall kein Problem vor, sondern es wird lediglich der Apache gestartet. Sie können also die Warnmeldung ruhig mit »**Nicht mehr blocken**« ausschalten. Klicken Sie nur nicht auf »**Weiterhin blocken**«, denn dann wird der Apache-Webserver nicht funktionieren, sondern von Windows an seiner Arbeit gehindert.



**10** Starten Sie Ihren Webbrowser, indem Sie in Ihrem Browser die Adresse `http://localhost/` aufrufen.

Jetzt wird es spannend: Es wird eine echte Webadresse aufgerufen. `localhost` ist dabei der Servername des lokalen Servers – also des Apache, den Sie gerade eben installiert haben. Und das ist ein sehr wichtiger Punkt: Immer wenn Sie jetzt mit PHP arbeiten, müssen Sie das PHP-Skript über `http://localhost/...` aufrufen, denn nur dann wird PHP über den Webserver ausgeführt. Ein Aufrufen über das Dateisystem – wie Sie es bei Ihren HTML-Dokumenten gemacht haben – funktioniert nicht, denn dann würde nur die Datei mit dem PHP-Code angezeigt, dieser aber nicht ausgeführt werden.

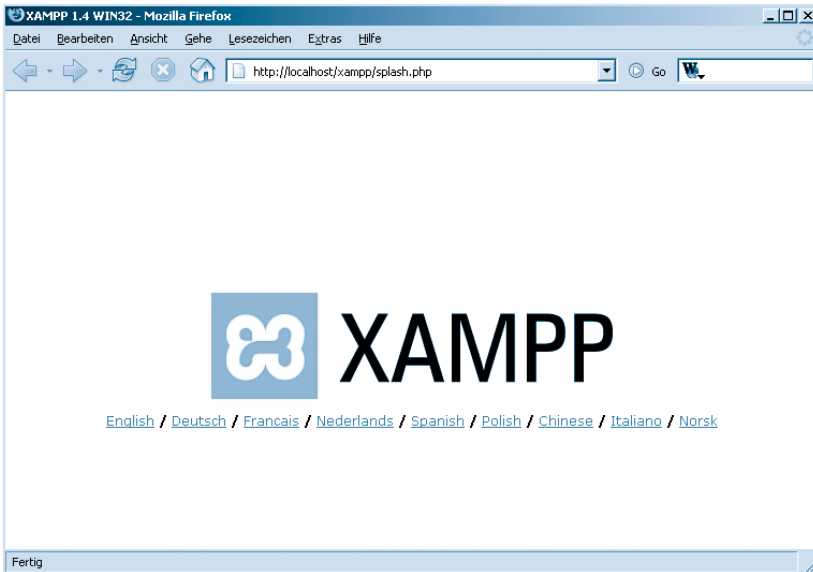
## Tipp

Alternativ können Sie den lokalen Server auch über **`http://127.0.0.1/`** aufrufen.





Es erscheint die Startseite der XAMPP-Installation, es hat also alles wunderbar funktioniert.



**11** Klicken Sie dort auf den Link *Deutsch*.

Es erscheint die Willkommenseite – es hat also tatsächlich alles geklappt.



## 12 Klicken Sie in der linken Navigationsleiste auf *Status*.

Sie erhalten eine Übersicht darüber, welche Komponenten von XAMPP alle installiert sind. So, wie wir XAMPP installiert und gestartet haben, laufen nicht alle Teile, beispielsweise kein MySQL und kein FTP-Server. Die wichtigsten Komponenten sind jedoch alle aktiv, allen voran PHP.

| Komponente                     | Status      | Hinweis |
|--------------------------------|-------------|---------|
| MySQL-Datenbank                | DEAKTIVIERT |         |
| PHP                            | AKTIVIERT   |         |
| HTTPS (SSL)                    | AKTIVIERT   |         |
| Common Gateway Interface (CGI) | AKTIVIERT   |         |
| Server Side Includes (SSI)     | AKTIVIERT   |         |
| SMTP Server                    | DEAKTIVIERT |         |
| FTP Server                     | DEAKTIVIERT |         |

Dieser Check funktioniert nur zuverlässig solange nichts an der Konfiguration des Apache geändert wurde. Durch bestimmte Änderungen kann das Ergebnis dieses Tests verfälscht werden. Mit SSL (https://localhost) funktionieren die Statuschecks nicht!

Wenn Sie dagegen Linux einsetzen sollten, geht die Installation ein klein wenig schneller, ist aber nicht ganz so intuitiv wie unter Windows.

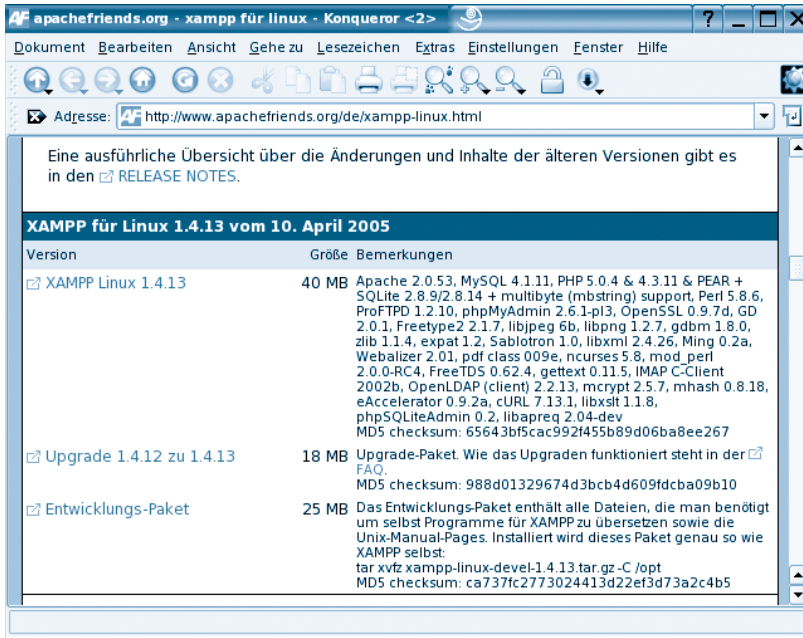
Zunächst einmal sollten Sie nachsehen, ob PHP und Apache nicht vielleicht bei Ihrer Linux-Distribution mit dabei sind. Starten Sie also die Paketverwaltung, beispielsweise YAST unter SuSE Linux, und suchen Sie dort nach Paketnamen wie *php* und *apache*. Falls Sie fündig werden, kontrollieren Sie, ob die Pakete bereits installiert sind. Falls nein, könnten Sie den Webserver auf diese Art und Weise installieren. Es gibt aber einen bequemer Weg, den Sie auf jeden Fall einschlagen sollten, wenn PHP und Apache noch nicht auf Ihrem System sind.



Dazu befolgen Sie diese Schritte:

**1** Laden Sie XAMPP für Linux unter <http://www.apachefriends.org/de/xampp-linux.html> herunter.

Sie benötigen das Paket *XAMPP Linux x.y.z*, das ungefähr 40 Mbyte groß ist.



**2** Starten Sie unter Linux die Konsole.

Dieses Tool befindet sich meist in einem Menü, das *Systemkomponenten* oder ähnlich heißt.

**3** Rufen Sie das Programm `su` auf und geben Sie bei der Aufforderung Ihr Root-Passwort ein.

Damit melden Sie sich als Super-User, der Nutzer mit den höchsten Rechten, auf dem System an. Denn nur dieser Nutzer hat das Recht, im System einen Webserver zu installieren.

**4** Entpacken Sie XAMPP mit folgendem Kommando:

```
tar xvfz xampp-linux-x.y.z.tar.gz -C /opt
```

Dabei ersetzen Sie natürlich *x.y.z* durch die Versionsnummer der heruntergeladenen XAMPP-Version, etwa 1.4.13.

Das System beginnt jetzt, XAMPP zu entpacken, und installiert so zahlreiche Dateien auf den Rechner. Sie benötigen ein klein wenig Geduld, doch es gibt auch eine gute Nachricht: Danach ist die Installation schon geschafft, XAMPP befindet sich auf dem System!

**5** Starten Sie XAMPP mit folgendem Kommando:

```
/opt/lampp/lampp start
```

XAMPP startet nacheinander den Apache-Webserver, die MySQL-Datenbank sowie einen FTP-Server.

#### Hinweis

*Wundern Sie sich nicht, dass das Programm jetzt `lampp` heißt und nicht `xampp`. Das ist eine Art historisches Relikt, denn früher hieß das Ganze tatsächlich LAMPP, mit L wie Linux (und das Windows-Paket hieß dementsprechend WAMPP – klang wohl zu sehr nach Wampe).*



```

Befehlsfenster - Konsole
Sitzung Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe

lampp/phpsqliteadmin/index.php
lampp/phpsqliteadmin/table_browse.php
lampp/phpsqliteadmin/mainframe.php
lampp/phpsqliteadmin/create_system_db.sql
lampp/phpsqliteadmin/editdb.php
lampp/phpsqliteadmin/table_create.php
lampp/phpsqliteadmin/dbaction.php
lampp/phpsqliteadmin/query.php
lampp/phpsqliteadmin/javascript.txt
lampp/phpsqliteadmin/SPSQLite.class-0.6.php
lampp/phpsqliteadmin/include.php
lampp/phpsqliteadmin/dbconfig.php
lampp/phpsqliteadmin/SPSQLite.class.diff
lampp/phpsqliteadmin/phpsla.css
lampp/cgi-bin/
lampp/cgi-bin/test-cgi
lampp/cgi-bin/printenv
linux:/home/christian # /opt/lampp/lampp start
Starte XAMPP fuer Linux 1.4.13...
XAMPP: Starte Apache mit SSL (und PHP5)...
XAMPP: Starte MySQL...
XAMPP: Starte ProFTPD...
XAMPP fuer Linux gestartet.
linux:/home/christian #

```

6 Rufen Sie im Webbrowser die Adresse *http://localhost/* auf.

Es erscheint die schon von Windows bekannte Startseite von XAMPP inklusive Statusansicht.



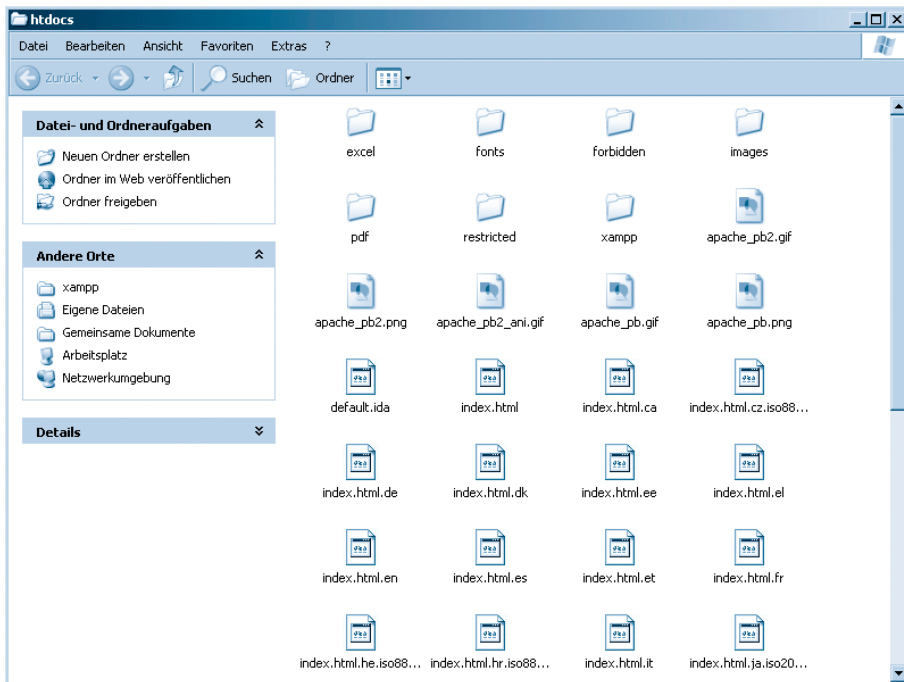
Sie sehen also – die Installation war gar nicht so schwer, für keines der beiden Betriebssysteme.

Jetzt können Sie ein wenig auf Entdeckungsreise gehen und unter *Demos* einige der Beispielprogramme ausprobieren. Doch Vorsicht: Nicht alle davon funktionieren, wenn Sie nicht alle XAMPP-Module aktiviert haben. Wundern Sie sich also nicht, wenn etwas nicht auf Anhieb klappt.

## Ein erster Test

Doch jetzt ist es an der Zeit, selbst tätig zu werden und einige PHP-Beispielprogramme zu schreiben. Dazu benötigen Sie zunächst eine sehr wichtige Information: was das Hauptverzeichnis des Webserverns ist.

Dazu ein Beispiel: Wenn Sie XAMPP für Windows unter *c:\apachefriends* installiert haben, ist *c:\apachefriends\xampp\htdocs* das Hauptverzeichnis. Wenn Sie also die Datei *index.html* in das Verzeichnis speichern, können Sie diese über den Webserver mit dem URL *http://localhost/index.html* aufrufen. Die (fiktive) Datei *c:\apachefriends\xampp\htdocs\easy\test.html* könnte parallel dazu über *http://localhost/easy/test.html* aufgerufen werden.



### Hinweis

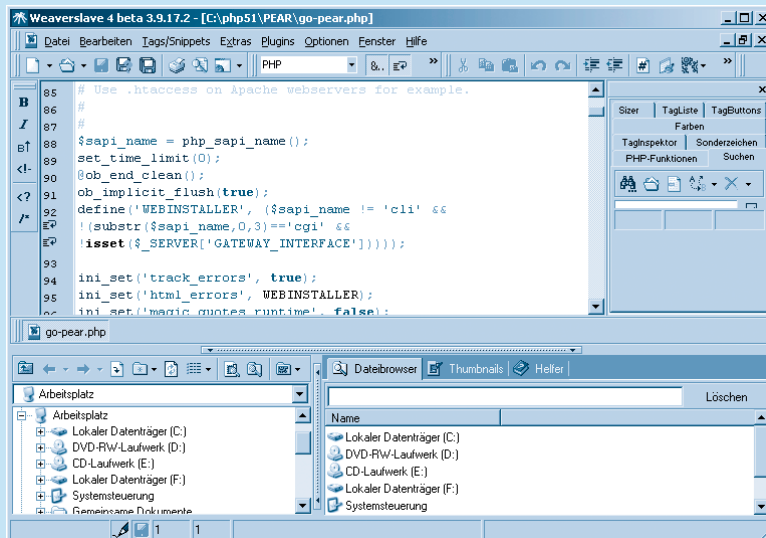
Noch einmal der Hinweis: PHP-Skripte können Sie nur über **http://localhost/...** aufrufen, nicht über das Dateisystem, also über **c:\verzeichnis\...** – das ist ein essenzieller Punkt für die weiteren Beispiele!

