Über diesen Text... Dies ist eine Einführung in HTML. Sie ist nicht vollständig, sondern behandelt nur eine Auswahl an wichtigen Elementen und Formateigenschaften. Sie setzt nahezu konsequent nur CSS ein (bis auf width/height), so dass nur eine Syntax für alle Elemente nötig ist. In violetten Kästen stehen Ergänzungen, die man auch gut überspringen kann.
Zur Vertiefung ist de.selfhtml.org zu empfehlen.
- H. Becker, 2007

HTML-Einführung

Eine HTML-Datei ist im Prinzip eine einfache Textdatei und kann daher auch mit einem einfachen Texteditor bearbeitet werden. Sie enthält allerdings spezielle **Tags**, um das Aussehen des Textes zu beeinflussen.

Beispiel:	Hier ist **ein Teil des Textes** fett gedruckt.
in HTML:	`Hier ist <b>ein Teil des Textes</b> fett gedruckt.`

In grün geschrieben ist das Tag, das in diesem Fall den Text fett gedruckt erscheinen lässt. Tags treten fast immer paarweise auf: Zu dem *öffnenden* Tag `<b>` gehört das *schließende* Tag `</b>`. Die beiden Tags eines Paares wirken auf alles, was zwischen ihnen steht.

Aufbau einer HTML-Datei: Grundgerüst

Jede HTML-Datei besitzt ein Grundgerüst - ohne dieses wird nichts angezeigt. Auch dieses Grundgerüst besteht aus Tag-Paaren:

```
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=ISO-8859-1">
  <title>Titel der Seite</title>
</head>
<body>

Alles, was irgendwie auf der Seite erscheinen soll,
muss hier stehen, zwischen dem öffnenden und dem
schließenden body-Tag!

</body>
</html>
```

Innerhalb des `<html>`-`</html>`-Paares steht alles, was irgendwie für die HTML-Datei von Bedeutung ist - und das sind zwei Abschnitte:

1. Der **Kopf** der HTML-Datei in `<head>`-`</head>`.
   Hier stehen *Informationen über die Seite*. In umserem Beispiel wird mit `<title>` der Titel der Seite genannt. *Achtung:* Dieser Titel erscheint nicht in der Seite selbst, sondern nur in der Titelleiste des Browser-Fensters!
   Außerdem definiert die `<meta>`-Zeile mit "`charset`", welcher Zeichensatz benutzt wird, das ist vor allem für Umlaute wichtig.
2. Der **Rumpf** der Seite in `<body>`-`</body>`.
   Im Rumpf steht der *Inhalt* der Seite. Alles, was in der Seite erscheinen soll, muss in diesem Bereich stehen - wirklich alles! Auch Überschriften, Tabellen, Listen, Autor, ...

Zeichenformatierung

Zunächst ein paar Elemente, die man auf Sätze oder Worte innerhalb eines Absatzes anwenden will:

- `<b>` für **fettgedruckt** (engl.: bold)
- `<i>` für *kursiv* (engl.: italics)
- `<u>` für unterstrichen

Man kann diese auch kombinieren. *Die schließenden Tags nicht vergessen!*

Beispiel:	Text in **fett**, *kursiv*, unterstrichen und ***allem zusammen***.
in HTML:	`Text in <b>fett</b>, <i>kursiv</i>, <u>unterstrichen</u> und <b><i><u>allem zusammen</u></i></b>.`

Überschriften

Überschriften sind bei der Gestaltung übersichtlicher Seiten ganz wichtig. Es gibt große und kleinere Überschriften:

Beispiel:

Große Überschrift

Text Text Text

Unter-Überschrift

Text Text Text

in HTML:

```
<h1>Große Überschrift</h1>
Text Text Text
<h2>Unter-Überschrift</h2>
Text Text Text
```

Es gibt auch die noch kleineren Überschriften `<h3>` bis `<h6>`.

