

Eine Homepage selbst erstellen

Inhalt:

- | | |
|---|--|
| <ul style="list-style-type: none">• Hypertext Markup Language• Der Aufbau von HTML-Befehlen• Die Grundstruktur von HTML-Dokumenten• Zeilenumbrüche• Schriftformatierung• Texthervorhebungen• Überschriften• Ausrichtung des Textes | <ul style="list-style-type: none">• Hintergrund• Trennlinien• Grafiken einbinden• Mit LINKS Verbindungen herstellen• Tabellen in HTML• Die Frame-Technik• Listen in HTML• Formulare gestalten |
|---|--|

Hyper Text Markup Language(HTML)

Die einfache Programmiersprache HTML ist die Standardsprache im World Wide Web. Mit ihr lassen sich Texte und andere Elemente wie Grafiken oder Tabellen einfach auf einer Seite formatieren. Zudem können Hyperlinks, also Querverweise, auf andere Web-Seiten verweisen.

Der Vorteil eines HTML-Dokumentes besteht darin, dass dieses Dokument auf jedem Computertyp mit einer beliebigen Textverarbeitung erzeugt bzw. bearbeitet werden kann.

Der Aufbau von HTML-Befehlen

HTML-Befehle (Tags) bestehen meistens aus einem Starttag und einem Endtag. Sie werden in spitze Klammern(<>) gefasst. Ein Starttag enthält nur das Befehlswort in seinen Spitzklammern; ein Endtag enthält zusätzlich einen Schrägstrich oder Slash (/) vor dem Befehlswort. Beispiel: Text (Der Text zwischen den Tags wird „fett“ gesetzt.)

Die Grundstruktur von HTML-Dokumenten

Ein Dokument besteht aus zwei Bereichen:

Der Kopf (head), in dem einige grundsätzliche Definitionen stehen; vor allem auch der Text, der in der Titelzeile des Fensters dargestellt wird.

Der Textkörper (body); er enthält den darzustellenden Text, Grafiken, Animationen und Java-Applets.

```
<html>
  <head>
    <title>
      hier steht der Titel
    </title>
  </head>
  hier ist der Kopfteil zu Ende
  <body>
    hier steht der eigentliche Inhalt der Seite
  </body>
</html>
```

Zeilenumbrüche

Zeilenumbruch sowie Leerzeilen werden per Tag eingegeben. Das gewohnte Betätigen der Return-Taste führt nicht zum Ziel, Returns werden ignoriert. Statt dessen werden

 (Break; zu deutsch: Zeilenumbruch) (gibt das Ende einer Zeile vor)

und

<p> (Paragraph; auf deutsch: Absatz) (erzeugt zusätzlich eine Leerzeile)

benutzt.

Schriftformatierung

Das Kommando für die Schriftformatierung lautet „font“.

Schriftgröße:

Schriftgröße 2

Es gibt sieben verschiedene Schriftgrößen (1-7), wobei 3 die Standardgröße ist.

Schriftfarbe:

Roter Text

Schriftart:

Arial-Schrifttyp

Es lassen sich auch mehrere Attribute gleichzeitig verwenden:

Dieser rote Text hat die Standardgröße 3 und ist in Times New Roman geschrieben

Texthervorhebungen

Der Text erscheint fett.

<i>Der Text erscheint kursiv.</i>

<u>Der Text erscheint unterstrichen.</u>

<s>Der Text erscheint durchgestrichen.</s>

<blink>Der Text erscheint blinkend.</blink>

Der Text erscheint<big>etwas größer</big> als normal.

Der Text erscheint<small>etwas kleiner</small> als normal.

Der Text erscheint_{tiefgestellt.}

Der Text erscheint^{hochgestellt.}

Überschriften

Man kennt in HTML sechs verschiedene Überschriften: <h1>...</h1> bis <h6>...</h6>, wobei h6 die kleinste Überschrift darstellt.

Ausrichtung des Textes

Mit dem Attribut ALIGN kann man Texte unterschiedlich ausrichten.

<h1 align=left>Linksbündig</h1>

<h1 align=center>zentriert</h1>

<h1 align=right>rechtsbündig</h1>

Hintergrund

Die Farbe für den Hintergrund des Dokumentes lässt sich verändern. Sie muss beim body-Tag eingegeben werden.