Zur Erinnerung: Wenn Sie Windows einsetzen, erhalten Sie über *Start/(Alle) Programme/apachefriends/xampp/xampp httpdoc folder* einen Zugriff auf den schon gut gefüllten Ordner, in dem die PHP-Dateien später liegen werden. Dort legen Sie die PHP-Dateien dann an – oder Sie erstellen der Übersicht halber einen Unterordner.

PHP-Dateien sind eigentlich reine Textdateien genauso wie auch HTML-Dateien. Sie können also einen beliebigen Texteditor verwenden. Nur eines ist wichtig: dass die PHP-Dateien die Endung *.php* haben. Nur dann werden sie vom PHP-Programm auf dem Webserver auch ausgeführt und beim Browser kommt HTML an (und kein PHP-Code, mit dem kein Browser etwas anfangen kann).

### Tipp

Unter **http://www.weaverslave.ws/** gibt es einen ziemlich guten und sogar kostenlosen PHP-Editor. Der bietet einige praktische Dinge, wie etwa eine farbliche Hervorhebung des Codes und vieles mehr.



PHP-Code wird – ähnlich wie JavaScript-Code – speziell gekennzeichnet, damit das PHP-Programm entscheiden kann, was PHP-Code und was HTML-Code ist.

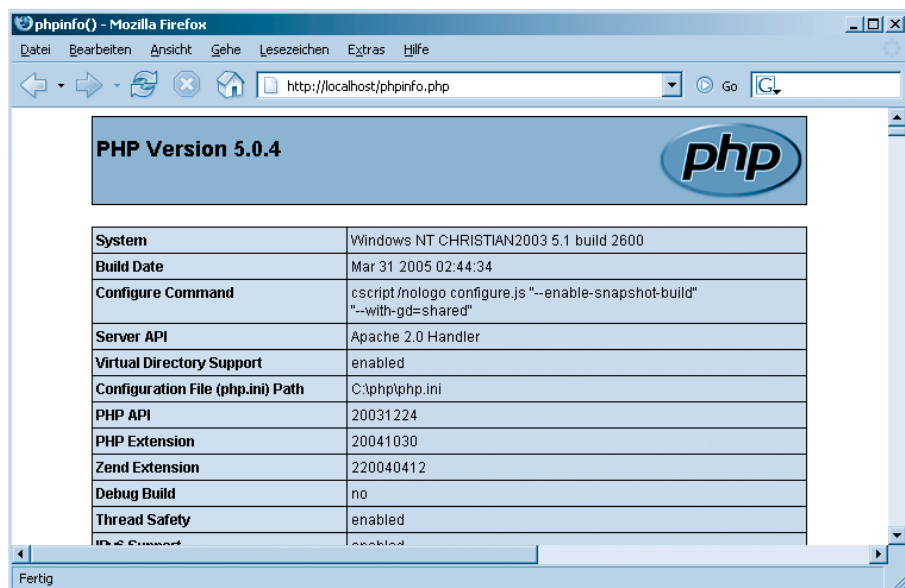
Es gibt mehrere Möglichkeiten, das zu tun, doch nur zwei davon funktionieren auf allen Systemen und unter allen Konfigurationen, egal was andere Bücher Ihnen weismachen möchten. Genau eine der beiden Möglichkeiten ist in der Praxis auch tatsächlich in Gebrauch:

- Mit `<?php` fängt der PHP-Code an.
- Mit `?>` hört der PHP-Code auf.

Dazwischen stehen die einzelnen PHP-Anweisungen. Wie bei JavaScript gilt auch: Alles endet mit einem Strichpunkt. Ganz allgemein sieht PHP häufig so ähnlich aus wie JavaScript: viele runde und auch einige geschweifte Klammern.

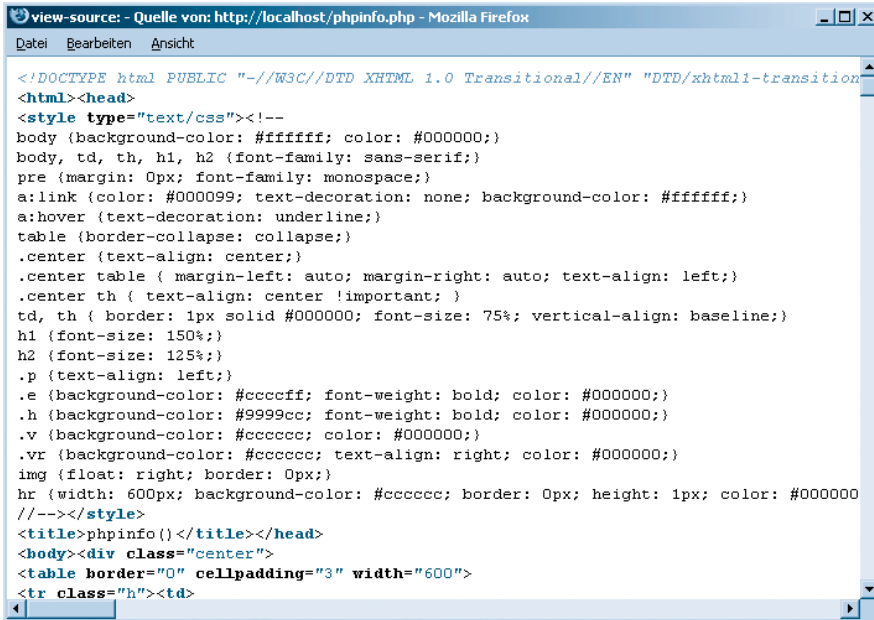
Die einfachste PHP-Anweisung heißt `phpinfo()`. Die ist nur neun Zeichen lang (beziehungsweise zehn Zeichen mit dem Strichpunkt), gibt aber viele Zeilen HTML-Code aus. Diese Anweisung nämlich erzeugt umfangreiche Informationen darüber, welche PHP-Version und welche Erweiterungen installiert sind sowie vieles mehr. Hier das komplette, sehr kurze Listing:

```
<!-- phpinfo.php -->
<?php
 phpinfo();
?>
```





Sehen Sie den Scrollbalken in der Abbildung? Es wird also eine ganze Menge Code erzeugt, aber wie Sie am Quellcode erkennen können, ist dort kein PHP-Code mehr vorhanden, sondern lupenreines (X)HTML und CSS.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional"
<html><head>
<style type="text/css"><!--
body {background-color: #ffffff; color: #000000;}
body, td, th, h1, h2 {font-family: sans-serif;}
pre {margin: 0px; font-family: monospace;}
a:link {color: #000099; text-decoration: none; background-color: #ffffff;}
a:hover {text-decoration: underline;}
table {border-collapse: collapse;}
.center {text-align: center;}
.center table { margin-left: auto; margin-right: auto; text-align: left;}
.center th { text-align: center !important; }
td, th { border: 1px solid #000000; font-size: 75%; vertical-align: baseline;}
h1 {font-size: 150%;}
h2 {font-size: 125%;}
.p {text-align: left;}
.e {background-color: #ccccff; font-weight: bold; color: #000000;}
.h {background-color: #9999cc; font-weight: bold; color: #000000;}
.v {background-color: #cccccc; color: #000000;}
.vr {background-color: #cccccc; text-align: right; color: #000000;}
img {float: right; border: 0px;}
hr {width: 600px; background-color: #cccccc; border: 0px; height: 1px; color: #000000}
--></style>
<title>phpinfo()</title></head>
<body><div class="center">
<table border="0" cellpadding="3" width="600">
<tr class="h"><td>
```

Wie gesagt, PHP und JavaScript ähneln sich in vielen Dingen. Allerdings gibt es natürlich zahlreiche kleine und große Unterschiede. Hier einige der wichtigsten:

- Variablennamen beginnen in PHP mit einem Dollarzeichen:  
\$variable = "Wert";
- Zur Ausgabe von Text gibt es kein document.write(), sondern echo().  
echo("ein bisschen Text ...");  
echo(\$variable);
- Wenn Zeichenketten aneinander gehängt werden, geschieht das mit einem Punkt, nicht mit einem Plus:  
echo "Die Variable hat den Wert " . \$variable;

Erinnern Sie sich noch an das Benzinpreis-Beispiel aus dem vorherigen Kapitel? Dort lief das Ganze mit JavaScript, jetzt wird es mit PHP umgeschrieben. Beachten Sie dazu einfach die vorangegangenen Unterschiede, dann geht es ganz leicht. Der Vorteil: Da PHP auf dem Server abläuft, funktioniert es auch bei deaktiviertem JavaScript.

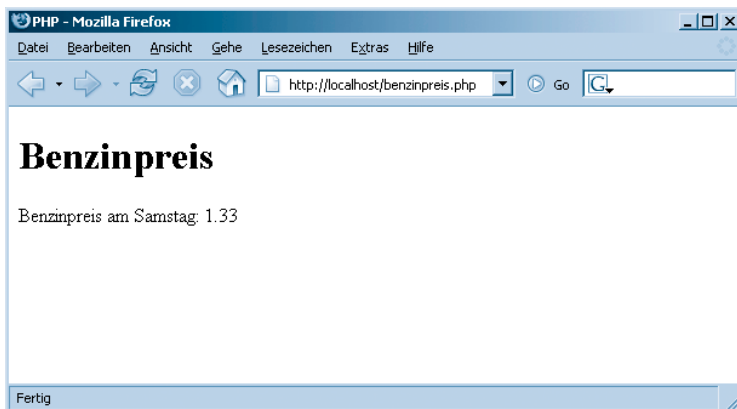
PHP-Code wird direkt in das HTML-Markup eingebunden. Das bedeutet: An den Stellen, an denen PHP steht, wird dieser Code ausgeführt und durch das Ergebnis dieser Ausführung (meist auch HTML) ersetzt.

Nachfolgend das Benzinpreis-Listing:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- benzinpreis.php -->
<html>
<head>
 <title>PHP</title>
</head>
<body>
 <h1>Benzinpreis</h1>
 <?php
 $benzinpreis = 1.23;
 $tag = "Samstag";

 if ($tag == "Samstag") {
 $benzinpreis = $benzinpreis + 0.10;
 } else if ($tag == "Sonntag") {
 $benzinpreis = $benzinpreis + 0.05;
 } else {
 $benzinpreis = $benzinpreis - 0.02;
 }
 echo("Benzinpreis am " . $tag .
 ": " . $benzinpreis);
 ?>
</body>
</html>
```

Andere Technologie, dasselbe Bild: ein hoher Benzinpreis am Samstag.



**Hinweis**

*Und ein allerletztes Mal der Hinweis, das Beispiel über den Webserver aufzurufen. Werfen Sie einen Blick auf die Abbildung: Der URL ist **http://localhost/benzinpreis.php**, die Datei liegt aber beispielsweise lokal in **c:\apache\friends\xampp\htdocs**.*

## Beispiel: Das aktuelle Datum ausgeben

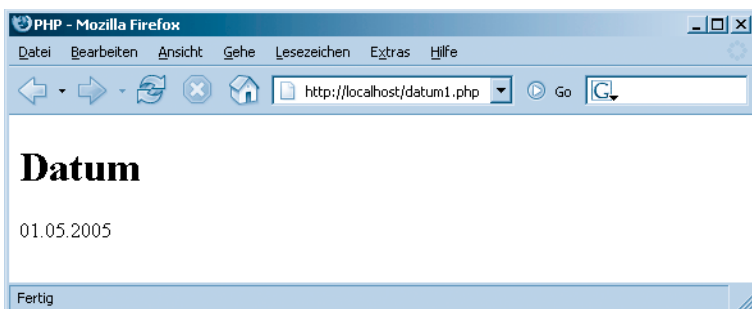
Ein Standardbeispiel ist die Ausgabe des aktuellen Datums – wirkt so eine Seite doch schön aktuell. PHP bietet dazu eine sehr mächtige, aber dennoch relativ einfach zu verwendende Funktion. Sie heißt `date()` und kann fast alles, was man sich im Bereich Datum vorstellen kann.

Als Parameter geben Sie für `date()` an, welchen spezifischen Datumswert Sie gerne hätten. Hier einige Beispiele:

- `date("d")` gibt den aktuellen Tag zurück (bei einstelligen Werten mit führender Null).
- `date("m")` gibt den Monat zurück (bei einstelligen Werten mit führender Null).
- `date("Y")` gibt das Jahr zurück (vierstellig).

Und jetzt wird es interessant. Was, denken Sie, gibt `date("d.m.Y")` zurück? Probieren Sie es aus, hier ein dazugehöriges Listing.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- datum1.php -->
<html>
<head>
 <title>PHP</title>
</head>
<body>
 <h1>Datum</h1>
<?php
 $datum = date("d.m.Y");
 echo($datum);
?>
</body>
</html>
```



Das *d* in "*d.m.Y*" wird durch den Tag, das *m* durch den Monat und das *Y* durch das Jahr ersetzt. Die beiden Punkte bleiben stehen. Sie können so also wunderbar das Datum in einem speziellen Format ausgeben.

Der Grund: `date()` erwartet als Parameter eine Zeichenkette. Einige Zeichen haben dort eine spezielle Bedeutung, wie die Tabelle zeigt:

| Zeichen | Beschreibung                                             | Wert am 1. Mai 2005, 07:08:09 Uhr |
|---------|----------------------------------------------------------|-----------------------------------|
| c       | Datum im ISO-8601-Format                                 | 2005-05-01T07:08:09+02:00         |
| d       | Tag, zweistellig (von 01 bis 31)                         | 01                                |
| g       | Stunde, einstellig (von 1 bis 12)                        | 7                                 |
| G       | Stunde, einstellig (von 0 bis 23)                        | 7                                 |
| h       | Stunde, zweistellig (von 01 bis 12)                      | 07                                |
| H       | Stunde, zweistellig (von 00 bis 23)                      | 07                                |
| i       | Minuten, zweistellig (von 00 bis 59)                     | 08                                |
| j       | Tag, einstellig (zwischen 1 und 31)                      | 1                                 |
| m       | Monat, zweistellig (von 01 bis 12)                       | 05                                |
| n       | Monat, einstellig (von 1 bis 12)                         | 5                                 |
| r       | Datum im RFC-2822-Format                                 | Sun, 01 May 2005 07:08:09 +0200   |
| s       | Sekunden, zweistellig (von 00 bis 59)                    | 09                                |
| t       | Anzahl Tag im aktuellen Monat (von 28 bis 31)            | 31                                |
| T       | Zeitzone                                                 | Westeuropäische Sommerzeit        |
| w       | Tag der Woche (0 = Sonntag, 1 = Montag, ... 6 = Samstag) | 0                                 |



| Zeichen | Beschreibung                | Wert am 1. Mai 2005, 07:08:09 Uhr |
|---------|-----------------------------|-----------------------------------|
| w       | Woche (von 1 bis 53)        | 17                                |
| y       | Jahr, zweistellig           | 05                                |
| Y       | Jahr, vierstellig           | 2005                              |
| z       | Tag im Jahr (von 0 bis 365) | 120                               |

Bleibt nur noch die Frage, was Sie tun müssen, wenn `date()` beispielsweise einen der Buchstaben ausgeben soll. Folgendes funktioniert nicht:

```
<?php
 $datum = date("Der d.m.Y");
 echo($datum);
?>
```

Denn unter anderem wird das `r` in "Der" durch das RFC-2822-Datum ersetzt. Abhilfe schafft es, wenn Sie vor die entsprechenden Zeichen zwei Backslashes (`\`, das erhalten Sie über `[AltGr] + [B]`) setzen:

```
<?php
 $datum = date("\\D\\e\\r d.m.Y");
 echo($datum);
?>
```

Hier eine kleine Übungsaufgabe: Geben Sie folgenden Text aus:

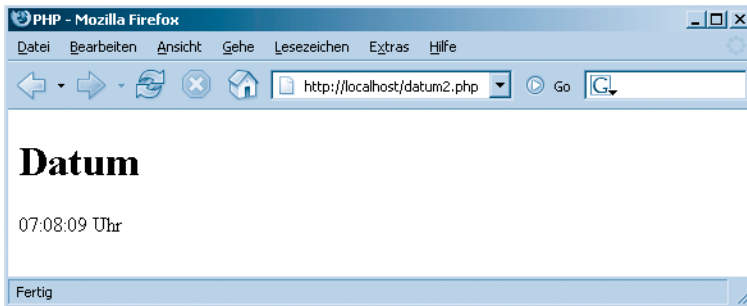
07:08:09 Uhr

Natürlich muss es sich um die aktuelle Uhrzeit handeln.

Wer spicken möchte: Hier die Auflösung:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- datum2.php -->
<html>
<head>
 <title>PHP</title>
</head>
<body>
 <h1>Datum</h1>
<?php
 $uhrzeit = date("H:i:s \\U\\h\\r");
 echo($uhrzeit);
?>
</body>
</html>
```

Die Buchstaben U, h und r müssen also durch je zwei Backslashes geschützt (maskiert) werden.



