

Thorsten Steinhoff
<http://www.lautwert.de>



Einführung in die Webseitengestaltung mit HTML

Seminar an der
Journalistenakademie
Dr. Hooffacker & Partner
München

Thorsten Steinhoff

Einführung in die
Webseitengestaltung mit HTML

Alle Rechte beim Autor
© 2001

PDF-Version 1.1
(4. September 2001)

Inhaltsverzeichnis

A. Wie sag ich's meinem Browser?	4
1. Wenn Daten sichtbar werden.....	4
a. Makeup für den Bildschirm	4
b. Das HTML-Grundgerüst: Wie beschreibt man eine Seite?.....	5
c. Als Objekt betrachtet – wie Elemente ansprechbar werden	7
B. Durch die (Blei-) Wüste: Keine Botschaft ohne Text	9
1. Immer nur Schwarz auf Weiß? Textgestaltung und Textfunktionen.....	9
a. Schriftart, Schriftschnitt und Typographie.....	9
b. Größe, Farbe, Font.....	10
c. Absätze, erzwungener Zeilenumbruch und Überschriften.....	11
d. Kodierung für den Rest der Welt – Alphabete jenseits des »mainstream«.....	11
e. No more missing links. Linkverweise, Anker, Sprungmarken und E-Mailing....	12
2. Listencheck.....	14
a. Römischer oder arabischer Stil? Nummerierte Listen	14
b. Scheibe um Scheibe, Punkt um Punkt. Graphische Listengestaltung.....	15
c. Memory in 3D: HTML verschachtelt Listen	16
C. Importschlager. An der Schwelle zum Multimedia-Overkill.....	19
1. Nur zur Illustration?.....	19
a. Mach Dir ein Bild. Aber wie?.....	19
b. Importeure und andere Schmuggler. Referenzierung und Urheberrecht.....	20
2. Grafik am Steuer und Multimedia-Overkill	20
a. Link(s)gerichtete Graphiken. Ganz- und Bereichsverweise.....	20
b. Techno unplugged – Animationen und Klangkulissen.....	21
D. Ins Netz gegangen: Screendesign mit HTML.....	22
1. Haftzellen für Webseiten: Layout mit Tabellen.....	22
a. Zeilen, Spalten, Zellen: Die Tabellenhausordnung.....	22
b. Im Tabellenknast – Komplexe Tabellenstrukturen.....	25
2. Formeller Umgang: Datenübergabe an Andere.....	26
a. Die richtige Dienststelle. Adressierung von Formulardaten.....	26
b. Worum geht's? - Keine Eingabe ohne Benennung!.....	28
c. Formulare im Gitternetz. Wie bringe ich das »form« in Form?.....	30
3. Alles im Rahmen. Kleine Kompositionslehre.....	31
a. Rahmenlose Aufhänger – Organisationstalent »Frameset«.....	31
b. Pro und Contra Frames. Argumente für die Abwägung.....	34
E. Ausblick.....	34
F. Auswahl von Tags mit ihren wichtigsten Attributen.....	35

A. Wie sag ich's meinem Browser?

1. Wenn Daten sichtbar werden

a. Makeup für den Bildschirm

Ursprünglich zielten Internetseiten weniger auf das Auge des Benutzers. Das weltweite Rechnernetz (WWW – *World Wide Web*) sollte den wissenschaftlichen und militärischen Datenaustausch erleichtern: Rechner mit verschiedenen Betriebssystemen konnten ein und dieselbe Datenquelle gleichzeitig lesen und interpretieren. Das gehört bis heute zu den Grundlagen des Internet.

Eine internationale Organisation, das W3-Konsortium (W3C), gibt die Internet-Standards vor. An diese Vorgaben müssen sich alle Hersteller von Webseitendarstellungsprogrammen (*Webbrowser* oder einfach *Browser*) halten. Wenigstens sollten sie es.

Die meisten Regeln des W3-Konsortiums finden tatsächlich allgemeine Beachtung. Die Hersteller von Browsern – am bekanntesten sind Microsoft und Netscape – kochen für »Ihre« Nutzer aber gerne nach eigener Rezeptur. Für die eigene Klientel gibt es besondere Darstellungsmöglichkeiten, beispielsweise blinkende oder „glühende“ Schrift.

Solche Gestaltungsmittel sind typographisch gesehen von zweifelhafter Natur. Vor allem aber haben Benutzer anderer Browser nichts von ihnen. Und es gibt weit mehr verbreitete Browserprogramme als die bekannten Flaggschiffe! Entwickler von Webseiten vergessen auch gerne, dass vielfach noch ältere Versionen der „großen“ Browser eingesetzt werden.

Was verbirgt sich hinter einer Internetadresse? Zunächst mal eine einfache Datei mit der Endung 'htm' oder 'html', mit enthaltenem Text und mit eingebauten Befehlen. Lassen Sie sich auch nicht durcheinanderbringen, wenn in der Adresszeile des Browsers plötzlich 'asp', 'php' oder 'cgi' und ein schier unendlicher Textschweif zu sehen sind. Hinter dem, was Sie gerade ansehen, steht auf jeden Fall eine HTML-Textseite.

Alle Befehle, die sogenannten *Tags*, richten sich an den Browser. Die Gesamtheit dieser Browseranweisungen bildet die Sprache HTML.

Bilder, die Sie auf den meisten Internetseiten finden, sind nicht etwa in der HTML-Datei selbst „verschlüsselt“. Im Grunde weist ein Tag-Befehl das Browserprogramm an, eine Bilddatei von einem (entfernten) Rechner abzurufen und an einer festgelegten Stelle einzublenden. Man spricht hier von Referenzierung. Sie werden damit noch Bekanntschaft machen – und sehen, dass es gar nicht kompliziert ist.

Obwohl Sie mit HTML Befehle geben können, ist es doch keine Programmiersprache. Eine echte Programmiersprache gibt Befehle an den Computerprozessor. HTML wendet sich dagegen nur an eine einzelne Programmanwendung: den Browser.

HTML ist eine sehr einfache Seitenbeschreibungssprache. Das Browserprogramm bekommt folgende Anweisungen:

- 1. Stell den folgenden Text bitte auf dem Bildschirm dar*
- 2. Ordne alles in einem bestimmten Layout an*
- 3. Blende an festgelegten Stellen Grafiken/Video/Hörmedien ein*
- 4. Setze bestimmte Textpassage mit besonderen Eigenschaften (etwa Farbe) aus*

b. Das HTML-Grundgerüst: Wie beschreibt man eine Seite?

Ein Browser kann stur sein. Wenn er schon Befehle entgegennimmt, dann müssen Sie ihm auch jede Kleinigkeit ausdrücklich befehlen. HTML-Dateien haben einen fest geregelten Bau. Dabei sind leichte Anlehnungen an die menschliche Anatomie sicher nicht zufällig.

Jetzt lernen Sie die ersten HTML-Befehle kennen! Jeder HTML-Befehl ist in Winkelklammern < > eingeschlossen. Nur so erkennt der Browser eine Passage als Anweisung.

Obenan in der HTML-Datei steht ein Kopf. Und der hat sogar ein Gesicht, das etwas über den Textinhalt der Seite sagt, den *title*:

```
<html>
<head>
<title>Eine Zeile zur Inhaltsangabe</title>
...
</head>
```

Alle HTML-Befehle sind englisch, meistens Abkürzungen. Die ältesten Wurzeln der Auszeichnungssprache liegen zwar in der Schweiz. Aber auch am CERN dachte man wohl schon international und hielt sich an die Lingua Franca der Computerbenutzer.

Der HEAD-Befehl und der TITLE-Befehl kommen doppelt vor: Einmal ohne '/' und einmal mit '/'. Die meisten Tags sind so aufgebaut: Sie klammern das ein, was sie mit Eigenschaften versehen. Die Anweisung an den Browser „Achtung! Jetzt kommt etwas zum Auszeichnen“ lautet <tag>, die Anweisung „Hier bitte mit der Auszeichnung aufhören“ heißt </tag> (sprich: *end of tag* oder deutsch: tag-Ende).

Einen geklammerten TITLE braucht jede HTML-Seite. Hier wird angegeben, was ganz oben im Browserfenster erscheint. Außerdem interessieren sich Suchmaschinen besonders für dieses „Gesicht“ des Webdokuments.

Nicht nur für Suchmaschinen können im Dateikopf aber noch mehr Angaben gemacht werden. Eine vernünftige HTML-Seite verfügt über die sogenannten META-Tags.

META-Tags bilden keine Klammern. Sie stehen einfach im Dateikopf und werden nicht geschlossen. Sie machen Aussage über den Dateiinhalt und richten sich teilweise an Suchmaschinen, teilweise an den Browser.

Teilweise haben Meta-Befehle eine eigenartige Syntax. An einem einfachen und sehr verbreiteten Meta-Befehl können Sie aber etwas beobachten, was es in vielen „normalen“ Tags auch gibt: die Attribute.

```
<html>
<head>
<title>Thema des Dokuments</title>
<meta name="date" content="2001-06-01">
<meta ...>
</head>
```

Das TITLE- und das HEAD-Tag sprechen für sich selber: Ein Kopf an sich und ein Thema sind – nach Computergeschmack – bereits Eigenschaft genug. Es gibt nur eine Art von Kopf, nämlich den Dateikopf – und nur eine Themenbezeichnung pro Datei.

Von META-Befehlen gibt es aber mehrere Sorten. Und deswegen muss – in diesem

Fall einer Suchmaschine – mitgeteilt werden, um welche Art von META-Tag es hier geht.

Viele META-Tags haben richtige Namen als Merkmal. Deswegen kennt die Anweisung „Achtung! Jetzt kommt ein META-Befehl!“ das Attribut 'name'. Ein möglicher Name für eine META-Anweisung ist 'date', das Datum also. Jedem Attribut eines Befehls (tags) wird ein Wert zugewiesen. Der Wert kann, je nach Zusammenhang, eine Zahl, eine Prozentangabe oder eine Buchstabenfolge sein.