<body bgcolor=white> Weißer Hintergrund

Auch lässt sich ein Hintergrundbild einfügen.

<body background="tropen.gif"> Einfügen des Bildes tropfen.gif

Trennlinien

Eine einfache Trennlinie über die gesamte Bildschirmbreite wird durch das Tag <hr> dargestellt.

Breite und Höhe der Trennlinie lassen sich vorgeben, wobei WIDTH die Länge in Prozent des Anzeigenfensters oder in Pixel (width=50% oder width=80), während SIZE die Breite oder Dicke in Pixel angibt.

<hr width=40% size=4>

<hr width=60 size 3>

Die Trennlinien lassen sich auch ausrichten.

<hr width=50 align=right>

Grafiken einbinden

Zum Einbinden einer Grafik benutzt man das -Tag. (für image; zu deutsch: Bild) und als Attribut „src“(source; zu deutsch: Quelle). Die Grafik muss im JPEG- oder GIF-Format vorliegen.

 (Die Grafik „bild.gif“ wird in das Dokument eingebunden, wobei vorausgesetzt wird, dass die Grafik im selben Verzeichnis steht wie das Dokument)

 (Wenn die Grafik bild.gif z.B. auf der Festplatte im Verzeichnis \bilder gespeichert ist.)

Die Größe der Grafik kann verändert werden; falls nur die Breite(Höhe) geändert wird, passt sich die Höhe(Breite) automatisch an. Man sollte nicht beide Attribute zusammen verwenden, da sonst die Grafik mit ziemlicher Sicherheit gestaucht oder gestreckt wird.

oder

Um die Grafik kann auch ein Rahmen gelegt werden, was mit dem Attribut BORDER erreicht wird, wobei sich noch die Rahmendicke festlegen lässt.

Selbstverständlich lässt sich auch der Abstand zwischen einer Grafik und der übrigen Umgebung bestimmen. Das Attribut VSPACE bezeichnet den Abstand zum vorhergehenden und nachfolgenden Text, das Attribut HSPACE den Abstand zum linken Rand.

Mit dem Attribut ALIGN lässt sich die Grafik auch rechts oder links zum Text ausrichten.

Mit LINKS Verbindungen herstellen

Verweise sind ein Kernstück und Grundmerkmal von HTML-Dokumenten. Sie definieren Querverbindungen(Hyperlinks) zu anderen Dokumenten, welche auf einem beliebigen WWW-Server abgelegt sind, oder zu anderen Stellen innerhalb desselben Dokumentes bzw. zu anderen Dokumenten der Website.

Ein Link wird prinzipiell mit dem Tag <a> eingeleitet und mit abgeschlossen. Der Text, der zwischen diesen beiden Tags steht, wird unterstrichen dargestellt, d.h. er kann angeklickt werden.

Verweis auf eine andere Seite der Website

Will man auf ein anderes Dokument seiner Website springen und liegt das Dokument im selben Verzeichnis, kann man das gewünschte Dokument mit folgender Adressierung ansprechen:

`Textname`

z.B. `Zur Klasse 9b`

Hinter dem Attribut HREF(**H**ypertext **R**eference) steht die Adresse(URL=Uniform Resource Locator) des Dokumentes.

Verweis auf ein Dokument einer anderen Website

Will man von seiner eigenen HomePage eine andere ansprechen, lautet die Adresse(URL) z.B. so:

`Deutsche Bundesbahn`

Lokaler Verweis

Wenn man von einer Stelle im Dokument zu einer anderen desselben Dokumentes springen möchte, muss man einen *Verankerungspunkt* setzen. Dazu wird dem Tag `<a> ... ` das Attribut NAME beigefügt.

Ein Beispiel: Man will in einem langen Text am Ende wieder zum Anfang springen.

```
<html>
  <head>
    <title>....</title>
  </head>
  <body>
    <a name="anfang"></a>
    Hier ist der Anfang des Textes
    ...
    ...
    ...
    <a href="#anfang">Zurück zum Anfang</a>
  </body>
</html>
```

Ein anderes Beispiel: Man hat vor einen langen Text ein Inhaltsverzeichnis gesetzt und will nun schneller zu den einzelnen Abschnitten gelangen.

```
<html>
  <head>
    <title>....</title>
  </head>
  <body>
    Inhalt:
    <a href="#einleit">Einleitung</a>
    <a href="#ueber">Zusammenfassung</a>
    <a href="#schluss">Schluss</a>
    ...
    <a name="einleit">Einleitung</a>
    ...
```

```
<a name="ueber">Zusammenfassung</a>
...
<a name="schluss">Schluss</a>
</body>
</html>
```

E-Mails schicken lassen

Ohne einen E-Mail-Link wäre die schönste HomePage nicht komplett. Schließlich möchte man ja auch Post bekommen.