### Tipp

*Sie können es natürlich auch einfacher haben, indem Sie einfach einzelne Zeichenketten aneinander hängen:*

```
<?php
 $uhrzeit = date("H:i:s") . " Uhr";
 echo($uhrzeit);
?>
```

Jetzt sind Sie gefragt: Viel Spaß beim weiteren Experimentieren mit `date()`!

## Beispiel: Formularversand per E-Mail

Was clientseitig nur schwerlich möglich ist, ist serverseitig kein größeres Problem mehr: der Versand von E-Mails. PHP hat eine Unterstützung für den E-Mail-Versand fest integriert. Sie müssen nur eines tun: PHP dafür konfigurieren.

Dazu gibt es im PHP-Verzeichnis (bei der Verwendung von XAMPP für Windows ist das standardmäßig `C:\apache\friends\xampp\php`) die Datei `php.ini`, die die komplette Konfiguration für PHP enthält.



Wenn Sie die Datei in einem Texteditor öffnen, finden Sie recht weit unten folgende Einträge:

```
[mail function]
; For Win32 only.
SMTP = localhost

; For Win32 only.
sendmail_from = postmaster@localhost

; For Unix only. You may supply arguments as well (default:
"sendmail -t -i").
;sendmail_path =
```

Sie müssen hier die folgenden Einstellungen tätigen:

- Wenn Sie Windows einsetzen, geben Sie bei SMTP den Namen des SMTP-Servers an. Diese Information erhalten Sie bei Ihrem Provider. Der Eintrag `localhost` bedeutet, dass Sie einen lokalen SMTP-Server verwenden. Starten Sie dazu im XAMPP Control Panel das Modul *Mercury*, das ist nämlich genau so ein Mailserver.
- Wenn Sie Windows einsetzen, geben Sie bei `sendmail_from` die Absenderadresse für die Mails an.
- Wenn Sie Linux einsetzen, geben Sie bei `sendmail_path` den Pfad zum Programm *sendmail* an.

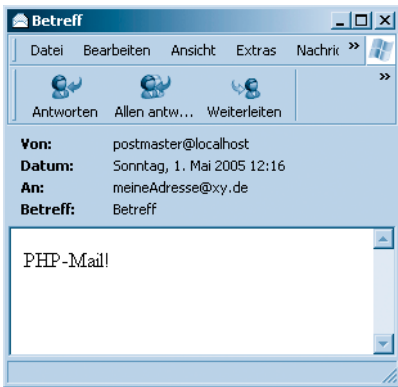
### Tipp

*Wenn Sie später PHP direkt bei Ihrem Hoster einsetzen, müssen Sie sich um solche Konfigurationsgeschichten nicht mehr kümmern, das hat der Hoster schon für Sie erledigt.*

Der Rest ist eigentlich einfach: Die Funktion `mail()` schickt E-Mails, Sie geben als Parameter einfach den Empfänger, den Nachrichtenbetreff und den Nachrichtentext an:

```
<?php
 mail("empfaenger@xy.de", "Betreff", "PHP-Mail!");
?>
```

Die Abbildung zeigt die daraus resultierende E-Mail.



### Hinweis

*Leider gibt es im Web immer mehr unerwünschte Werbemails (Spam), so dass zahlreiche Mailanbieter rigoros verdächtige Mails aussortieren. Eine von lokal verschickte Mail ist insofern verdächtig, als dass der Empfängerserver feststellen kann, dass diese von einer Einwahlleitung aus verschickt wird – genau so, wie Spammer das auch machen. Aus diesem Grund kann es sein, dass die Mail nicht ankommt. Allerdings funktioniert der Versand von einem Webserver aus dann umso zuverlässiger.*

In der Regel wollen Sie aber keine statische Mail verschicken, sondern den Inhalt eines Formulars. Aus diesem Grund finden Sie im Downloadarchiv dieses Kapitels die Datei *mail.php*. Diese verschickt E-Mails mit Formulardaten. Mit den folgenden Schritten setzen Sie sie ein:

**1** Ändern Sie in Ihrem Formular, das Sie in *Kapitel 6* erstellt haben, die Versandmethode auf POST und das action-Attribut auf das Mailskript.

```
<form method="post" action="mail.php">
```

**2** Passen Sie in der Datei *mail.php* in der zweiten Zeile des PHP-Teils die Adresse des Empfängers an (Variable *\$empfaenger*):

```
$empfaenger = "meineAdresse@xy.de";
```

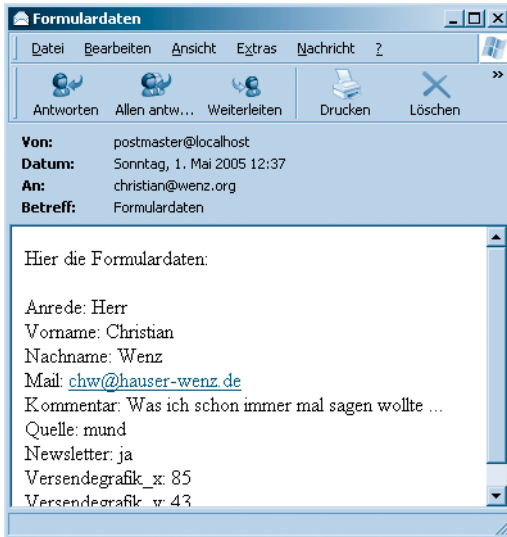
**3** Passen Sie, wenn Sie mögen, in der Datei *mail.php* in der dritten Zeile des PHP-Teils noch den Betreff der E-Mail an (Variable *\$betreff*):

```
$betreff = "Formulardaten";
```





Testen Sie das Ganze mit dem Beispielformular aus *Kapitel 6* (Sie finden es auch noch einmal im Downloadcode zu diesem Kapitel). Nach Befolgung der vorherigen Schritte und korrekter Konfiguration trudelt schon kurze Zeit später eine entsprechende E-Mail bei Ihnen ein.



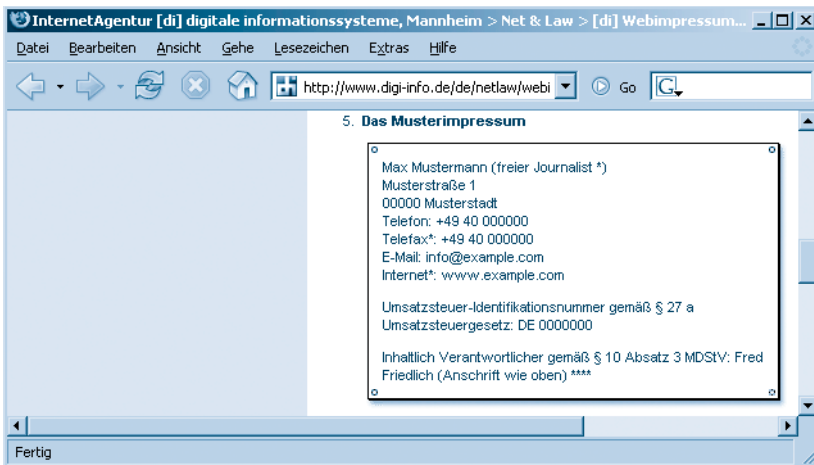
Damit ist der erste Einstieg in die »wunderbare Welt von PHP« schon einmal geschafft. Es gibt hier noch viel mehr zu entdecken. Wenn Sie das Thema interessiert: Bei Markt+Technik ist aus unserer Feder ein umfangreiches Werk rund um PHP erschienen, das »PHP Kompendium«.

## Nutzloses Wissen

Ungarn ist Europas größter Exporteur von Nilpferden.

## Kapitel 11

# Die Site online stellen



*Sie haben es fast geschafft: Die Site ist fertig. Jetzt müssen Sie sie nur noch auf den Webserver bekommen. Davor ist allerdings noch eine Bedingung zu erfüllen, Sie benötigen unter Umständen ein Impressum. Doch auch das ist recht schnell erledigt, es gibt mal wieder im Internet kostenlos Hilfe.*

**Das können Sie schon:**

|                                                |     |
|------------------------------------------------|-----|
| Formatierung von Text                          | 56  |
| Einsatz von Links                              | 75  |
| Aufzählungslisten                              | 88  |
| Grafiken einbauen                              | 116 |
| Frames erstellen                               | 134 |
| Formulare verwenden                            | 158 |
| Mit Tabellen und CSS positionieren             | 194 |
| Gästebuch, Diskussionsforum und Weblog anlegen | 212 |
| Einstieg und Verwendung von JavaScript         | 238 |
| PHP kennen lernen und verwenden                | 274 |

**Das lernen Sie neu:**

|                        |     |
|------------------------|-----|
| Impressum erstellen    | 304 |
| Website online stellen | 309 |

## Ein Impressum erstellen

Beginnen wir ausnahmsweise mit ein wenig Juristerei. Seit 2002 ist das Teledienstgesetz, kurz TDG, in Kraft. Das verpflichtet alle kommerziellen Homepages, unter anderem dazu, ein leicht zu findendes Impressum aufzuweisen.

Ab wann eine Homepage kommerziell ist, ist schwer zu sagen. Sobald Sie Werbung einblenden oder für irgendetwas werben, und wenn es nur Ihr Arbeitgeber ist, könnte man Ihre Website als kommerziell einstufen. Aber auch sonst ist ein Impressum eine Sache des guten Stils.

Doch was gehört in ein solches Impressum? Unter <http://bundesrecht.juris.de/bundesrecht/tdg/> finden Sie den Originaltext. Klicken Sie auf Abschnitt 2 und lesen Sie dort § 6 durch. Dort steht unter anderem Folgendes:

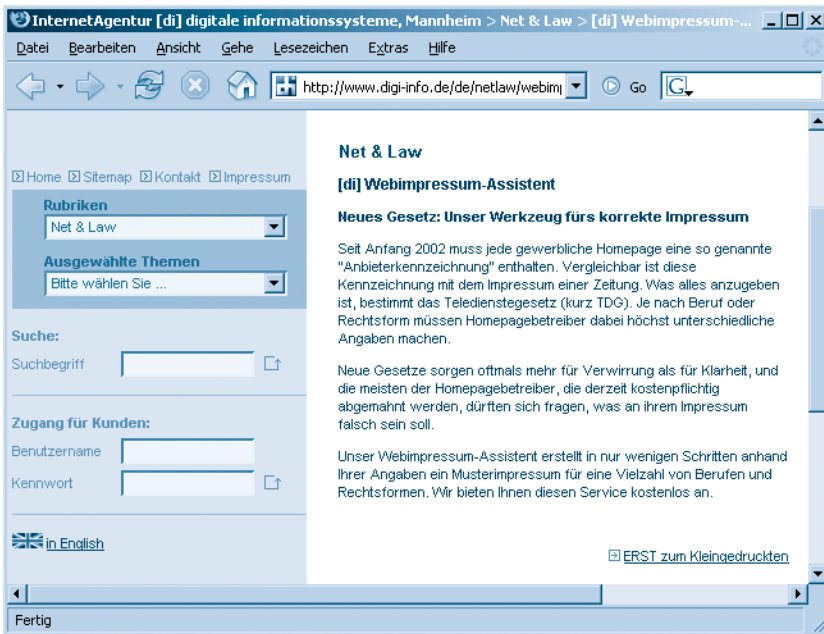
Diensteanbieter haben für geschäftsmäßige Teledienste mindestens folgende Informationen leicht erkennbar, unmittelbar erreichbar und ständig verfügbar zu halten:

- 1. den Namen und die Anschrift, unter der sie niedergelassen sind, bei juristischen Personen zusätzlich den Vertretungsberechtigten,*
  - 2. Angaben, die eine schnelle elektronische Kontaktaufnahme und unmittelbare Kommunikation mit ihnen ermöglichen, einschließlich der Adresse der elektronischen Post,*
  - 3. soweit der Teledienst im Rahmen einer Tätigkeit angeboten oder erbracht wird, die der behördlichen Zulassung bedarf, Angaben zur zuständigen Aufsichtsbehörde,*
  - 4. das Handelsregister, Vereinsregister, Partnerschaftsregister oder Genossenschaftsregister, in das sie eingetragen sind, und die entsprechende Registernummer,*
- [...]*
- 6. in Fällen, in denen sie eine Umsatzsteueridentifikationsnummer nach § 27a des Umsatzsteuergesetzes besitzen, die Angabe dieser Nummer.*



Oha. Und einen Unterpunkt, Nummer 5, haben wir sogar noch weggelassen (da geht es um die Angabe der Berufsbezeichnung). Also müssen all diese Punkte irgendwie aufgeführt werden. Damit Sie auch nichts vergessen, lohnt sich der Einsatz eines Online-Tools. Im Folgenden gehen wir davon aus, dass ein Freiberufler das Tool ausfüllt und sich sein Online-Impressum generieren lässt. Sie müssen den Vorgang also unter Umständen leicht anpassen.