Suchmaschine, die auf den oben genannten Dateikopf stoßen, wissen jetzt schon, dass der META-Befehl etwas über das Erstellungsdatum der Datei aussagt. Mit dem Attribut 'content' verrät das META-Tag hier nun das Jahr, den Monat und den Tag der letzten Dateispeicherung. Ein META-Befehl mit dem Attribut name="author" übergibt als Wert des Attributs 'content' den Namen des Dateiverfassers.

Mit der beinahe schon unüberschaubaren Menge von META-Tags sollten Sie sich erst beschäftigen, wenn Sie in HTML schon firm sind. Das sollte in etwa zwei Wochen der Fall sein. Wertvolle Hinweise finden Sie, wie immer in HTML-Fragen, in der großen Referenz »SelfHTML« von Stefan Münz (<http://www.teamone.de/selfhtml> – dort auch zum kostenlosen Download). Die META-Tags werden in der Datei 'tcbc.htm' behandelt.

Welche Tags sonst noch im Dateikopf stehen können, dürfen Sie an dieser Stelle vernachlässigen. Das W3-Konsortium empfiehlt, dem Dateikopf noch eine Mütze überzustülpen. Die Zeile `<!doctype html public "-//W3C//DTD HTML 4.0 //EN">` wird noch vor dem öffnenden HTML-Tag gesetzt. Sie zeigt dem Browser an, an welcher HTML-Version (hier ist es die Version 4.0) sich die Datei richtet. Bald dürfte diese Zeile fester Bestandteil jeder HTML-Datei werden, bisher wird sie aber von keinem Browser vorausgesetzt.

Was dagegen keiner HTML-Datei fehlen darf ist der Mittelteil und der Dateifuß. Bei über 90% aller HTML-Seiten wird der Mittelteil mit der Body-Tagklammer eingefasst:

```
<html>
<head>
  <title>Thema der Seite</title>
</head>
<body>
Hier steht der eigentliche Text der Seite mit allen Elementen.
</body>
```

Selbstredend kann das BODY-Tag mit einer ganzen Reihe von Attributen versehen werden. Einen großen Teil der möglichen BODY-Attribute werden Sie bald kennenlernen. Was Sie sich aber jetzt schon merken sollten: Das BODY-Tag kann Attribute bekommen, braucht sie aber nicht unbedingt. Eine ganze Reihe von Tags benötigen unbedingt Attribute, um überhaupt eine Wirkung zu haben!

Nachdem Kopf und Körper der HTML-Datei eingegrenzt sind, fehlt nur noch der Dateifuß. Die meisten modernen Browser akzeptieren HTML-Seiten auch dann, wenn der Fuß ganz fehlt. Machen Sie sich aber trotzdem die kleine Mühe, nach dem End-of-Body-Tag noch den allerersten Tag der Datei mit einem Endpendant zu versehen:

```
</html>
```

```
<head>
  <title>Thema der Seite</title>
</head>
<body>
...
</body>
</html>
```

Voilà, das ist zugleich das Grundgerüst einer jeden HTML-Datei. Es kann nicht schaden, wenn Sie dieses Schema ein paar Mal auf Papier und Bildschirm aufzeichnen! So geht Ihnen die Grundstruktur ganz leicht in Fleisch und Blut über.

c. Als Objekt betrachtet – wie Elemente ansprechbar werden

Als Websurfer wissen Sie: Eine HTML-Seite lässt sich meistens nicht auf einem einzelnen DIN A4-Blatt ausdrucken. Die Seitenlänge auf dem Bildschirm ist variabel, zum Weiterlesen gibt es einen Scrollbalken oder einen „Weiter“-Link.

Es zeugt von gutem Schreibstil, wenn Sie Ihre Texte so kurz halten, dass ein Scrollbalken unnötig wird. Aber auch sonst brauchen Sie sich keine Sorge um den Scrollbalken zu machen. Der Browser setzt ihn ganz von selbst, wenn eine Seite mehr Text enthält, als auf den Bildschirm passt.

Alles, was Ihnen eine einzelne HTML-Datei auf den Bildschirm zaubert, lässt sich als ein Dokument betrachten – ganz ähnlich wie eine Word-Datei. Schließlich umfasst ein Word-Dokument meistens auch mehrere Seiten, auf denen vielleicht Bilder, Tabellen und anderes vorkommen.

Wie gesagt: Im Quelltext einer HTML-Datei finden Sie nirgendwo sichtbare Bilder. Grafische und klingende Elemente werden grundsätzlich referenziert: der Browser bekommt per Tag den Befehl, eine Grafik- oder Audiodatei einzublenden. Mit diesen „Importwaren“ werden Sie sich im Abschnitt C. (ab Seite 19) beschäftigen.

Was für Elemente gibt es aber noch in der Textverarbeitung wie auf Webseiten?

- Textpassagen: Absätze und Zeichen
- Listen: so wie hier
- Tabellen: sie spielen auf Webseiten eine enorm wichtige Rolle!

Bei Webseiten kann noch eine vierte Art von Elementen vorkommen:

- Formular: dient der Datensammlung und Übermittlung.

In der folgenden Tabelle (sic!) finden Sie alle Elemente, die auf Webseiten vorkommen können.

HTML-Strukturelemente	Beschreibung ab Seite	Importfähige Elemente	Beschreibung ab Seite
Textpassagen (auch Links)	9	Bilder/Grafiken (auch Buttons)	19
Listen	14	Audio	21
Tabellen	22	Video / Animationen	21
Formulare	26	Bereiche des Browserfensters	31

Viele Elemente von Webseiten kann man als Objekte bezeichnen: Damit ist der Brückenschlag in den Bereich der Programmierung gemacht.

Tatsächlich gibt es zwei waschechte und weit verbreitete Programmiersprachen, die Daten aus HTML-Seiten verarbeiten. Die Sprache Perl wird sehr oft für sogenannte CGI-Scripts (*Common Gateway Interface*) verwendet. Mit diesen Scripts (kurze Programmen) lassen sich etwa aus Benutzereingaben dynamisch HTML-Dokumente oder E-Mails erzeugen.

Oft findet sich im Kopf einer HTML-Datei folgende Tag-Klammer:

```
<script language="javascript">
```

```
...
```

```
</script>
```

JavaScript ist eine „kleine“ Programmiersprache. Das heißt: sie ist nicht sehr komplex. Ihre Möglichkeiten sind längst nicht so groß wie etwa die von Perl oder von der „Muttersprache“ Java. Mit Java verbinden JavaScript einige Ähnlichkeiten in der Befehlssyntax. Ansonsten sind die beiden völlig verschieden nach Aufbau und Leistungsfähigkeit.

Javascript zeigt einige Ansätze zum objektorientierten Programmierungsmodell OOP (*Object Orientated Programming*). In echten OOP-Sprachen sind Objekte bestimmbar Daten, denen besondere Eigenschaften und Verarbeitungsmöglichkeiten (Methoden) zugeordnet werden. Wird auf einer Webseite JavaScript verwendet, so sieht sich das Programm unaufgefordert in der Datei nach Objekten um, die JavaScript kennt. Solche Objekte sind etwa Formulare und Formularfelder, Tabellen und Tabellenzellen. Für alle diese Elemente kann das entsprechende Tag mit einem Attribut name="" versehen werden. Alle Objekte, die JavaScript erkennt, werden in Datencontainern registriert. Sie verschwinden dadurch nicht aus der HTML-Datei. Die Datencontainer samt Inhalt bleiben vollkommen unsichtbar. Wenn ein Element über das Tag-Attribut name="" gekennzeichnet wurde, wird es mit diesem Namen im Datencontainer registriert. Ansonsten nummeriert JavaScript die Elemente eines Objekttyps einfach durch, bei 0 beginnend.

In HTML-Dateien mit eingebundenem JavaScript sind fast alle Elemente gezählt und benannt. Und zwar als Objekte in einer festgelegten Hierarchie. Von allen so registrierten Objekten einer Webseite können Script-Programme Werte auslesen. Oft lassen sich die Inhalte von Bereichen sogar beeinflussen. Dann ändert sich auch der Quelltext einer HTML-Datei – allerdings nur im Arbeitsspeicher des Rechners. Zu den Objektprinzipien

lesen Sie mehr im Heft zum Kurs »JavaScript und CGI in der praktischen Anwendung«.

B. Durch die (Blei-) Wüste: Keine Botschaft ohne Text

1. Immer nur Schwarz auf Weiß? Textgestaltung und Textfunktionen

a. Schriftart, Schriftschnitt und Typographie

Der Begriff »Bleiwüste« stammt aus der Setzer- und Druckersprache. Als Illustrationen satztechnisch noch sehr kompliziert waren, mussten Drucktexte mit anderen Mitteln aufgelockert werden. Dass der Einsatz von möglichst vielen verschiedenen Schriftarten zur blanken Katastrophe führte, stellten die professionellen Setzer schon vor gut 500 Jahren fest.

Im Internet ist der Einsatz von schmückenden oder illustrierenden Bildelementen technisch kein Problem. Trotzdem ist hier Zurückhaltung dringend geboten: Zum einen sollte sich eine Webseite nicht aus lauter Blickfängern zusammensetzen, die sich ihre Beute gegenseitig streitig machen. Zum anderen verbrauchen Bilder, die ja vom Browser noch „abgeholt“ und eingeblendet werden müssen, viel Datenleitungskapazität.

Was für den klassischen Drucksatz gilt, gilt ebenso für Webseiten. Vermeiden Sie den Einsatz von mehr als zwei Schriftarten. Schriftarten sind etwa 'Times New Roman' oder 'Arial'.

Man unterscheidet auch im Web zwischen Serifenschriften und serifenlosen Schriften. Dieses Skript ist in einer Serifenschrift gesetzt, der Schriftart Garamond. An den unteren und gelegentlich oberen Enden der meisten Buchstaben können Sie kleine Schnörkel erkennen, die in der Fachsprache Serifen heißen.

Serifenlose Schriftarten zeigen nur den nackten Buchstaben, ohne Verzierung.

