

HTML

Vorab

Wir stecken voll in den Problemen, denn: Anfang 1996 war alles noch ganz einfach – es gab nur wenige Programme zur Erstellung von Web-Sites, etwa WebWeaver 1.0, PageSpinner 1.0 oder PageMill 1.0. Und ebenso wenige Nutzerprogramme: Netscape 2.0, MacWeb 1.0 oder Internet Explorer 1.0. Dann begannen die Probleme:

- 1) Grafiker und Layouter hielten die Ergebnisse – mit recht – für miserabel;
- 2) Anwender fanden die Programme zu kompliziert;
- 3) Der Kommerz (Stichwort: Bill Gates) fand die Programm-Seite zu simpel und vor allem waren sie unverschämterweise fast gratis.

- Zu 1) Aus HTML wurden HTML2 und HTML3, jetzt HTML4; hinzu trat PDF;
zu 2) Die »User«-Seite wurde mit Angeboten seither eingedeckt: Site Mill, GoLive und andere traten auf den Plan;
zu 3) Der Wettkampf der »Titanen« zwischen IE und NS entbrannt e voll, Safari, Firefox und andere traten auf den Plan.

Ein weiteres Problem: Wie koordiniere ich dies?

Ich denke: »Back tot the roots« – Ein Vergleich: Bei der Einführung des Fotosatzes und der elektronischen Redaktion in den Zeitungshäusern galt auch: Es geht mehr, als in den Handbüchern steht, wenn man das Prinzip kennt – man muß es nur herausfinden. Also erst die »Pflicht« lernen, dann die »Kür« machen. D.h. hier, die teilautomatischen HTML-Programme und -Editoren individuell beeinflussen können.

HyperText Markup Language

HTML-Texte sind als »Nur Text + Zeilenschaltung« (auch ASCII genannt) abgefaßt.

HTML ist eine SGML (Standard Generalized Markup Language).

Vorbemerkung

Die Arbeit mit HTML unterscheidet sich in einigen Punkten von der mit »normaler« Textverarbeitung:

- Die Darstellung beim Empfänger richtet sich nach den bei ihm gewählten Grundeinstellungen was z.B. Zeilenbreite, Schriftart, Schriftgröße u.ä. betrifft (vgl. auch direkte oder physische und indirekte oder logische Formatierung weiter unten).
- Der Zeilenschalter hat keine Auswirkung, der Text wird also »endlos« eingegeben. Der Browser, also das Betrachtungsprogramm, reguliert die Zeilenbreite automatisch, d.h. abhängig von der gewählten Breite des Dokuments auf dem Bildschirm, so daß auch Wort-/Silbentrennung nicht erforderlich ist. Die Einrichtung eines Absatzes erfordert ein »Tag«.
- Ebenso werden keine mehrfachen Leerzeichen vom Browser berücksichtigt.

Tags

- Zur Angabe spezieller Elemente (vergleichbar etwa dem Auszeichnen in der Typografie) sind besondere Befehle/Steuerzeichen (»Tags«) nötig.
- Manche Tags lassen sich kombinieren/verschachteln, andere (besser) nicht.

- Jeder Tag besteht aus einer Linken spitzen Klammer (<), dem Namen und einer rechten spitzen Klammer (>). Der Name ist meist der bezügliche Befehl oder dessen Abkürzung auf englisch, oft noch mit einem Attribut versehen.
- Die Tags treten üblicher- und meist sinnvollerweise als Paar auf, um den Befehl einzuleiten, z.B. <p>, und zu beenden, hier dann als </p>; das end-Tag hat also einen Querstrich.
- HTML unterscheidet, von einigen Ausnahmen abgesehen, bei den Tags nicht zwischen Groß- und Kleinschreibung; <title> ist also dasselbe wie <TITLE> wie <tITle>.

Skelett

Jedes HTML-Dokument benötigt als Basis einige Tags, um als solches vom Browser erkannt zu werden (einige neuere Browser schaffen das allerdings auch ohne – aber man weiß ja nie, welches der angesprochene Benutzer hat!).

<html> Mit diesem Tag wird jedes Dokument eingeleitet

</html> Mit diesem Tag wird es logischerweise beendet

Jedes HTML-Dokument besteht aus einem Kopf (head) und einem Körper (body), die durch entspr. Tags eingeleitet bzw. beendet werden.

Head

<head>
</head> Im HEAD-Bereich befinden sich allg. Informationen zum Dokument; dieser Text gehört nicht zum »eigentlichen«, also sichtbaren Dokument. Hier sind folgende Tags zu finden (eines oder mehrere):

<title>
</title> Name des Dokuments, der auch im Titelbalken des Browsers auftritt bzw. in die Liste der bookmarks (Lesezeichen) aufgenommen wird. Er sollte das Dokument ausreichend identifizieren und nicht mehr als 64 Anschläge umfassen.

<base>
</base> Mit diesem Steuerzeichen wird die vollständige URL des Dokuments angegeben.

<isindex>
</isindex> Kennzeichnet ein HTML-Dokument als ein Such-Dokument

<Link>
</Link> Gibt mit seinen Attributen eine Verbindung zu anderen Dokumenten an

Body

<body>
</body> Im BODY-Bereich befinden sich die übrigen HTML-Elemente, also das »eigentliche« Dokument.

Überschriften (Headings)

HTML kennt – bei der indirekten Formatierung – sechs Gliederungsebenen für die Überschriften von 1 bis 6 (1 die größte, 6 die kleinste bezüglich Schriftgrad und -schnitt). Ihre jeweilige aktuelle Darstellung hängt von den Grundeinstellungen des Browsers beim Betrachter ab. Die Syntax ist:

`<Hy>Text der Überschrift</Hy>` wobei y für die Ebenen 1 bis 6 steht, das H für Heading.

Text (Font/Size)

Wie bei den Überschriften gilt auch hier: Bei der indirekten Formatierung bestimmt der Betrachter mit seiner Grundeinstellung im Browser Schriftgrad und -art der Darstellung des Mengentextes. Er kann also prinzipiell ohne besonderen Tag eingegeben werden. Etliche HTML-Erstellungsprogramme kennen die Möglichkeit, den Text zu variieren: Web Weaver z.B. Tags für die Schriftgrößen 1 (kleinste) bis 7 (größte) aufsteigend. Ein Syntax-Beispiel wäre:

`Text des Textes` [Ø=3 für den Grundtext]

Absatz (Paragraph)

Da HTML den Zeilenschalter ignoriert, kann ein neuer Absatz nur durch ein Tag erreicht werden; dies ist `<P>` Für die erste Textzeile nach einer Überschrift ist dieser Tag nicht erforderlich. Bei neueren Browsers ist es ebenfalls nicht nötig, ein `</P>` zu setzen, da ein einleitendes `<P>` ja einen neuen Absatz erzwingt – es sei denn, der betreffende Absatz wäre speziell ausgerichtet (etwa zentriert). `<P>` setzt einen Abstand von ca. einer Leerzeile zwischen die Absätze.

Neue Zeile (Break)

Will man keinen Abstand zwischen Absätzen oder zwischen Überschrift und erster Textzeile, so kann man den Tag `
` für break statt des `<P>` verwenden.