**1** Rufen Sie die Adresse <http://www.digi-info.de/webimpressum/> im Webbrowser auf und klicken Sie auf *ERST zum Kleingedruckten*.

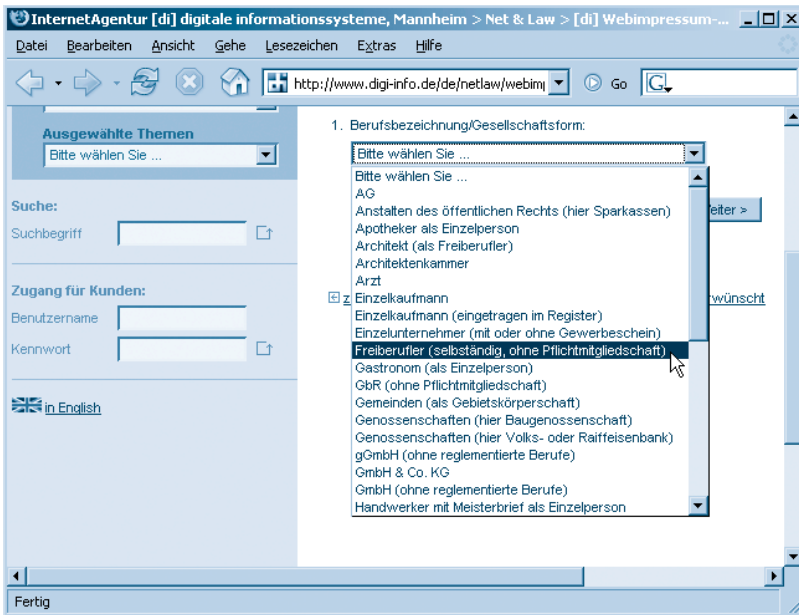


Sie müssen sich nun durch eine Menge Text, das »Kleingedruckte«, quälen. Danach klicken Sie auf den Link *zum Webimpressum-Assistenten*, dann kann es endlich losgehen.

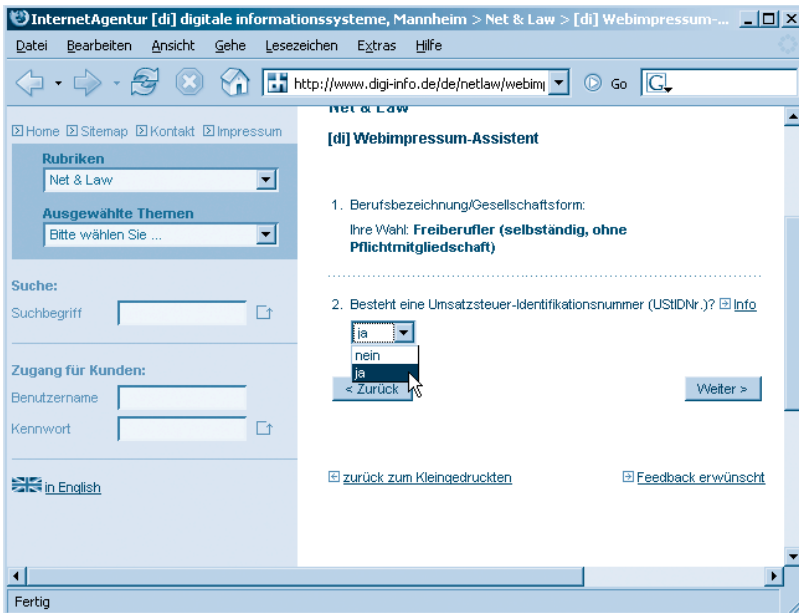
**2** Wählen Sie die Berufsbezeichnung aus.

Dieser Punkt ist entscheidend, denn aufgrund Ihrer Angabe an dieser Stelle fragt der Online-Assistent nach allen benötigten Informationen – und nach keinen, die unnötig sind.

Wir wählen den Eintrag *Freiberufler*, so dass sich für Ihre Anforderungen möglicherweise die nächsten Schritte leicht anders gestalten könnten.



3 Geben Sie an, ob Sie eine Umsatzsteuer-ID-Nummer (UStIDNr.) haben.  
Im Beispiel behaupten wir einfach mal ja.



**4** Geben Sie an, ob Sie journalistische oder redaktionelle Texte auf Ihrer Website haben (dazu könnte auch ein Weblog zählen).

Auch hier nehmen wir an, dass das der Fall ist.

InternetAgentur [di] digitale informationssysteme, Mannheim > Net & Law > [di] Webimpressum-...

Net & Law

Ausgewählte Themen  
Bitte wählen Sie ...

Suche:  
Suchbegriff

Zugang für Kunden:  
Benutzername  
Kennwort

In English

1. Berufsbezeichnung/Gesellschaftsform:  
Ihre Wahl: **Freiberufler (selbständig, ohne Pflichtmitgliedschaft)**

2. Besteht eine Umsatzsteuer-Identifikationsnummer (UStIDNr.)? [Info](#)  
Ihre Wahl: **ja**

3. Enthält der Webaufrtritt journalistisch-redaktionelle Texte? [Info](#)  
☒ ja  
☐ nein  
☐ ja

[< Zurück](#) [Weiter >](#)

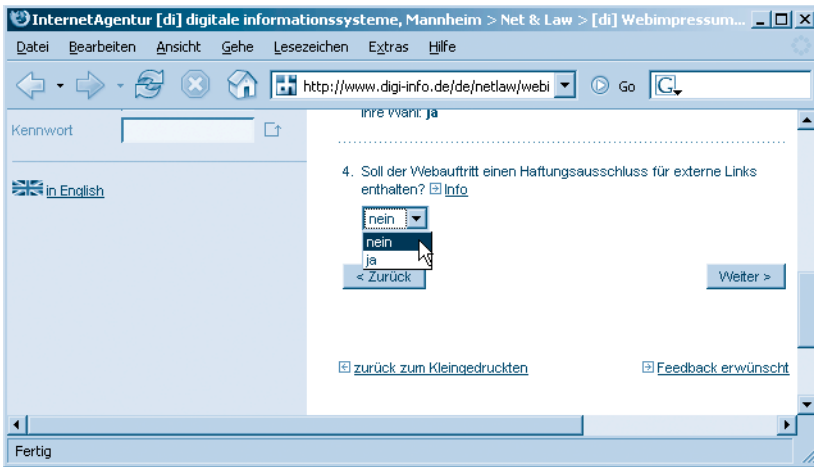
[zurück zum Kleingedruckten](#) [Feedback erwünscht](#)

Fertig

**5** Geben Sie an, ob Sie für externe Links einen Haftungsausschluss wünschen.

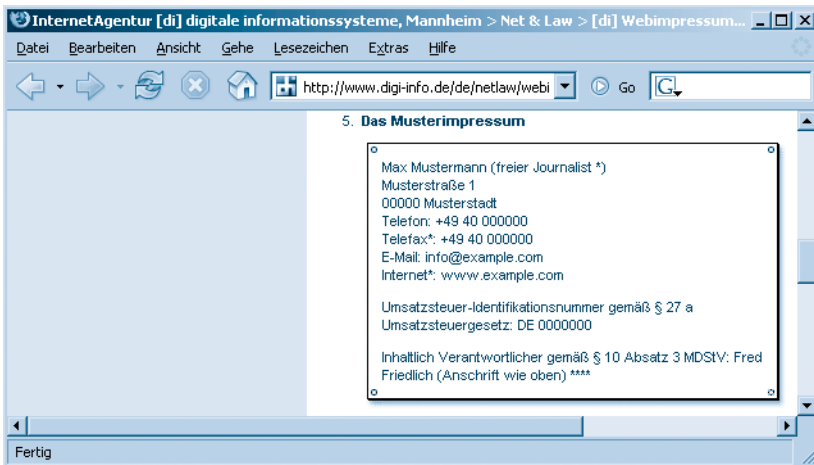
### Tipp

Auch wenn Sie einen solchen Haftungsausschluss mittlerweile auf sehr vielen Seiten finden: Er ist nach aktueller juristischer Auffassung komplett nutzlos. Sehr lesenswerte Informationen dazu finden Sie unter <http://schneegans.de/distanzierung-von-links/>.



## 6 Bauen Sie das erzeugte Impressum auf Ihre Website ein.

Vergessen Sie dabei aber natürlich nicht, die Platzhalter (etwa »Max Mustermann«) durch die richtigen Daten zu ersetzen.



Jetzt müssen Sie nur noch dafür sorgen, dass das Impressum auch gut gefunden werden kann. Dafür gibt es eine einfache Merkregel: Der Link auf das Impressum soll auch mit »Impressum« beschriftet und beim Laden der Webseite bei einer üblichen Bildschirmauflösung sofort sichtbar sein – also nicht rechts unten in einer Minischrift unter einem Werbebanner verstecken.





## Die Seite online stellen

Jetzt ist die Seite fertig, nun muss sie nur noch ins Internet. Dazu benötigen Sie einen **Hoster**. Das ist jemand, der einen Rechner direkt ans Internet angeschlossen hat. Sie kaufen bei diesem Hoster so genannten **Webspace** ein. Das ist Speicherplatz auf dem Rechner des Hosters.

Bei einigen Online-Diensten wie etwa AOL oder T-Online ist bereits Webspace mit dabei; Ihr Anbieter berät Sie sicherlich gerne, wie Sie Ihre Seiten dort online stellen können.

Richtig schick ist es aber, wenn Sie zusätzlich noch eine eigene Domain haben. Wenn Sie etwa bei T-Online eine Website haben, hat die eine Adresse nach dem Muster *http://<IhrName>.bei.t-online.de/*. Das ist zwar schon ganz nett, allerdings stört das T-Online ein wenig (bei anderen Anbietern ist es natürlich etwas Vergleichbares). Eine eigene Domain wie etwa *http://www.<IhrName>.de/* ist da schon deutlich hübscher.

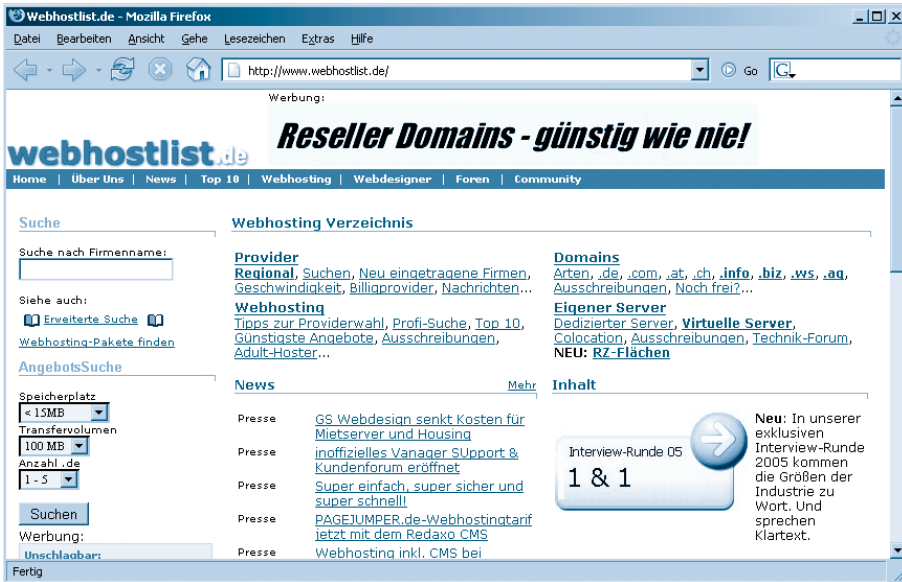
Sie müssen also mehrere Dinge erledigen:

- bei einem Hoster Webspace mieten
- eine Domain kaufen
- dem Hoster sagen, dass Sie diese Domain bei ihm »lagern« möchten

Früher war es so, dass Domains sehr, sehr teuer waren, und Hoster ebenso. Mittlerweile hat sich das geändert. Viele Hoster besorgen im Auftrag ihrer Kunden Domains, und das in großem Umfang, so dass sie die Domains zu einem wirklich sehr günstigen Preis erhalten. Und so ist es mittlerweile möglich, dass Sie Domain plus Webspace für wenige Euro im Monat bekommen können.

Es fällt uns schwer, eine spezifische Empfehlung für einen Hoster abzugeben, da sich der Markt zum einen sehr schnell wandeln kann und es zum anderen so viele Hoster gibt, dass es schwer fällt, einen Überblick zu behalten. Wir sind unter anderem mit *http://www.webhostone.de/* recht gut gefahren; außerdem gibt es da ja noch die Massenhoster *http://www.strato.de/* und *http://www.1und1.info/*, die irrsinnig viele Kunden haben, was ja auch einen Grund haben müsste.

Statt einer spezifischen Empfehlung können wir Ihnen einen universellen Tipp geben: Schauen Sie bei *http://www.webhostlist.de/* vorbei. Dort finden Sie eine umfangreiche Übersicht des Webhoster-Markts, wobei jeder Webhoster von seinen Kunden bewertet worden ist. So gibt es unter anderem auch eine Top 10 der besten Anbieter sowie eine Übersicht der günstigsten Anbieter. Hier können Sie ja schon mal mehrere Hoster in die nähere Auswahl nehmen.



Da der Hoster-Markt wirklich sehr groß ist, sind die Angebote nicht so ohne Weiteres vergleichbar. Aber einige Begriffe tauchen immer wieder auf, deswegen hier eine Liste nebst Erläuterungen:

- **Speicherplatz** oder **Webpace**: Wie viel Platz Ihre Webseiten belegen dürfen, also HTML-, CSS-, JavaScript-Dateien, PHP-Skripte und Grafiken zusammen.
- **Transfervolumen** oder **Traffic**: Wie viel Datenverkehr pro Monat im Pauschalpreis enthalten ist. Wenn also Ihre Website insgesamt 500 Kbyte groß ist und im Monat tausend Mal komplett abgerufen wird, ist ein Traffic von ca. 500 Mbyte entstanden. Wenn Sie mehr Traffic haben als vorgesehen, wird es häufig empfindlich teuer. Aber auch über diese Preise sollten Sie sich informieren.
- **Domains**: Eine Domain ist in der Regel bei jedem Paket dabei, manchmal auch mehrere. Wichtig: Achten Sie darauf, dass der Provider Sie als administrativer Ansprechpartner (»Admin-C«) der Domain(s) einträgt. Denn nur dann können Sie, wenn Sie mal den Provider wechseln möchten, auch die Domain(s) mitnehmen.
- **E-Mail-Adressen**: Wenn Sie die Domain *xy.de* registrieren, ist es standardmäßig so, dass *irgendwas@xy.de* per Mail bei Ihnen ankommt (das nennt man auch »Catch-All«). Allerdings ist es natürlich viel praktischer, wenn Sie eigene E-Mail-Postfächer für die verschiedenen Personen in der



Firma oder im Verein haben. Je nach Paket können Sie eine bestimmte Anzahl dieser spezifischen Postfächer anlegen. Praktisch ist es natürlich auch, wenn der Anbieter eine Form des Spam-Schutzes anbietet.