Absätze

Schon mal versucht, einen Text mit mehreren Absätzen ohne Überschriften dazwischen zu schreiben? Das hat wahrscheinlich nicht geklappt... HTML kümmert sich nämlich nicht um Zeilenumbrüche und Absätze im Text einer HTML-Datei. Denn der Browser passt den Zeilenumbruch der Fensterbreite an. Das kann man sehen, wenn man das Browser-Fenster mal schmaler oder breiter macht. Um echte Absätze zu erzeugen, dient das Tag `<p>` (engl.: paragraph).

```
test0 test1 test2 test3
test4 test5

test6
test7 test8     test9
```

test0 test1 test2 test3 test4 test5 test6 test7 test8 test9

```
<p>
test0 test1 test2 test3
test4 test5
</p>

<p>
test6
test7 test8     test9
</p>
```

test0 test1 test2 test3 test4 test5

test6 test7 test8 test9

```
<p>a</p><p>b</p><p>c</p>
```

a

b

c

Manchmal möchte man einfach nur eine neue Zeile beginnen - dann benutzt man `<br>` (engl.: line **br**eak). `<br>` ist eines der wenigen Tags, für die es *kein schließendes Tag* gibt; es kommt immer alleine vor. Es bezieht sich auch nicht auf einen *Bereich*, sondern wirkt nur an einer *Stelle*, indem es dort einen Zeilenumbruch erzeugt.

```
test0 test1 test2 test3
<br>test4 test5
<br>
<br>test6
<br>test7 test8     test9
```

test0 test1 test2 test3
test4 test5

test6
test7 test8 test9

Leer- und Sonderzeichen

Auch mehrere Leerzeichen zwischen zwei Worten werden zu einem zusammengefasst. Wenn man mal mehr Abstand benötigt, kann man `&nbsp;` benutzen. Das ist kein Tag, sondern ein Sonderzeichen (**n**on **b**reakable **sp**ace - untrennbares Leerzeichen). Es hat noch eine weitere Funktion: Wenn es statt eines normalen Leerzeichens zwischen zwei Worten steht, erfolgt dort kein Zeilenumbruch (daher der Name).

`etwas mehr&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Platz`	etwas mehr Platz

Sonderzeichen... Da die Zeichen <, > und & in HTML besondere Bedeutung haben - wie kann man sie dann sichtbar auf die Seite schreiben? Das geht mit `&lt;` (**l**ess **t**han - kleiner als), `&gt;` (**g**reater **t**han - größer als) und `&amp;` (**amp**ersand - das Und-Zeichen).

`Die Zeichen &lt;, &gt; und &amp; in HTML`	Die Zeichen <, > und & in HTML

Horizontale Linie

Eine horizontale Linie, wie die über und unter diesem kleinen Abschnitt, erhält man mit `<hr>` (engl.: **h**orizontal **r**uler). Sie passt sich der Fensterbreite an. Auch dieses Tag steht alleine, hat also kein schließendes Tag.

Aufzählungen

Es gibt numerierte und nicht-numerierte Aufzählungen in HTML:

```
Aufzählung
  (<b>u</b>nnumbered <b>l</b>ist)
<ul>
 <li>Ein Punkt</li>
 <li>ein anderer Punkt</li>
 <li>noch einer</li>
</ul>
```

Aufzählung (**u**nnumbered **l**ist)

- Ein Punkt
- ein anderer Punkt
- noch einer

```
Nummerierte Aufzählung
  (<b>o</b>rdered <b>l</b>ist)
<ol>
 <li>Erster Punkt</li>
 <li>zweiter Punkt</li>
 <li>dritter
   <br>mit Zeilenumbruch</li>
</ol>
```

Nummerierte Aufzählung (**o**rdered **l**ist)

1. Erster Punkt
2. zweiter Punkt
3. dritter
   mit Zeilenumbruch

`<li>` steht übrigens für **l**ist **i**tem.

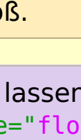
Bilder

Bilder lassen sich in HTML-Seiten auch wieder über ein Tag einfügen: `<img src="...">` (engl.: Bild - **im**a**g**e; Quelle - **s**ou**rc**e). Da Bilder an einer *Stelle* eingefügt werden (und nicht in einem Text*bereich*), steht auch dieses Tag allein, hat also *kein schließendes Tag*.