Kein Zeilenschalter (No Break)

Soll eine Zeile auf jeden Fall erhalten bleiben, ist ein `<NOBR>` zu setzen (das `</NOBR>` keinesfalls vergessen!). [Hierin sehe ich nach dem zuvor Gesagten jedoch wenig Sinn.]

Worttrennung (Word Break)

Der entspr. gesetzte Tag `<WBR>` wirkt bei langen Wörtern wie eine Silbentrennung, wenn der Browser in eine neue Zeile gehen muß (aber wer weiß schon, wann das wo nötig wird?!).

Wortabstand

Der Wortabstand lässt sich im HTML (natürlich) nicht durch Betätigen der Leertaste vergrößern; hierzu bedarf es des Codes ` `, der die »Wortpause« jeweils um etwa den Normalwert vergrößert.

Ausrichtung (Alignment)

Genau wie in der Textverarbeitung oder beim Satz, gibt es bei HTML neben der hier normalen Linksbündigkeit die Möglichkeit des Ausrichtens: Überschriften und Text können auch zentriert oder (nur bei neueren Browsers) rechtsbündig gesetzt werden. Die Attribute sind »center« oder »right«. Für Überschriften gibt es zwei Möglichkeiten der Sytax: Bei älteren Browsers (nur zentriert)

`<CENTER><H1>Überschrifttext</H1></CENTER>` hierbei ist wichtig, den Ausrichtungs-Befehl wieder aufzuheben!

Für neuere Browsers geht es auch so:

`<H1 ALIGN="CENTER">Überschrifttext</H1>` (hier könnte auch "right" stehen)

Für Text lautet der Tag:

`<P ALIGN="CENTER">` (oder auch "right"; wichtig sind die ""
Der Befehl muß nicht aufgehoben werden; dies erledigt das nächste `<P>`.)

Auflistungen (Lists)

HTML unterstützt nicht-numerierete, numerierete und definierende Listen. Diese können auch verschachtelt sein (vorsichtig, daß die Übersicht nicht verloren geht).

Nicht-numerierete Liste:

(a) Beginnt mit dem unnumbered list Tag ``

(b) Jede Position wird mit `` (für list item) eingeleitet; ein `` ist nicht nötig

(c) Die Liste wird mit `` beendet

Als Kategoriezeichen für die items kann nach dem einführenden `` der Typ gewählt werden:

`<UL TYPE="BULLET / CIRCLE / SQUARE">` für Vollpunkt (•), Leerpunkt (o) oder Geviert; der Vollpunkt ist automatisch Standard. Absätze in der Liste werden mit dem `<P>`-Tag erreicht

Beispiel: aus

```
<UL>
<LI> Äpfel
<LI> Birnen
<LI> Bananen
</UL>
```

wird

```
•Äpfel
•Birnen
•Bananen
```

Numerierete Liste:

Eine numerierete Liste (numbered oder auch ordered list, daher der Tag-Name) ist mit einer nicht-numerierten identisch, außer daß hierbei `` statt `` verwendet wird.

Die Standardnumerierung ist 1., 2., 3. ... Sie kann variiert werden durch `>OL TYPE="A /a / I / i">` für Großbuchstaben Kleinbuchstaben, Römische Zahlen groß oder klein. Soll das erste item nicht die 1 sein, wird `<OL START=n>` eingegeben, wobei n für die erste **Zahl** steht.

Beispiel: aus

```
<OL>
<LI> Äpfel
<LI> Birnen
<LI> Bananen
</OL>
```

wird

1. Äpfel
2. Birnen
3. Bananen

Definierende Liste:

Eine solche Liste (definiert als <DL>) besteht aus abwechselnd einem Stichwort (definition term, <DT>) und einer Erklärung (definition definition, <DD>). Die Browser setzen die Erklärung üblicherweise in eine neue Zeile. Die DDs und DTs können auch Absätze (<P>), Listen u.ä. enthalten. Es ist zu überlegen, ob man den DTs einen größeren Schriftgrad zuweist: entweder als »Header 1–6« oder als »Font Size=2–6« (beim header gibt es dann allerdings einen entspr. Abstand zur 1. Zeile der DD).

Beispiel: aus

<DL>

<DT> BJS

<DD> BJS, die Berliner Journalisten-Schule, hat ihren Sitz im Hause
des Berliner Verlags in Berlin-Mitte.

<DT> DJS

<DD> DJS, die Deutsche Journalisten-Schule, hat ihren Sitz im Zentrum
von München.

wird

BJS

BJS, die Berliner Journalisten-Schule, hat ihren Sitz im Hause
des Berliner Verlags in Berlin-Mitte.

DJS

DJS, die Deutsche Journalisten-Schule, hat ihren Sitz im Zentrum
von München.

Sind das Stichwort DT und die Definition DD sehr kurz, kann die Liste auch »kompakt« angelegt werden. Beispiel: aus

<DL COMPACT>

<DT> -a

<DD> geht oder geht nicht;

<DT> -b

<DD> ist auch egal;

<DT> -c

<DD> wie man es eben nimmt.

</DL>

wird

-a geht oder geht nicht;

-b ist auch egal;

-c wie man es eben nimmt.

Verschachtelte Listen:

Listen können auch – in verschiedenen Hierarchien und/oder Absätzen – verschachtelt werden (sofern man die Übersicht behält).

Beispiel: aus

 Einige westliche Bundesländer:

 Hamburg

 Bremen

 Hessen

 Zwei neue Bundesländer:

```
<UL>
<LI> Brandenburg
<LI> Sachsen
</UL>
</UL>
wird:
```

- Einige westliche Bundesländer:
 - o Hamburg
 - o Bremen
 - o Hessen
- Zwei neue Bundesländer:
 - o Brandenburg
 - o Sachsen

Vorfomattierter Text (Preformatted text)

HTML berücksichtigt, wie bereits erwähnt, normalerweise keine Zeilenschaltung, keine mehrfachen Leerzeichen/Tabulatoren oder Laufweiten im Text. Soll doch einmal eine bestimmte Auftrittform beibehalten werden (etwa bei Tabellen oder MS-Originalen), so müssen die Tags `<PRE>` bzw. `</PRE>` verwendet werden. Die Browser stellen derartigen Text allerdings nur mit nicht-proportionalen Schriften dar. Zudem dürfen in dem Text keine (Sonder-)Zeichen enthalten sein, die in der HTML eine spezielle Bedeutung haben: dies sind vor allem `>`, `<` und `&`. Dasselbe gilt häufig auch für Umlaute (wer kennt sie nicht, die »unsauberen« Texte im Browser, bei denen z.B. statt eines »Ü« ein »`>`« steht?). Solche Zeichen müssen durch ihre sog. Escape-Sequenzen (auch »Character Entities«, s.u. Sie hier alle aufzuführen, ist jedoch unmöglich, da sie gut 70 umfassen) ersetzt werden. Es empfiehlt sich, innerhalb einer PRE-Passage keine weiteren HTML-Tags zu setzen; (Hyper-)Links sind jedoch möglich. Ein PRE-Tag kann zusätzlich mit dem Attribut `WIDTH` versehen werden, das von einigen Browsern verstanden wird und sie dazu veranlaßt, die damit vorgegebene Buchstabenanzahl pro Zeile einzuhalten und eine entspr. geeignete Schrift auszuwählen.