Serifenschrift „Times New Roman“	Serifenlose Schrift „Arial“
A T R Z	A T R Z
a t r z	a t r z

Für jede Schriftart gibt es eine Reihe von Schriftschnitten. Die bekanntesten sind:

- (halb)fett
- regulär
- kursiv
- fett-kursiv

Zuletzt macht die Größe die Schrift mit aus. Im Druck wird in Didot- oder Pica-Punkt gemessen, im Web in Pixeln (Bildschirmbildrasterpunkten). HTML unterscheidet selbst nur fünf relative Schriftgrößen: 3 , 4 , 5 , 6 und 7. Wenn Sie nichts anderes angeben, setzt der Browser den Lauftext in der Größe 3. Über die physikalisch-konkrete Darstellungsgröße entscheidet der Benutzer mit der Browservoreinstellung.

Mit Hilfe von Stylesheets lassen sich auch Schriftgrößen nach Pixel-Punkt definieren.

Zum Thema Stylesheets gibt es einen eigenen Kurs in der Journalistenakademie.

Eine typographische Grundregel besagt, dass auf einer Seite nicht mehr als fünf verschiedene Schriften vorkommen sollten. Zum Beispiel wäre demnach folgender Schriftsatz gerade noch denkbar:

- eine halbfette Arial in Größe 6 für die Kapitelüberschrift,
- eine reguläre Arial in Größe 4 für Zwischenüberschriften,
- drei webtaugliche Times New Roman-Schnitte in Größe 3 für den Lauftext.

Ein weiteres Schriftgestaltungsmittel hat das Web aus dem modernen Zeitschriftensatz übernommen und zu zweifelhafter Blüte gebracht. Schriftpassagen lassen sich mit HTML-Befehlen ohne Schwierigkeiten einfärben.

b. Größe, Farbe, Font

Für alle drei Gestaltungsmöglichkeiten sind Attribute des FONT-Tags zuständig:

```
<font size=" " color=" " face=" ">...</font>
```

Das FONT-Tag braucht notwendig mindestens ein Attribut, um Wirkung zu zeigen. Es muss mit einem End-of-Tag abgeschlossen werden.

Dem Attribut 'size' können Zahlen zwischen 3 und 7 zugeordnet werden. Der Wert 3 ist Standard, 7 ist die größtmögliche Schrift.

Zum Attribut 'color' gehören mit '#' beginnende hexadezimale Angaben (etwa #FFFFFF für 'weiß') oder englische Farbnamen. Einen Überblick zum Thema 'webtaugliche Farben' gibt Stefan Münz in der Datei 'tcae.htm'.

Das Attribut 'face' nimmt Schriftart-Namen auf. Dabei ist zu beachten, dass Rechner verschiedener Betriebssysteme mit verschiedenen Schriftarten arbeiten. Standardschriftart für einen Windows-Browser ist 'Times New Roman', für die Macintosh-Version 'Times'. Jeder Browser kann nur Schriften darstellen, die auf dem Benutzercomputer installiert sind. Mit einer 'Times' können die wenigsten Windows-Systeme etwas anfangen, mit einer 'Times New Roman' kommt so mancher Apple-Fan ins Schwitzen. Statt der auf Windows-Rechnern sehr beliebten Arial ist auf Mac-Computern standardmäßig Helvetica als serifenlose Schriftart vorhanden. Dafür fehlt sie auf den meisten Windows-Rechnern.

Wer sichergehen möchte, dass eine Textpassage auf jedem System mit einer serifenlosen Schrift dargestellt wird, kann dem Browser über das face-Attribut eine Auswahl anbieten:

```
<font face="Arial, Helvetica">...</font>
```

Für eine Serifenschrift lohnt sich ein solcher Aufwand nicht. Standardmäßig ist bei nahezu jedem Browser eine Serifenschrift eingestellt. Bei ausgefallenen Typen, etwa der Garamond oder einer Schreibschrift, dürfen Sie sicher davon ausgehen, dass mindestens zwei Drittel Ihrer Sitegäste bloß eine normale Serifenschrift zu sehen bekommen.

c. Absätze, erzwungener Zeilenumbruch und Überschriften

Absätze werden unter HTML immer mit einer Leerzeile am Anfang versehen. Das „zuständige“ Tag ist `<p>`. Dieses Tag darf geschlossen werden, braucht es aber (noch) nicht unbedingt.

Eine neue Zeile erzwingt man mit dem Tag `
`. Das BR-Tag wird niemals geschlossen!

In HTML sind sechs Typen von Überschriften vordefiniert. Sie erzwingen alle eine Leerzeile über und eine Leerzeile unter sich. Sie unterscheiden sich sonst nur in der Größe und im Schriftschnitt. Wie sie genau aussehen, ist in den Voreinstellungen des Browsers eingestellt und somit fast von Rechner zu Rechner verschieden.

Die Überschriften-Tags sind alle paarig. Die größte eingestellte Überschrift erreichen Sie mit der Tag-Klammer

`<h1>Überschrift-Text</h1>`,

die kleinste mit

`<h6>Überschrift-Text</h6>`.

d. Kodierung für den Rest der Welt – Alphabete jenseits des »mainstream«

Aber was ist mit fremden Alphabeten? Was tun, wenn etwa eine russische, griechische oder polnische Seite ins Web soll?

Der Weg führt hier nicht über das FONT-Tag, sondern über einen Meta-Befehl. Alle Schriftsätze für das Web fußen bislang auf der ASCII-Zeichentabelle. In dieser Tabelle haben genau 256 Zeichen Platz. Die ersten 128 Tabellenplätze sind mit Buchstaben, Zahlen und Sonderzeichen belegt, die von amerikanischen Internetnutzern regelmäßig benötigt werden. Die Belegung der übrigen 128 Tabellenplätze unterscheidet sich von Zeichentabelle zu Zeichentabelle. Glücklicherweise sind alle Zeichentabellen benannt.

Das erwähnte META-Tag verrät dem Browser, welche Zeichentabelle er für den Text im HTML-Dokument verwenden soll.

`<meta http-equiv="document-type" content="text/html; charset=Windows-1253">`

Damit der Browser die Sonderzeichen aus der griechischen Tabelle Windows-1253 darstellen kann, muss natürlich eine Schriftart installiert sein, die griechische Buchstaben auf den letzten 128 Tabellenplätzen hat. Im Zweifelsfall können solche Schriften aus dem Internet heruntergeladen werden. Der Browser muss allerdings selbst auch noch mitspielen: Nicht jeder versteht das genannte META-Tag!

Die griechische Tabelle Windows-1253 ist sichtlich für Microsoft-Systeme gemacht. Was machen aber Mac-Besitzer und Linux-Freunde? Entweder ihre Browser kennen diese Tabellen und entsprechende Schriftarten sind vorhanden – oder der Webautor hat freundlicherweise eine andere Kodierung verwendet. Es gibt beispielsweise zwei griechische, drei japanische und vier kyrillische Zeichentabellen. Mit jeder einzelnen kann man sicher sein, einen nennenswerten Anteil des Webpublikums nicht zu erreichen.

In Zukunft sollen alle Browser Unicode unterstützen. Das ist ein System, in dem alle (!) weltweit benötigten Alphabete und Zeichensysteme vertreten sind. Was das Unternehmen aufhält: Wie sollen etwa die allgemein anerkannten Schnitte für ägyptische Hieroglyphen aussehen?

Denken Sie bitte daran: Für unsere Freunde in aller Welt wirken auch die deutschen Umlaute 'ä', 'ö', 'ü' und das 'ß' wie Letter aus einer anderen Welt. Deswegen sollten Sie Ihre deutschsprachigen Seiten mit folgendem META-Tag ausstatten:

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

Das kann auf keinen Fall schaden. Und dann sollten Sie die besagten Sonderzeichen sicherheitshalber auch noch kodieren.

Viele Editor-Programme nehmen Ihnen diese Arbeit ab. Sie sollten aber trotzdem die Codes für die wichtigsten Sonderzeichen kennen.

Buchstabe	HTML-Code	Sonderzeichen	HTML-Code
Ä	Ä	“	"
ä	ä	»	»
Ö	Ö	«	«
ö	ö	<	<
Ü	Ü	>	>
ü	ü	&	&
ß	ß	§	§

Besonders wichtig ist noch das geschützte Leerzeichen. Wenn zwischen Wörtern etwa zwei oder mehr normale Leerzeichen (blanks) stehen, interpretieren das alle Browser als nur ein Leerzeichen. Will man bestimmte Abstände zwischen Wörtern oder Grafiken auf der horizontalen Achse erzwingen, braucht man geschützte Leerzeichen. Diese Leerzeichen lassen sich in beliebiger Zahl aneinanderreihen. Der Code ist etwas seltsam, Sie sollten ihn sich aber dringend merken: ** **

e. No more missing links. Linkverweise, Anker, Sprungmarken und E-Mailing

Es ist wohl der größte Clou in HTML: Man „klickt“ auf farblich hervorgehobenen Text oder einen „Button“, und schon landet man auf einer anderen Seite im World Wide Web. Sogar bestimmte Passagen auf beliebigen Seiten lassen sich direkt auf diese Weise erreichen.

Um eine Textpassage zum Link zu machen, brauchen Sie das A-Tag:

`...`

Das A-Tag braucht notwendig mindestens ein Attribut und muss geschlossen werden. Das Attribut href=" " " gibt das Linkziel an. Verweist der Link zu einer Seite außerhalb des eigenen Webangebots spricht man – logisch nicht ganz korrekt – von einem externen Link. Der Wert des href-Attributes muss dann eine komplett ausgeschriebene Webadresse sein, also etwa

`Link zu der Homepage der Journalistenakademie`

Bei HTML-Dokumenten innerhalb des eigenen Angebots genügt als Attributswert der Dateiname, wenn nötig mit Pfad:

```
<a href="index.html">Link zurück zur Startseite</a>
```

```
<a href="/texte/beitrag1.htm">Link zum Beitrag 1 im Unterverzeichnis 'texte'</a>
```

Das A-Tag kann aber nicht nur Links bestimmen! Mit einem anderen Attribut setzt es einen unsichtbares Sprungziel (*anchor*) in einer HTML-Datei, auf das sich ein Link ausrichten lässt.