- Wenn das Bild im gleichen Verzeichnis steht wie die HTML-Datei, muss man für `src` einfach nur den Dateinamen angeben, beispielsweise `<img src="bild3.jpg">`.
- Befindet sich das Bild in einem Unterverzeichnis namens `images`, so kann man den relativen Pfad angeben, z. B. `<img src="images/bild3.jpg">`.
- Windows zeigt standardmäßig Dateiendungen nicht an - hier aber muss man sie unbedingt mit angeben!
  Tip: Im Explorer "Extras" / "Orderoptionen" / "Ansicht" den Punkt "Erweiterungen bei bekannten Dateitypen ausblenden" ausschalten.
- In Windows ist die Groß-/Kleinschreibung des Dateinamens egal, in Unix-basierten Betriebssystemen nicht! Webserver setzen meist solche Systeme ein. Wenn die Seite also auch auf einem Webserver laufen soll, lieber die genaue Schreibweise des Dateinamens beachten.
- Man kann auch Bilder von anderen Webseiten einbinden. Dann muß man deren volle Adresse angeben, also etwa `<img src="http://www.server.com/Seite/bild3.jpg">`

```
Ein kleines Bild <img src="mail.gif"> mitten im Text, wie ein Icon.
<br>Und ein animiertes Bild in einer extra Zeile
<br><img src="neko.gif">
<br>Denn sonst werden Bilder in die aktuelle Textzeile eingefügt.
```

Ein kleines Bild ✉ mitten im Text, wie ein Icon.
Und ein animiertes Bild in einer extra Zeile:



Denn sonst werden Bilder in die aktuelle Textzeile eingefügt.

Nicht immer hat ein Bild gerade die Größe, die man in der HTML-Seite haben möchte. Natürlich lassen sich Bilder mit Bildbearbeitungsprogrammen verkleinern, aber diesen Aufwand muss man nicht immer betreiben. In HTML kann man nämlich auch die gewünschte Größe eines Bildes angeben, und der Browser stellt das Bild dann verkleinert (oder vergrößert) dar. Die wichtigsten Möglichkeiten sind:

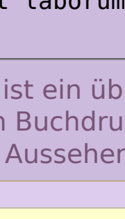
- Angabe der **Breite** in Pixeln: `<img width="150" src="bild.jpg">`
  Das Bild wird auf eine Breite von 150 Pixeln umgerechnet; die Höhe wird mit angepasst.
- Angabe der **Höhe** in Pixeln:`<img height="100" src="bild.jpg">`
  Das Bild wird auf eine Höhe von 100 Pixeln umgerechnet; die Breite wird mit angepasst.

```
Dieses Bild ist verkleinert dargestellt,

<br><img height="80" src="regenbogen.gif">

<br>denn im Original ist es zu groß.
```

Dieses Bild ist verkleinert dargestellt,

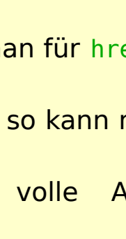


denn im Original ist es zu groß.

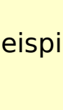
Ergänzung für Neugierige... Man kann Bilder auch von Text umfließen lassen. Dafür sorgt ein `style="float:left;"` im `img`-Tag; in diesem Fall ist das Bild links. Man kann das Bild auch rechts plazieren mit `style="float:right;"`.
Damit das Bild ein wenig Abstand zum umgebenden Text hat, kann man noch `margin:8px;` ergänzen; hier sind 8 Pixel Randabstand angegeben.
Am besten wirken umflossene Bilder, wenn der Text links- und rechtsbündig ausgerichtet ist (Blocksatz). Das geht mit `<p style="text-align:justify">` (zu deutsch etwa: Stil="Text-Ausrichtung:Blocksatz"). Damit wird ein Absatz (`<p>`) im Blocksatz gesetzt.

```
<p style="text-align:justify">
Lorem ipsum dolor sit amet, consectetur adipisicing elit,
<img style="float:left;" height="90" src="regenbogen.gif">sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
consequat.
<img style="float:right; margin:20px;" src="neko.gif"> Duis aute irure dolor
in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla
pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui
officia deserunt mollit anim id est laborum.
</p>
```

Ubrigens... Dieses "*Lorem ipsum*" ist ein üblicher Blindtext, also ein bedeutungsloser Beispiel-Text, der von Buchdruckern schon seit dem 16. Jahrhundert als Platzhalter benutzt wurde, um das Aussehen von Schrift oder Layout zu beurteilen.