Blockquote

Zur optischen Heraushebung einer – längeren – Textstelle, etwa eines Zitates, eignet sich der Tag `<BLOCKQUOTE>` bzw. `</BLOCKQUOTE>`: Hierdurch wird die Passage als separater Block, meist eingerückt, im Browser gezeigt.

Adresse

Das Steuerzeichen `<address>` wird allgemein zur Benennung des Verfassers, einer Kontaktmöglichkeit (v.a. Email-Anschrift) und/oder eines Versionshinweises zum Text verwendet und ist üblicherweise die letzte Zeile eines HTML-Dokuments. Hiermit wird nicht die Postadresse versehen! Dies geschieht mit der erzwungenen Zeilenschaltung `
` anstelle des Absatzbefehls `<P>`, der ja größere Zeilenabstände hervorruft:

```
Dr. Michael Meissner<BR>
Zu den Erlen 19<BR>
14979 Großbeeren<BR>
```

Trennlinie (Horizontal Rule)

Mit dem Tag `<HR>` wird eine helle horizontale Linie gezogen, die über die gesamte Breite der Seite im Browser geht. Mit dem Zusatz `SIZE=n` kann die Linienstärke in Pixeln modifiziert werden (bis maximal ca. 80), mit `WIDTH=n%` ihre Breite (zentriert gesehen): `<HR SIZE=4 WIDTH=50%>` meint eine Linie von 4facher Stärke und halber Seitenbreite. Mit dem Attribut `ALIGN="RIGHT"`

oder "LEFT" läßt sich die Linie entspr. Ausrichten. Der Zusatz NOSHADE erzeugt eine volle/schwarze Linie.

Zeichenformatierung

HTML kennt zur Formatierung von Zeichen, Wörtern oder Sätzen zwei Stilformen: die logische und die physische. Wenn z.B. in einem Browser ein Wort *kursiv* erscheint, kann der Verfasser des Dokuments dies mit dem Tag <DFN> (logisch) definiert oder mit einem anderen Steuerzeichen versehen haben, das dem Browser (physisch) sagt: »Setze das folgende Wort kursiv!«

Der Vorteil der logischen Vorgehensweise liegt darin, daß HTML z.B. eine Überschrift der Kategorie 1 als ebensolche versteht und nicht als etwa 24p Times Bold zentriert. Die typografischen Merkmale kann der Benutzer in seinem Browser voreinstellen oder – vergleichbar mit einer Druckformatliste in Textverarbeitungs- und Layoutprogrammen – kollektiv ändern. Oder die Hervorhebung: Der Tag veranlaßt die meisten Browser automatisch dazu, den Textteil gefettet hervorzuheben; wünscht der Leser stattdessen solche Passagen etwa in rot, stellt er dies selber ein. Für den Verfasser trägt diese Stilform zur Konsistenz in seinem Dokument und zur Arbeitserleichterung bei: er braucht sich nicht ständig um die typografischen Eigenheiten der Text- und Überschriftenkategorien zu kümmern.

Der Vorteil der physischen Tags liegt vor allem darin, daß sie die Einstellungen des Lesers »überlaufen«: Will der Verfasser also seine Hervorhebungen etwa in kursiv, setzt er das entspr. Steuerzeichen und trägt somit zu einer gewissen Konsistenz bei. Hinzu tritt, daß zahlreiche logische Definitionen gleichartig, d.h. in der Regel kursiv erscheinen.

Zwei Hinweise aus der Literatur: Zum einen wird empfohlen, in einem Dokument die beiden Stile nicht zu mischen, zum anderen darauf verwiesen, daß zukünftige HTML-Generationen physische Tags eventuell nicht mehr werden interpretieren und Browsers somit nicht mehr darstellen können.

Logische Stile

- <DFN> ein definiertes Wort; typischerweise kursiv.
- für emphasis, Betonung; üblicherweise ebenfalls kursiv.
- <CITE> (zitiert mit "..." im Text) für Titel von Büchern, Filmen etc.; üblich als kursiviert.
- <CODE> für Computercodes; erscheint in nichtproportionaler Schrift.
- <KBD> Hinweis für Tastatur-(keyboard)-befehle des Nutzers/Lesers; typischerweise in nichtproportionaler Schrift.
- für (starke) Betonung; erscheint üblicherweise gefettet.
- <VAR> für eine Variable, die durch eine spezifische Information ersetzt wird; erscheint normalerweise kursiviert.

Physische Stile

- gefetteter (bold) Text.
- <I> kursivierter (italic) Text.
- <TT> nichtproportionale Schrift wie Schreibmaschine (typewriter).

Escape-Sequenzen (auch Character Entities).

Sie haben zwei Funktionen: (a) um Buchstaben/Zeichen (characters) darzustellen, die mit dem ASCII nicht aufrufbar sind; etwa Umlaute oder Buchstaben mit diakritischen Zeichen; (b) um speziellen Buchstaben/Zeichen zu »entkommen«, die in HTML eine spezifische Funktion haben: v.a. < und >, die Anfang bzw. Ende eines HTML-Befehls definieren, und &, das eine Escape-Sequenz einleitet:

< für < (left angle bracket).

> für > (right angle bracket).

& für & (ampersand).

ö für ö.

ü für ü.

ä für ä.

ñ für ñ.

È für È.

Hinweise: 1. Diese »character entities« unterscheiden, anders als bei üblichen HTML-Befehlen, zwischen Groß- und Kleinschreibung! Es MUSS also < oder > sein, aber ö ist ö, Ö ist Ö.
2. Sie werden anstelle des von ihnen erzeugten Buchstabens/Zeichens direkt in den Text gesetzt; »läuten« ist also in HTML »läuten«. Hieran erkennt man auf jeder web site die Experten und die Ignoranten bzw. Faulen (bei diesen steht »l,uten«).

Verbindungen (Links)

Das Herausragende an HTML ist ihre Fähigkeit, andere Texte/Textstellen und/oder Bilder und Illustrationen (neuerdings auch Töne und Bewegtbilder) mit einem HTML-Dokument oder -Absatz zu verbinden. Ein Browser hebt die identifizierten Verbindungen (durch Unterstreichung oder farbliche Markierung) hervor um zu zeigen, daß ein hypertext Link (auch HyperLink oder Link) existiert.

Der einzige Tag, der solch eine Verbindung herstellt, ist <A> für anchor (Anker). Um einen solchen Anker im eigenen Dokument herzustellen, geht man so vor:

1. Man startet den Anker an der betreffenden Stelle mit <A (danach ein Leerzeichen);
2. man spezifiziert das betreffende Dokument mit dem Parameter HREF="Dokumentname des Bezugs" (wobei die Anführungszeichen wichtig sind) und schließt mit >;
3. man schreibt den Text/das Wort, das im aktuellen Dokument diesen Bezug herstellt;
4. man endet mit .