Zum Beispiel wird in einem zoologischen Fachartikel über allerlei Getier und auch Ameisen geschrieben. Ein beliebiges Wort am Anfang des Ameisen-Abschnitts kann zum Sprungziel werden.

Der kleine Bär ist nur scheinbar ein harmloses Tier. In Wirklichkeit gibt es kein Wesen, das gefährlicher wäre. Denn wo das Junge ist, ist Mutter Bär ganz bestimmt nicht weit.

```
<a name="ameise">Ameisen</a> sind  
possierliche und freundliche Tiere,  
solange man sich nicht auf Ihr  
Nest setzt. ...
```

Nehmen wir an, diese Passage über Bären und Ameisen fände sich in der Datei "tiere.htm" des Angebots <http://www.dr-grzimek-lebt.tv>. Von einer Seite 'uebersicht.htm' dieses Webangebots aus wird auf die Datei als solche verwiesen.

```
<a href="tiere.htm">Link zum Tiereartikel</a>
```

Sie aber sind beim eifrigen Quellcode-Studium auf den benannten Anker (*named anchor*) `` gestoßen und wollen Ihren Link ausdrücklich zur Ameisenpassage legen. Das sieht dann so aus:

```
<a href="http://www.dr-grzimek-lebt.tv/tiere.htm#ameise">Link zur  
Ameisenpassage</a>
```

Links zu benannten Ankern sind auch innerhalb einer Datei möglich! Vielleicht haben Sie an die zehn DIN A4-Seiten in einer HTML-Datei verpackt. So viel Text sollten Sie nicht in ein Dokument packen, aber es kann vorkommen. Um den Leser nicht zu überfordern, stellen Sie an den Anfang der Seite eine Sammlung aller Zwischenüberschriften. Und diese Überschriften sind – natürlich! - Sprungmarken mit benannten Ankern als Zielen. Ein benannter Anker selbst ist für den Benutzer unsichtbar. Wird er von einem Link oder einer Sprungmarke „angepeilt“, beginnt die Bildschirmseite mit der im Quelltext ausgezeichneten Stelle.

Eine Sprungmarke selbst ist aber ebenso gefärbt wie ein ganz normaler Link, der auf eine andere Webseite verweist.

Die Syntax einer Sprungmarke sieht der eines Links nicht zufällig zum Verwechseln

ähnlich:

```
<a href="#benannteranker">Seiteninterner Link – alias Sprungmarke</a>
```

Achten Sie darauf, dass die Raute # im 'name'-Attribut eines A-Tags nichts verloren hat! Sie kann nur in einem 'href'-Attribut vorkommen.

Neben Linkadressen und Sprungmarken kann das 'href'-Attribut noch eine dritte Art von Link aufnehmen: Eine E-Mailadresse.

In HTML ist dafür der mailto-Befehl vorgesehen:

```
<a href="mailto:blossnichtso@hateemel.de">Mail an blossomnichtso@hateemel.de</a>
```

Auf nahezu jeder guten Website finden Sie solche Mail-Links. Diese Methode der Kontakterleichterung hat aber einen großen Nachteil: Sie funktioniert nur, wenn der Surfer ein funktionierendes E-Mail-Programm an seinen Browser gekoppelt hat. Davon kann man aber leider nicht ausgehen. Die Konfiguration eines Mailprogramms gehört zu den Übungen für Fortgeschrittene. Besser ist es, den Seitengästen ein eigenes Feedback-Formular anzubieten und dieses mit einem CGI-Programm auf dem Server zu verbinden. Für alle Surfer ist der Umgang mit einem solchen Formular angenehm unproblematisch. Und Sie bekommen damit auch keine Schwierigkeiten mehr, nachdem Sie das Skript »JavaScript und CGI in der praktischen Anwendung« gelesen haben.

2. Listencheck

a. Römischer oder arabischer Stil? Nummerierte Listen

Als die Webtexte laufen lernten, wurden sie zum Laufftext. Um den Leser bei der Stange zu halten, musste und muss der Laufftext irgendwie gegliedert werden. Überschriften und Absätze sind eine Möglichkeit. Gerade für Inhalte aber, die man kurz und übersichtlich fassen möchte, bieten sich Aufzählungen in Listen an.

HTML hat vieles aus seiner naturwissenschaftlichen und militärischen Zeit bewahrt. Da verwundert es kaum, dass man Listeneinträge auch automatisch vom Browser durchnummerieren lassen kann.

Nummerierte Listen werden, unabhängig von der Zahl ihrer Einträge, vom Geordneten-Listen-Tag `...` eingeschlossen. Die einzelnen Einträge werden durch das Tag `` eingeleitet und sollten mit `` geschlossen werden. Das LI-Tag darf auch nicht-paarig verwendet werden. Wenn HTML aber künftig mit der Spracherweiterung XML kombiniert wird, führen ungeschlossene LI-Tags zur sicheren Katastrophe.

```
<ol>
  <li>Eintrag A</li>
  <li>Eintrag B</li>
  <li>Eintrag C</li>
</ol>
```

sieht auf dem Bildschirm so aus:

1. Eintrag A
2. Eintrag B

3. Eintrag C

Was aber, wenn Sie keine arabische Zählung, sondern eine lateinische haben wollen? Zweifeln Sie nicht an den Lateinkenntnissen der Interneteltern! Auch für das OL-Tag gibt es schließlich Attribute:

<code><ol type="I">...</code>	= lateinische Zählung
<code><ol type="i">...</code>	= lateinische Zählung mit Kleinbuchstaben (Gemeinen)
<code><ol type="A">...</code>	= „Zählung“ mit Großbuchstaben (Versalien)
<code><ol type="a">...</code>	= Auflistung mit Kleinbuchstaben

Dem OL-Attribut 'type' begegnen wir gleich wieder, wenn wir das UL-Tag für nicht-nummerierte Listen genauer ansehen.

b. Scheibe um Scheibe, Punkt um Punkt. Graphische Listengestaltung

Listen müssen nicht immer nummerierte Einträge haben. Aus gedruckten Medien kennt man auch den »dicken Punkt« oder einfache Striche links neben der jeweiligen Textzeile.

Es steht Ihnen frei, Ihren HTML-Text mit festen Zeilenumbrüchen auszustatten (das geht mit dem BR-Tag) und einfach jede Zeile mit einem Zeichen Ihrer Wahl anzufangen:

```
<br>- 1. Eintrag  
<br>- 2. Eintrag  
<br>- 3. Eintrag  
...
```

Die Einträge werden dann aber nicht so schön eingerückt wie bei einer ordentlichen HTML-Liste. Das fällt gerade dann auf, wenn Listeneinträge mehrzeilig sind.

Auch bei nicht-nummerierten Listen werden die Einträge mit dem LI-Tag eingeklammert. Diese Ähnlichkeit mit dem OL-Tag findet einen weiteren Niederschlag im Tag selbst: Auch das UL-Tag kennt das type-Attribut.

Für nicht-nummerierte Listen haben Sie drei mögliche Eintragskennzeichnungen, die Sie mit dem type-Attribut angeben:

<code><ul type="square">...</code>	=Aufzählung mit schwarzen Quadratflächen n
<code><ul type="disc">...</code>	=Aufzählung mit schwarzen Kreisflächen l
<code><ul type="circle">...</code>	=Aufzählung mit leeren Kreislinien ;

Wenn Sie im UL-Tag kein type-Attribut angeben, beginnt der Browser die Listeneinträge mit dem *Circle*-Muster. Im Stylesheet-Kurs werden Sie eine Möglichkeit kennenlernen, die schwarzen Quadrate und Kreise etwas schöner einzufärben.

c. Memory in 3D: HTML verschachtelt Listen

Stellen Sie sich eine Liste vor, deren Einträge noch Unterpunkte haben. Und die Unterpunkte leiten vielleicht noch eine weitere Aufzählung ein. Dank HTML ist das alles überhaupt kein Problem – weil HTML wunderbar verschachteln kann. Sie müssen nur darauf achten, dass Sie Ihre Orientierung nicht verlieren.

Es gibt übrigens ein eigenwilliges Tag für Anmerkungen und Kommentare. Kommentare können beliebig lang sein und machen den Quelltext leichter zu durchschauen, ohne vom Browser angezeigt zu werden. Das Besondere am Kommentar-Tag: es gibt keine Attribute, dafür liegt der ganze Inhalt innerhalb der Tag-Klammern:

<!-- Kommentar oder Anmerkung -->

Wenn Sie Listen schachteln, verwenden Sie getrost solche Kommentare. Andernfalls werden Sie sich bei einem Neuaufbau Ihrer Seiten nach ein paar Monaten nicht mehr zurechtfinden.


Nun zu einem ersten Beispiel. Folgende Listenschachtelung sollen Sie in HTML abbilden:

- Zoologie
 - 1. Bären
 - Braunbären
 - Eisbären
 - 2. Insekten
 - Ameisen
 - Fliegen
 - Mücken
- Botanik

Mit Biologie hat diese Listenschachtelung nun wirklich nichts zu tun. Aber sehr wohl mit HTML.

Zählen wir zuerst die Ebenen dieser komplexen Liste. 'Zoologie' und 'Botanik' sind gleichrangig in einer nicht-nummerierten Liste vom Square-Typ organisiert:

```
<ul type="square">  
  <li>Zoologie</li>  
  <li>Botanik</li>  
</ul>
```



Wenn Sie Listen verschachteln, dürfen Sie sehr grafisch vorgehen. In HTML gibt es auch schachtelbare Elemente, bei denen Sie zuerst einen genauen und logisch durchdachten Plan ausarbeiten müssen. Das gilt insbesondere für komplexe Tabellen (siehe Kapitel »Im Tabellenknast – Komplexe Tabellenstrukturen« ab Seite 25).