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.



Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Links

Die Bezeichnung **Hypertext** bedeutet, dass verschiedene Seiten miteinander verknüpft sind und so ein Netzwerk aus Informationen bilden. Diese Verknüpfungen heißen **Hyperlinks**. Mit anderen Worten: Auf einer Seite klickt man auf einen Link und landet auf einer anderen Seite, die sich auch auf einem ganz anderen Server befinden kann. Das passende Tag dazu ist `<a href="...">Text</a>` (engl: a - **a**ction; href - **h**yper **ref**erence). Der Text zwischen dem öffnenden und dem schließenden Tag ist anklickbar und führt auf die gewünschte Seite. Für `href` muß man die Adresse der Zielseite angeben; dabei gilt das gleiche wie für Bilder:

- Wenn die Zielseite im gleichen Verzeichnis steht wie die aktuelle HTML-Datei, muss man für `href` einfach nur den Dateinamen angeben, beispielsweise `<a href="sonstwas.html">`.
- Befindet sich die Zielseite beispielsweise in einem Unterverzeichnis namens `projekt`, so kann man den relativen Pfad angeben, z. B. `<a href="projekt/sonstwas.html">`.
- Befindet sich die Zielseite auf einem anderen Server, gibt man ihre volle Adresse an, also beispielsweise `<a href="http://www.server.com/projekt/sonstwas.html">`

Beispiel:	Auf dieser Seite ist ein Regenbogen... Und hier geht es zu selfHTML
in HTML:	`Auf <a href="link.html">dieser Seite</a> ist ein Regenbogen...` ` Und hier geht es zu <a href="http://de.selfhtml.org">selfHTML</a>`

Auch Bilder können als Link benutzt werden:

```
<a href="link.html">
<img height="80" src="regenbogen.jpg"> </a>
<br>Auf das Bild klicken, um eine größere Version davon zu sehen.
```

Auf das Bild klicken, um eine größere Version davon zu sehen.

Kommentare

Kommentare sind erläuternde Texte in der HTML-Datei, die nicht im Browser angezeigt werden. Sie sind nur für den Autor (oder den Betrachter) der HTML-Datei (in einem Editor) gedacht, genau so wie Kommentare in Programmiersprachen.

Beispiel:	Einfach nur ein kleiner Text. Sonst nix.
in HTML:	`Einfach nur ein kleiner Text. <!-- besseren Text suchen --> Sonst nix.`

Man kann damit auch ganze Abschnitte des HTML-Codes auskommentieren (also zu einem Kommentar machen), die man gerade nicht anzeigen lassen, aber auch nicht löschen möchte.

Tabellen

Tabellen sind wichtige Gestaltungselemente in HTML. Sie dienen einerseits dazu, Inhalte tabellarisch darzustellen, wie diese Seite hier hier oft demonstriert. Andereseits kann man sie auch benutzen, um die ganze Seite beispielsweise in Bereiche für Navigation und Inhalt aufzuteilen (obwohl diese Zweckentfremdung dem eigentlichen Sinn von Tabellen zuwiderläuft!).
Eine HTML-**Tabelle** (`<table>`) besteht von ihrer Struktur her aus **Zeilen** (`<tr>`, **t**able **r**ow), die wiederum in ihren **Zellen** die Daten (`<td>`, **t**able **d**ata) enthalten. Daraus ergibt sich, dass die Tabelle **zeilenweise** aufgebaut wird.

```
<table border=1>
<tr>  <td>Aaa</td>  <td>Bbb</td>  <td>Ccc</td>  </tr>
<tr>  <td>Ddd</td>  <td></td>     <td>Fff</td>  </tr>
<tr>  <td>Ggg</td>  <td>Hgg</td>  <td>Iii</td>  </tr>
<tr>  <td>Jjj</td>  <td>Kkk</td>       </tr>
</table>
```