Eine Hypertextverbindung zu einer Datei namens »Test.html« etwa sähe so aus:

```
<A HREF="Test.html">Versuch</A>
```

Dieser Tag macht das Wort »Versuch« im aktuellen Text zum HyperLink zur Datei »Test.html«, die sich in diesem Fall auf derselben (Ordner-/Verzeichnis-)Ebene befindet.

Relative versus absolute Pfadbezeichnungen

Man kann einen Link zu Dokumenten in anderen Verzeichnissen herstellen, indem der relative Pfad vom aktuellen zum verbundenen Dokument angegeben wird. Die Verbindung zu einer Datei namens »Test.html« im Unterverzeichnis »HTML lernen« wäre z.B.:

```
<A HREF="HTML lernen/Test.html">Versuch</A>
```

Dies ist eine »relative« Verbindung, da der Pfad zum verbundenen Dokument relativ zur Lage des aktuellen Textes angegeben wird. Man kann auch den »absoluten« Pfadnamen (d.h. die komplette URL) angeben, aber relative Links sind vorzuziehen, da es:

1. einfacher ist, ein Dokument oder eine Gruppe von Dokumenten an einen anderen Ort zu bewegen (denn deren relative Pfadname bleibt weiterhin gültig);
2. effizienter ist, eine Verbindung zum Server herzustellen;
3. weniger zum Eintippen gibt.

Andererseits sollten jedoch absolute Pfadnamen verwendet werden, wo Verbindungen zu Dokumenten hergestellt werden, die nicht in einem direkten Zusammenhang stehen. So z.B. in einer Gruppe von Texten, die eine Benutzerführung darstellen: Links innerhalb dieser Gruppe sollten relative sein, solche zu anderen Dokumenten (etwa Hinweise zu entspr. Software) jedoch absolute (denn wenn die Gruppe in ein anderes Verzeichnis verlegt wird, muß keiner dieser Links aktualisiert werden).

Email-Funktion (mailto)

Wird Wert gelegt auf Reaktionen per Email zu einem Dokument/Text, kann man die Arbeit dem Leser erleichtern, wenn ein »Mailto«-Attribut als HyperLink angelegt ist. Das Format lautet:

```
<A HREF="mailto:emailinfo@host">Name</A>
```

Wird z.B. eingegeben

```
<A HREF="mailto:mmei@zedat.fu-berlin.de">Michael Meissner</A>
```

, so öffnet sich bei der Aktivierung dieses Links ein Fenster, das bereits für Email an mich konfiguriert ist.

URL

Das WWW benutzt Uniform Resource Locators (für U manchmal auch Unique), um die Position von Dateien auf anderen Servern zu spezifizieren. Eine URL umfaßt die Bezeichnung des Ressourcenzugangstyps bzw. Des Übertragungsprotokolls (etwa Web, gopher), die Adresse des Servers, optional den »port« und die dortige Position der Datei. Die Syntax ist:

Protokollbezeichnung://Server.Adresse [:Port]/Pfad/Pfad/Dateiname.Erweiterung

Als »Protokollbezeichnung« gibt es folgende Kennungen:

file	für eine Datei auf dem lokalen System
ftp	File Transfer Protocol für den Zugriff auf einen »anonymen Ftp-Server«
gopher	für eine Datei auf einem Gopher-Server
http	Hypertext Transfer Protocol für den Zugang zum WWW
news	für eine Newsgroup im Usenet
telnet	für einen Telnet-Zugriff auf einem anderen Rechner
WAIS	für eine Quelle auf einem Wide Area Information Service-Server

Die »Port«-Nummer kann generell ausgelassen werden – es sei denn, man wird ausdrücklich zu ihrer Verwendung aufgefordert).

Bilder (Inline images)

Die meisten Browser können inline images (d.h. Bilder neben dem Text) darstellen, die in den Formaten X Bitmap (XBM), Portable Network Graphic PNG), GIF oder JPEG vorliegen. Da das »Laden« von Bildern den Aufbau des Dokumentes verlangsamt, sollte die Zahl der Bilder überlegt werden. Um ein Bild einzubeziehen, lautet der HTML-Befehl

```
<IMG SRC="Name des Bildes">
```

wobei der Name der URL der Bilddatei entspricht; die Syntax ist somit die eines Anker-HREF. Je nach Format des Bildes muß dem Bildnamen unmittelbar die Endung .xbm, .png, .gif oder .jpeg bzw. .jpg angefügt werden.

Für die Bildauflösung (beim Scannen oder der Bildbearbeitung) reichen 72 dpi völlig aus, weil dies die Bildschirmausgabe ist. Auch bei den Farben kann man sich ruhig auf 256 beschränken. Denn je höher die Auflösung und/oder die Farbdichte, desto höher der Speicherbedarf (z.B. auf dem Server) und die Ladezeit beim Betrachter. Gleiches gilt für die Bilddimension (Höhe und Breite, gemessen in Pixeln). In der Mehrzahl der Fälle bietet sich das JPEG-Format an, denn wo dasselbe Bild als TIFF 300 (nicht inline verwendbar) und als GIF noch 30 Bytes braucht, kommt es als JPEG mit 10 Bytes aus. GIF-Dateien können im übrigen maximal 256 Farben, JPEG-Dateien hingegen bis zu 16,7 Millionen.

Dennoch hat GIF einen – psychologischen – Vorteil: Es verfügt über das GIF89a-»Interlaced«-Format, mit dem ein Bild beim Betrachter nicht auf einmal geladen wird (und »ewig nix« zu sehen ist), sondern stufenweise langsam aufgebaut und ständig verfeinert wird.

Bildgröße

Dem -Tag sollten zwei weitere Attribute zugefügt werden, um dem Browser die gewünschte Bildgröße vorzugeben – sofern die Bilddatei nicht bereits als 1:1 angelegt wurde: HEIGHT und WIDTH nennen sie in Pixeln (die Pixelgröße erfährt man aus dem Bildbearbeitungsprogramm wie z.B. Photoshop).

Ein Porträtfoto in einem Dokument erhalte etwa diesen Tag:

```
<IMG SRC="Selbstporträt.gif" HEIGHT=100 WIDTH=65>
```

Daneben können die Werte zur Grafikdarstellung auch in Prozent angegeben werden; so bei einem quadratischen Foto, das verkleinert werden soll:

```
<IMG SRC="Viereck.jpeg" WIDTH=50% HEIGHT=50%>
```

Und schließlich können Pixel- und Prozentangaben gemischt erfolgen. Einige Browser werden durch diese Attribute veranlaßt, das Bild zu strecken oder zu stauchen, wenn die angegebenen Werte nicht exakt sind. Anderen wiederum reicht die Angabe der Höhe oder Breite, so daß sie das Bild automatisch skalieren.

Bildrahmen

Man kann den Browser dazu veranlassen, um das Bild einen Rahmen zu setzen:

```
<IMG SRC="Viereck.jpeg" BORDER=20>
```

Die Angabe erfolgt in Pixeln und benennt die Rahmenstärke.

Wenn »Mapping-Dateien« erstellt werden, kann man bei der Grafikreferenz »BORDER=0« angeben. Andernfalls zeichnet der Browser dabei einen Rahmen in der eingestellten Verweisfarbe um das Bild um zu signalisieren, daß es sich um einen Link handelt.