- **PHP und MySQL:** Wenn Sie PHP-Skripte verwenden möchten, muss der Provider auch PHP unterstützen, was meist ein wenig mehr kostet als ein Paket ohne PHP. Wollen Sie zusätzlich auch noch Datenbanken verwenden, benötigen Sie ein Paket, bei dem auch noch MySQL mit dabei ist, was wiederum mehr kostet. Das müssen Sie also genau abwägen. Allerdings gestatten es die meisten Webhoster, das Paket zu wechseln. Erkundigen Sie sich vorher.
- **Laufzeit und Kündigungsfrist:** Vergleichen Sie auch die Kündigungsfristen der Verträge. Wenn die Gebühr für das ganze Jahr am Anfang der Laufzeit eingezogen wird, ist das in Ordnung; allerdings sollte es trotzdem möglich sein, das Hosting-Paket während der Laufzeit zu kündigen und die restlichen Gebühren zurückzuerhalten.

Sie sehen, das ist eine ganze Menge an Informationen, die Sie abfragen und vergleichen können. Der wichtigste Punkt ist vielleicht noch die Auswahl des Domain-Namens. Die meisten Hoster bieten online die Möglichkeit an, abzufragen, ob eine Domain noch frei ist. Bei manchen Anbietern ist es sogar möglich, einfach nur einen Namen wie »Schmidt« anzugeben und es werden dann die Domains *schmidt.de*, *schmidt.at*, *schmidt.ch*, *schmidt.com* und so weiter auf Verfügbarkeit geprüft (übrigens alle vergeben, eine davon bis vor Kurzem an einen bekannten Talkshowmaster). Oder, Sie sehen direkt bei der Vergabestelle für die Domains nach:

- Für Deutschland ist das die DENIC, Abfrage unter <http://www.denic.de/de/whois/index.jsp>.
- Für Österreich ist das nic.at, Abfrage unter <http://www.nic.at/de/index/index/index.asp>.
- Für die Schweiz ist das SWITCH, Abfrage unter <http://www.switch.ch/de/id/>.

Haben Sie eine Domain registriert (was wie gesagt der Hoster in der Regel für Sie übernimmt) und auch ein Hosting-Paket, erhalten Sie die Zugangsdaten für Ihre Domain per E-Mail. Sie können dann – in der Regel per Webbrowser – die E-Mail-Adressen und -Konten konfigurieren. Viel interessanter sind jedoch die FTP-Zugangsdaten, die in der Regel auch per Mail übermittelt worden sind.

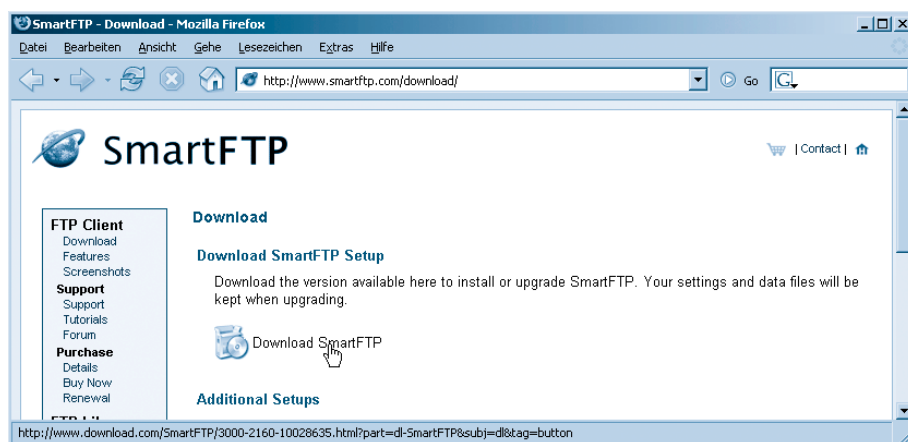
Bei FTP (**File Transfer Protocol**) handelt es sich um ein Internetprotokoll wie HTTP. FTP ist auf die Übertragung von Dateien optimiert. Deswegen gibt es für fast jeden Webspace einen FTP-Zugang. Sie verbinden sich über das Protokoll mit dem Server, auf dem Ihre Webseite liegt, und übertragen so die Dateien.

Alles, was Sie dazu benötigen, ist ein FTP-Programm. Bei den meisten Linux-Distributionen ist ein solches Programm dabei, auch beim Mac werden Sie schnell fündig. Unter Windows gibt es auch einiges, aber wenn Sie es recht bequem haben wollen, benötigen Sie zusätzliche Software aus dem Internet.

Es gibt mehrere empfehlenswerte FTP-Programme. Wir möchten SmartFTP vorstellen, eine englische Software, die es aber auch in Deutsch gibt. So installieren Sie sie und setzen sie ein:

**1** Gehen Sie auf <http://www.smartftp.com/download/> und klicken Sie auf den *Download*-Link.

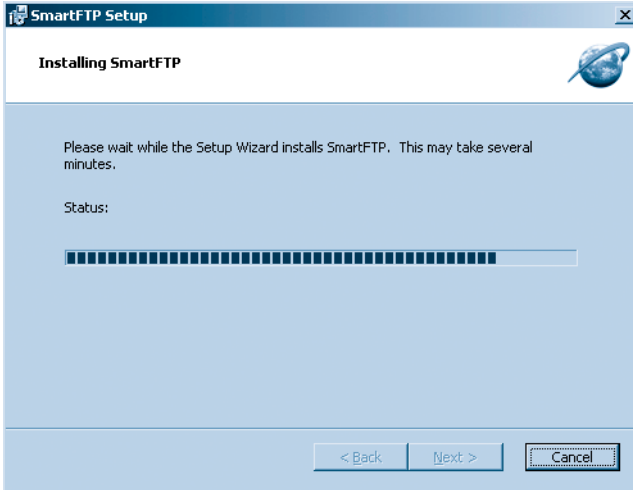
Auf der nächsten Seite klicken Sie auf *Download Now*, und nach kurzer Zeit können Sie das Programm herunterladen.



**2** Führen Sie die heruntergeladene Datei aus. Das Installationsprogramm ist sehr einfach gehalten, klicken Sie sich einfach durch.



Die Lizenzbedingungen akzeptieren Sie mit dem Radiobutton *I accept the terms in the Licence Agreement*. Beim Installationsmodus verwenden Sie am besten *Typical*, eine typische Installation mit allen wichtigen Komponenten.



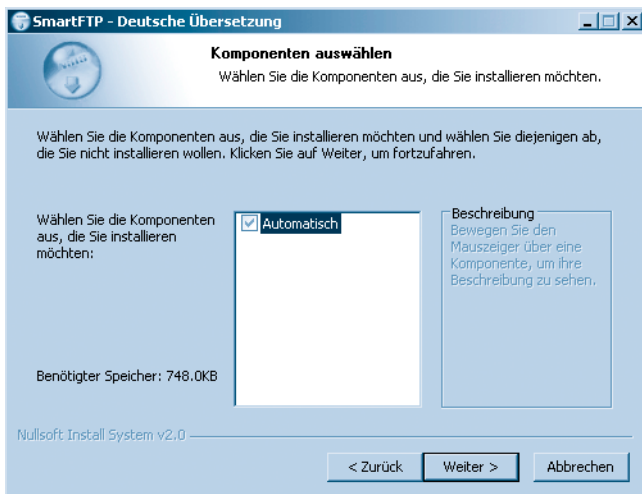
Deaktivieren Sie nach der Installation die Checkbox *Launch SmartFTP*, denn zunächst wollen wir noch die Oberfläche des Programms eindeutschen.

**3** Gehen Sie auf <http://www.smartftp.com/localization/projects/german/> und klicken Sie auf den Downloadlink des deutschen Sprachmoduls.



**4** Klicken Sie sich auch hier durch die Installation durch.

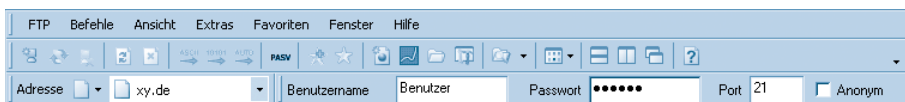
Am Ende lassen Sie das Häkchen in der Checkbox *Run SmartFTP*, denn dann startet SmartFTP automatisch – in Deutsch.



**5** Geben Sie die FTP-Zugangsdaten in SmartFTP oben in der Adressleiste ein. Sie sollten die folgenden Daten erhalten haben:

- Servername (oder Host)
- Benutzername
- Passwort

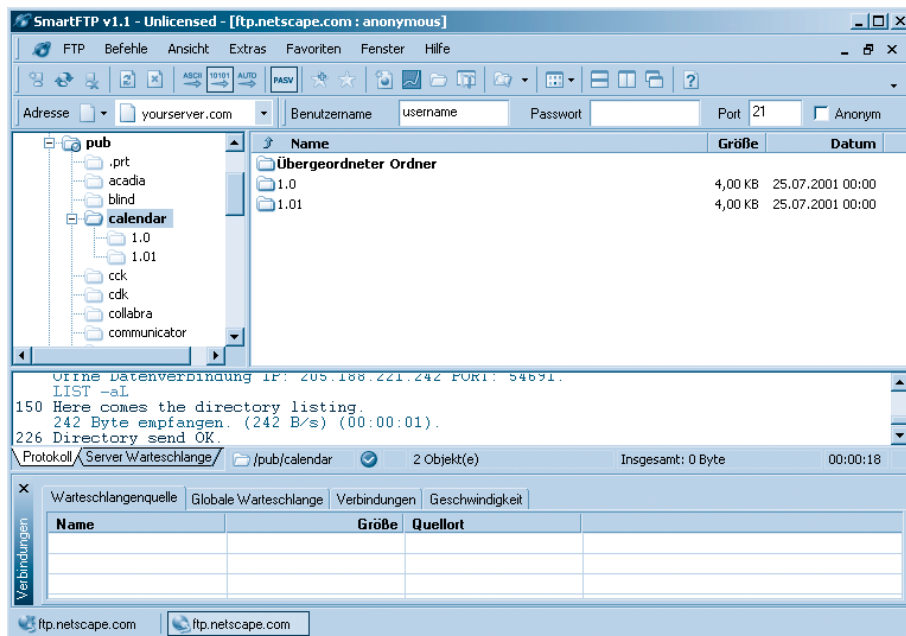
Diese drei Daten geben Sie ein, lassen bei *Port* die 21 stehen und drücken dann einfach auf .



**6** Übertragen Sie per Drag&Drop die Dateien von Ihrer Festplatte auf den Webserver.



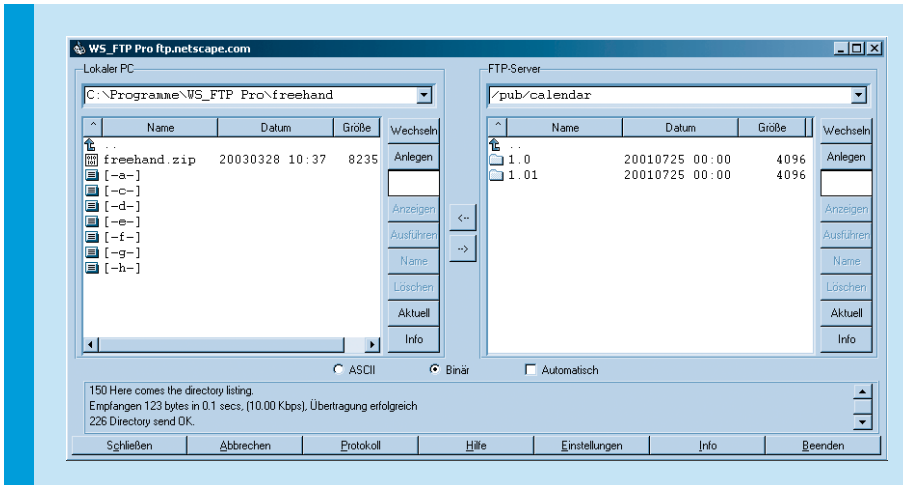
Was SmartFTP anzeigt, ist eine Explorer-Ansicht Ihres Webspaces. Sie können dort also per Drag&Drop Dateien und Ordner ablegen und auch Dateien und Ordner auf Ihre lokale Festplatte übertragen.



### Tipp

Ein Klassiker in Sachen FTP-Programmen ist **WS\_FTP**, erhältlich unter <http://www.wsftp.com/>. Allerdings kostet es mittlerweile etwas. Es gab – nur für den privaten Gebrauch – eine Freeware-Version **WS\_FTP LE**, doch die ist mittlerweile recht schwer zu finden. Tipp: Unter [http://www.wiesweg.de/ws\\_ftp/](http://www.wiesweg.de/ws_ftp/) ist sie immer noch verfügbar.

WS\_FTP bietet eine geteilte Explorer-Ansicht: links die lokale Festplatte, rechts der Webserver. Mit den Schaltflächen mit den Pfeilen drauf können Sie Dateien und Ordner in beide Richtungen schieben. Das ist also auch sehr intuitiv zu bedienen. Die Abbildung zeigt die kostenpflichtige Variante **WS\_FTP Pro**, die etwas mehr Features bietet, doch auch WS\_FTP LE genügt den meisten Ansprüchen.



Geschafft! Sie sollten nun glücklicher Webmaster sein. Das Dokument, das *index.html* heißt (oder bei manchen Hostern auch *index.php*), wird beim Aufruf der Domain automatisch geladen und im Browser angezeigt. Viel Erfolg bei Ihren weiteren Experimenten im Web!

## Nutzloses Wissen

Die US-Synchronstimme von Bugs Bunny hat eine Karottenallergie.





# Lexikon

## Accessibility

Zugänglichkeit einer Website für Menschen mit Behinderung. Beispielsweise soll Blinden mittels Screen Readern, die Inhalte der Seite vorlesen, die Möglichkeit gegeben werden, im *Web* zu surfen.

## Anker

siehe *Textmarke*

## Apache

Open Source-*Webserver* der Apache Foundation. Neben dem *IIS* der wichtigste *Webserver* mit – je nach Statistik – etwa zwei Dritteln Marktanteil. Der Apache ist für Windows, Linux und Mac OS X verfügbar.