Aaa	Bbb	Ccc
Ddd		Fff
Ggg	Hhh	Iii
Jjj	Kkk	

- Wie viele Spalten die Tabelle hat, richtet sich ganz einfach nach der Anzahl der Zellen (`<td>`...`</td>`) in den Zeilen.
- Wie man sieht, gibt es sowohl leere Zellen (`<td></td>`) als auch gar nicht vorhandene Zellen (im Beispiel am Ende der vierten Zeile).
- Statt `<td>`...`</td>` kann man auch `<th>`...`</th>` (**t**able **h**ead - Kopfzelle) benutzen. Diese Zellen sind hervorgehoben (meist zentriert, fett, kursiv) und dienen üblicherweise der Spalten- bzw. Zeilenbeschreibung (in der ersten Zeile bzw. Spalte).
- Innerhalb der Zellen kann man alle Elemente verwenden, die man auch sonst verwenden kann: Absätze, Überschriften, Aufzählungen, Bilder - und auch weitere Tabellen.
- Die Ergänzung `border=1` sorgt dafür, dass die Rahmen bzw. Gitternetzlinien der Tabelle angezeigt werden. Lässt man es weg, werden die Begrenzungen nicht gezeichnet, und man spricht von einer blinden Tabelle.
- Die Größen der Zellen werden automatisch vergeben - man kann aber mit `<td width="500">` eine Breite (hier 500 Pixel) zuweisen, oder mit `<td height="150">` eine Höhe, oder beides.

Ausrichtung

- **Horizontale Ausrichtung:** Den Inhalt einer Zelle kann man links, zentriert oder rechts ausrichten mit `style="text-align:center;"` (Text-Ausrichtung: zentriert). Die Möglichkeiten sind `left`, `center` und `right`.
- **Vertikale Ausrichtung:** Manche Zellen haben weniger Inhalt als andere und nutzen nicht die volle Höhe aus. Deshalb möchte man gelegentlich den die vertikale Position des Inhalts bestimmen: oben, mittig oder unten. Das geht mit `style="vertical-align:middle;"` (hier mittig). Die Möglichkeiten sind `top`, `middle` und `bottom`
- beides ist kombinierbar, z.B.
  `<td style="text-align:center; vertical-align:middle;">...</td>`

text-align vertical-align	left	center	right
top	X	X	X
middle	X	X	X
bottom	X	X	X

In den neun dunklen Zellen sind alle neun Kombinationen von text-align und vertical-align aufgeführt.

Zellen zusammenfassen... Es gibt die Möglichkeit, mehrere Zellen zusammenzufassen, wie dies auch in Tabellenkalkulationen möglich ist. Um beispielsweise drei benachbarte Zellen einer Zeile zusammenzufassen, dient `<td colspan=3>...</tr>` (**col**umn **span** - Spalten-Bereich, erstreckt sich über mehrere Spalten). Der Inhalt dieser Zelle erstreckt sich dann über 3 ursprüngliche Zellen:

```
<table border=1>
<tr>  <td>Aaa</td>  <td>Bbb</td>  <td>Ccc</td>  <td>Ddd</td>  </tr>
<tr>  <td>Eee</td>  <td colspan=3>Fffffffff</td>               </tr>
<tr>  <td colspan=2>Ggg</td>      <td>Hhh</td>  <td>Iii</td>  </tr>
</table>
```

Aaa	Bbb	Ccc	Ddd
Eee	Fffffffff		
Ggg		Hhh	Iii

Um mehrere übereinanderliegende Zellen zusammenzufassen, benutzt man `rowspan` (**row span** - Zeilen-Bereich, erstreckt sich über mehrere Zeilen). Beispielsweise fasst `<td rowspan=2>...</tr>` zwei Zellen zusammen. Wichtig: In der/den darunterliegenden Zeile/n muss diese Zelle ausgelassen werden!

```
<table border=1>
<tr>  <td>Aaa</td>             <td>Bbb</td>  <td rowspan=2>Ccc</td>  </tr>
<tr>  <td rowspan=2>Ddd</td>  <td>Eee</td>                          </tr>
<tr>                           <td>Fff</td>  <td>Ggg</td>            </tr>
</table>
```

Aaa	Bbb	Ccc
Ddd	Eee	
	Fff	Ggg

Natürlich lässt sich beides nicht nur gleichteitig in einer Tabelle einsetzten, sondern auch miteinander kombinieren; eine Zelle kann also mehrere Spalten *und* Zeilen umfassen, wenn sowohl `colspan` als auch `rowspan` angegeben werden.