Bildausrichtung

Bei der Positionierung von Bildern hat man einige Flexibilität. Bilder lassen sich getrennt vom Text oder am Text rechts, links oder zentriert ausrichten. Hier muß man ausprobieren, wie die Information am besten aussieht.

Bei der Ausrichtung von Text mit Bild wird grundsätzlich die Bildunterkante mit dem folgenden Text ausgerichtet. Mit den Attributen ALIGN="TOP / MIDDLE / BOTTOM" im IMG SRC-Tag können Bilder (jeweils links) an den Beginn, die Mitte oder das Ende eines Absatzes ausgerichtet werden, sofern das Bild nicht durch <P> als eigener absatz deklariert ist.. Wird der Text mit einem <P> eingeleitet, beginnt er unterhalb der Grafik.

Um Bilder ohne direkte Textverbindung (z.B. ein Logo o.ä.) zu plazieren, wird es in einen separaten Absatz <P> verwandelt und mit einem ALIGN=-Attribut versehen; also z.B.

```
<P ALIGN="CENTER">  
<IMG SRC="Firmenlogo.gif">  
</P>
```

Namentlich beim Netscape genügt auch:

```
<CENTER>  
<IMG SRC="Firmenlogo.gif">  
</CENTER>
```

Auch ist es möglich, mehrere/verschiedene Bilder in derselben Zeile darzustellen, bspw.:

```
<HTML>  
<IMG SRC="Porträt.gif" ALIGN="LEFT" HSPACE=10>  
<IMG SRC="Porträt.gif" ALIGN="RIGHT" HSPACE=10>  
</HTML>
```

Mit dem Parameter HSPACE wird der (horizontale) Abstand der Grafik zum nächsten Objekt bestimmt, dementsprechend mit VSPACE der vertikale.

Natürlich ließe sich zwischen die beiden Grafiken in unserem Beispiel noch Text unterbringen. das sähe dann etwa so aus:

```
<HTML>  
<IMG SRC="Porträt.gif" ALIGN="LEFT" HSPACE=10>  
<IMG SRC="Porträt.gif" ALIGN="RIGHT" HSPACE=10>  
<CENTER> Das rechts und Links bin ich, denn es ist ja mein Portr&auml;t! </CENTER>  
</HTML>
```

Hintergrundbilder

Die neueren Browsers können Bilder laden und sie als Dokumenthintergrund zeigen. Ob man dies verwendet oder nicht ist wohl Geschmackssache; jedenfalls sollte der Text in seiner Lesbarkeit nicht beeinflußt werden. Solche Bilder können z.B. eine Textur oder ein Objekt (etwa das Firmenlogo) sein.

Das Hintergrundbild wird wie jedes andere kreiert. Man braucht jedoch nur ein kleines Bildchen herzustellen: Durch die Benutzung des »tiling« nimmt der Browser das Bild auf und verteilt es horizontal und vertikal über die Bildfläche des Betrachters. Dies erfolgt automatisch, wenn der folgende Tag im »Body-Bereich« des Dokumentes gesetzt wird (hier ein »GIF«-Format:

```
<BODY BACKGROUND="Bildname.gif">
```

Exkurs: Grafiken referenzieren ab HTML 3.0

Statt des -Tags steht hier der wesentlich leistungsfähigere <FIG> zur Verfügung. Beim SRC= gelten die gleichen Regeln. Beispiel:

```
<FIG SRC="freundin.gif">
<H1> Ulla </H1>
<P> Das nebenstehende Bild zeigt meine Freundin
</FIG>
```

Im Gegensatz zum IMG braucht FIG einen schließenden Tag. Zwischen beiden können beliebige Textelemente stehen; im Beispiel eine Überschrift 1. Ordnung und ein normaler Textabsatz. Diese Elemente werden so ausgerichtet, daß der Text um die Grafik fließt.

Hinzu tritt, daß mit diesem Tag bei verweis-sensitiven Grafiken (s.u.) keine Server-Kommunikation mehr nötig ist.

Grafiken ausrichten

Mit dem <FIG>-Tag gibt es folgende Möglichkeiten:

```
<FIG SRC="freundin.gif" ALIGN="LEFT"> ... </FIG>
    richtet die Grafik am linken Textrand aus
<FIG SRC="freundin.gif" ALIGN="BLEEDLEFT"> ... </FIG>
    richtet die Grafik am linken Fensterrand aus
<FIG SRC="freundin.gif" ALIGN="CENTER"> ... </FIG>
    richtet die Grafik zentriert aus
<FIG SRC="freundin.gif" ALIGN="RIGHT"> ... </FIG>
    richtet die Grafik am rechten Textrand aus
<FIG SRC="freundin.gif" ALIGN="BLEEDRIGHT"> ... </FIG>
    richtet die Grafik am rechten Fensterrand aus
```

Der Unterschied zum »bleed« ist in HTML 3x nötig wegen der Möglichkeit der Absatzeinrückung.

Grafikbeschriftung

Mit dem <FIG>-Tag können zwischen Start und Ende Angaben zur Grafibeschriftung gemacht werden. Beispiel:

```
<FIG SRC="freundin.gif" CAPTION>
<CAPTION> Bild von Ulla </CAPTION>
<P> Das nebenstehende Bild zeigt meine Freundin
<CREDIT> aufgenommen am 5.12,1997 </CREDIT>
</FIG>
```

Mit »caption« wird eine Bildüberschrift ein- und mit »/caption« ausgeleitet. Für Zusatzangaben zum Bild gilt »credit« entsprechend.

Beide Tags sind logische, d.h. der Browser entscheidet, wie die Elemente im Verhältnis zum Bild ausgerichtet werden.

Hintergrundfarbe

In der Grundeinstellung zeigen Browser schwarzen Text auf grauem Hintergrund. Beides läßt sich auf Wunsch ändern (gleiches gilt für die Farbe von Links, besuchten Links und aktiven Links). Auch hierbei ist auf den Erhalt der Lesbarkeit zu achten. Der entsprechende Tag gehört in den »Body«-Bereich; z.B.:

```
<BODY BGCOLOR="#000000" TEXT="#FFFFFF" LINK="#9690CC">
```

Dieser Tag erzeugt ein Fenster mit schwarzem Background (BGCOLOR), weißem Text (TEXT) und silbrigen HyperLinks (LINK). Die Farbzuzuweisungen für besuchte (visited) und aktive Links erfolgen mit VLINK bzw. ALINK.

Die sechsstellige Zahlen-Buchstaben-Kombination gibt die RGB-Werte (rot, grün, blau) hexadezimal zwischen 000000 (schwarz bzw. keine Farbe) und FFFFFFFF (weiß; alle drei Farben voll gesättigt) an. Dies funktioniert allerdings nicht für die HR

Einige übliche Dokumententypen und ihre Suffixe

Nurtext	.txt
HTML-Dokument	.html
GIF-Bild	.gif (Graphic Interchange Format)
TIFF-Bild	.tiff (Tagged Image File Format)
X Bitmap-Bild	.xbm
JPEG-Bild	.jpeg oder .jpg (Joint Photographers Engineering Group)
PostScript-Dokument	.ps
AIFF-Tondokument	.aiff (Audio Interchange File Format)
AU-Tondokument	.au (Sund Audio Fle)
WAV-Tondokument	.wav
QuickTime-Film	.mov
MPEG-Film	.mpeg oder mpg (Motion Picture/Photographers Engineering Group)

Es gilt bei diesen Links natürlich zu berücksichtigen, daß nicht jeder Betrachter über die entsprechende Software verfügt!