Will man, dass ein E-Mail aus der WWW-Seite heraus geschickt wird, sieht das notwendige Tag dazu z.B. folgendermaßen aus:

```
<a href="mailto:fritz.finster@t-online.de">E-Mail an mich</a>
```

Hyperlinks hinter Grafiken legen

Selbstverständlich ist es auch möglich, einen Hyperlink hinter eine Grafik zu legen.

Beispiel: Man möchte, um auf das Dokument HOBBY.HTM zu springen, auf die Grafik BILD.GIF anklicken.

```
<a href="hobby.htm"></a>
```

Tabellen in HTML

Bei allen Darstellungen, bei denen Texte und Grafiken an bestimmte Bildschirmpositionen gestellt werden sollen, ist die Verwendung des Tags `<table>` (zu deutsch "Tabelle") notwendig.

Eine Tabelle besteht immer aus einzelnen Zellen. In die einzelnen Zellen lassen sich beliebige HTML-Elemente einsetzen, z.B. Texte, Links und Grafiken.

- Das Grundgerüst beginnt mit dem einleitenden Tag `<table>`.
- Jede Zeile wird mit `<tr>` (für "table row"/Tabellenzeile) eingeleitet.
- Man unterscheidet noch zwischen Zellen im Tabellenkopf für Spaltenbeschriftungen und Zellen im Tabellenrumpf für Daten.
- Zellen des Kopfes werden zwischen `<th>` und `</th>` definiert.
- Jede Zelle im Tabellenrumpf wiederum beginnt mit dem Tag `<td>` ("table data"), der in aller Regel mit dem Attribut `WIDTH` ("Breite") angegeben wird. (Beispiel: `<td width=200>` d.h. Zelle mit einer Breite von 200 Pixeln). Die Zelle wird mit `</td>` abgeschlossen.
- Hat man alle Zellen einer Zeile aufgeführt, wird die Zeile mit `</tr>` beendet.
- Hat man alle Zeilen fertig, so endet die Tabelle mit `</table>`.

Beispiel:

```
<table>
  <tr>
    <th>Kopfzeile, 1.Spalte</th>
    <th>Kopfzeile, 2.Spalte</th>
    <th>Kopfzeile, 3.Spalte</th>
  </tr>
  <tr>
    <td width=50>Zelle 1</td>
    <td width=60>Zelle 2</td>
    <td width=50>Zelle 3</td>
  </tr>
  <tr>
    <td width=50>Zelle 4</td>
    <td width=60>Zelle 5</td>
    <td width=50>Zelle 6</td>
  </tr>
</table>
```

Will man das Gitternetz der Tabelle sehen, fügt man in den <table>-Tag das Attribut "border" ein; z.B. <table border=3>.

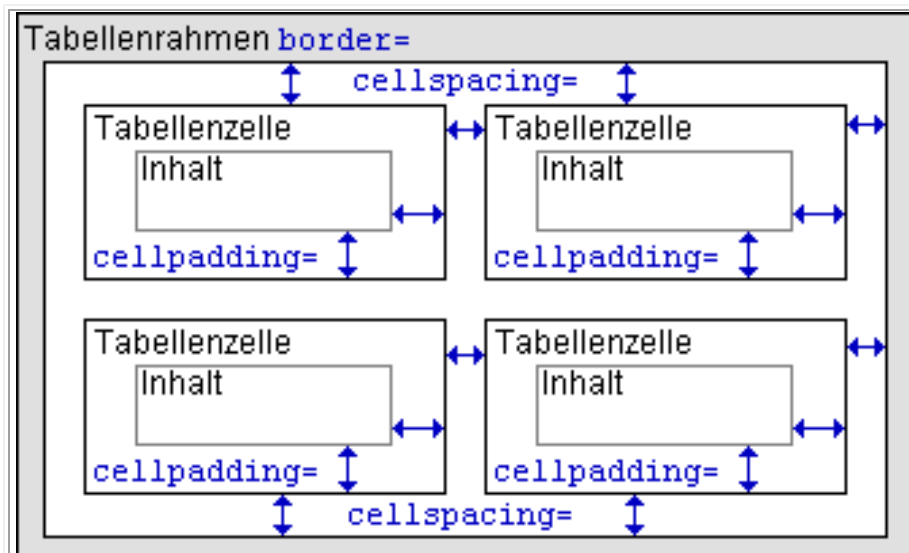
Auch zum Einrücken eines Textes bietet sich ein <table>-Tag an. Es muss dazu eine Leerzellen definiert werden. Beispiel:

```
<table border=1>
  <tr>
    <td width=50></td>
    <td width=200>Hallo, hier bin ich.</td>
  </tr>
</table>
```