Bis jetzt haben wir es aber noch nicht allzu schwer. Es gibt eine Liste, die unter Zoologie beginnt, aber noch vor dem zweiten Eintrag der umfassenden Liste endet: Die

Liste mit den zwei Einträgen 'Bären' und 'Insekten'. Diese Liste lassen wir schlicht unter 'Zoologie' beginnen und schließen sie vor 'Botanik' wieder ab:

```

<ul type="square">
  <li>Zoologie</li>
  <ol>
    <li>Bären</li>
    <li>Insekten</li>
  </ol>
  <li>Botanik</li>
</ul>

```

Jetzt fehlt nur noch eine Liste unter 'Bären' und eine unter 'Insekten'. Dass diese beiden sich ähneln sollen (indem beide UL-Listen vom Circle-Typ sind) liegt nur an unserem Ordnungssinn. So lange beide Unterlisten nach den Regeln abgeschlossen werden, sind sie nach HTML-Maßstäben unabhängig voneinander.

```

<ul type="square">
  <li>Zoologie</li>
  <ol>
    <li>Bären</li>
    <ul type="circle">
      <li>Braunbären</li>
      <li>Eisbären</li>
    </ul>
    <li>Insekten</li>
    <ul type="circle">
      <li>Fliegen</li>
      <li>Mücken</li>
    </ul>
  </ol>
  <li>Botanik</li>
</ul>

```

Und so können Sie nach Belieben weerschachteln. Denken Sie daran, die Listen in der richtigen, der symmetrischen Reihenfolge zu schließen! Eine Querschachtelung nach folgendem Schema ist in HTML grundsätzlich verboten und führt zu unvorhersehbaren Ergebnissen:

```

<ul><li>...</li><ol><li>...</li></ol></ul></ol>

```

Mit der symmetrischen Schachtelung sind Sie immer auf der sicheren Seite:

```
<ul><li>...</li><ol><li>...</li></ol><li>...</li></ul>
```

C. Importschlager. An der Schwelle zum Multimedia-Overkill

1. Nur zur Illustration?

a. Mach Dir ein Bild. Aber wie?

Grundsätzlich werden Grafiken nicht als Buchstabensalat in eine HTML-Datei geschrieben. Statt dessen räumt ein spezielles Tag Platz für eine Grafik ein und bildet dann eine Bilddatei ab, die auf irgendeinem Serverrechner liegt. Weil an sich nur ein Bezug hergestellt wird, spricht man von Referenzierung. Eine Kopie der Grafikdatei wandert allerdings in den Arbeitsspeicher des User-Rechners! Der HTML-Text verändert sich kein bisschen. Das Tag zur Grafikreferenzierung ist

```

```

Das Attribut 'src' ist notwendig. Bei Fremdservern wird hier die komplette URL angegeben, bis zum Dateinamen mit seiner Endung. Bei Grafiken auf dem eigenen Server genügt der Dateiname (mit Endung) und gegebenenfalls der Verzeichnispfad.

Eine ganze Reihe von weiteren Attributen ist möglich. Mit dem Attribut 'alt' tut man allen Webpuristen und Benutzern älterer Modems einen großen Gefallen.

Über die Attribute 'width' und 'height' kann für die Grafik ein Raum reserviert werden. Mit diesen Maßen lassen sich Grafiken auch nach dem Laden in den Arbeitsspeicher verkleinern oder vergrößern. Beides sollten Sie dringend vermeiden. Eine Verkleinerung kostet nur Rechenzeit zusätzlich zur Ladezeit der Grafik. Eine Vergrößerung von komprimierten Grafiken fällt immer „pixelig“ aus: Das Raster wird gröber und stärker sichtbar. Wenn Sie 'width' und 'height' als Attribute zum IMG-Tag verwenden, dann ordnen Sie als Maße bitte die Originalmaße der Grafik zu. Das spart sogar etwas Rechenzeit!

Mit zwei Arten von Grafikdateien können Browser etwas anfangen:

- JPG- alias JPEG-Dateien
- GIF-Dateien

Beide Formate zeichnen sich dadurch aus, dass die Bildinformation auf das notwendige Mindestmaß reduziert ist. Das JPG- und das GIF-Format sind komprimierende Formate.

Ein weiteres Dateiformat beginnt sich zu etablieren – das PNG-Format. Bisher können alle gängigen Browser dieses Bildformat nur über kleine Zusatzprogramme, sogenannte *plugins*, interpretieren und darstellen. Die nächste HTML-Version könnte das PNG-Format bereits als Standard-Webformat anerkennen.

Wenn Sie selbst Webgrafiken bereitstellen wollen, gehen Sie wahrscheinlich nicht von einem komprimierenden Format aus. Jedenfalls sollten Sie das nicht. Ein komprimierte Grafik lässt sich nur noch mit großem Qualitätsverlust in der Größe oder in der Farbigkeit beeinflussen. Um Grafikdateien aus nicht-komprimierenden Formaten (zum Beispiel BMP, TIF, PSD, CPT, CDR) in GIF- oder JPG-Dateien zu erzeugen, gibt es eine Vielzahl von Werkzeugen.

Große Grafikprogramme wie Photoshop oder Coreldraw enthalten diese Werkzeuge bereits. Ein kostenloser Grafikbetrachter und -konvertierer ist etwa Irfanview von Irfan Skiljan (<http://www.irfanview.de>).

b. Importeure und andere Schmuggler. Referenzierung und Urheberrecht

Da Grafiken von beliebigen Servern im WWW referenziert werden können, ohne dass die Eigentümer etwas davon merken, wackeln im Internet die Grundfesten des Urheberrechts. Das heisst aber nicht, dass dieses grundlegende Recht im Netz keine Gültigkeit hätte. Wer ohne ausdrückliche Genehmigung fremde Grafiken auf seinen Seiten einbaut, und sei es auch mit einem Copyright-Vermerk, macht sich strafbar.

Das Gleiche gilt selbstverständlich für fremde Texte. Verfremdungen an Bild oder Wort schützen nicht unbedingt vor einer strafrechtlichen Verfolgung.

2. Grafik am Steuer und Multimedia-Overkill

a. Link(s)gerichtete Graphiken. Ganz- und Bereichsverweise

Im Internet können Grafiken nicht nur illustrieren oder das „Satzbild“ auflockern. Zwar gehören beide Aufgaben zu den ursprünglichen Zwecken. Grafiken können – als *buttons* – über HTML aber auch mit Links versehen werden.

Soll eine Grafik als Ganzes eine Webseite verlinken, setzen Sie das A-Tag einfach um die Grafikreferenz herum:

```
<a href="http://www.sonstwo.htm"></a>
```

Wenn Sie eine Grafik als Link einsetzen, geben Sie dem IMG-Tag unbedingt das Attribut border="0" mit auf den Weg. Es verhindert, dass um die Grafik ein hässlicher violetter Rand gezogen wird. Machen Sie aber auch irgendwie deutlich, dass die Grafik ein Link *ist* !

Eine selten gebrauchte Finesse liegt darin, ein Bild in Bereiche aufzuteilen und aus den verschiedenen Bereichen unterschiedliche Links zu formen.

Das Verfahren ist zugegebenermaßen aufwändig. Sehr gut erläutert finden Sie es bei Stefan Münz in der Datei 'tcgh.htm' unter dem Titel „Verweissensitive Grafiken (Image Maps)“.

b. Techno unplugged – Animationen und Klangkulissen

Wie Grafikdateien werden Animationen jeder Art und Hintergrundmusik nicht eigentlich ein-, sondern angebunden. Trotzdem heisst das entsprechende Tag:

```
<embed src="Datei mit URL oder Verzeichnispfad">&nbsp;  </embed>
```

Wenn es sich um Flashfilme (*.swf), Videos (*.avi, *.mov) und Musikdateien (*.wav, *.au, *.mp3) handelt, wird automatisch das passende Plugin-Programm gestartet. So es denn installiert ist. Und von einer satten Hintergrundmusik haben Site-Gäste ohne Soundcard nichts als eine deutlich verlängerte Ladezeit.

Wenn Sie Multimedia in Ihre Seiten einbauen wollen, denken Sie bitte an folgende Einschränkungen:

- Noch sind die Datenleitungskapazitäten sehr eng und Multimediadateien ziemlich groß. ISDN hat sich nicht flächendeckend durchgesetzt und DSL steckt erst in den Kinderschuhen.

- Zu viel Animation macht die Seite unruhig. Endlosbewegungen nerven nachweislich jeden. Machen Sie „Ihren“ Surfern keine Kopfschmerzen.

- Passende Plugin-Programme sind nicht so weit verbreitet, wie ihre Hersteller weismachen wollen.

D. Ins Netz gegangen: Screendesign mit HTML

1. Haftzellen für Webseiten: Layout mit Tabellen

a. Zeilen, Spalten, Zellen: Die Tabellenhausordnung

Wenn HTML eines nicht ist, dann ein gutes Layoutwerkzeug. Bei der Grundsteinlegung am CERN dachte noch niemand an buntes Entertainment oder an Bildschirmjunkies.

Eine vernünftige Aufteilung des Bildschirms ist nur mit (unsichtbaren) Tabellen möglich. In dieser Einführung machen Sie nur Bekanntschaft mit den wichtigsten Attributen, die in den vier Tabellen-Tags vorkommen können. Gute Tabellenkonstrukteure unter HTML werden Sie aber nur durch Übung und viele, viele Fehler. Lassen Sie sich von Rückschlägen nicht entmutigen. Echte Profis lernen täglich aus ihren Tabellen-»Altbauten«.

Der Grundaufbau einer Tabelle ist denkbar einfach. Immerhin. Bekanntlich hat eine Tabelle Zeilen und Spalten, die sich in Zellen gliedern. HTML macht es Ihnen hier sogar ein Stück weit einfach: Es kennt nur Zeilen und Zellen. Die Spalten zählt es selbsttätig. Wenn Sie einer Zeile vier Zellen geben, dann weiss der Browser: Hier habe ich eine vierspaltige Tabelle.

Gehen Sie gleich an das konkrete Beispiel heran:

```
<table>
  <tr>
    <td>Inhalt 1A</td>
    <td>Inhalt 1B</td>
  </tr>

  <tr>
    <td>Inhalt 2A</td>
    <td>Inhalt 2B</td>
  </tr>
</table>
```

Das sieht dann so aus:

Inhalt 1A	Inhalt 1B
Inhalt 2A	Inhalt 2B

Das TABLE-Tag eröffnet eine Tabelle und wird geschlossen, nachdem alle Inhalte festgelegt sind. Das TR-Tag umgrenzt eine Tabellenzeile (*table row*). Die Daten einer einzelnen Zelle werden vom TD-Tag (*table data*) eingeschlossen.

