

## Crashkurs Webseitenerstellung mit HTML

### Ziel

Das Ziel dieser Einführung in die Webseitenerstellung ist das Kennenlernen der Seitenbeschreibungssprache HTML und die Nutzung für einfach strukturierte Seiten, die untereinander verbunden (verlinkt) sind.

Nicht Bestandteil dieses Kurses sind Elemente der Programmierung in Java, Javascript oder Active-X.

### Grundlegendes

Einige Grundregeln, die sich aus der Technik und der einfachen Bereitstellung im WWW ergeben muß man beachten, diese gelten unabhängig von den Seiteninhalten.

- Vermeiden Sie Großschreibung in den Dateinamen – die meisten Webserver sind auf UNIX-Basis und unterscheiden diese, die wenigsten Probleme haben Sie mit nur kleingeschriebenen Namen.
- Leerzeichen, Sonderzeichen (wie & # + \$ / §) haben nichts in Dateinamen zu suchen, ebenso wenig wie Umlaute und ß.
- Strukturieren Sie Ihre Dokumente so, das man aus Unterverzeichnissen und Namen auch gleich den Inhalt „erahnen“ kann dies erspart viel Suche bei Aktualisierungen.
- Verwenden Sie eine relative Struktur bei der Verlinkung Ihrer Dokumente, d.h. geben Sie Pfade ausgehend von ihrem aktuellen Standort ein. Somit haben Sie die Möglichkeit Ihre Dokumente recht schnell auf ein anderes System zu übertragen.
- Testen Sie Ihr Ergebnis mit mehreren – vor allem auch älteren Browsern. Sie setzen damit den Grundstein, das ihr Dokument von vielen Nutzern gelesen werden kann und entdecken dabei auch Fehler.
- Texte haben Vorrang vor Bildern. Texte sind auswertbar und kleiner als entsprechend in Bilder gegossene Informationen.

### Struktur von HTML

HTML steht für HyperText Markup Language – es ist wie der Name schon sagt eine Text-Auszeichnungssprache. Häufig wird auch von HTML-Texte programmieren gesprochen, was ich aber für übertrieben halte.

Im Prinzip hinterlegt man mit sogenannten Text zusätzliche Anweisungen (Tags) was man an der Stelle hier machen will und wann man damit zuende ist.

Tags werden immer in spitzen Klammern (Kleiner-Als und Größer-Als-Zeichen) gepackt. Der beginnende Tag einfach nur so, das Ende beginnt innerhalb mit einem Schrägstrich.

Ein Beispiel:

In einem Satz soll ein Wort **fett** geschrieben werden.

Das Tag für Fett ist b

Die Groß- und Kleinschreibung spielt bei den Tags keine Rolle – im Skript werden klein geschriebene Tags benutzt.

Das Ergebnis in HTML ist dann:

In einem Satz soll ein Wort `<b>fett</b>` geschrieben werden.

Einige der Anweisungen haben nur ein Tag, das dann wie ein Beginner-Tag geschrieben wird.

Beispiel:

Das Tag `<hr>` fügt eine horizontale Linie in das Dokument ein.

Mehrfache Leerzeichen oder Leerzeilen werden (bis auf ein bestimmtes Tag) zu einem Leerzeichen zusammengezogen, spielen also keine Bedeutung und können zur einfacheren Pflege benutzt werden. Sollen Leerzeichen oder Leerzeilen verwendet werden, dann muß dies extra angewiesen (durch Tags) werden.

### **Grundstruktur eines HTML-Dokumentes**

Ein einfaches HTML-Dokument hat die folgende Struktur:

```
<html>
  <head>
    <title>
      Hier steht der Titel
    </title>
  </head>

  <body>
    Hier steht der Text.
  </body>
</html>
```

Das Tag `<html>` kennzeichnet den Beginn (und natürlich auch das Ende) des Dokumentes.

Innerhalb eines Tags können auch wieder weitere Tags folgen – hier ist dann von Schachtlung die Rede. Als Beispiel ist hier das `title`-Tag innerhalb des `head`-Tags zu nennen.

**ACHTUNG!** Die Schachtlung darf nicht überlappend sein!

Das `head`-Tag bildet den Kopf, in dem Angaben stehen, die nicht innerhalb des Browsers dargestellt werden. Solche Angaben können z.B. der Titel einer Seite sein, die im Kopf des Browsers angezeigt wird – dafür wird das `title`-Tag benutzt.

Der eigentliche Text kommt in den `Body`-Bereich. Dort kommen zur Darstellung die Tags, die im folgenden aufgeführt werden zum Einsatz.

## HTML-Tags im Einsatz

### **Textformatierung**

Zur Textformatierung kommen die Tags `b`, `i`, `u` zum Einsatz.

