

HOME PAGE I

HTML

Schulungsunterlagen

Von Dietmar Becker



<http://www.8ung.at/mcp.becker>

Tel.: 0172 / 25 748 25

INHALT:

	Seite
1. Hypertext Markup Language	3
2. Syntax	4
3. Meine erste Seite	6
4. Formatierungen	7
5. Bilder einfügen	12
6. Hyperlinks - relativ oder absolut	17
7. Tabellen – Layout-Gestaltung	19
8. Frames	25
9. FTP	28
10. Tags und Attribute	31

HTML

HyperText Markup Language ist die Programmiersprache im World Wide Web. Der Quellcode wird in einfachem ASCII-Format gespeichert, also als einfache Textdatei (z. B. mit Wordpad). Dieser Quellcode enthält neben dem angezeigten Text auch noch alle Formatierungsbefehle, also Schriftgröße, Form, Zeilenumbrüche, Absätze, Tabellen, Rahmen, etc...

Um sich die Inhalte des WWW ansehen zu können, brauchen wir ein Programm, das diesen Quellcode interpretieren kann. Das ist der Browser (Internet Explorer, Netscape Navigator, Mozilla, etc...).

Das WWW breitet sich extrem aus, und auch die Programmiersprache HTML wird in seinem Leistungsumfang ständig erweitert. Die Koordinierung der Standards und die Weiterentwicklung dieser Sprache wird vom W3-Consortium (<http://www.w3c.org>) wahrgenommen. Hier werden die Normen festgelegt - nicht nur für HTML - auch für XML, Java, etc...XML ist die Abkürzung für eXtensible Markup Language, die von Sun "erfunden" wurde und viele namhafte Hersteller unterstützen die Entwicklung (Microsoft, IBM, Novell, Sun, Netscape, Hewlett-Packard). Die Möglichkeiten sind nahezu unbegrenzt - doch die Vorteile von HTML liegen auf der Hand:

- Es ist ohne großartige Programmierungskennntnis schnell zu erlernen.
- Für die Erstellung reicht ein einfacher Texteditor.
- Für die meisten Webseiten völlig ausreichend.
- Die Browser werden auch in Zukunft problemlos HTML-Quelltexte interpretieren können.

Nachdem der Quellcode mit einem Editor geschrieben ist, muss man nur noch die Dateiendung von z.B. txt in htm (oder html) umbenennen.

Syntax

Das ist der Seitenaufbau einer HTML-Seite, also welche Regeln eingehalten werden müssen, damit die Seite korrekt dargestellt wird.

Die Befehle

bei HTML werden *Tags* genannt und stehen immer in spitzen Klammern, also

`<bold>fetter Text</bold>`

und ist meist die Abkürzung eines englischen Begriffs (bold = fett).

Die meisten *Tags* (`<bold>`) benötigen einen Endtag (`</bold>`) für die Beendigung des Befehls, sonst gilt er bis zum Schluss der Seite.

Entities

Das sind Sonderzeichen, die man nicht in der Art und Weise schreiben sollte, wie man es gewohnt ist, um Fehlinterpretationen auszuschließen. Dafür gibt es die Entities.

Jedes Entity beginnt mit einem &, dann kommt die Umschreibung des Sonderzeichens und wird mit einem Semikolon abgeschlossen.

Die wichtigsten Entities:

Sonderzeichen	Entity	Beschreibung
ä	<code>&auml;</code>	(a Umlaut)
Ä	<code>&Auml;</code>	(A Umlaut)
ö	<code>&ouml;</code>	(ö Umlaut)
Ö	<code>&Ouml;</code>	(Ö Umlaut)
ü	<code>&uuml;</code>	(ü Umlaut)
Ü	<code>&Uuml;</code>	(Ü Umlaut)
ß	<code>&szlig;</code>	
Leerzeichen	<code>&nbsp;</code>	non breakable space
"	<code>&quot;</code>	
&	<code>&amp;</code>	
<	<code>&lt;</code>	
>	<code>&gt;</code>	

Die Kommentare

gerade bei umfangreichen Seiten ist eine Kommentierung des Quellcodes nützlich, um die Übersicht zu behalten. Auch sie werden mit spitzen Klammern eingeleitet, jedoch von einem Ausrufezeichen und zwei Minuszeichen gefolgt - beendet werden sie mit -->.

`<!-- Hier steht ein Kommentar -->`

Ein Solcher Text wird vom Browser ignoriert, wird also nicht angezeigt.