Für die Tabellentags gibt es eine ganze Reihe von Attributen. Hier die wichtigsten:

- Attribute von <table>

align	Ausrichtung der Tabelle (center, left oder right). Normalerweise richtet sich die Ausrichtung nach der Textausrichtung.
bgcolor	Hintergrundfarbe der Zellen definieren.
border	Breite des Außenrahmens in Pixeln. Mit border=0 wird eine „blinde“ Tabelle erzeugt.
cellpadding	Abstand des Zelleinhaltes vom Zellenrand in Pixeln.
cellspacing	Breite der Gitternetzlinien in Pixeln.



height	Höhe der Zellen in Pixeln oder in Prozent der Anzeigegröße.
width	Breite der Zellen in Pixeln oder in Prozent der Anzeigegröße.

- Attribute von <tr>

align	Ausrichtung des Inhalte aller Zellen der Zeile.
bgcolor	Hintergrundfarbe aller Zellen der Zeile.
nowrap	Automatischen Zeilenumbruch in den Zellen verhindern.
valign	Vertikale Ausrichtung der Inhalte aller Zellen der Zeile. Werte sind top, bottom oder center.

- Attribute von <td> und <td>

align	Ausrichtung der Zelle.
bgcolor	Hintergrundfarbe der Zelle.
colspan	Verbinden mehrerer Zellen in einer Reihe zu einer einzigen. Der Wert gibt die Anzahl der zu verbindenden Zellen an.
height	Zellenhöhe in Pixeln oder Prozent der Anzeigegröße. Die Zellenhöhe wird automatisch für die anderen

	Zellen der Zeile übernommen.
nowrap	Automatischer Zeilenumbruch in der Zellen verhindern.
rowspan	Verbinden mehrerer Zellen in einer Spalte zu einer einzigen. Der Wert gibt die Anzahl der zu verbindenden Zellen an.
valign	Vertikale Ausrichtung der Zelle.
width	Breite der Zelle in Pixeln oder in Prozent der Anzeigegröße. Die Breite wird automatisch für die anderen Zellen der Spalte übernommen.

Außerdem gibt es noch den `<caption>`-Tag, mit dem Tabellenunter- bzw. überschriften definiert werden. Er besitzt ein Attribut `align`, mit dem angegeben wird, an welcher Stelle die Beschriftung erscheinen soll. Möglich sind `top` für eine Überschrift, `bottom` für eine Unterschrift. Auch `left` und `right` sind möglich.

Beispiel:

```
<table border=1 align=right>
<caption align=bottom>Manschaftsaufstellung</caption>
  <tr bgcolor=blue>
    <th>Name</th>
    <th>R&uuml;cknummer</th>
  </tr>
  <tr>
    <td>Enke</td>
    <td>1</td>
  </tr>
</table>
```

Tabellen mit COLSPAN und ROWSPAN

11 und 12		13
21	22	23

```
<table border="1" cellspacing="0" cellpadding="10">
  <tr>
    <td colspan="2">11 und 12</td>
    <td>13</td>
  </tr>
  <tr>
    <td>21</td>
    <td>22</td>
    <td>23</td>
  </tr>
</table>
```

11	12
21	22 und 32
31	

```
<table border="1" cellspacing="0" cellpadding="10">
  <tr>
    <td>11</td>
    <td>12</td>
  </tr>
  <tr>
    <td>21</td>
    <td rowspan="2">22 und 32</td>
  </tr>
  <tr>
    <td>31</td>
  </tr>
</table>
```

Die Frame – Technik

Mit Frames (Rahmen) hat man die Möglichkeit, im Fenster des Browsers mehrere HTML-Dokumente gleichzeitig anzuzeigen.