## ASP.NET

*Serverseitige Technologie* von Microsoft, die ähnliche Aufgaben erledigt wie *PHP*. ASP.NET basiert auf dem .NET Framework, einer großen Bibliothek aus Klassen. Diese Klassen stellen die Funktionalität zur Verfügung. ASP.NET kann mit unterschiedlichen Programmiersprachen, z.B. Visual Basic .NET, C# oder J#, geschrieben werden. ASP.NET läuft auf dem *Webserver IIS* und damit hauptsächlich unter Windows.

## Attribut

Ein *Tag* kann eines oder mehrere Attribute zugewiesen bekommen. Jedes Attribut hat einen *Wert*.

## Banner

Werbeform im *Web*. Banner werden meist oben oder unten auf einer Webseite angezeigt und führen den Nutzer bei Klick auf die Seite des Werbers.

## Banneraustausch

Kleinere Websites tauschen untereinander Banner aus. Einer der ersten Vertreter dieser Marktnische war LinkExchange, das mittlerweile von Microsoft übernommen wurde.

## Berners-Lee, Tim

Gilt als Erfinder des Web. Entwickelte 1991/1992 *HTML*. Heute Vorsitzender des W3C.

## Browser

siehe *Webbrowser*

## Browserkrieg

Zeitspanne des großen Konkurrenzkampfes zwischen den *Webbrowsern* *Netscape Navigator* und *Microsoft Internet Explorer*, aus dem sich der Internet Explorer als Sieger herauskristallisiert hat. Die aktuellen Zugewinne von *Mozilla* und *Firefox* werden auch als zweiter Browserkrieg gewertet.

## CGI

Common Gateway Interface. Eine Art Übertragungsprotokoll bzw. Schnittstelle für serverseitige Skripte. Diese werden dann in einer serverseitigen Programmiersprache wie beispielsweise Perl erstellt.

## CSS

Cascading Style Sheet. Format für Style Sheets. Style Sheet-Dateien haben die Dateiendung *.css*. Stile können in den Kopf der *HTML*-Seite, in eine externe Datei oder in ein Tag eingebunden werden und dienen zum Formatieren von *Webseiten*. CSS ist ein Standard des *W3C*.

## Client

Bezeichnet den Nutzer bzw. den Rechner des Nutzers. Ist z.B. Teil des *Client-Server-Modells*. Client kann allerdings auch den *Webbrowser* bezeichnen.

## Client-Server-Modell

Im Client-Server-Modell bezieht der *Client* Daten und Dienste vom *Server*. Im Web basiert das Client-Server-Modell auf *HTTP*. Der *Webserver* sendet *HTML*-Dateien an den Client, wenn dieser sie anfordert.

## Datenbank

Eine Datenbank speichert Daten in einer für den Nutzer logischen Ordnung. Die Datenspeicherung kann auf verschiedene Arten erfolgen: Es gibt hierarchische, relationale, multidimensionale und objektorientierte Datenbanken. Heute sind die relationalen Datenbanken am weitesten verbreitet, *XML*-Datenbanken und objektorientierte Datenbanken gewinnen allerdings an Bedeutung. Eine Datenbank besitzt ein Datenbankmanagementsystem (*DBMS*), das für die Verwaltung zuständig ist.

## Denic

Zentrale Registrierungsstelle für *Domains* mit dem deutschen Länder-*Toplevel* *.de* (<http://www.denic.de/de/>).

## DocType

Angabe der verwendeten *DTD* in einer HTML- oder XHTML-Seite, d.h. des verwendeten »Dokumenttyps«. Gegen diese DTD wird die Seite *validiert*. Außerdem bestimmt der DocType das *Rendering*.

## Domain

Eindeutiger Name für eine *Website* im *Internet*; beispielsweise in der Form *www.domain.de*. Die Domain besteht aus dem Host (oft *www*), dem Domain-Namen an sich (*domain*) und der *Top Level Domain* (*de*). Die Top Level Domain ist ein Länderkürzel oder eine Gattungsbezeichnung (z.B. *com* für *commercial*).

## DTD

Document Type Definition. Sprache zur Festlegung von *DocTypes*. Der Standard stammt ursprünglich von SGML.

## ECMA

European association for standardizing information and communication systems (<http://www.ecma-international.org/>). Standardisierungsgremium, das unter anderem *ECMAScript* standardisiert.

## ECMAScript

Standardisiert den Sprachkern von *JavaScript*. Die Standardisierung begann erst im Nachhinein, als der *Browserkrieg* schon tobte. ECMAScript wird unter anderem auch als Sprachkern für *ActionScript*, die Skriptsprache von *Flash*, verwendet.

## Editor

Ein Editor ist ein Programm, um Programme oder *HTML*-Seiten zu erstellen. Im *Web*-Bereich unterscheidet man normale Text- und *HTML*-Editoren, die die Bearbeitung des *HTML*-Codes erlauben, und so genannte *WYSIWYG*-Editoren.

## Firefox

Standalone-Browser des *Mozilla*-Projekts.

## Flash

Vektorgrafiktechnologie von Macromedia, die sich im Internet als Quasi-standard durchgesetzt hat. Gleichzeitig auch der Name der Entwicklungsumgebung für Macromedia. Flash-Filme haben den Dateityp *SWF*.

## Formular

Eingabemaske, in der der Nutzer seine Angaben machen kann. Ein Formular kann aus verschiedenen Elementen wie Kontrollkästchen, Auswahlmenüs

und Textfeldern bestehen. Formulare werden meist mittels einer *serverseitigen Technologie* versandt oder in eine Datenbank gespeichert.

### Frame

Eine *Webseite* wird in mehrere Teile wie beispielsweise Navigationsleiste und Inhaltsframe unterteilt. Jeder dieser Teile ist eine eigene *HTML*-Seite und unabhängig vom Rest scrollbar.

### FTP

File Transfer Protocol. Übertragungsprotokoll für Daten. Mit diesem Protokoll laden Sie beispielsweise Ihre *Website* auf einen *Webserver*.

### GIF

CompuServe Graphics Interchange Format. Ursprünglich von CompuServe entwickeltes Bildformat, das im Internet eingesetzt wird. Es unterstützt nur 256 Farben, kann dafür eine Transparenzfarbe enthalten. Außerdem lässt sich mit dem Format eine *GIF-Animation* speichern.

### GIF-Animation

Folge von Einzelbildern, wird im *GIF*-Format als Animation abgespeichert. Die Animation kann mit Verzögerung einmal, mehrmals oder unbegrenzt abgespielt werden.

### Hexadezimal

Notation von Farben in *HTML*; beispielsweise #ffffff für Weiß. Das Muster funktioniert folgendermaßen: Das Doppelkreuz # markiert den Beginn der Farbnotation. Die nächsten zwei Ziffern geben den *RGB*-Rot-Wert, die folgenden zwei den Grün-Wert und die letzten zwei den Blau-Wert an. Der Wert wird mittels Multiplikation der ersten Ziffer mit 16 und dann Addition der zweiten Ziffer gebildet. Um 256 Werte abbilden zu können, muss jede Ziffer 16 Werte annehmen können, deshalb reicht der Wertebereich einer Ziffer nicht aus. Alternativ werden 0 bis 9 und dann a (10) bis f (15) verwendet. ff ist also  $15 * 16 = 240$ . Dann wird das zweite f = 15 addiert und heraus kommt der Wert 255 für Weiß.

### Hintergrund

In *HTML* oder *CSS* können Sie für eine Seite einen Hintergrund definieren. Dieser besteht entweder aus einer Farbe oder einem Bild.

### Homepage

Die Einstiegsseite einer *Website*. Bei kleinen, meist privaten Websites auch als Gattungsbegriff gebräuchlich.

## Hoster

Anbieter von Speicherplatz auf einem *Webserver* meist zusammen mit einer oder mehreren *Domains*. Sie haben die Wahl zwischen einem eigenen Server, bei dem Sie vollen Zugriff haben, und einem *Shared Hosting*-Paket, bei dem Sie den Webserver mit anderen teilen.

## HTML

HyperText Markup Language. Seitenbeschreibungssprache für Webseiten. HTML enthält *Tags* (Befehle), die vom Browser der Reihe nach interpretiert werden. Um HTML und den Nachfolger XHTML geht es in diesem Buch.

## HTTP

Hypertext Transfer Protocol. Übertragungsprotokoll für Daten im *Web*. Mit HTTP werden beispielsweise HTML-Seiten übertragen. Ein Übertragungsvorgang besteht aus der Anfrage (Request) des *Webbrowsers* und der Antwort (Response) des *Webservers*.

## HTTPS

*HTTP* mit *SSL*-Verschlüsselung.

## HyperText

Nichtlineare Organisation von Texten, die über *Links* netzartig miteinander verbunden sind. HyperText ist das Grundprinzip des *Web*.

## IIS

Internet Information Service. Webserver, der bei den *Microsoft*-Serverbetriebssystemen mit ausgeliefert wird. Der IIS ist in verschiedenen Versionen bei Windows NT, 2000, XP Professional und 2003 dabei. Windows XP Home besitzt keinen IIS.

## Imagemap

In *HTML* enthaltene Funktion, die es Ihnen erlaubt, ein Bild in mehrere Bereiche zu unterteilen und jeden Bereich mit einem *Link* oder sonstigen Funktionen zu versehen. Das *Tag* (Befehl) für eine Imagemap in *HTML* ist `<map>`, das für einen Bereich `<area>`.

## Indizierte Farben

Farben, die in einer Farbpalette zusammengefasst und aus einem größeren Farbraum herausgeschnitten wurden. Eine Farbpalette mit indizierten Farben kann maximal 256 Farben haben. Jeder Farbwert hat einen eigenen Index, der seine Position in der Palette festlegt. Indizierte Farben werden hauptsächlich beim Speichern im Dateiformat *GIF* eingesetzt.

## Interlaced

Abspeicheroption für *GIF*-, *JPEG*- und *PNG*-Dateien. Die Dateien werden im Browser in mehreren Stufen aufgebaut. Bei *GIF* wird zunächst jede achte Zeile dargestellt. Bei *JPEG* und *PNG* wird die Datei von unscharf zu scharf abgebildet.

## Internet

Weltumspannendes Rechnernetz. Beinhaltet mehrere Dienste wie E-Mail, *FTP* und das *World Wide Web*.

## Internet Information Service

siehe *IIS*

## Internet Explorer

Webbrowser von Microsoft. Der Internet Explorer hat den *Browserkrieg* gewonnen. Er ist heute der am weitesten verbreitete Browser. Vom Internet Explorer gibt es eine Windows- und eine Mac-Version. Letztere wird allerdings nicht weiterentwickelt.

## IP

1. Internet Protocol. Netzwerkprotokoll, das zusammen mit *TCP* die Protokollfamilie bildet, die die Basis des Internets ist.
2. IP-Adresse. Eine Nummer, unter der ein Rechner, egal ob *Client* oder *Server*, im *Internet* erreichbar ist.

## Java

Von Sun entwickelte Programmiersprache, die sich plattformunabhängig ausführen lässt. Dafür ist allerdings eine so genannte Virtual Machine nötig. Java kommt im Web auch *clientseitig* als Java-Applet zum Einsatz.

## JavaScript

Programmiersprache, die von *Webbrowsern* interpretiert wird. Insofern auch oft als Skriptsprache bezeichnet. JavaScript ist eine Ergänzung zu *HTML* und wurde von Netscape ins Leben gerufen. JavaScript enthält wichtige Funktionen zum Programmieren wie beispielsweise Schleifen und Variablen.

## JPEG

Joint Picture Experts Group. Bildformat für das *Web*. JPEG hat eine verlustbehaftete Komprimierung und eignet sich besonders für Fotos und Grafiken mit vielen Farbabstufungen.

## JScript

Die *JavaScript*-Variante des *Internet Explorers*. Der Name wurde auf Grund von Namensrecht-Problemen geändert.

### Kaskadieren

Vorgang des Verschachtelns von *CSS*. Das Kaskadieren bestimmt auch die Reihenfolge, in der Stile präferiert werden.

### Link

Verweis von einer Webseite auf eine andere. Ein Link kann relativ (z.B. `../start.html`) oder absolut (`http://www.addison-wesley.de`) erfolgen.

### MathML

Auf *XML* basierende Sprache, die die Darstellung von mathematischen Formeln gewährleisten soll. MathML kann mit *XHTML* verbunden werden.

### Microsoft

Unter anderem von Bill Gates gegründete Firma, die mit dem Betriebssystem Windows eine monopolähnliche Stellung im Desktop-Markt hat.

### Mozilla

Open Source-Projekt, das die Basis für den Firefox und die neuen Netscape-Browser (Netscape 6/7 und 8) legt. Das Besondere ist, dass der Quellcode offen ist und sich jeder an der Entwicklung beteiligen kann. Der aktuelle Zwischenstand kann unter <http://www.mozilla.org> begutachtet werden.

### Multimedia

Bezeichnet den Einsatz von verschiedenen Medien. Im Web bedeutet Multimedia die Verwendung von *Sound* und *Video*, unter Umständen kombiniert mit Interaktivität.

### Netscape

Von Jim Clark und Marc Andreessen gegründet. Mittlerweile von AOL aufgekaufter Softwarehersteller. Von ihm wird der *Webbrowser* Netscape Navigator programmiert. Dieser liegt mittlerweile in Version 8 vor.

### Opera

*Webbrowser* der norwegischen Firma Opera. Leichtgewichtig und mit viel Funktionalität. Der Opera wird kostenpflichtig ohne Werbung und kostenfrei mit Werbung angeboten.

### PNG

Format für Bilder im Web. Wurde entwickelt, um Copyright-Probleme mit dem *GIF*-Format zu umgehen, hat sich dann aber über GIF hinausentwickelt. Unterstützt als PNG-8 256 Farben, als PNG-24 sogar über 16,7 Millionen. PNGs werden verlustfrei mit dem LZW-Algorithmus komprimiert und erlauben Transparenz sowie Animationen. Allerdings unterstützt der *Internet Explorer* Transparenz nur mit Anpassungen.

## PHP

PHP: Hypertext Preprocessor. *Serverseitige Technologie* mit der größten Verbreitung. PHP ist ein Open Source-Projekt. Seine Marktstellung gewann PHP durch die einfache Bedienbarkeit und den großen Funktionsumfang.

## Rendering

Bezeichnet die Darstellung von Inhalten im *Webbrowser*. Eine HTML-Seite muss vom Quellcode in eine Webseite gerendert werden.

## RGB-Farbsystem

RGB steht für Rot, Grün und Blau. Das RGB-Farbsystem fügt die drei Farbkanaäle zusammen und bildet daraus die Farben. Das Zusammenfügen der Farben wird auch als additive Farbmischung im Gegensatz zur subtraktiven des Drucks bezeichnet. Monitore verwenden das RGB-Farbsystem. Deswegen setzen *HTML* und *CSS* Farben aus diesem System ein.

## Selektor

In *CSS* verfügbare Zuweisungsmöglichkeit von Stilen zu bestimmten Elementen.