Wichtig: die Zahl der TD-Tags innerhalb der Zeilen (TR) muss in einer Tabelle immer gleich bleiben! Sonst weiss der Browser nicht, wieviele Spalten die Tabelle denn nun haben soll.

Was aber tun, wenn eine Zelle über mehrere Zellen einer anderen Zeile hinausgehen soll? Für diesen Fall gibt es ein Attribut des TD-Tags: Das **colspan** (Spaltenübergriff).

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

```

<td colspan="2">Überschrift</td>
</tr>

<tr>
  <td>Inhalt 2A</td>
  <td>Inhalt 2B</td>
</tr>
</table>

```

Diese Tabelle sieht so aus:

Überschrift	
Inhalt 2A	Inhalt 2B

Sie können die Colspan-Werte von Zeile zu Zeile variieren. Das ist bei drei- und mehrspaltigen Tabellen interessant. In einer Zeile werden aber auch einmal alle Zellen einzeln definiert sein. In jeder Zeile müssen alle Spalten der Tabelle berücksichtigt sein!

```

<table border="1">

<!-- Die folgende Zeile organisiert insgesamt 5 Spalten -->

<tr>
  <td colspan="2">Rubrik X</td> <!-- Spalten I + II -->
  <td>Nichts</td> <!-- Spalte III -->
  <td colspan="2">Rubrik Y</td> <!-- Spalten IV + V -->
</tr>

<tr>
  <td>Inhalt 2A</td> <!-- Spalte I -->
  <td>Inhalt 2B</td> <!-- Spalte II -->
  <td>Inhalt 2C</td> <!-- Spalte III -->
  <td>Inhalt 2D</td> <!-- Spalte IV -->
  <td>Inhalt 2E</td> <!-- Spalte V -->
</tr>
</table>

```

Und so sieht das aus:

Rubrik X		Nichts	Rubrik Y	
Inhalt 2A	Inhalt 2B	Inhalt 2C	Inhalt 2D	Inhalt 2E

Vielleicht müssen Sie die Tabelle später um einige Zeilen erweitern. Und dabei kommen Sie womöglich zu dem Schluss, dass Sie noch eine weitere Spalte einbauen sollten. - Es ist möglich, aber Sie müssen dann in jeder einzelnen Zeile eine Zelle einbauen.

Neben dem 'colspan'-Attribut vertragen TD-Tags auch das Attribut 'rowspan'. Es verbindet untereinanderliegende Zellen verschiedener Zeilen. Das 'rowspan'-Attribut sollten Sie nur verwenden, wenn eine Tabelle für alle Ewigkeit unverändert bestehen bleibt. In der Regel tun das Tabellen nie. Wenn Sie gar noch 'rowspan' und 'colspan' in einem TD-Tag kombinieren – viel Vergnügen bei der Nachbearbeitung. Aber möglich ist es.

Für das TD-Tag gibt es noch andere wählbare Attribute, die stärker verbreitet und wirklich sehr nützlich sind. Nachfolgend die wichtigsten:

TD-Attribut	Werte	Funktion
align=	left, center, right	Textausrichtung in der Zelle: linksbündig, zentriert, rechtsbündig
bgcolor=	#+Hexadezimalwert, engl. Farbbezeichnung	Einfärbung einer Zelle
height=	Pixelzahl, Prozentangabe	Höhe der ganzen Zeile, muss nur in einer Zelle der Zeile definiert werden, wird „gesprengt“, wenn der Inhalt einer Zeilenspalte mehr Platz benötigt.
width=	Pixelzahl, Prozentangabe	Breite der Zelle, wird unter Umständen vom Browser ignoriert.
valign=	top, middle, bottom	Vertikale Ausrichtung der Zelleninhalte, wenn eine andere Zelle die Zeilenhöhe vergrößert hat und in der betroffenen Zelle „Luft“ entstanden ist.

Nach den Vorgaben des W3-Konsortiums können Sie dem TR-Tag keine nennenswerten Attribute zuweisen. Microsofts Internetexplorer ist so fehlertolerant, dass er typische TD-Attribute auch im TR-Tag akzeptiert. Trotz der weiten Verbreitung des Browsers sollten Sie sich aber nicht auf die Gutmütigkeit von Browsern verlassen. Die W3C-Rache kommt bestimmt.

Über eine erkleckliche Anzahl von möglichen Attributen verfügt das TABLE-Tag. Eines, das 'border'-Attribut haben Sie schon kennengelernt. Bitte tun Sie sich und Ihren Gästen einen Gefallen: Setzen Sie möglichst immer border="0".

Auch hier wieder die wichtigsten Attribute in der Übersicht:

TABLE-Attribut	Werte	Funktion
-----------------------	--------------	-----------------

bgcolor=	#Hexadezimalwert, engl. Farbbezeichnung	Einfärbung der ganzen Tabelle, Einzelfärbungen von Zellen haben Vorrang.
border=	Pixelzahl	Breite von Rand- und Gitterlinien.
cellpadding=	Pixelzahl	Abstand zwischen Zelleninhalt und innerem Zellenrand.
cellspacing=	Pixelzahl	Abstand zwischen den Zellen einer Tabelle.
height=	Pixelzahl, Prozentangabe	Höhe der Tabelle, wird von größeren Inhalten gesprengt.
width=	Pixelzahl, Prozentangabe	Breite der Tabelle, kann unter Umständen von Tabelleninhalten gesprengt werden, wenn sie unter 100% liegt.

b. Im Tabellenknast – Komplexe Tabellenstrukturen

Wozu der ganze Aufwand mit Tabellen? Oder konkreter gefragt: Wie soll mit diesem Wirrwarr ein vernünftiges Layout aufgebaut werden?

Layout kann man tatsächlich mit unsichtbaren Tabellen gestalten: Weil sich Tabellen verschachteln lassen. In einer Zelle können notfalls mehrere Tabellen verpackt werden, vielleicht sogar noch mit Textabschnitten und Grafiken aufgelockert. In jede Zelle dieser inneren Tabelle kann wiederum Inhalt mit Tabellen arrangiert werden. Wenn Sie eine komplexe Tabelle über vier oder fünf Ebenen aufbauen, sollten Sie besonderen Wert auf die Kommentare legen.

Wie lässt sich eine Tabelle als „Arbeitsblatt“ über den ganzen Bildschirm legen? Das bekommen Sie über den gezielten Einsatz von nur zwei TABLE-Attributen problemlos zustande:

```
<table width="100%" height="100%">
```

...

```
</table>
```

Wenn die Inhalte die Bildschirmanzeige vertikal übersteigen, passt sich die Tabelle automatisch an. Sind Sie sich schon darüber im klaren, auf wieviele Spalten Ihr Layout aufgebaut sein soll und wie sich diese Spalten zueinander quantitativ verhalten? Dann bietet sich der nächste Schritt an:

```
<table width="100%" height="100%">
```

```
<tr>
  <td width="25%">...</td>
  <td width="50%">...</td>
  <td width="25%">...</td>
</tr>

</table>
```

Jetzt haben Sie ein weißes Blatt vor sich. Das sollten Sie jedenfalls, denn komplexe Tabellen plant man am besten auf dem Reißbrett und setzt sie erst danach in HTML-Code um. Wenn Sie Ingenieurin oder Architekt sind: Denken Sie an die flexiblen Maße, denen HTML-Seiten leider unterliegen. Gerade horizontale Verhältnismäßigkeiten können sich je nach Browsertyp und Bildschirmauflösung stark verschieben.

Trotzdem: Planen Sie Ihre komplexen Tabellen bitte sehr sorgfältig. In vielen Fällen kommen Sie mit dem 'colspan'-Attribut bereits über die Runden. Tabellen in Tabellenzellen sorgen immer für sehr unübersichtlichen Quelltext. Die Fehlersuche, die nicht nur am Anfang zu Ihrem täglichen Brot gehören wird, ist in komplexen Tabellen äußerst aufwändig.

2. Formeller Umgang: Datenübergabe an Andere

a. Die richtige Dienststelle. Adressierung von Formulardaten

Listen und Tabellen dienen der Übersichtlichkeit, wenigstens der für den Leser der Webseiten. HTML kann aber noch viel mehr als nur etwas darstellen. Es kann auch Daten sammeln und kontrolliert an einen Adressaten weiterleiten. Eingabemöglichkeiten für den Surfer bieten meistens Formulare.

Viele Internetbenutzer schrecken vor diesem Fachausdruck „Formular“ zurück. Die meisten denken sofort an irgendeinen amtlichen Wisch, auf grauem Papier gedruckt und mit Blockbuchstaben auszufüllen. Ein HTML-Formular können Sie als HTML-Setzer aber sehr phantasievoll gestalten. Denken Sie dabei nur an die Benutzer, die durchaus erkennen sollten, wo sie welche Angaben machen müssen oder dürfen.

Im Quelltext erkennt man ein Formular am FORM-Tag:

```
<form action="" method="">
```

...

```
</form>
```

Das FORM-Tag muss geschlossen werden und braucht notwendig zwei Attribute: 'action' und 'method'. Wenn Sie anfangen mit Formularen zu arbeiten, werden Sie dem

Attribut 'method' immer den Wert "post" zuweisen. Die Alternative wäre action="get". Und die ist sehr selten. Welchen Wert das 'method'-Attribut zugewiesen bekommt, hängt von der 'action' ab. Das 'action'-Attribut kann nur einen von zwei Werten erhalten:

- action="mailto:adresse@sonstwo.tv"
- URL eines CGI-Programms auf einem Server

Im ersteren Fall verwenden Sie grundsätzlich method="post" und zusätzlich ein Attribut enctype="text/plain". Im anderen Fall ist meistens ebenfalls method="post" gefordert. Wenn es anders ist, steht das in der Dokumentation zum betreffenden CGI-Programm.