Soll der komplette Bildschirm mit Frames aufgebaut sein, wird eine Steuerdatei(index.htm oder steuer.htm) geladen. Die Steuerdatei besteht in der Regel aus wenigen Zeilen, z.B.:

```
<html>
  <head>
    <title>Steuerdatei</title>
  </head>
  <frameset rows="30%,70%">
    <frame src="inhalt.htm" name="oben">
    <frame src="willkommen.htm" name="unten">
  </frameset>
</html>
```

Eine Frame-Definition leitet man mit dem Tag <frameset> ein und schließt diese mit dem Ende-Tag </frameset>. Durch die Angabe des Attributs "rows" wird der Bildschirm in zwei Frames aufgeteilt, und zwar horizontal. Ersetzt man "rows" durch "cols" kommt es zu einer Teilung in vertikaler Richtung. Die Prozentzahlen geben an, in welchem Größen-Verhältnis die beiden Rahmen den Bildschirm aufteilen.

Mit <frame src="..."> wird definiert, mit welchen Dateien die beiden Teile des Bildschirms gefüllt werden. Frames können Namen (z.B. name="oben") erhalten, was sich als sinnvoll erweist, wenn man z.B. unterschiedliche Dateien in einem Frame aufrufen können möchte.

Standardmäßig werden die Frame-Bereiche durch eine Trennlinie hervorgehoben. Will man diese ausblenden, fügt man folgendes Attribut hinzu: <frameset rows="30%,70%" frameborder=no>.

Durch das Ziehen mit der Maus kann man Trennlinien verschieben. Auch dies läßt sich unterbinden: `<frame scr="inhalt.htm" name="oben" noresize>`.

Sinnvoll ist es häufig, wenn noch ein dritter Frame erstellt wird, z.B. um eine Inhaltsseite anzuzeigen, von der aus man alle anderen Seiten aufrufen kann. Eine solche Steuerdatei könnte folgendermaßen aussehen:

```
<frameset rows="30%,70%">
  <frame src="logo.htm" name="oben">
  <frameset cols="20%,80%">
    <frame src="inhalt.htm" name="links">
    <frame src="willkommen.htm" name="rechts">
  </frameset>
</frameset>
```

Hat man z.B. im linken Frame ein Inhaltsverzeichnis seiner Website und möchte im rechten Frame die entsprechenden Seiten darstellen, so muss man ein Link angeben und diesem das Attribut "target="rechts" zuweisen:

```
<a href="hobby.htm" target="rechts">Meine Hobbyseite</a>
```

So könnte z.B. eine Inhaltsverzeichnis aussehen:

```
<html>
  <head>
    <title>Inhalt</title>
  </head>
  <body>
    <a href="welcome.htm" target="rechts">Homepage</a>
    <a href="familie.htm" target="rechts">Meine Familie</a>
    <a href="freunde.htm" target="rechts">Meine Freunde</a>
    <a href="hobby.htm" target="rechts">Meine Hobbies</a>
    <a href="adressen.htm" target="rechts">Interessante Adressen</a>
  </body>
</html>
```

Man kann sicherlich auch mehr als drei Frames definieren, doch sollte man bedenken, dass zu viele Frames den Bildschirm unübersichtlich erscheinen lassen.

Listen in HTML

Ungeordnete und geordnete Listen sind ein recht einfach zu beschreibender Teil von HTML. Deshalb gibt es auch hierzu **keine** Beispielseite.

Ungeordnete Liste UL

- Text A
- Text B
- Text C

Notiert wird es so:

```
<ul>
    <li>Text A</li>
    <li>Text B</li>
    <li>Text C</li>
</ul>
```

Den BULLET-Typ festlegen (Bullet ist der Aufzählungspunkt vor dem Listeneintrag)

- Ein Kreis
- ...
- Ein Viereck
- ...
- Ein gefüllter Kreis/Punkt
- ...

```
<ul type=circle> Ein Kreis
    <li>...
</ul>
<ul type=square>      Ein Viereck
    <li>...
</ul>
<ul type=disc> Ein gefüllter Kreis/Punkt
    <li>...
</ul>
```