## Server

1. Programm, das einen bestimmten Dienst oder bestimmte Protokolle implementiert, z.B. ein *Webserver*, der *HTTP* verwendet.
2. Rechner, auf dem das Programm läuft.

Oft werden beide Begriffe auch synonym verwendet.

## Serverseitige Technologie

Programmiersprache bzw. -technologie, die anspruchsvollere Aufgaben im Rahmen einer *Website* übernimmt. Dazu zählt beispielsweise der Mailversand oder die Datenbankanbindung.

## Shared Hosting

Bezeichnet Pakete bei *Hostern*. Ein Paket ist eine Einheit auf einem *Webserver*, der mit mehreren anderen *Websites* geteilt wird.

## SSL

Secure Socket Layer. Ursprünglich von *Netscape* entwickeltes Verschlüsselungsverfahren für die Übertragung per *Web*. Bei einer sicheren Verbindung trägt *HTTP* den Namen *HTTPS*.

## Style Sheet

Style Sheets dienen in *HTML*-Seiten der Formatierung und Positionierung von Elementen. Das Besondere an Style Sheets ist, dass eine Definition einmal an zentraler Stelle vorgenommen wird und dann auf verschiedene Bereiche angewendet werden kann.



Für Style Sheets muss eine spezielle Sprache eingesetzt werden. Bei normalen *HTML*- und *XHTML*-Seiten ist das *CSS*. Für *XML* gibt es unter dem Oberbegriff *XSL* noch eine weitere Stilsprache, die allerdings weitreichendere Möglichkeiten bietet.

### SVG

Scalable Vector Graphics. Auf *XML* basierendes Vektorgrafik- und Animationsformat. Vom *W3C* betreut und als Konkurrent von *Flash* bzw. *SWF* angetreten. Hat durchaus Nischen z.B. in der Kartographie gefunden, kann sich aber gegen *Flash* bisher nicht durchsetzen.

### SWF

Flash File Format. Das Format von *Flash*-Filmen. Es wird vom *Flash-Player*, einem *Webbrowser*-Plug-In interpretiert und dargestellt. *SWF* hieß in früheren Versionen *Shockwave File Format* und *Small Web Format*.

### Tabelle

Eine Tabelle in *HTML* besteht aus Zeilen mit darin enthaltenen Zellen. Neben dem klassischen Einsatz zur Präsentation von Daten können Tabellen auch zum Positionieren von Objekten und zum Strukturieren einer *Webseite* verwendet werden.

### Tag

Ein *HTML*-Befehl wird Tag genannt. Die meisten Tags bestehen aus einem Anfangs-Tag und einem End-Tag.

### TCP

Transmission Control Protocol. Verbindungsorientiertes Transportprotokoll, das zusammen mit *IP* die *TCP/IP*-Protokollfamilie bildet. Diese wiederum ist die Basis des Internets.

### Textmarke

Anker innerhalb einer *HTML*-Seite. Sie definieren mit einem Anker die Stelle, an die Sie springen, also das Verweisziel. Über einen Link mit angehängtem *#* verweisen Sie darauf.

### Top Level Domain

Länder- oder Gattungskennung für eine *Domain*. Die von der *Denic* verwaltete Toplevel für Deutschland ist z.B. *.de*. Die österreichische Toplevel ist *.at*, die schweizerische *.ch*.

### Transparentes GIF

*GIF*-Grafik, die aus einer einzigen Farbe besteht, welche transparent geschaltet wurde. Sie wird häufig in Tabellen als Platzhalter und zum Positionieren eingesetzt.

### URI

Uniform Resource Identifier. Heute oft synonym zum *URL* gebrauchter Begriff für eine eindeutige Adresse im Internet. Das *W3C* definiert die Unterschiede (<http://www.w3.org/Addressing/>).

### URL

Uniform Resource Locator. Die Internetadresse, die man auch bei einem *Link* angeben muss, also beispielsweise *www.addison-wesley.de*.

### URN

Uniform Resource Name. Oft mit *URL* verwechselt. Heute eigentlich als Bezeichnung für *XML*-Namensräume eingesetzt.

### Validieren

Feststellen der strukturellen Korrektheit eines Dokuments. *HTML*- und *XHTML*-Dokumente können mit ihrem *DocType* validiert werden.

### W3C

World Wide Web Consortium. Konsortium zur Findung und Festlegung von Internetstandards wie beispielsweise *HTML 4.01* und *XHTML*. Zu erreichen ist das *W3C* unter der Adresse <http://www.w3c.org>.

### Web

Das World Wide Web (WWW oder W3) ist der grafische Teil des Internets. Um Seiten des World Wide Web betrachten zu können, benötigen Sie einen Webbrowser, der die *HTML*-Seiten interpretiert.

### Webbrowser

Der Webbrowser stellt die *HTML*-Seiten dar. Dazu interpretiert er den *HTML*-Code und *rendert* die Seite. Außerdem zeigt er die Grafiken an. Mit Plug-Ins, die meist schon mitgeliefert werden, können Sie zusätzliche Formate darstellen wie beispielsweise *Flash*.

### Webseite

Einzelne *HTML*-Seite mit den zugehörigen Bildern. Eine Webseite ist meist Teil einer *Website*.

### Webserver

1. Rechner, der ständig an das Internet angebunden ist.
2. Software, die die Webserver-Funktionalität bereitstellt, z.B. *IIS* und *Apache*.

Beide Begriffe werden oft auch synonym verwendet. Per *URL* kann jeder Nutzer mit Internetanschluss auf die *Webseiten* auf dem Webserver zugreifen. Um Daten auf den Webserver zu laden, wird meist *FTP* eingesetzt.

### Websichere Farben

Farbpalette mit 216 Farben, die es bei 8-Bit Farbauflösung sowohl auf dem Macintosh als auch unter Windows gibt. Sie sorgt für sichere Farbdarstellung auch bei Monitoren mit nur 256 Farben. Die websichere Farbpalette wurde ursprünglich von dem Browserhersteller *Netscape* entwickelt. Sie wird allerdings heute nur noch wenig eingesetzt.

### Website

Eine Website ist ein komplettes Internetangebot, das sich unter einem Domain-Namen verbirgt. Eine Website besteht also meist aus mehreren *Webseiten*. Die Einstiegsseite wird als *Homepage* bezeichnet.

### Wert

Der Wert ist die Information, die einem *Attribut* zugewiesen wird. Ein Wert kann aus Zahlen oder Zeichen bestehen. Dies hängt von der Art des *Attributs* ab.

### Wohlgeformtheit

Bezeichnet ein syntaktisch korrektes Dokument gemäß der *XML*-Regeln. Syntaxfehler und andere Regelverletzungen stören die Wohlgeformtheit. Sagt allerdings nichts über die korrekte Struktur, also den Einsatz der richtigen Tags und Attribute, eines Dokuments aus. Letzteres wird über das *Validieren* festgestellt.

### WYSIWYG

What You See Is What You Get. Bezeichnet *Editoren*, die eine Design-Ansicht bieten, so dass Webentwickler ihre *HTML*-Seite zusammenklicken können, während sie ihr Ergebnis bereits sehen.

### XHTML

Extensible Hypertext Markup Language. Unter diesem Oberbegriff wird die Standardisierung von *HTML* nach *XML*-Richtlinien zusammengefasst. Nachfolger von *HTML*.

### XML

Extensible Markup Language. Metasprache, die nicht nur im *Web*-Umfeld zur Datenhaltung eingesetzt wird. Mittels *XML* kann man andere Sprachen definieren. Der Standard wird vom *W3C* gepflegt. *XML* arbeitet mit *Tags* und *Attributen*.



## Liebe Leserin, lieber Leser,

herzlichen Glückwunsch, Sie haben es geschafft. Sie können sich nun eine eigene Website aufbauen und einrichten. Ist es Ihnen nicht viel leichter gefallen, als Sie am Anfang dachten? Genau das ist das Ziel unserer Bücher aus der easy-Reihe. Sie sollen helfen, erfolgreich die ersten Schritte zu gehen, und den Leser auf keinen Fall mit unverständlichem Fachchinesisch überhäufen.

Als Lektorin hoffe ich, dass Sie durch das Buch die richtige Unterstützung bekommen haben. Denn für Ihre Zufriedenheit stehen alle Beteiligten mit ihrem Namen: der Verlag, die Autoren, die Druckerei.

Aber niemand ist perfekt. Wenn Sie Fragen haben: Fragen Sie. Wenn Sie Anregungen zum Konzept haben: Schreiben Sie uns. Und wenn Sie uns kritisieren wollen: Kritisieren Sie uns.

Ich verspreche Ihnen, dass Sie Antwort erhalten.

Denn nur durch Sie werden wir noch besser.

Ich freue mich auf Ihr Schreiben!

Birgit Ellissen  
Lektorin Markt + Technik  
Pearson Education Deutschland GmbH  
Martin-Kollar-Str. 10-12  
81829 München

E-Mail: [bellissen@pearson.de](mailto:bellissen@pearson.de)  
Internet: <http://www.mut.de>

# Stichwortverzeichnis

## Symbols

- !important 51
- # 50, 79
- \$szlig 39
- % 58
- & 37
- &amp; 39
- &Auml 39
- &auml 37, 39
- &copy 39
- &euro 39
- &gt; 39
- &lt; 39
- &nbsp; 102
- &Ouml 39
- &ouml 39
- &quot; 39
- &Uuml 39
- &uuml 39
- .css 21, 44
- .htm 26
- .html 21, 26
- .php 291
- // 245
- //--> 244
- :active 85
- :first-letter 74
- :first-line 74
- :hover 85
- <!-- 32, 244
- <?php 292
- <a> 30, 75, 79, 127
- <area> 129
- <b> 61
- <body> 27, 132, 197
- <br /> 28
- <caption> 114
- <col> 113
- <colgroup> 113
- <div> 52, 56, 202
- <dl> 94
- <dt> 94
- <font> 56
- <form> 161, 188
- <frame> 142
- <frameset> 142
- <h1> 49, 64
- <h2> 49, 64
- <h3> 64
- <h4> 64
- <h6> 64
- <head> 27, 42
- <hr> 31
- <html> 27
- <i> 61
- <iframe> 155
- <img> 121
- <input type= 163, 166, 173, 185, 192
- <input> 163
- <li> 90, 94
- <link> 45
- <map> 129
- <meta> 38
- <ol> 92
- <option> 179
- <p> 26, 56
- <script> 243
- <select> 179
- <span> 52
- <style> 42
- <sub> 64
- <sup> 64
- <table> 96
- <tbody> 111
- <td> 96
- <textarea> 168
- <tfoot> 111
- <th> 96
- <thead> 111
- <title> 27
- <tr> 96
- <ul> 90
- > 32
- ?> 292
- @import 45
- @media 52
- \_blank 30, 78, 153
- \_parent 78, 153

\_self 78, 153

\_top 78, 153

## A

Absätze 56

Abstände 72

ausrichten 69

formatieren 68

Größe 70

Pseudo-Klassen 74

Rahmen 70

absolute 202

Abstände 72

Accessibility 201

action 161, 188, 267, 300

Adobe GoLive 21

Adresse 30

alert() (JavaScript) 243

align 69, 105, 114, 125, 198

align= 105

all 53

alt 123, 130, 185

Anker 79, 317

Antialiasing 120

ASCII 38

Attribute 30

Regeln 34

Auflösung 120

Aufzählung 90

Aufzählungszeichen 91

aural 53

Außenabstände 72

Auswahllisten 178

Mehrfach 183

auto 149, 208

## B

background 132

background-attachment 132

background-color 104, 132

background-image 132

background-position 132

background-repeat 132

baseline 105

bcc 82

Berners-Lee, Tim 12, 26

bgcolor 132

Bildbearbeitungsprogramm 22

Bilder

Abstände 126

Alternativtext 123

Einbau 121

Formate 118

Größe 121

Links 127

optimieren 120

Rahmen 128

Titel 123

Verweissensitiv 129

Bitstream 56

blink 63

Blockelemente 32

Abstände 72

Blocksatz 69

Blog 232

blue 60

body 82

bold 41, 61

bolder 61

border 71, 96, 101, 185, 196

border-bottom 72

border-collapse 102

border-color 71, 102, 147

border-left 72

border-right 72

border-style 71

border-top 72

border-width 71

bottom 105, 125

Boxmodell 70

braille 53

Browser 14

HTML öffnen 28

## C

capitalize 62

caption-side 114

Cascading Style Sheets 40

cc 82

cellpadding 101, 196

cellspacing 101, 196

center 69, 105

CGI 318  
 Checkboxes 175  
 checked 174, 176  
 circle 92, 130  
 class 48  
 Clientseitig 13, 276  
 cm 40, 58  
 collapse 102  
 color 48, 59  
 cols 111, 142, 168  
 colspan 109  
 CompuServe Graphics Interchange Format 118  
 coords 129  
 CSS 40, 318  
   DocType 37  
   einbinden 41  
   Externe Datei 41, 44  
   Im Kopf 41, 42  
   Klassen 48  
   Maßeinheiten 58  
   Medienspezifisch 52  
   Positionieren 201  
   Präferenzreihenfolge 51  
   Pseudo-Klassen 74  
   Schriftformatierung 57  
   Vererbung 51  
 cursive 57

## D

date() (PHP) 295  
 Datum 258  
   mit PHP 295  
 default 130  
 Definitionsliste 94  
 deprecated 36, 56  
 disc 92  
 Diskussionsforum 222  
 DocType 34, 36  
 Document Object Model 29  
 document.write() (JavaScript) 253  
 Dokumenttypdefinition 36  
 DOM 29  
 Domain 310  
 Download 80, 84

## E

echo() (PHP) 293  
 Editor 21  
 else (JavaScript) 254  
 em 40, 58  
 E-Mail  
   Formularversand 187  
   Links 80  
 enctype 188  
 ex 58  
 eXtensible Markup Language 33

## F

fantasy 57  
 Farben 59  
   256 118  
   Farbname 60  
   Hexadezimal 59  
   websicher 59  
 Fenster  
   Abstand zum Rand 197  
   Im neuen öffnen 78  
 Fett 61  
 File Transfer Protocol 83, 312  
 fixed 132, 202  
 font-family 57  
 font-size 40, 58  
 font-style 61  
 font-variant 62  
 font-weight 41, 61  
 Formatierung 56  
 Formmailer 190  
 Formulare 158  
   Auswahllisten 178  
   Checkboxes 175  
   Kontrollkästchen 175  
   Mehrfache Auswahllisten 183  
   Optionsfelder 173  
   Passwortfeld 170  
   Per E-Mail 187  
   per PHP versenden 298  
   Radiobuttons 173  
   Textfelder 165  
   übermitteln 160  
   Überprüfung 263