Tag	Woher stammend	Bedeutung	Bemerkungen
<code>&lt;b&gt;</code>	Bold	Fett	Besser <code>&lt;strong&gt;</code> verwenden
<code>&lt;i&gt;</code>	Italic	Kursiv	
<code>&lt;u&gt;</code>	Underline	Unterstrichen	

#### Beispiel

Ein Text kann `<b>fett</b>`, `<i>kursiv</i>`, `<u>unterstrichen</u>` oder auch `<b><i><u>fett und kursiv und unterstrichen</u></i></b>` sein.

Eine weitere auch inhaltliche Hervorhebungsart ist die Kennzeichnung von Überschriften.

Diese sind hierarchisch aufgebaut.

`<h1>` Überschrift der Ebene 1

`<h2>` Überschrift der Ebene 2

....

Die entsprechende Darstellung übernimmt der Browser. Es ist aber für Suchmaschinen beispielsweise auch eine inhaltliche Hervorhebung.

### **Umlaute**

Umlaute sollten in HTML immer kodiert werden, die Kodierung lautet wie folgt:

ä	<code>&amp;auml;</code>
Ä	<code>&amp;Auml;</code>
ü	<code>&amp;uuml;</code>
Ü	<code>&amp;Uuml;</code>
ö	<code>&amp;ouml;</code>
Ö	<code>&amp;Ouml;</code>
ß	<code>&amp;szlig;</code>

Sonst ist die ordentliche Darstellung nicht in allen Browsern möglich.

### **Absatzformatierung**

Man sollte sich – auch in Hinblick auf fortgeschrittenere Techniken – daran gewöhnen auch Absätze zu kennzeichnen.

`<p>`Daher sollte jeder Absatz mit dem `p`-Tag begonnen und beendet werden.`</p>`  
Häufig sieht man die Nutzung des `p`-Tags als Zeilenumbruch, was aber nicht exakt ist.

`<p>`Der Zeilenumbruch`<br>`

ist das `br`-Tag, das innerhalb eines Absatzes steht.`</p>`

Möchte man Leerzeilen einfügen, so kann man dies durch Einfügen leerer Absätze, die exakt

`<p></p>` sein müssten, aber auch die Form `<p>` ist recht oft zu finden.

Benötigt man in einem Absatz mehrere Leerzeichen hintereinander oder möchte bestimmte Angaben, die ein Leerzeichen enthalten zusammenlassen, so muß man auf ein geschütztes Leerzeichen, das in der Form

&nbsp;

notiert wird, zurückgreifen. Diese Leerzeichen werden auf jeden Fall eingefügt.

```
<pre>
```

```
Haben Sie einen Text, der so stehen bleiben soll,  
wie er ist, inkl. Zeilenumbrüche
```

```
    Leerzeichen und in einer Schreibmaschinenartigen  
Formatierung ist, so  
müssen Sie das pre-Tag (pre formatted= vorformatiert) nutzen.  
</pre>
```

## **Bilder**

Bilder werden an einer Stelle über das img-Tag eingefügt.

Es sieht z.B. so aus:

```

```

Dabei bedeuten die sogenannten Attribute:

src... die Quelle (source) woher das Bild kommt

alt... die alternative Angabe, falls das Bild nicht dargestellt wird

width... die angezeigte Breite des Bildes (falls dies nicht die Originalgröße ist, wird das Bild verkleinert oder vergrößert)

height... die angezeigte Höhe des Bildes (auch hier gilt: falls dies nicht die Originalgröße ist, wird das Bild verkleinert oder vergrößert).

## **Tabellen**

Eine schon etwas fortgeschrittenere Form ist die Nutzung von Tabellen, sie stellt auch eine Alternative zur Nutzung von Frames dar, die hier nicht behandelt werden.

Eine Tabellenstruktur ist wie folgt aufgebaut:

```
<table border=1>
```

```
<tr>
```

```
    <td> A1
```

```
    </td>
```

```
    <td> B1
```

```
    </td>
```

```
</tr>
```

```
</table>
```

A1	B1
----	----

Diese Tabelle beginnt mit dem Tag table – das Attribut border legt die Stärke der Linien fest. In unserem Fall hier die kleinste Stärke. Der Wert kann auch 0 sein, dann wird die Linie nicht dargestellt.

Das Tag tr legt die beginnende Zeile (row) fest. Innerhalb der Zeile legt das Tag td die Datenzelle fest.

Hier finden Sie eine erweiterte Darstellung einer Tabelle.

Durch das Attribut `colspan` erreichen Sie, dass sich eine Zelle über mehrere Spalten hinweg erstreckt (`colspan` = `column span` = Spalten spannen).

Das Attribut `rowspan` gilt auf gleiche Weise für Zeilen.

Tabellen können auch ineinander geschachtelt werden, d.h. Eine Tabelle kann auch innerhalb einer Zelle einer äußeren Tabelle sich befinden.

## Weiterführende Literatur

Auf jeden Fall zu empfehlen ist SELFHTML, das im Original unter <http://selfhtml.teamone.de> zu finden ist. Einen Mirror finden Sie unter <http://www.schmidma.de/selfhtml> .