Tabellen

Vor der Entwicklung der entsprechenden Tags konnten Tabellen nur als <PRE>-Tags mit anderwärts vorbereiteten Tabellen in HTML-Dokumenten präsentiert werden. Doch nun lassen sie sich direkt mit HTML erzeugen (HTML 3.2 und 4 sehen noch weitere Vereinfachungen vor).

Tabellen bestehen prinzipiell aus Köpfen (in den gesagt wird, was die Spalten/Reihen beinhalten, den Reihen für die Informationen und den Zellen für jeden Informationspunkt. In HTML sind die Elemente und ihre Tags im wesentlichen diese:

Elemente

<TABLE> </TABLE>	definiert eine Tabelle; wenn das Attribut BORDER=n hinzugefügt ist, wird sie entspr. umrahmt.
<CAPTION> </CAPTION>	definiert den Tabellentitel; er steht grundsätzlich zentriert am Tabellenkopf, kann aber durch das Attribut ALIGN=BOTTOM ans Ende gesetzt werden. (Zur Caption können weitere Tags gesetzt werden.)
<TH> </TH>	definiert einen Reihenkopf; grundsätzlich ist der Text gefettet und zentriert, doch lassen sich auch hier Attribute beifügen.
<TR> </TR>	spezifiziert eine Reihe/Kolumne; für die gesamte Reihe lassen sich Attribute vergeben: ALIGN= (LEFT, CENTER, RIGHT) und/oder VALIGN= (TOP, MIDDLE, BOTTOM); zu den Attributen siehe weiter unten.
<TD> </TD>	definiert eine Datenzelle; grundsätzlich ist der Zellentext linksbündig und vertikal ausgerichtet, andere Attribute sind möglich.

Attribute

(Attribute in <TH> und <TD> überlaufen die zu <TR> gegebenen)

ALIGN (LEFT, CENTER, RIGHT)	horizontale Ausrichtung einer Zelle
VALIGN (TOP, MIDDLE, BOTTOM)	vertikale Ausrichtung einer Zelle
COLSPAN=n	Breite der Tabellenzelle in Spalten
ROWSPAN=n	Höhe der Tabellenzelle in Reihen

Aufbau

Das generelle Format einer Tabelle sieht so aus:

<TABLE>	Beginn der Tabelle(ndefinition)
<CAPTION>Text</CAPTION>	Tabellenname (wird oft weggelassen und als »normaler« Text abgesetzt)

<TR> Start der ersten Reihenzeile
 <TH>Zellinhalt</TH> erste Zelle in Reihe 1 (Reihenkopf)
 ...
 <TH>Zellinhalt</TH> letzte Zelle in Reihe 1 (Reihenkopf)
 </TR> Ende der ersten Reihenzeile

 <TR> Start der zweiten Reihenzeile
 <TD>Zellinhalt</TD> erste Zelle in der zweiten Zeilenreihe
 ...
 <TD>Zellinhalt</TD> zweite Zelle in der zweiten Zeilenreihe
 </TR> Ende der zweiten Reihenzeile

 <TR> Start der letzten Reihenzeile
 <TD>Zellinhalt</TD> erste Zeile in der letzten Zeilenreihe
 ...
 <TD>Zellinhalt</TD> letzte Zeile der letzten Zeilenreihe
 </TR>

 </TABLE>

Beispiel für eine ganz simple Tabelle:

Nachname	Vorname	Telefon
Maier	Josef	01234/5678
Müller Heinz	02345	6789

Der HTML-Code dafür sieht so aus:

```

<TABLE BORDER=1>

<TH COLSPAN=40 ROWSPAN=1>Nachname
<TH COLSPAN=40 ROWSPAN=1>Vorname
<TH COLSPAN=40 ROWSPAN=1>Telefon
</TH>
<TR>
<TD COLSPAN=40 ROWSPAN=1>Maier
<TD COLSPAN=40 ROWSPAN=1>Josef
<TD COLSPAN=40 ROWSPAN=1>01234/5678
<TR>
<TD COLSPAN=40 ROWSPAN=1> Muuml;ller
<TD COLSPAN=40 ROWSPAN=1>Heinz
<TD COLSPAN=40 ROWSPAN=1>02345/6789
</TD>
</TABLE>

```

Breite und Höhe

Tabellen werden so breit, wie es ihr Inhalt verlangt, oder so breit wie das Fenster des Benutzers. Es läßt sich jedoch eine Maximalbreite im Verhältnis zur Größe des Fensters bestimmen. Zudem kann eine Tabellenhöhe definiert werden.

Beispiele:

```
<TABLE BORDER WIDTH=60% HEIGHT=90%>
```

```
<TABLE BORDER WIDTH=400 HEIGHT=800>
```

Die Zahlenwerte geben die Dimensionen in Pixeln an (2 Pixel = ca. 1mm). Diese Angaben haben im Konfliktfall mit erzwungenen/vorgegebenen Spaltenbreiten oder Zeilenhöhen Vorrang. Gleiches gilt für die Zeilenhöhe und Spaltenbreite.

Linien, Abstände und Farben

Alle als Attribute zum <table>

Wie bei den übrigen Linien, z.B. der <HR>, können der Charakter des Außenrahmens und der Gitternetzlinien definiert werden:

```
<TABLE BORDER=6 CELLSPACING=3>
```

»Border« bestimmt die Dicke der Außenlinie in Pixeln, »cellspacing« die der Gitternetzlinien. Mit dem Attribut »bordercolor="#xxxxxx"« läßt sich die Farbe bestimmen. Zudem kann der Abstand zwischen Zellenrand und Zelleninhalt als »cellpadding« definiert werden:

```
<TABLE BORDER=6 CELLSPACING=3 CELLPADDING=5>
```

Mit dem Attribut »BGCOLOR="#xxxxxx"« und kann die Hintergrundfarbe der Tabelle bzw. einzelner Zellen(gruppen) und die Farbe der Schrift bestimmt werden, mit dem Tag »FONT COLOR="#xxxxxx"« ebenjene.