Jetzt weiss das Formular schon, wohin es seine Daten auf welche Weise schicken soll. Es stellt sich nur die Frage, *was* es zu verschicken hat...

b. Worum geht's? - Keine Eingabe ohne Benennung!

In einem Formular dürfen Sie so viele Eingabefelder unterbringen, wie Sie möchten.

Achten Sie aber von Anfang an darauf, dass auch jedes sichtbare Feld eine Beschriftung bekommt, damit der User weiss, was er eingeben soll.

Die folgende Tabelle zeigt die gängigen Typen für Eingabefelder mit ihren Attributen. Notwendige und dringend empfohlene Attribute sind mit * gekennzeichnet:

Gruppe	Tag	Attribute
Auswahlliste	<select> <option>... <option>... ... </select>	*name="Bezeichnung" *size="Sichtbare Zeilen" multiple
Checkbox (Mehrfachauswahl zum Ankreuzen)	<input>	*name="Bezeichnung" *type="checkbox" *value="Ausgewählter Wert"
Einzeilige Texteingabe	<input>	maxlength="maximale Zahl der Zeichen" *name="Bezeichnung" size="Breite in Anschlägen" *type="text"
Mehrzeilige Texteingabe	<textarea></textarea>	*cols="Breite in Anschlägen" *name="Name" *rows="Höhe in Zeilen" wrap="virtual"
Passworteingabe (mit Sternchen auf dem Bildschirm – aber Klartext in der Nachricht!)	<input>	*name="Bezeichnung" *type="password"
Radio-Auswahl (Einfachauswahl zum Ankreuzen)	<input>	*name="Bezeichnung" *type="radio" *value="Ausgewählter Wert"
Verborgene Eingabe	<input>	*name="Bezeichnung" *type="hidden" *value="vorherbestimmter Inhalt"

- Auswahlliste: Mehrere Wahlmöglichkeiten stehen in einem grauen Kasten untereinander. Die vom W3-Konsortium vorgesehene Möglichkeit der Menüstruktur

wird nur von wenigen Browsern umgesetzt. Das Attribut 'multiple' ist unregelmäßig und braucht keine Wertzuweisung. Es erlaubt eine Mehrfachauswahl über Klick mit der linken Maustaste bei gedrückter Strg-Taste.

- Checkbox: die konservative Version der Auswahlliste. Dem Benutzer bietet sich pro Eingabefeld ein kleines Feld zum Ankreuzen. Zusammengehörige Auswahlfelder bekommen die selbe Bezeichnung über das Attribut 'name'. Eine Mehrfachauswahl von Feldern mit der selben Bezeichnung ist möglich.

- Einzeilige Texteingabe: der Surfer kann Text eintippen.

- Mehrzeilige Texteingabe: ebenso, nur wird die Eingabe auf mehrere Zeilen verteilt. Wenn das Attribut wrap="virtual" gesetzt ist, erfolgt der Zeilenumbruch automatisch.

- Passworteingabe: hier nur der Vollständigkeit halber aufgenommen.

- Radioauswahl: wie Checkbox, wobei aber nur eines von den gleichnamigen Feldern angekreuzt werden kann.

- Verborgene Eingabe: viele CGI-Programme fordern ein solches Feld. Hier geben Sie als HTML-Setzer/in einen Wert vor, an dem der User nichts so einfach ändern kann. Das kann etwa eine E-Mailadresse sein, an die die Inhalte des Formulars geschickt werden sollen.

Zuletzt braucht das Formular noch einen Knopf (*button*) zum Abschicken und einen Knopf für Gäste, die sich das Ganze noch einmal anders überlegt haben.

```
<input type="submit" value="Absenden">  
<input type="reset" value="Nicht absenden">
```

c. Formulare im Gitternetz. Wie bringe ich das »form« in Form?

Eingabefelder auf eine HTML-Seite zu bringen ist gar nicht so schwer. Sie sollten aber

für eine klare Beschriftung der einzelnen Eingabefelder sorgen. Und das können Sie am besten mit einer (eventuell komplexen) Tabelle.

Es ist sinnvoll, wenn das Formular die Tabelle umschließt, in der die Daten gesammelt werden. Sonst riskieren Sie Tag-Überschneidungen, die kein Browser mag.

Checkbox- und Radioauswahl-Eingaben werden meistens in eigene kleine Tabellen gefasst. Angenommen eine Radioauswahl befindet sich als eigene Tabelle in der rechten Spalte einer einzeiligen Tabelle:

```
<table border="0" cellspacing="5">
  <tr>
    <td>Hier steht ein längerer Text, vielleicht zur allgemeinen Erklärung.</td>
    <td><form action="mailto:thsf@gmx.de" method="post"
enctype="text/plain"><table border="0">
      <tr>
        <td colspan="2">Was schmeckt Ihnen besser?</td>
      </tr>
      <tr>
        <td>Rohkost</td>
        <td><input type="radio" name="Kost"
value="Rohkost"></td>
      </tr>
      <tr>
        <td>BSE-Knochen</td>
        <td><input type="radio" name="Kost"
value="BSE"></td>
      </tr>
      <tr>
        <td><input type="submit"
value="Absenden"></td>
        <td><input type="reset"
value="Nicht abschicken"></td>
      </tr>
    </table></form></td>
  </tr>
</table>
```

Das Ergebnis sieht etwa so aus:

Hier steht ein längerer Text, vielleicht zur allgemeinen Erklärung.	Was schmeckt Ihnen besser?
	Rohkost ..
	BSE-Knochen ..
	<input type="button" value="Absenden"/> <input type="button" value="Nicht absenden"/>

3. Alles im Rahmen. Kleine Kompositionslehre

a. Rahmenlose Aufhänger – Organisationstalent »Frameset«

Sie haben schon gelesen, wie Sie dem Bildschirm mit Hilfe komplexer Tabellen ein Layout geben können. Es gibt noch einen anderen Weg, der den Einsatz von Tabellen keineswegs ausschließt, Ihnen als HTML-Setzer/in aber Umstellungsarbeiten erleichtern kann.

Wenn Sie eine umfangreiche Rubrikenzahl immer im auf dem Bildschirm sichtbar halten wollen oder vielleicht sogar mit Untermenüs arbeiten wollen – immer dann bieten sich Framesets an. Durch Frameset-Dateien wird der Bildschirm in Bereiche (Rahmen, engl.: *frames*) aufgeteilt, in denen dann verschiedene HTML-Dateien angezeigt werden.

Wenn am Anfang des Kurses gesagt wurde, dass über 90% der Webseiten dem normalen HTML-Grundschema folgen, dann ist jetzt von dem Rest die Rede. Auch Frameset-Dateien haben die Dateiendung HTM oder HTML, können META-Tags beinhalten und haben Tags. Nur anstelle eines BODY-Tags steht bei ihnen ein FRAMESET-Tag. Und in den können noch weitere FRAMESET-Tagpaare verschachtelt sein.

Beginnen Sie mit einem ganz einfachen Frameset. Der Bildschirm soll in folgende Bereiche aufgeteilt werden:

Menü: 1. Link 2. Link 3. Link 4. Link 5. Link Diese Datei heisst 'menu.htm'	Begrüßungstext. In diesem Rahmen erscheinen die Zielformen der fünf Links im Menü, das ständig sichtbar bleibt. Der Begrüßungstext steht in der Datei 'gruss.htm'.
---	--

Die Framesetdatei sieht so aus:

```
<html>  
  <head>  
    <title>...</title>  
  </head>
```

```
<frameset cols="25%,*">  
  <frame src="menu.htm" name="Menu">  
  <frame src="gruss.htm" name="Haupt">  
</frameset>
```

```
</html>
```

Das FRAMESET-Tag definiert über das Attribut 'cols' zwei Spalten. Das Attribut zur Bildung von Zeilen heisst 'rows'. Der erste (hier linke, bei Zeileneinteilung: oberste) Rahmen bekommt 25% der hier waagerechten Bildschirmgröße zugewiesen, der zweite den Rest (im Frameset immer angezeigt durch '*'). Als Werte für 'cols' oder 'rows' können Sie Prozentverhältnisse oder Pixel festlegen.

Nachdem die Grenzen für beide Bildschirmbereiche gesetzt sind – und sie bleiben unverändert, egal was in den angezeigten Dateien sichtbar wird – wird jedem Rahmen (FRAME) eine Dateiquelle (Attribut 'src' für engl. *source*) gegeben, aus der er „seine“ HTML-Datei beziehen kann.

Von entscheidender Bedeutung ist, dass beide Rahmen benannt werden! So kann ein Link in der Datei "menu.htm" eine Webseite im großen Rahmen abbilden ohne sich selbst zu verändern. Der Begrüßungstext verschwindet. Üblicherweise setzt man im Menü einen Link, der die Begrüßungsseite wieder in den Hauptrahmen setzt.

Am Beispiel gezeigt: der erste Link in der Datei 'menu.htm' verweist auf die Datei 'teil1.htm'. Im Quelltext der Datei 'menu.htm' steht:

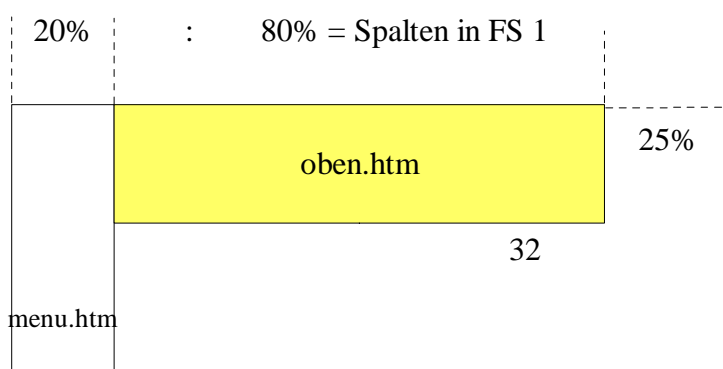
```
<a href="menu.htm" target="Haupt">1. Link</a>
```

Wenn Sie mit Framesets arbeiten, brauchen alle Links das Attribut 'target'. Als Wert des target-Attributes erscheint immer der Name eines Frames oder – wenn Sie eine ganz neue Seite aufrufen wollen – der Wert „_top“ oder „_parent“.

