Einführung in html

Das Grundgerüst:

```
1 <!doctype html public "-//W3C//DTD HTML 4.0 //EN">
2 <html>
3 <head>
4 <title>Text des Titels</title>
5 </head>
6 <body text="#000000" bgcolor="#FFFFFF" link="#FF0000" alink="#FF0000"
       vlink="#FF0000">
7 </body>
8 </html>
```

Erläuterung der einzelnen Zeilen:

```
1        kennzeichnet den Typ von html - Version, der verwendet wird.
2,8      Beginn und Ende des html - Textes.
3,5      Beginn und Ende des Kopfes. Hier stehen allgemein für die Seite
         gültige Anweisungen.
4        Beginn und Ende des Titels. Der Titel wird in der Fenster - Leiste
         am oberen Rand des Fensters dargestellt.
6,7      Beginn und Ende des (Text- usw.) - Körpers der html - Seite,
         dazwischen im Körper steht der Inhalt der html - Seite.
```

tags

Die Definition der Anweisungen, mit denen das Aussehen der Seite durch html beschrieben wird erfolgt in den ***tags***.
Grundsätzlich gilt, dass alle tags paarweise auftreten, also aus einem ***öffnenden tag*** *(z.B.<title>)* und einem ***schließenden tag*** *(z.B.</title>)* bestehen. Zusätzliche besondere Anweisungen *(Attribute, Eigenschaften)* stehen im öffnenden tag *(z.B. im body – tag die Definitionen der Farben von Text und links)*. Das schließende tag ist durch den Schrägstrich („slash“) vor dem Namen gekennzeichnet.
Einige tags sind von ihrem Zweck her nicht paarweise sinnvoll. Das sind z.B. das tag für den Zeilenumbruch und das tag für eine horizontale Trennlinie. Im Hinblick auf die Erweiterung des html – Standard auf den xhtml – Standard sollte man hier in jedem Fall am Ende des tags einen Schrägstrich mit einbauen. Derzeit können alle ***browser*** (= Programme zur Darstellung von html – Seiten) beide Varianten erkennen. Also:

```
zwangsweiser Zeilenumbruch   <br />
horizontale Trennlinie       <hr />
```

<body> das tag für den „Körper“ der Seite

Der Köper einer html – Seite wird in der Regel eine oder mehrere Überschriften haben und nachfolgende Absätze, die sowohl Text als auch Bilder als Inhalte haben. Die entsprechenden tags sind:

```
Überschriften:               <h1> ...<h6>
Absätze                      <p>
```

Attribute zur Ausrichtung

Schon diese Grundelemente lassen sich aber mit vielen zusätzlichen Eigenschaften gestalten. Bei den Überschriften sind natürlich die Ausrichtungen besonders wichtig. Standardmäßig sind Überschriften linksbündig ausgerichtet, man kann sie aber auch mit (Z.B.)

`<h1 align="center">` zentriert,
`<h1 align="right">` rechtsbündig,
`<h1 align="justify">` im Blocksatz ausrichten.
`<h1 align="left">` bedeutet linksbündig. Dies ist der Standard und braucht daher in der Regel nicht angegeben zu werden..
Dieselben Attribute sind auch in <p> - tags zulässig.

Grundlegende Gliederungselemente

<p> das tag für einen Absatz

Am Beispiel eines Steckbriefes aller Schülerinnen und Schüler des Kurses können wir die Anwendung grundlegender Gliederungselemente kennen lernen. Das Grundelement Absatz haben wir bereits benutzt. Der Text eines Absatzes wird durchgehend dargestellt. Wenn im html – Text Umbrüche vorhanden sind, werden sie genau so wie mehrere Leerzeichen oder Einrückungen als einzelnes Leerzeichen behandelt! Will man einen Umbruch in einem Absatz, muss man das `<br />` - tag verwenden.
Unbefriedigend ist dabei das Einfügen von Bildern: In den Text eingefügte Bilder werden als normales Zeilenelement aufgefasst. Da sie in der Regel ein deutlich größere Höhe als die Textzeile haben, wird der Text in eigenartiger Weise umgebrochen.
Wie man das besser machen kann, lernen wir später.