Erweiterte Einsatzmöglichkeiten

<table> dient nicht nur zur Darstellung von Tabellen im engeren Sinne, sondern auch als Layout-Hilfe, da die HTML-Möglichkeiten ja beschränkt sind. Ein Zelleninhalt kann z.B. auch eine Illustration sein, die durch die entspr. Konstruktion einer Tabelle an eine gewünschte Position plaziert wird. Man denke daran, daß eine Tabelle mit unsichtbaren Linien (»border« und »cellspacing« mit gleicher Farbe wie der Fensterhintergrund) ausgestattet werden kann, so daß nur die Zellen(inhalte) sichtbar sind. Oder betrachten Sie folgendes Beispiel zur Darstellung eines dreispaltigen Textes mithilfe einer »Tabelle«:

```
<HTML>
<HEAD>
<TITLE> &Uuml;bung 3 </TITLE>
</HEAD>
<BODY>
<CENTER><FONT SIZE=7>Digital Dispatch</FONT><BR>
<FONT SIZE=4>The newsletter of CNET</FONT><P>
</CENTER>
<TABLE WIDTH="100%">
<TR>
<TD WIDTH="20%" VALIGN="TOP" ALIGN="LEFT"><FONT SIZE=2>
<B>CONTENTS:</B>
```


1. Read all about it! NEWS.COM launches
2. Windows nightmares, how to win a digital camera, and other cool new gizmos
3. Saving you time and money: SHAREWARE.COM
4. Fun stuff: "The best game site on the Net"
5. TV: wired politics, 'zines, finding a home online, "The Web" goes local
6. SEARCH.COM: fun fix!
7. "Your turn": all the news that's fit to surf?
8. Top ten signs that you're a computer virus
9. Help wanted: travel coordinator, office manager, and more
10. Subscribe and unsubscribe

</TD>
<TD WIDTH="20%" VALIGN="TOP" ALIGN="LEFT">

1. READ ALL ABOUT IT! NEWS.COM LAUNCHES

If you missed our special Digital Dispatch announcing the launch of CNET's groundbreaking new site, NEWS.COM, log in and check it out. NEWS.COM is the first online, free, up-to-the-minute news source devoted entirely to technology. If you need to know what's happening in computing, on the Net, and in technology business, NEWS.COM is what you need. Updated as news occurs throughout the day, NEWS.COM is presented using the absolute latest Internet multimedia technologies. See for yourself:

</TD>

<TD WIDTH="20%" VALIGN="TOP" ALIGN="LEFT">

2. WINDOWS NIGHTMARES, HOW TO WIN A DIGITAL CAMERA, AND OTHER COOL NEW GIZMOS

You wake in the middle of the night. You can't breathe. Your pulse is racing. Fresh in your memory is an apparition of pure horror that came to you in your sleep. Was it Freddy of Elm Street? Jason on a bad-luck Friday? The spectacle of yourself on-stage in front of everyone you know...without your pants? No, it's worse. It's Windows 95. It's lost files, a deadly virus, a corrupt registry...CNET has the soothing wisdom to get you through the night. Because it's not just a bad dream.

</TD>

</TR>

</TABLE>

</BODY>

</HTML>

Das Ergebnis ist ein dreispaltiger Text mit zentrierter Überschrift nebst Unterzeile. Das Problem ist, daß die drei Schenkel des »Artikels« unterschiedlich lang sind und ihre Länge sich mit der Breite des Fensters im Browser ändert: Die Werte sind schließlich prozentual angegeben. (Wieso »width=20%«, wo doch 3x20 nur 60 ergibt? Weil es rechts, links und zwischen den Spalten auch noch Raum gibt!) Ein »richtiges« Layout läßt sich erreichen, indem statt der % feste Werte eingegeben und die <TD>-Tags durch Kontrolle im Browser an den entspr. Stellen gesetzt werden.

Rahmen

Jeder Browser kann, genügend Arbeitsspeicher vorausgesetzt, mehrere HTML-Dokumente gleichzeitig mit voneinander unabhängigen Fenstern öffnen. Mit dem Setzen von Rahmen (»frames«) dagegen kann ein Fenster in mehrere Abteilungen für separate Dokumente gegliedert werden. Mit der Einführung des »framing« wird zwischen zwei Dokumenttypen unterschieden:

- (a) Frameset Documents enthalten (lediglich) die Informationen, die der Browser für das Rahmenlayout benötigt, sowie die URL, die auf das Dokument hinweist, das den jeweiligen Rahmen füllt.

- (b) Body Documents beschreiben den Inhalt eines Rahmens, sind also »herkömmliche« HTML-Dokumente wie bisher diskutiert.

Während <frameset> nur die Existenz von Rahmen angibt, erfolgt deren konkrete Definition mit dem <frame>-Tag; <frame> ist ein singulärer Tag, d.h. er benötigt keinen </frame>-Tag.

Rahmen-URL

Nur wer leere Rahmen äußerst interessant findet, verzichtet auf deren Inhalt. Alle anderen ergänzen den Tag durch das Attribut »SRC="URL"«, wobei die URL wie gewohnt die »Adresse« des Dokumentes ist, das den Rahmen füllen soll.

Rahmen-Name

(Fast) jeder Rahmen sollte mit einem Namen versehen werden; dies erfolgt durch das Attribut »NAME="frame_name"« im <frame>-Tag. Im Beispiel erhielt der obere Rahmen keinen, der untere jedoch heißt »bottom«. Diese Benennung macht es möglich, den Rahmen anzuzielen (»target«), wenn mehrere Rahmen durch HyperLinks »kommunizieren« sollen.

Rahmen anzielen (Targeting Frames)

HyperLinks in Rahmen

In einem »normalen« HTML-Dokument stellt ein aktivierter (Hyper)Link zu einem anderen Dokument her, das gegenwärtige Browser-Fenster wird durch das neue *völlig* ersetzt.

Bei Rahmen ist dies anders: Ein Link kann den Browser veranlassen,

- * den Rahmen mit dem Link oder
- * einen anderen Rahmen oder
- * eine Rahmengruppe oder
- * das gesamte Browser-Fenster zu modifizieren.

HyperLink-Ziele (Targets)

Damit der HTML-Autor spezifizieren kann, welche der o.a. Aufgaben ein Link erfüllt, gibt es das Stichwort »target«. Damit ist es nicht nur möglich zu bestimmen, *welches* Dokument geladen werden soll, sondern auch, *wo* – also bei welchem Ziel.

Ein solcher Link hat dieses Format:

```
<A HREF=Dokument_URL TARGET="Target_Name"> HyperLink-Text </A>
```

Der variable Target-Name kann sein:

- (a) Ein expliziter Name, der einem Rahmen in einem Frameset Document gegeben wurde, wie »bottom« in unserem Beispiel oben
- (b) Ein impliziter Name, der durch die Verbindung des Zielrahmens zum Rahmen mit dem Link bestimmt wird.

Implizite Namen sind reservierte Wörter, die mit einer Unterstreichung (_) beginnen; einige sind:

<u>_self</u>	veranlaßt den Browser, den Rahmen, der den Link enthält, neu aufzubauen. Dies ist gewöhnlich die Grundeinstellung, wenn kein Ziel spezifiziert ist.
<u>_parent</u>	veranlaßt den Browser, den »parent frame« (s.o.) neu aufzubauen, sofern der Rahmen, der den Link enthält, ein »child« (verschachtelter Rahmen) ist. Andernfalls erzeugt <u>_parent</u> denselben Effekt wie <u>_self</u> .
<u>_top</u>	veranlaßt den Browser, das gesamte Fenster unabhängig vom gegenwärtigen Rahmen-Layout neu aufzubauen.