Bitte beachten Sie: Obwohl es technisch problemlos möglich ist, sollten Sie niemals fremde Seiten in Ihre Rahmenkonstruktion einblenden. Sie verstoßen damit gegen das Urheberrecht.

Der Browser liest die Framesetdatei streng von oben nach unten. Deswegen können Sie in einen Rahmen anstelle einer HTML-Datei auch ein weiteres Frameset einstellen. Sie brauchen dafür keine andere Framesetdatei »in den Rahmen zu holen«.

Die folgende Bildschirmaufteilung ist sehr beliebt:





Die Dateinamen können selbstverständlich ganz andere sein. Hier beispielhaft die Framesetdatei:

```
<html>
  <head>
    <title>...</title>
  </head>

  <frameset cols="20%,*">
    <frame src="menu.htm" name="li_menu">
    <frameset rows="25%,*">
      <frame src="oben.htm" name="ob_menu">
      <frame src="start.htm" name="haupt">
    </frameset>
  </frameset>

</html>
```

Es ist möglich, die Bereiche ohne Zwischenlinien ineinander übergehen zu lassen. Zu diesem Zweck braucht jedes FRAMESET-Tag drei Attribute:

```
<frameset cols="... , ..." border="0" frameborder="0" framespacing="0">
```

Wenn die Datei in einem Rahmen mehr Breite oder Höhe braucht, als das Frameset ihr gibt, kann sie das Frameset nicht verändern. Dafür blendet sie Scrollbalken ein, die manchmal sehr störend wirken. Mit einem Attribut im FRAME-Tag lassen sich Scrollbalken verhindern:

```
<frame src="..." name="..." scrolling="no">
```

Sie sollten diese Vorgabe allerdings meiden. Sie haben keinen Einfluss auf die Bildschirmauflösung Ihrer Gäste.

b. Pro und Contra Frames. Argumente für die Abwägung

Der Einsatz von Framesets ist umstritten. Für Framesets spricht ein wesentlich geringerer Aufwand, wenn in einem umfangreichen Webangebot Rubriken ergänzt oder gestrichen werden. In beiden Fällen müssen nur ein oder zwei Menüdateien geändert

werden. In einer reinen Tabellenkonstruktion sind Änderungen in allen Dateien fällig, weil die Menüs in allen Dateien festgeschrieben sind.

In Kombination mit Framesets läuft JavaScript erst zur Hochform auf. Fenster von festlegbarer Größe können erzeugt und veränderliche (dynamische) Inhalte hineingestellt werden.

Einen spürbaren Aufwand verursachen Framesetkonstruktionen bei Seiten mit wenigen Rubriken und Unterseiten. Scrollbalken zwischen sonst nahtlos aneinander anschließenden Rahmen ist Screen-Designern ein Dorn im Auge.

Der größte Nachteil von Framesets ist sehr ärgerlich: Es ist nicht möglich, ein Frameset mit einer anderen Datei als der Begrüßungsseite im Hauptrahmen aufzurufen. Verweist eine Suchmaschine oder ein Lesezeichen (*bookmark*) auf ein Einzeldokument, so wird nur dieses vom Browser aufgerufen – ohne den Navigationskontext.

E. Ausblick

Dieser einführende Kurs konnte und wollte nur einige grundlegende Möglichkeiten von HTML darstellen. Cascading Stylesheets (CSS) eröffnen HTML-Seiten ein typographisch besseres Design, JavaScript und CGI-Scripts ergänzen interaktive Elemente. Der HTML-Aufbau XML etabliert sich allmählich und erleichtert die Einbindung von datenbankähnlichen Strukturen. Die Skriptsprache PHP ist bereits jetzt eine gut durchschaubare Schnittstelle zu echten Datenbanken. Die Integration von Bestell- und sogar Redaktionssystemen in Ihr Webkonzept ist damit möglich.

F. Auswahl von Tags mit ihren wichtigsten Attributen

Einführung in die Webseitengestaltung mit HTML

Befehl (tag)	Attribute	Werte	Anmerkungen
<a>...	(*)href="..."	Dateiname URL #Ankername	Dateiname für HTML-Dateien im selben Serverbereich, evtl. mit Pfad, URL mit http://..., Ankername muss in einer Datei über festgelegt sein.
	(*)name="..."	ANSI-Zeichen	Ankername, JavaScript-Objektbenennung wenn <a>-Tag mit Attribut href="..." ausgestattet!
	(*)target="..."	ANSI-Zeichen	Nur in Frameset-Konstruktion, dort notwendig für Links: zielt auf Namensattribut aus <frame> oder auf "_top", "_parent" oder "_new"
<body>...</body>	alink="..."	engl. Farbbez. #RGB-hexadezimal	
	background="..." / bgcolor="..."	Grafikdatei mit Pfad / engl. Farbbez. #RGB-hexadezimal	
	link="..."	engl. Farbbez. #RGB-hexadezimal	
	text="..."	"	
	vlink="..."	"	
<div>...</div>	*align="..."	center right	Ersetzt ausscheidendes Zentrierungs-Tag <center>...</center>
...	(*)color="..."	engl. Farbbez. #RGB-hexadezimal	
	(*)face="..."	Schriftart, alter nativeSchriftart ,...	Standards der verschiedenen Betriebssysteme beachten!: Serifenschrift Win / Linux: Times New Roman, Mac: Times; Serifenlose Schrift: Win / Linux: Arial, Mac: Helvetica
	(*)size="..."	3-7	Standard-Voreinstellung ist 3

Befehl (tag)	Attribute	Werte	Anmerkungen
<form>...</form>	*action="..."	mailto:E-Mailadresse URL eines CGI-Scripts	Mailto-Action setzt funktionierendes E-Mail-Programm voraus
	(*enctype="..."	text/plain	Nur wenn action="mailto...", dann notwendig!
	*method="..."	post get	Wert 'get' nur, wenn CGI-Programm das ausdrücklich erfordert!
<frame>.....	marginheight="..."	Pixel	
	marginwidth="..."	Pixel	
	*name="..."	ANSI-Text	
	scrolling="..."	yes no	Voreinstellung ist 'yes', Wert 'no' bitte reiflich überlegen!
	*src="..."	Dateiname.htm , evtl. mit Pfad	
<frameset>...</frameset>	border="..."	0	Wert '0' dringend empfohlen, muss mit Attributen frameborder und framespacing kombiniert werden.
	*cols="..." oder *rows="..."	Pixel Prozentangabe %	Prozentangaben sind meistens günstiger als Pixelwerte.
	frameborder="..."	0	Siehe Anm. zum Attribut border="..."
	framespacing="..."	0	"
.....	align="..."	left right	left: Lauftext umfließt Bild rechts, right: Lauftext umfließt Bild links.
	alt="..."	ANSI-Text	Alternativtext, der angezeigt wird, wenn Browser keine Grafiken unterstützt
	border="..."	0	Wert 0 dringend empfohlen.
	name="..."	ANSI-Text	Nützlich nur bei Einsatz von JavaScript
	src="..."	Grafikdatei (.jpg, * .gif), ggf. mit Pfad	
	usemap="..."	#Imagemap-Name	

Einführung in die Webseitengestaltung mit HTML

Befehl (tag)	Attribute	Werte	Anmerkungen
<code><input>.....</code>	<code>maxlength="..."</code>	Ganzzahl	Maximale Anzahl der einzugebenden Zeichen
	<code>*name="..."</code>	ANSI-Text	
	<code>size="..."</code>	Ganzzahl	Länge des Eingabefeldes in Anschlägen
	<code>*type="..."</code>	Siehe S. 28	
	<code>*value="..."</code>	“	
<code><map>...</map></code>	<code>name="..."</code>	ANSI-Text	
<code>...</code>	<code>type="..."</code>	A a I i	A: Zählung mit Großbuchstaben, a: Zählung mit Kleinbuchstaben I: Lateinische Zählung i: Lateinische Zählung mit Kleinbuchstaben
<code><p>...</p></code>	<code>align="..."</code>	center right	Wert richtet nur den aktuellen Absatz aus, p-Tag muss nicht, kann aber geschlossen werden.

Befehl (tag)	Attribute	Werte	Anmerkungen
<code><table>...</table></code>	<code>bgcolor="..."</code>	engl. Farbbezeichnung #RGB-Hexadezimal	Einzelne Zellen können andere Hintergrundfarbe bekommen
	<code>border="..."</code>	0	Wert 0 empfohlen!
	<code>cellpadding="..."</code>	Pixel	
	<code>cellspacing="..."</code>	Pixel	
	<code>height="..."</code>	Pixel Prozentwert%	
	<code>width="..."</code>	"	
<code><td>...</td></code>	<code>align="..."</code>	center right	Richtet die Elemente in der Zelle bündig aus
	<code>colspan="..."</code>	Ganzzahl	Verbindet horizontal benachbarte Zellen
	<code>height="..."</code>	Pixel Prozentangabe %	Kann von Zelleninhalt gesprengt werden
	<code>rowspan="..."</code>	Ganzzahl	Verbindet vertikal benachbarte Zellen
	<code>valign="..."</code>	top bottom	richtet den Zelleninhalt nach dem oberen ('top') oder unteren ('bottom') Zellenrand aus
	<code>width="..."</code>	Pixel Prozentangabe %	kann vom Zelleninhalt gesprengt werden

Einführung in die Webseitengestaltung mit HTML

Befehl (tag)	Attribute	Werte	Anmerkungen
<textarea></textarea>	cols="..."	Ganzzahl	Sichtbreite des Eingabefeldes in Anschlägen, Tag wird ohne (Leer)zeichen (!!) geschlossen
	*name="..."	ANSI-Text	
	rows="..."	Ganzzahl	Sichtbare Höhe des Eingabefeldes in Zeilen
	wrap="..."	virtual	Erzeugt automatischen Zeilenumbruch bei der Eingabe
...	type="..."	disc square	disc : punktförmiges Aufzählungszeichen, square : quadratförmiges Aufzählungszeichen