 das tag für eine Aufzählung

Ein häufig eingesetztes Gliederungselement in Texten sind Aufzählungen, wie sie gleich anschließend hier verwendet werden: Sie sind

- vielseitig anwendbar
- nützlich
- übersichtlich
- einfach einzusetzen

Ein solches Gliederungselement wird daher auch standardmäßig angeboten. Es lautet:

```
Aufzählung                    <ul>
```

wobei die aufgezählten Zeilen dann noch mit dem Listentag gekennzeichnet werden müssen:

```
Listenzeile                   <li>
```

Unser o.a. Beispiel lautet damit:

```
<ul>
        <li>vielseitig anwendbar</li>
        <li>nützlich</li>
        <li>übersichtlich</li>
        <li>einfach einzusetzen</li>
</ul>
```

Als einfaches Attribut kann das Aufzählungszeichen bestimmt werden. Möglich sind `type="disc"`, `square` oder `circle`, Standardwert *(default)* ist disc.

 das tag für eine Nummerierung

Das Symbol `<ul>` kommt von „ungeordneter Liste“, es gibt natürlich auch die „geordnete Liste“ `<ol>`, also eine Nummerierung. Machen wir das gleiche Spiel wie bei der o.a. Aufzählung:

1. vielseitig anwendbar
2. nützlich
3. übersichtlich
4. einfach einzusetzen

so lautet der html – Text dazu:

```
<ol>
        <li>vielseitig anwendbar</li>
        <li>nützlich</li>
        <li>übersichtlich</li>
        <li>einfach einzusetzen</li>
</ol>
```

Mögliche Attributwerte sind: `type="1", a, A, i` und `I`. Durch sie wird der Typ der Nummerierung gekennzeichnet. Standardwert ist 1. Zusätzlich kann noch der Startwert definiert werden.

<table> das tag für Tabellen

Grundsätzlich sind Tabellen natürlich dafür gedacht, „normale“ Tabellen wie die folgende darzustellen:

```
<table border>
<tr>
 <th colspan="2">Wochenplan    </th>
</tr>
<tr>
 <th>Wochentag </th>
 <th>Essen </th>
</tr>
<tr>
 <td>Montag </td>
 <td>Frikadellen </td>
</tr>
<tr>
 <td>Dienstag </td>
 <td>Hähnchen </td>
</tr>
...
</table>
```

Wochenplan	
Wochentag	Essen
Montag	Frikadellen
Dienstag	Hähnchen
Mittwoch	Nudeln
Donnerstag	Salat
Freitag	Fisch

Blinde Tabellen

Eine der häufigsten Anwendungen ist aber die sogenannte „blinde“ Tabelle, bei der die Tabelle keinen Rahmen hat und nur genutzt wird, um verschiedene Elemente auf einer Seite gezielt zu Positionieren. Dafür wieder eine Beispiel:

```
<table>
 <tr>
  <th width=100 height=80>linke Spalte </th>
  <th width=100 height=80>mittlere Spalte</th>
  <th width=100 height=80>rechte Spalte</th>
 </tr>
 <tr>
  <td width=100 height=80>  </td>
  <td align=center><img src="Bilder/oben.gif" width="33" height="28"
     border="0" alt=""> </td>
  <td width=100 height=80>  </td>
 </tr>
 <tr>
  <td align=center><img src="Bilder/links.gif" width="28" height="33"
     border="0" alt=""> </td>
  <td width=100 height=80>  </td>
  <td align=center><img src="Bilder/rechts.gif" width="28" height="33"
     border="0" alt=""> </td>
 </tr>
 <tr>
  <td width=100 height=80>  </td>
  <td align=center><img src="Bilder/unten.gif" width="33" height="28"
     border="0" alt=""> </td>
  <td width=100 height=80>  </td>
 </tr>
</table>
```

Zeile

Überschrift

Zelle

Zusätzlich wurden bei dieser Tabelle noch weitere **Attribute** eingesetzt. Das sind die Attribute

width=100	zum Einstellen der Breite einer Zelle,
height=80	zum Einstellen der Höhe einer Zelle und
align=center	um die Bilder mittig auszurichten.