Formateigenschaften

Bislang wurden hier HTML-Gestaltungs**elemente** vorgestellt - Absätze, Überschriften, Tabellen-Zellen etc., aber auch `<b>` oder `<i>`. Diese Elemente sind üblicherweise mit **Formateigenschaften** verbunden, beispielsweise Überschriften groß, `<b>` fettgedruckt, Absätze in ihrer Standard-Schrift...
Doch das lässt sich alles verändern: Farbe, Hintergrundfarbe, Rahmen, Abstände, Schriftgröße, kursiv, fett,...
Dazu ergänzt man `style="..."` in das öffnende Tag. Hier ein paar Beispiele, die nahezu selbsterklärend sind:

`<p>Ein normaler Text</p>`	Ein normaler Text
`<p style="font-size:130%;">größerer Text</p>`	größerer Text
`Hier ein <i style="color:blue;">blauer kursiver</i> Text.`	Hier ein *blauer kursiver* Text.
`<h2 style="border:1px solid black;">Überschrift mit Rahmen</h2>`	**Überschrift mit Rahmen**
`<p style="background-color:silver;">Absatz mit Hintergrundfarbe</p>`	Absatz mit Hintergrundfarbe
`<td style="background-image:url(marble.gif)">Eine Tabellenzelle mit Hintergrundbild</td>`	Eine Tabellenzelle mit Hintergrundbild
`<p style="border:1px solid black; padding:15px;"> Text mit Rahmen und Abstand</p>`	Text mit Rahmen und Abstand
`<h2 style="border:2px solid red; background-color:navy; color:yellow; padding:15px;">Überschrift</h2>`	**Überschrift**
`<body style="font-family:sans-serif;color:navy;">Ganze Seite in anderer Schiftart</body>`	Ganze Seite in anderer Schiftart

Dies sind nur Beispiele für die vielfältigen Kominationsmöglichkeiten, die es gibt. Durch `style="..."` kann man nahezu jedes Element mit jeder Formateigenschaft versehen. Das klappt auch für den ganzen Rumpf (`<body>`) der Seite: So kann man die beispielsweise Standardschriftart, Standardfarbe oder ein Hintergrundbild für die Seite festlegen.

Hier eine Zusammenstellung der wichtigsten Formateigenschaften:

`color`	Schriftfarbe `color:yellow;`	siehe unten
`background-color`	Hintergrundfarbe `background-color:blue;`	siehe unten
`background-image:url(...);`	Hintergrundbild `background-image:url(bild.jpg);`	Eine JPG- oder GIF-Bilddatei
`border`	Rand: Breite, Typ und Farbe `border:1px solid blue;`	Randtyp: none (keiner), solid (durchgezogen), dotted (gepunktet), dashed (gestrichelt), double (doppelt)
`padding`	Abstand zwischen Rahmen und Inhalt `padding:15px;`	Abstand in Pixeln, siehe oben
`font-size`	Schriftgröße `font-size:170%;`	Angabe in Prozent der normalen Schriftgröße
`font-family`	Schriftart `font-family:Arial,sans-serif;`	Angabe einer Schriftart. Für den Fall, dass diese nicht existiert, sollte eine weitere Schriftart angegeben werden oder ein Schrifttyp: serif, sans-serif, monospaced, cursive, fantasy
`font-style`	*kursiv* oder normal `font-style:italics;`	normal, italics
`font-weight`	**fett** oder normal `font-weight:normal;`	normal, bold
`text-decoration`	unterstrichen, überstrichen, ~~durchgestrichen~~ `text-decoration:underline;`	none, underline, overline, line-through, blink
`text-align`	Text-Ausrichtung: linksbündig, rechtsbündig, zentriert, Blocksatz `text-align:center;`	left, right, center, justify