- versenden 162
- Verstecktes Feld 192
- frameborder 147
- Frames 136
  - Frameset 142
  - Iframes 155
  - Mehrere ändern 267
  - Nachladen 270
  - verlinken 152
  - verschachteln 150
  - verschönern 146
- Frameset 36
- FTP 83, 312
  - Links 80
- ftp:// 83
- function (JavaScript) 256
- Funktionen 256

## G

- Gästebuch 214
- Geordnete Liste 92
- GET 160
- GIF 118, 320
- GIF-Animation 320
- Grafiken
  - Abstände 126
  - Alternativtext 123
  - Einbau 121
  - Formate 118
  - Größe 121
  - Links 127
  - Rahmen 128
  - Titel 123
  - Verweissensitiv 129
- green 60
- Großbuchstaben 62
- groups 111

## H

- handheld 53
- Hash 50
- height 99, 101, 121, 155, 168
- Hexadezimal 59
- Hexadezimalsystem 320
- Hintergründe 104, 132
  - wiederholen 132

- Hoster 22, 309
- href 30, 45, 75, 79, 81, 130
- hspace 102, 126
- HTML 26
  - Aufzählung 90
  - Befehle 26
  - Blockelement 32
  - Blocksatz 69
  - CSS einbinden 41
  - Grundgerüst 27
  - Im neuen Fenster 78
  - Inline-Element 32
  - Tabellen 95
  - validieren 35
- HTTP 12, 321
- HTTPS 12
- Hyperlinks 75
- HyperText Markup Language 26
- HyperText Transfer Protocol 12

## I

- ID 50
- id 79, 129
- Identifizier 50
- if (JavaScript) 254
- Iframes 155
- Imagemaps 129
- Impressum 304
- in 58
- Inch 58
- Inline-Element 32
- Inline-Stil 41
- Innenabstände 72
- Internet Explorer 14, 246
- Interpretersprache 241
- IP-Adresse 13
- ISO-8859-1 38
- ISO-8859-15 38
- italic 61

## J

- Java 240
- JavaScript 240
  - Datum 258
  - einbauen 241
  - Entscheidungen 254



- Externe Datei 245
- Frames nachladen 270
- Funktionen 256
- Grenzen 273
- Mehrere Frames ändern 267
- Methoden 256
- QuickNavigation 260
- Windows XP Service Pack 2 246
- Joint Picture Experts Group 118
- JPEG 118, 119
- justify 69

## K

- Kapitälchen 62
- KDE 19
- KHTML 19
- Klassen 48
- Kleinbuchstaben 62
- Komprimieralgorithmus 118
- Konqueror 19
- Kontrollkästchen 175
- Kursiv 61

## L

- language 243
- Layout
  - Mit CSS 209
  - Mit Tabellen 200
- left 69, 105, 202
- leftmargin 197
- lighter 61
- line-height 63
- line-through 62
- Links 75
  - Absolut 76
  - Arten von 75
  - Download 80
  - E-Mail 80
  - Formatieren 84
  - Frames 152
  - FTP 80
  - Für Bilder 127
  - Newsgroups 80
  - Nicht unterstrichen 63

- PDF 84
- Pseudo-Klassen 84
- Relativ 76
- Ziel 78
- Linkziel 78
- Linux 19
- Listen 90
  - Aufzählungszeichen 91
  - Definitions- 94
  - Formatieren 95
  - Nummerierte 92
  - ungeordnet 90
- list-style-image 95
- LiveScript 240
- localhost 284
- lowercase 62
- Lynx 20

## M

- Macintosh 19
- Macromedia Dreamweaver 21
- mail() (PHP) 299
- mailto 81, 188
- margin 72, 102, 124, 197
- marginheight 197
- margin-left 208
- margin-right 208
- marginwidth 197
- maxlength 166
- media 52
- Medienspezifisch 52
- Mehrzeilige Textfelder 168
- method 161, 188
- Methoden 256
- Microsoft FrontPage 21
- middle 105, 125
- min-height 101
- min-width 101
- mm 58
- monospace 57
- Mosaic 16
- Mozilla 15, 323
- multiple 183

## N

- name 79, 124, 144, 155, 163, 166, 170, 173, 176, 179, 185, 265
- Navigation
  - mit JavaScript 260
- Netscape 16, 240
  - Navigator 4.x 16
- news 84
- Newsgroups
  - Links 80
- no 149
- no-repeat 132
- noresize 146
- normal 48
- Nummerierte Liste 92

## O

- oblique 61
- Opera 17
- Optionsfelder 173
- overline 63

## P

- padding 72, 124, 197
- Paragraph 56
- Passwortfeld 170
- pc 58
- PDF 84
- PHP 276
  - Datum 295
  - Formularversand 298
  - Installation 278
  - Möglichkeiten 276
  - Testen 290
- phpinfo() (PHP) 292
- Pixel 58
- PNG 118, 119
  - 24 119
  - 8 119
- polygon 130
- Portable Network Graphics-Format 118
- position 202
- Positionierung 194
  - Mit CSS 201
  - Überlappung 206

- POST 160
- Präferenzreihenfolge 51
- print 53
- Processing-Instruction 38
- projection 53
- Pseudo-Klassen 74
  - Links 84
- pt 40, 58
- px 58

## Q

- Quicknavigation 260
- QuirksMode 37

## R

- Radiobuttons 173
- Rahmen 70
  - Für Bilder 128
- rect 129
- red 60
- Reihe 96
- relative 202
- Rendern 27
- repeat 132
- repeat-x 132
- repeat-y 132
- Request 12
- Response 12
- RGB 59
- right 69, 105
- rows 111, 145, 168
- rowspan 109
- rules 111

## S

- Safari 19
- sans-serif 57
- Schaltflächen
  - Grafisch 184
  - Versende- 162
- Schrift 56
  - farbe 58
  - Fett 61
  - größe 58
  - Kursiv 61

- Typographie 62
- Unterstrichen 62
- Zeilenhöhe 63
- screen 53
- scroll 132
- scrolling 149
- selected 179
- Selektoren 41, 46
  - Als Überschriften 66
  - ID 50
  - Klassen 48
  - Mehrere Tags 46
  - Tag 46
  - Verschachtelte Tags 46
- sendmail\_from (PHP) 299
- sendmail\_path (PHP) 299
- separate 102
- serif 57
- Serverseitig 14, 276
- shape 129
- size 166, 183
- Skriptsprache 241
- small-caps 62
- SmartFTP 312
- SMTP 299
- Sonderzeichen 37
- speech 53
- square 92
- src 121, 143, 155, 185, 246
- static 202
- Strict 36
- style 41, 169, 179
- subject 82
- Sun Microsystems 240
- T**
- Tabellen 95
  - Abstände 101
  - gliedern 111
  - größe 99
  - Hintergründe 104
  - Inhalte ausrichten 105
  - Inhalte formatieren 106
  - Positionieren mit 196
  - Rahmen 96, 101
  - Spalten 113
  - Spaltenübergreifend 109
  - Unsichtbar 196
  - verschachteln 97
  - vorstrukturieren 113
  - Zeile 96
  - Zeilenübergreifend 109
  - Zellen verbinden 109
- Tag 26, 30
  - Regeln 33
  - Selektoren 46
- target 30, 78, 130, 153, 155
- TDG 304
- Teledienstgesetz 304
- Text
  - Am Bild ausrichten 125
  - Blinkend 63
  - Einzüge 73
  - Hochgestellt 64
  - In Tabellen 106
  - Tiefgestellt 64
- text-align 69, 105
- text-decoration 62
- Textfelder 165
  - Mehrzeilig 168
  - Passwortfeld 170
- text-indent 73
- Textmarken 79
- text-transform 62
- Thumbnail 122
- title 123, 130
- top 105, 125, 202
- topmargin 197
- Traffic 310
- Transfervolumen 310
- Transitional 36
- TrueDoc 56
- type 42, 91, 93, 163, 171, 176, 243
- Typographie 62
- U**
- Überschriften 64
  - Tabellenzellen 96
- Umlaute 37
- underline 62
- Ungeordnete Liste 90
- uppercase 62

URL 30  
url() 132  
usemap 129  
UTF-8 38

## V

Validierung 35  
valign 105  
value 94, 163, 166, 171, 173, 176, 179  
var (JavaScript) 251  
Vererbung 51  
Verschachtelte Tabellen 97  
Versendeschaltfläche 162  
vertical-align 105, 125  
vspace 102, 126

## W

W3C 26, 35, 326  
Weaverslave 291  
Web 12  
    Links 75  
Web Embedded Fonts 56  
Webadresse 30  
Weblog 232  
Webseite  
    Dateiendung 26  
    Elemente positionieren 194  
    Kopf 27  
    Körper 27  
    Text 54  
Websichere Farben 59

Website  
    Online stellen 309  
Webospace 309  
width 70, 99, 101, 113, 121, 155, 164,  
    166, 168  
Windows 14  
Windows XP  
    Service Pack 2 246  
World Wide Web Consortium 26, 35  
WS\_FTP 315  
WYSIWYG 21

## X

XAMPP 278  
XHTML 33, 327  
    Regeln 33  
XML 33, 327

## Y

yes 149

## Z

Zeichensätze 37  
Zeile 96  
Zeilenhöhe 63  
Zelle 96  
Zentrieren 198, 208  
Ziel für Links 78  
z-index 206  
Zoll 58

# Wichtige HTML-Tags

| Tag                                            | Erklärung                             |
|------------------------------------------------|---------------------------------------|
| <b>HTML-Gerüst</b>                             |                                       |
| <code>&lt;html&gt;...&lt;/html&gt;</code>      | Äußerer Rahmen                        |
| <code>&lt;head&gt;...&lt;/head&gt;</code>      | Kopfbereich                           |
| <code>&lt;title&gt;...&lt;/title&gt;</code>    | Seitentitel (im Kopfbereich)          |
| <code>&lt;style&gt;...&lt;/style&gt;</code>    | Stilangaben (im Kopfbereich)          |
| <code>&lt;body&gt;...&lt;/body&gt;</code>      | Inhaltsbereich                        |
| <b>Textformatierung</b>                        |                                       |
| <code>&lt;p&gt;...&lt;/p&gt;</code>            | Absatz                                |
| <code>&lt;br /&gt;</code>                      | Umbruch                               |
| <code>&lt;hr /&gt;</code>                      | Horizontale Linie                     |
| <code>&lt;b&gt;...&lt;/b&gt;</code>            | Fettdruck                             |
| <code>&lt;i&gt;...&lt;/i&gt;</code>            | Kursivdruck                           |
| <code>&lt;sup&gt;...&lt;/sup&gt;</code>        | Hochstellen                           |
| <code>&lt;sub&gt;...&lt;/sub&gt;</code>        | Tiefstellen                           |
| <code>&lt;div&gt;...&lt;/div&gt;</code>        | Container                             |
| <b>Listen</b>                                  |                                       |
| <code>&lt;ol&gt;...&lt;/ol&gt;</code>          | Geordnete Liste (1., 2., 3., ...)     |
| <code>&lt;ul&gt;...&lt;/ul&gt;</code>          | Ungeordnete Liste (Aufzählungspunkte) |
| <code>&lt;li&gt;...&lt;/li&gt;</code>          | Listenelement (innerhalb einer Liste) |
| <b>Links</b>                                   |                                       |
| <code>&lt;a href="..."&gt;...&lt;/a&gt;</code> | Link                                  |
| <code>&lt;a name="..."&gt;...&lt;/a&gt;</code> | Textmarke                             |

# Wichtige HTML-Tags

| Tag                                               | Erklärung                                     |
|---------------------------------------------------|-----------------------------------------------|
| <b>Grafiken</b>                                   |                                               |
| <code>&lt;img src="..." /&gt;</code>              | Grafik                                        |
| <b>Tabellen</b>                                   |                                               |
| <code>&lt;table&gt;...&lt;/table&gt;</code>       | Tabelle                                       |
| <code>&lt;tr&gt;...&lt;/tr&gt;</code>             | Tabellenzeile (innerhalb einer Tabelle)       |
| <code>&lt;td&gt;...&lt;/td&gt;</code>             | Tabellenzelle (innerhalb einer Tabellenzeile) |
| <b>Formulare</b>                                  |                                               |
| <code>&lt;form&gt;...&lt;/form&gt;</code>         | Formular                                      |
| <code>&lt;input type="text" /&gt;</code>          | Texteingabefeld (einzellig)                   |
| <code>&lt;input type="password" /&gt;</code>      | Passwortfeld                                  |
| <code>&lt;input type="hidden" /&gt;</code>        | Verstecktes Feld                              |
| <code>&lt;textarea&gt;...&lt;/textarea&gt;</code> | Mehrzeiliges Texteingabefeld                  |
| <code>&lt;input type="checkbox" /&gt;</code>      | Checkbox                                      |
| <code>&lt;input type="radio" /&gt;</code>         | Radiobutton                                   |
| <code>&lt;select&gt;...&lt;/select&gt;</code>     | Auswahlliste                                  |
| <code>&lt;option&gt;...&lt;/option&gt;</code>     | Element einer Auswahlliste                    |
| <code>&lt;input type="file" /&gt;</code>          | Datei-Upload                                  |
| <code>&lt;input type="button" /&gt;</code>        | Schaltfläche                                  |
| <code>&lt;input type="reset" /&gt;</code>         | Zurücksetzen-Schaltfläche                     |
| <code>&lt;input type="submit" /&gt;</code>        | Versenden-Schaltfläche                        |
| <code>&lt;input type="image" /&gt;</code>         | Grafische Versenden-Schaltfläche              |



### **Copyright**

Daten, Texte, Design und Grafiken dieses eBooks, sowie die eventuell angebotenen eBook-Zusatzdaten sind urheberrechtlich geschützt. Dieses eBook stellen wir lediglich als persönliche Einzelplatz-Lizenz zur Verfügung!

Jede andere Verwendung dieses eBooks oder zugehöriger Materialien und Informationen, einschliesslich

- der Reproduktion,
- der Weitergabe,
- des Weitervertriebs,
- der Platzierung im Internet, in Intranets, in Extranets,
- der Veränderung,
- des Weiterverkaufs
- und der Veröffentlichung

bedarf der schriftlichen Genehmigung des Verlags.

Insbesondere ist die Entfernung oder Änderung des vom Verlag vergebenen Passwortschutzes ausdrücklich untersagt!

Bei Fragen zu diesem Thema wenden Sie sich bitte an: [info@pearson.de](mailto:info@pearson.de)

### **Zusatzdaten**

Möglicherweise liegt dem gedruckten Buch eine CD-ROM mit Zusatzdaten bei. Die Zurverfügungstellung dieser Daten auf unseren Websites ist eine freiwillige Leistung des Verlags. Der Rechtsweg ist ausgeschlossen.

### **Hinweis**

Dieses und viele weitere eBooks können Sie rund um die Uhr und legal auf unserer Website



herunterladen