Barrierefreiheit

Das Problem ist, dass diese Darstellung schlecht von Programmen zum Vorlesen der Bildschirminhalte verstanden wird, da die Tabelle nicht für ihren eigentlichen Zweck eingesetzt wird.

Formatvorlagen (css)

css bietet andere Möglichkeiten, Positionierungen zu erreichen und sollte nach Möglichkeit dafür heute genutzt werden.
Da dieses Werkzeug aber noch viel weiter reicht, werden wir es ausführlich noch später behandeln.

links

Ein link ist eine Verknüpfung zu einer anderen[1] Datei. Diese Verknüpfung wird durch den Mausklick aktiviert und der Server, auf den der link verweist (`href=`... s.u.) wird angewiesen, die Datei zu übermitteln.
Je nach dem Dateityp reagiert der browser dann unterschiedlich:
Ist die Datei eine html- oder htm-Datei, dann ersetzt er die aktuelle Seite durch die neu geholte. Falls es in dem link besondere Anweisungen dafür gab (`target=`...) kann die neue Seite auch in einem neuen Fenster oder einem speziellen Ausschnitt des Fensters (frame) dargestellt werden.

- Bei Dateitypen, zu denen der browser ein plugin (Hilfsfunktion für Fremde Anwendungen) installiert hat, wird dies gestartet und das plugin behandelt die Datei.
- Für andere Dateitypen fragt der browser nach, ob der Benutzer die Datei abspeichern möchte – und wenn er ordentlich konfiguriert ist – auch wohin und dann wird die Datei dort abgelegt.

Besonderheiten

Bilder als links

Anstelle von Texten können auch Bilder für links verwendet werden. Es ist sogar möglich, besondere Bereiche in Bildern (hot-spots) zu definieren, die als links verwendet werden. Beispielsweise kann man eine Landkarte verwenden und links für einige der Orte definieren.

Anker

Wird im href – Attribut ein leerer String als Wert angegeben (href=""), bezieht sich der link auf die aktuelle Seite, so dass im Prinzip nichts passiert. Interessant wird diese Möglichkeit, wenn die Seite einen Anker (Sprungmarke) enthält: Der browser stellt dann den Bereich mit der Marke am oberen Fensterrand dar. Ein Beispiel:

```
<a href="#hierhin">weiter bei der Marke</a>
... viele Zeilen ...
<a name="hierhin">an dieser Stelle geht es weiter</a>
```

refresh

Eine weitere Besonderheit ist der erzwungene refresh einer Seite. Enthält die Seite im Kopf eine Anweisung zum refresh, dann wird nach der definierten Zeit automatisch weiter geleitet. Dies habt ihr sicher schon sehr oft erlebt, wenn die von euch eingegebene URL des links automatisch durch eine andere ersetzt wurde. Dann hat der browser eine Seite geladen, die sofort durch die andere ersetzt wurde.
Diese Möglichkeit kann man geschickt ausnutzen, um Seiten aufeinander wie bei einer automatisch ablaufenden Präsentation einzusetzen. Ein Beispiel:

```
<meta http-equiv="refresh" content="3; URL=die-neue-Seite.html">
```

ersetzt die aktuelle Seite nach 3 Sekunden durch die angegebene URL.

1 In einigen Fällen bezieht sich diese Verknüpfung auf dieselbe Datei, man will dann in der Regel zu einer anderen Position auf derselben Seite springen. Ein Beispiel dafür ist eine umfangreiche Erläuterung, die in mehrere Abschnitte gegliedert ist. Sprungziele sind dann jeweils die Überschriften der Absätze.