Einige Farben:

black	maroon	green	olive	navy	purple	teal	silver
gray	red	lime	yellow	blue	fuchsia	aqua	white

span und div... Wenn man einfach mal ein paar Worte in einer anderen Farbe oder Schrift darstellen will, ohne dass es auch gleichzeitig auch fett oder kursiv sein soll, kann man `<span>`...`</span>` (span - Bereich) benutzen. Es funktioniert wie `<b>`...`</b>` oder `<i>`...`</i>`, ist aber eigenschaftslos, hat also keinerlei eigene vordefinierte Formateigenschaften wie fett oder kursiv. Aber Formateigenschaften lassen sich mit `style="..."` ergänzen:

`Ein <span style="color:red;">rotes</span> Wort.`	Ein rotes Wort.

Etwas ähnliches gibt es für ganze Abschnitte oder Blöcke, in denen beispielsweise mehrere Absätze und Überschriften zentriert werden sollen. Das entsprechende eigenschaftslose Block-Element heißt `<div>`...`</div>`:

```
<div style="text-align:center">
<h3>Überschrift</h3>
<p>Ein zentrierter Absatz</p>
<p>Noch einer.</p>
</div>
```

Überschrift

Ein zentrierter Absatz

Noch einer.

Auf diese Weise kann man Abschnitte auch mit Rahmen, Hintergrundfarbe und eigenen Hintergrundbildern versehen. Dieser Abschnitt hier ist ein Beispiel dafür.

Zentrale Formateigenschaften - CSS

Wenn eine komplette Seite in einem durchgehenden Design gestaltet werden soll (beispielsweise jede Überschrift in einer bestimmten Schriftart, mit Rahmen und Hintergrund-Bild), ist es sehr unpraktisch, diese Angaben jedes mal zu wiederholen. Noch unpraktischer ist es, wenn das Design geändert werden soll: Dann müßte man alle Überschriften einzeln ändern. Wenn dann noch mehrere Elemente geäandert werden sollen, artet das in langwierige und auch fehleranfällige Arbeit aus.

```
<html>
<head>
<title>Titel der Seite</title>
</head>
<body>

<h1 style="border:2px solid black; background-image:url(marble.gif);
font-family:sans-serif;">Überschrift1</h1>

Text text text

<h1 style="border:2px solid black; background-image:url(marble.gif);
font-family:sans-serif;">Überschrift2</h1>

Text text text

</body>
</html>
```

Überschrift1

Text text text

Überschrift2

Text text text

Viel besser ist es, diese Formateigenschaften **einmal** zentral für alle diese Überschriften zu definieren. Genau aus diesem Grund wurde **CSS** (Cascading Style Sheets) erfunden.
Damit definiert man am Anfang beispielsweise, daß alle `<h1>`-Überschriften ein bestimmtes Aussehen haben sollen. Diese Definition erfolgt im `<head>`-Abschnitt der HTML-Datei, in dem dazu hinter dem `<title>`-`</title>` ein neuer `<style>`-`</style>`-Abschnitt eingefügt wird.

Hier das obige Beispiel mit CSS:

```
<html>
<head>
<title>Titel der Seite</title>
<style type="text/css">
h1  {border:2px solid black; background-image:url(marble.gif);
     font-family:sans-serif;}
</style>
</head>
<body>

<h1>Überschrift1</h1>
Text text text

<h1>Überschrift2</h1>
Text text text

</body>
</html>
```

Überschrift1

Text text text

Überschrift2

Text text text

...und das sieht im body-Abschnitt auch gleich viel übersichtlicher aus, insbesondere, wenn man an größere Seiten denkt. Man achte auf die geänderte Schreibweise innerhalb `<style>`-`</style>`: Das Element `h1` steht ohne spitze Klammern, und die eigentlichen Formateigenschaften stehen in geschweiften Klammern statt in `style="..."`
Wenn man das Erscheinungsbild mehrerer Elemente ändern will, schreibt man einfach weitere dazu. Zum Beispiel:

```
...
<style type="text/css">
body  {background-image:url(sand.jpg);}
h1    {border:2px solid black; background-image:url(marble.gif);
       font-family:sans-serif;}
h2    {border:1px solid black; font-style:italics;}
</style>
...
```

Hier wird auch das Erscheinungsbild von `<h2>` geändert und die gesamte Seite (`<body>`) mit einem Hintergrundbild versehen.