Nummerierte Listen

1. Listeneintrag, bekommt "1." vorangestellt
2. Listeneintrag, bekommt "2." vorangestellt
3. Listeneintrag, bekommt "3." vorangestellt

```
<ol>
    <li>Listeneintrag, bekommt "1." vorangestellt</li>
    <li>Listeneintrag, bekommt "2." vorangestellt</li>
    <li>Listeneintrag, bekommt "3." vorangestellt</li>
</ol>
```

OL =geordnete Liste

Alphabetische Listen

- A. Listeneintrag, bekommt ein "A." vorangestellt
 - B. Listeneintrag, bekommt ein "B." vorangestellt
 - C. Listeneintrag, bekommt ein "C." vorangestellt
-
- a. Listeneintrag, bekommt ein "a." vorangestellt

- b. Listeneintrag, bekommt ein "b." vorangestellt
- c. Listeneintrag, bekommt ein "c." vorangestellt

```
<ol type=A>
  <li>Listeneintrag, bekommt ein "A." vorangestellt
  <li>Listeneintrag, bekommt ein "B." vorangestellt
  <li>Listeneintrag, bekommt ein "C." vorangestellt
```



```
<ol type=a>
  <li>Listeneintrag, bekommt ein "a." vorangestellt
  <li>Listeneintrag, bekommt ein "b." vorangestellt
  <li>Listeneintrag, bekommt ein "c." vorangestellt
```


<ol type=A> =Großbuchstaben

<ol type=a> =Kleinbuchstaben

Listen römisch nummerieren

- I. Listeneintrag, bekommt ein "I." vorangestellt
- II. Listeneintrag, bekommt ein "II." vorangestellt
- III. Listeneintrag, bekommt ein "III." vorangestellt
- IV. Listeneintrag, bekommt ein "IV." vorangestellt

- i. Listeneintrag, bekommt ein "i." vorangestellt
- ii. Listeneintrag, bekommt ein "ii." vorangestellt
- iii. Listeneintrag, bekommt ein "iii." vorangestellt
- iv. Listeneintrag, bekommt ein "iv." vorangestellt

```
<ol type=I>
  <li>Listeneintrag, bekommt ein "I." vorangestellt
  <li>Listeneintrag, bekommt ein "II." vorangestellt
  <li>Listeneintrag, bekommt ein "III." vorangestellt
  <li>Listeneintrag, bekommt ein "IV." vorangestellt
```



```
<ol type=i>
  <li>Listeneintrag, bekommt ein "i." vorangestellt
  <li>Listeneintrag, bekommt ein "ii." vorangestellt
  <li>Listeneintrag, bekommt ein "iii." vorangestellt
  <li>Listeneintrag, bekommt ein "iv." vorangestellt
```


Nummerierte Liste mit römischen Zahlen bei z. B. X(=10) beginnend

- X. Freunde treffen
- XI. Zur Schule gehen
- XII. chillen

```
<ol type="I">
  <li value=10>Freunde treffen </li>
  <li>zur Schule gehen</li>
  <li>chillen</li>
```


Listenstarteinträge bestimmen

Hierbei kann festgelegt werden mit welchem Startwert die Liste beginnt

7. Listeneintrag, bekommt "7." vorangestellt
8. Listeneintrag, bekommt "8." vorangestellt
9. Listeneintrag, bekommt "9." vorangestellt

10000. Listeneintrag, bekommt "10000." vorangestellt
10001. Listeneintrag, bekommt "10001." vorangestellt
10002. Listeneintrag, bekommt "10002." vorangestellt
10003. Listeneintrag, bekommt "10003." vorangestellt

```
<ol start=7>
  <li>Listeneintrag, bekommt "7." vorangestellt
  <li>Listeneintrag, bekommt "8." vorangestellt
  <li>Listeneintrag, bekommt "9." vorangestellt
  <li value=10000>Listeneintrag, bekommt "10000." vorangestellt
  <li>Listeneintrag, bekommt "10001." vorangestellt
  <li>Listeneintrag, bekommt "10002." vorangestellt
  <li>Listeneintrag, bekommt "10003." vorangestellt
</ol>
```

Farbige Listenpunkte

```
<html>
  <head>
    <title>Aufzählungszeichen farbig</title>
  </head>
  <body>
    <ul style="">
      <li style="color: green;">Sascha</li>
      <li style="color: blue;">Dennis</li>
    </ul>
  </body>
</html>
```

FORMULARE

Um ein Formular zu erzeugen, verwendet man das **<FORM>...</FORM>**-Tag. Innerhalb der einleitenden und abschließenden **<FORM>**-Tags befinden sich dann die individuellen Formular-Elemente plus aller anderer HTML-Inhalte wie z.B. Überschriften, Absätze, Tabellen usw., mit denen man dem Formular eine Struktur gibt.