Rahmen-Namen

Wenn alle Links implizite Namen zur Angabe des Ziels benutzen, gäbe es nicht die Notwendigkeit, Rahmen bestimmte Namen zuzuordnen. Es empfiehlt sich auch, wo immer möglich, implizite Namen zu verwenden. Aber es gibt Fälle, in denen ein Rahmen keinen Zugang zu einem anderen über einen impliziten Namen hat. So ist es u.U. nötig, das Ziel mit einem zugeordneten Namen zu spezifizieren.

Jedem Rahmen kann mit dem Attribut »NAME="Rahmen-Name"« im <frame>-Tag ein Name gegeben werden.:

Frame-Tags und ihre Elemente

Zusammenfassend und ergänzend noch ein Blick auf die wesentlichen Tags bei der Arbeit mit Rahmen:

<FRAMESET> kennt 2 Attribute, »rows« (Reihen) und »cols« (Spalten); beide haben dieselben Ergänzungen (»values«) hinsichtlich ihrer Höhe bei den Reihen bzw. der Breite bei den Spalten. Es sind dies:

value n	n steht für einen einfachen numerischen Wert, der die feste Größe in Pixeln vorgibt (wir erinnern uns: 3 Pixel entspr. ca. einem Millimeter). Dies wäre die »gefährlichste« Angabe, da die Bildschirmgrößen der Betrachter erheblich variieren können.
value %	der eingegebene Prozentwert bezieht sich auf die Fenstergröße des Dokuments auf dem Bildschirm des Betrachters. Ist die Summe der Werte größer als 100, wird sie auf 100 heruntergerechnet; ist sie kleiner, werden die Räume zwischen den Rahmen entspr. vergrößert.
value *	bezeichnet eine relative Rahmengröße, die vom Browser so interpretiert wird, daß er den größtmöglichen Raum im Fenster zur Verfügung stellt .

Bei mehreren Reihen oder Spalten werden die Werte durch Kommas getrennt.

Bei verschachtelten Rahmen nehmen die »child«-Rahmen zusammen den Raum ein, den der nicht unterteilte Rahmen alleine einnehmen würde.

Der <frame>-Tag kennt 6 mögliche Attribute:

SRC="URL" bezeichnet die Adresse des Dokumentes, das den Inhalt des gemeinten Rahmens darstellen soll. Rahmen ohne SRC bleiben leer.

NAME="Rahmen-Name" wird benutzt, um den Rahmen mit Links in anderen Dokumenten (gemeinhin aus anderen Rahmen desselben Dokuments) anzuzielen. Das Attribut ist optional. Die Namen müssen mit einem alfanumerischen Zeichen beginnen.

MARGINWIDTH="value" wird optional angegeben, wenn der HTML-Autor einen horizontalen Rand zwischen Rahmengrenze und Rahmeninhalt definieren will; andernfalls präsentiert der Browser einen geeigneten Abstand.

MARGINHEIGHT="value" das entsprechende für die vertikalen Abstände.

SCROLLING="yes|no|auto" definiert optional, ob ein Rahmen mit einem Rollbalken versehen wird. »yes« sorgt dafür daß immer (wenn nötig), »no« daß niemals ein Rollbalken sichtbar wird. »auto« läßt den Browser entscheiden, ob er nötig ist; ist aber auch die Grundeinstellung.

NORESIZE dies optionale Attribut hat keinen »value«. Es zeigt an, daß der Benutzer die Rahmengröße entgegen der Grundeinstellung der Browsers nicht verändern kann. »noresize« kann auch Auswirkungen auf die Modifikationsmöglichkeiten anderer Rahmen im selben Dokument haben.

Das Tag-Paar <noframes> und </noframes> bildet einen »Container«, dessen Inhalt von Browsers angesprochen wird, die keine Rahmen darstellen können. Browsers, die dazu in der Lage sind, ignorieren alle Tags und Inhalte darin. Ganz »höfliche« HTML-Autoren verwenden diese Möglichkeit, um auch Besitzer solcher alter Software in den Genuß ihres Dokumentes kommen zu lassen – wenn auch eben beschränkt. Die Tags werden nach dem letzten </frameset> gesetzt, also:

```
.....  
</FRAMESET>  
<NOFRAMES>  
<BODY>  
.....  
</BODY>  
</NOFRAMES>  
</HTML>
```

Praktische Übung mit Rahmen

Nach allem Grundsätzlichen eine kleine Übung für ein Rahmendokument.

Zunächst wird das Frameset Document »Target-Versuch.htm« angelegt:

```
<HTML>  
<HEAD>  
<TITLE>Target-Versuch.htm</TITLE>  
</HEAD>  
<FRAMESET COLS="80,*">  
<FRAMESET ROWS="150,*">  
<FRAME SRC="Banner.htm" NORESIZE SCROLLING="NO">  
<FRAME SRC="Inhaltsverzeichnis.htm" NORESIZE>  
</FRAMESET>  
<FRAMESET ROWS="*,30">  
<FRAME SRC="Info-Fenster.htm" NAME="Fenster">  
<FRAME SRC="Toolbar.htm" NORESIZE SCROLLING="NO">  
</FRAMESET>  
</FRAMESET>  
</HTML>
```

Die nun gewählte Reihenfolge ist prinzipiell egal.
Das nächste ist das »Banner.htm«:

```
<HTML>  
<HEAD>  
<TITLE>Banner.htm</TITLE>  
</HEAD>  
<BODY>  
<H4 ALIGN="CENTER">Hier sollte eine kleine Graphik zur Corporate Identity  
beitragen</H4>  
</BODY>  
</HTML>
```

Es folgt das Dokument »Inhaltsverzeichnis.htm«:

```
<HTML>
<HEAD>
<TITLE>Inhaltsverzeichnis.htm</title>
</HEAD>
<BODY>
<H2>Inhalt</H2>
<A HREF="Text_1.htm" TARGET="Fenster"> Inhalt 1 </A><BR>
<A HREF="Text_2.htm" TARGET="Fenster"> Inhalt 2 </A><BR>
</BODY>
</HTML>
```

Daraufhin die »Text 1.htm« und »Text 2.htm«:

```
<HTML>
<HEAD>
<TITLE>Text_1.htm</TITLE>
</HEAD>
<BODY>
Dies ist der Inhalt des Fensters f&uuml;r den ersten Text.
</BODY>
</HTML>
```

```
<HTML>
<HEAD>
<TITLE>Text_2.htm</TITLE>
</HEAD>
<BODY>
Dies ist der Inhalt des Fensters f&uuml;r den zweiten Text.
</BODY>
</HTML>
```

Es folgt der Rahmen »Toolbar.htm«:

```
<HTML>
<HEAD>
<TITLE>Toolbar.htm</TITLE>
</HEAD>
<BODY>
<H3 ALIGN="CENTER">Und hier eine E-Mail-Adresse zum Anklicken und
Zum&uuml;llen.</H3>
</BODY>
</HTML>
```

Und schließlich der Hauptrahmen »Info-Fenster.htm«:

```
<HTML>
<HEAD>
<TITLE>Info-Fenster.htm</TITLE>
</HEAD>
<BODY>
<h1 ALIGN="CENTER">Willkommen im Target-Versuch.</H1>
</BODY>
</HTML>
```

28.5.1997
(rev. 15.12.05)
Dr. Michael Meissner