Auf diese Weise lassen sich alle Elemente in HTML umgestalten, also auch h2 bis h6, i, b, u, p, body, table, td, th, ol, ul, li, hr, div, ...
Kursiv soll immer auch auch gleichzeitig grün sein? Fettgedrucktes grundsätzlich in einer anderen Schriftart? Alle Tabellenzellen mit Hintergrundfarbe? Mit CSS kein Problem. Probieren Sie es aus...

CSS - Trennung von Inhalt und Gestaltung... CSS verfolgt auch die Grundidee von HTML, den eigentlichen Inhalt der Seite vom Aussehen zu trennen. Das `h1`-Tag soll eigentlich nur den umschlossenen Text logisch als Überschrift auszeichnen, ihm also die Bedeutung einer Überschrift geben, ohne sich um das Design zu kümmern, während die CSS-Formateigenschaften die optische Gestaltung der Überschrift definieren. So kann man den Inhalt ändern, ohne sich um die Gestaltung zu kümmern, oder andersherum das Design unabhängig vom Inhalt ändern.

Ein Web-Auftritt besteht üblicherweise nicht nur aus einer HTML-Seite, sondern aus einer Anzahl von vielen Seiten. Oft sollen alle diese Seiten ein gemeinsames Design haben, da wäre es unpraktisch, das Design in jeder Datei einzeln definieren oder ändern zu müssen. Deshalb gibt es auch hier eine Lösung: Die CSS-Formateigenschaften werden in eine besondere Datei ausgelagert und von jeder HTML-Seite mit einem speziellen Aufruf eingebunden:

```
<html>
<head>
<title>Titel der Seite</title>
<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>

...

</body>
</html>
```

```
<html>
<head>
<title>Titel der Seite</title>
<link rel="stylesheet" type="text/css" href="style.css">
<style type="text/css">
h1  {border:2px solid black; background-image:url(marble.gif);
     font-family:sans-serif;}
</style>
</head>
<body>

...

</body>
</html>
```

- Das Beispiel links zeigt, wie eine externe CSS-Datei in eine HTML-Datei eingebunden wird. Die HTML-Datei benutzt dann die in `style.css` definierten Formateigenschaften (der Dateiname ist beliebig).
- Das rechte Beispiel zeigt, daß man auch beides gleichzeitig machen kann, also sowohl einen externen Style einbinden als auch eigene Formateigenschften nur für diese HTML-Datei definieren. Wird ein Element doppelt definiert, überschreibt die in der HTML-Datei definierte Eigenschaft die aus der externen Datei.

Die Datei `style.css` sieht dann beispielsweise so aus:

```
body  {background-image:url(sand.jpg);}
h1    {border:2px solid black; background-image:url(marble.gif);
       font-family:sans-serif;}
h2    {border:1px solid black; font-style:italics;}
```

Damit gibt es drei Möglichkeiten, wo mit Formateigenschaften die Gestaltung von HTML-Elementen geändert werden kann:

1. in einer externen CSS-Datei. Gut geeignet, um ein gemeinsames Design für mehrere Seiten festzulegen. Durch das Ändern der CSS-Datei ändert man dann das Design aller HTML-Seiten gleichzeitig.
2. im `style`-Abschnitt einer HTML-Seite - nur für diese Seite. Wenn man eine CSS-Datei benutzt, kann man so für diese Seite einige Elemente abweichend gestalten.
3. direkt bei einem HTML-Element mit `style="..."`. So kann man ein einzelnes Element abweichend vom zentralen Design umgestalten.

ToDo:

- CSS-Selektoren (class, subtype, id)
- beliebige Farben (#RRGGBB)
- #id
- frames

© H. Becker 2007++