Man kann beliebig viele Formulare auf einer Seite unterbringen, aber man kann sie nicht ineinander verschachteln, d.h. man kann keinen <FORM>...</FORM>-Tag innerhalb eines <FORM>...</FORM>-Tags benutzen.

Die einleitende Kennung des <FORM>-Tags enthält zwei Elemente: das Attribut **METHOD="..."**, das für die Übermittlung der Daten an den angegebenen Ort steht und das Attribut **ACTION="..."**, das den Namen des auszuführenden Programmes wiedergibt - hier wird angegeben, was mit den Daten geschehen soll, wenn der Leser das Formular abschickt.

Mit **METHOD="..."** gibt man die Übertragungsmethode an. Hier gibt es zwei Möglichkeiten:

- Mit **METHOD="GET"** werden die Daten des ausgefüllten Formulars auf WWW-Servern mit installiertem http-Protokoll in der Standard-Umgebungsvariablen QUERY_STRING gespeichert. Das CGI-Script kann den Inhalt der Umgebungsvariablen dann auslesen und weiterverarbeiten.
- Mit **METHOD="POST"** werden die Daten des ausgefüllten Formulars verschickt.

Lässt man sich Formulardaten per E-Mail zuschicken, benutzt man immer METHOD="POST". Außerdem sollte man bei E-Mail-Empfang immer die Angabe **ENCTYPE="TEXT/PLAIN"** mit angeben. Mit dieser Angabe erhält man ordentlich formatierte E-Mails, die einfach und schnell gelesen werden können.

Die Angabe innerhalb der Anführungszeichen bei **ACTION="..."** ist entweder eine E-Mail-Adresse, bei der das mailto: vorangestellt wird oder es ist der Aufruf eines CGI-Scripts, das die Angaben des Lesers im Formular weiterverarbeiten soll.

Hier zwei Beispiele:

```
<FORM METHOD="POST" ACTION="mailto:m.mustermann@t-online.de"
ENCTYPE="TEXT/PLAIN">...</FORM>
```

```
<FORM METHOD="POST" ACTION="http://www.???de/cgi-bin/post">...</FORM>
```

Einzeilige Eingabefelder

In einzeilige Eingabefelder kann man Zahlen oder einzelne Wörter eingeben.

Ein kurzes Beispiel:

```
<HTML>
  <HEAD>
    <TITLE>Wie lautet Ihr Name?</TITLE>
  </HEAD>
  <BODY>
    <H2>Wer sind Sie?</H2>
    <FORM METHOD="POST" ACTION="http://www.URL.de/cgi-bin/post">
      <P>Geben Sie hier bitte Ihren Namen ein: <INPUT TYPE= "TEXT"
NAME="Nachname" SIZE="60" MAXLENGTH="60"></P>
      <P>...</P>
    </FORM>
```

```
</BODY>  
</HTML>
```

<INPUT> kennzeichnet ein einfaches Formularelement. Es hat normalerweise wenigstens die beiden Attribute **TYPE="..."** und **NAME="..."**.

TYPE="..." beschreibt die Art des Formularelements. Es gibt dabei die Möglichkeit für **TEXT** für Textfelder, **RADIO** für Optionsfelder und **CHECK** für Auswahl-Felder. Wird das TYPE-Attribut weggelassen, so wird aus dem Feld normalerweise ein Textfeld.

Mit **NAME="..."** wird dem Eingabefeld ein interner Elementname gegeben. Der Name sollte nicht zu lang sein und darf auch keine Leerzeichen oder deutschen Umlaute oder Sonderzeichen enthalten.

SIZE="..." bezeichnet die Länge des Eingabefeldes in Zeichen; voreingestellt sind 20 Zeichen.

MAXLENGTH="..." ermöglicht es, die Anzahl der Zeichen zu begrenzen, die der Leser in das Textfeld eingeben kann. Überzählige Zeichen werden dabei zurückgewiesen. Ist MAXLENGTH= kleiner als SIZE=, zeichnen manche Browser ein Textfeld, das so groß ist wie MAXLENGTH=.

Textvorbelegung bei einzeiligen Eingabefeldern

Jedes Formularelement kann einen Vorgabewert haben, der automatisch eingegeben oder ausgewählt wird, wenn das Formular angezeigt wird:

- Bei Textfeldern kann das **VALUE="..."**-Attribut mit einer Zeichenfolge als Vorgabewert verwendet werden. Der Inhalt von VALUE wird automatisch eingegeben, wenn das Formular angezeigt wird.
- Bei Auswahl- und Options-Feldern sorgt das **CHECKED="..."**-Attribut dafür, dass das Feld automatisch ausgewählt wird.

Mehrzeilige Eingabefelder (Textbereiche)

Mehrzeilige Eingabefelder dienen zur Aufnahmen von größeren Textmengen, die über mehrere Zeilen gehen können. Möchte man beispielsweise ein Formular entwerfen, mit dem Leser einer Web-Seite E-Mails erstellen können, so könnte man einen Textbereich für den Inhalt der Nachricht verwenden.

Um einen Textbereich in ein Formular aufzunehmen, benötigt man den **<TEXTAREA>...</TEXTAREA>**-Tag. Dieser Tag enthält drei Attribute:

NAME="...": Der Name des Feldes, der dem Gateway-Script übergeben wird, wenn der Formularinhalt übermittelt wird. Der Name sollte nicht zu lang sein und darf keine Leerzeichen, Sonderzeichen oder deutsche Umlaute enthalten.

ROWS="...": Die Höhe des Textbereichs, im Formular angegeben in Textzeilen.

COLS="...": Die Breite des Textbereichs, im Formular angegeben in Zeichen.

Die Angaben ROWS= und COLS= bestimmen lediglich die Anzeigegröße des Textfeldes, nicht aber die Länge des erlaubten Textes. Diese Länge ist unbegrenzt.

Beispiel:

```
<TEXTAREA NAME="COMMENT" ROWS="15" COLS="50">Hier können Sie Ihre  
Kommentare abgeben.</TEXTAREA>
```

Radio-Buttons

Options-Felder kennzeichnen eine Liste von Einträgen, von denen nur einer ausgewählt werden kann. Wird eines dieser Felder aus der Liste gewählt, werden alle anderen Felder der Liste deaktiviert.

Options-Felder verwenden **RADIO** als ihr TYPE-Attribut. Zusammengehörnde Gruppen solcher Felder kennzeichnet man, indem man dasselbe NAME-Attribut für alle benutzt. Darüber hinaus muss jedes Optionsfeld ein eigenes VALUE="..."-Attribut haben, das die jeweilige Auswahl bezeichnet.

```
<OL>
```

```
  <INPUT TYPE="RADIO" NAME="AGE" VALUE="MGLK1">unter 18<BR>
```

```
  <INPUT TYPE="RADIO" NAME="AGE" VALUE="MGLK2">über 18<BR>
```

```
</OL>
```

In der Voreinstellung sind alle Options-Felder ausgeschaltet. Da man bei Optionsfeldern im Allgemeinen aber immer eine Auswahlmöglichkeit hat, wäre es gut, wenn man eines der Felder per Voreinstellung aktiviert. Dies geschieht mit dem CHECKED-Attribut.

```
<OL>
```

```
  <INPUT TYPE="RADIO" NAME="AGE" VALUE="MGLK1">unter 18<BR>
```

```
  <INPUT TYPE="RADIO" NAME="AGE" VALUE="MGLK2" CHECKED>über 18<BR>
```

```
</OL>
```

Wird der Formularinhalt übermittelt, so wird dem Script ein einzelnes Name/Wert-Paar für die Felder-Gruppe übergeben. Dieses Paar enthält das NAME-Attribut der ausgewählten Gruppe.

Check-Boxes

Mit Auswahlfeldern kann man mehrere Einträge zugleich in einer Liste auswählen. Auswahlfelder benutzen **CHECKBOX** als TYPE-Attribut.

```
<OL>
```

```
  <INPUT TYPE="CHECKBOX" NAME="ROT">ROT<BR>
```

```
  <INPUT TYPE="CHECKBOX" NAME="BLAU">BLAU<BR>
```

```
</OL>
```