Der Seitenaufbau

Das Grundgerüst einer Seite kann z. B. so aussehen:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">
<!-- Hier werden die Spezifikationen der Seite (HTML 4.0) angegeben -->
<html>
<!-- Hier wird das Dokument geöffnet. Es ist der erste Tag, und wird erst am Schluss wieder
geschlossen -->
<head>
<!-- Hier wird der Kopf geöffnet. Dort können Informationen der Seite, wie z. B. der Titel
angegeben werden.-->
<title>Seitentitel</title>
</head>
<!-- Hier wird der Kopf geschlossen -->
<body>
<!-- Hier steht der angezeigte Seiteninhalt -->
</body>
<!-- Hier wird die Seite geschlossen -->
</html>
<!-- Hier wird das Dokument geschlossen -->
```

Meine erste Seite

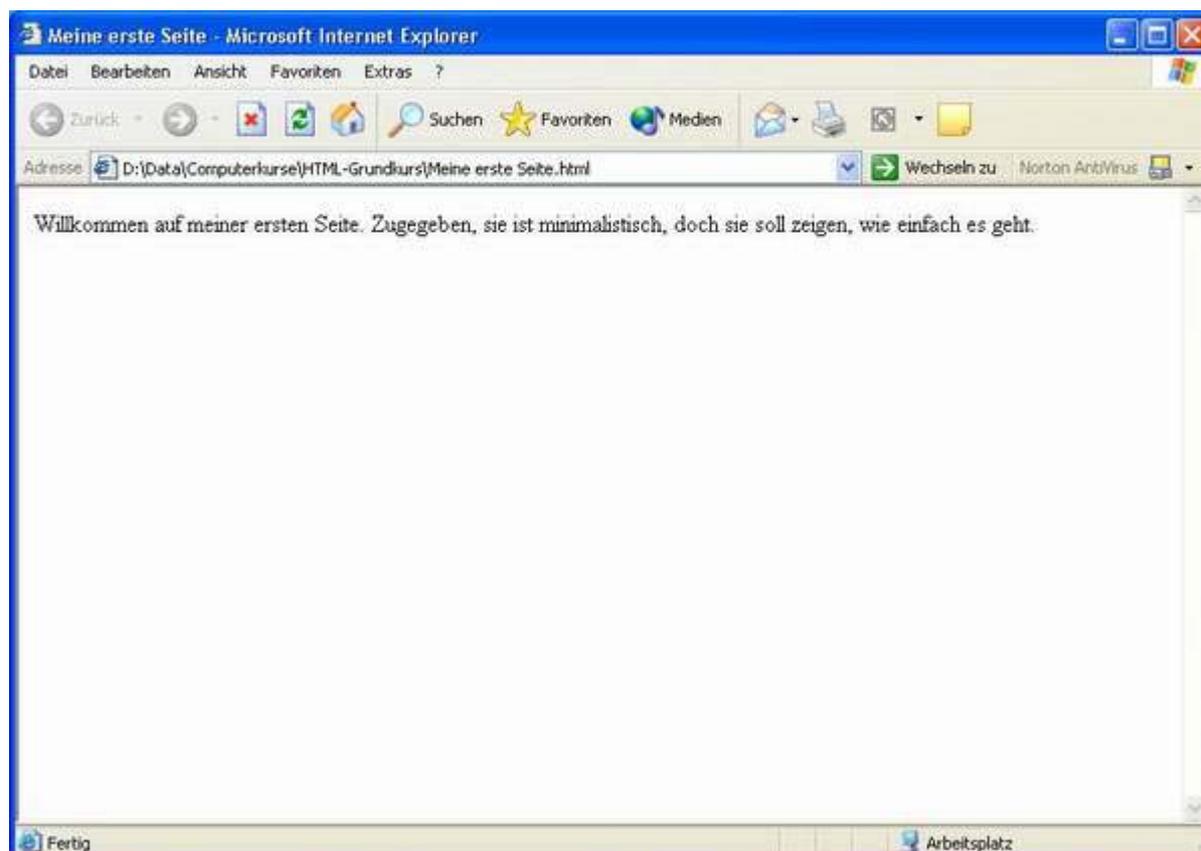
```
<!DOCTYPE HTML PUBLIC „-//W3C//DTD HTML 4.0//EN//“>
<html>
<head>
<title>Meine erste Seite</title>
</head>
<body>
```

Willkommen auf meiner ersten Seite.

Zugegeben, sie ist minimalistisch,
doch sie soll zeigen, wie einfach es geht.

```
</body>
</html>
```

Schreiben Sie diese Zeilen in eine ASCII - Datei (einfacher Text, z. B. mit Text-Editor von Microsoft) und ändern die Dateiendung von Meine erste Seite.txt in Meine erste Seite.html
Mit einem Doppelklick wird dieses Dokument vom Browser (z. B. Internet Explorer oder Netscape) geöffnet und sieht dann so aus:



In der Titelleiste (oben, blau) wird nun der im **<title></title>** eingegebene Titel angezeigt.
Texte innerhalb des **<body></body>** Tags werden im Seiteninhalt angezeigt.
Nur wo sind die Zeilenumbrüche und Absätze geblieben? Dies und mehr im Kapitel
Formatierungen.

Formatierungen

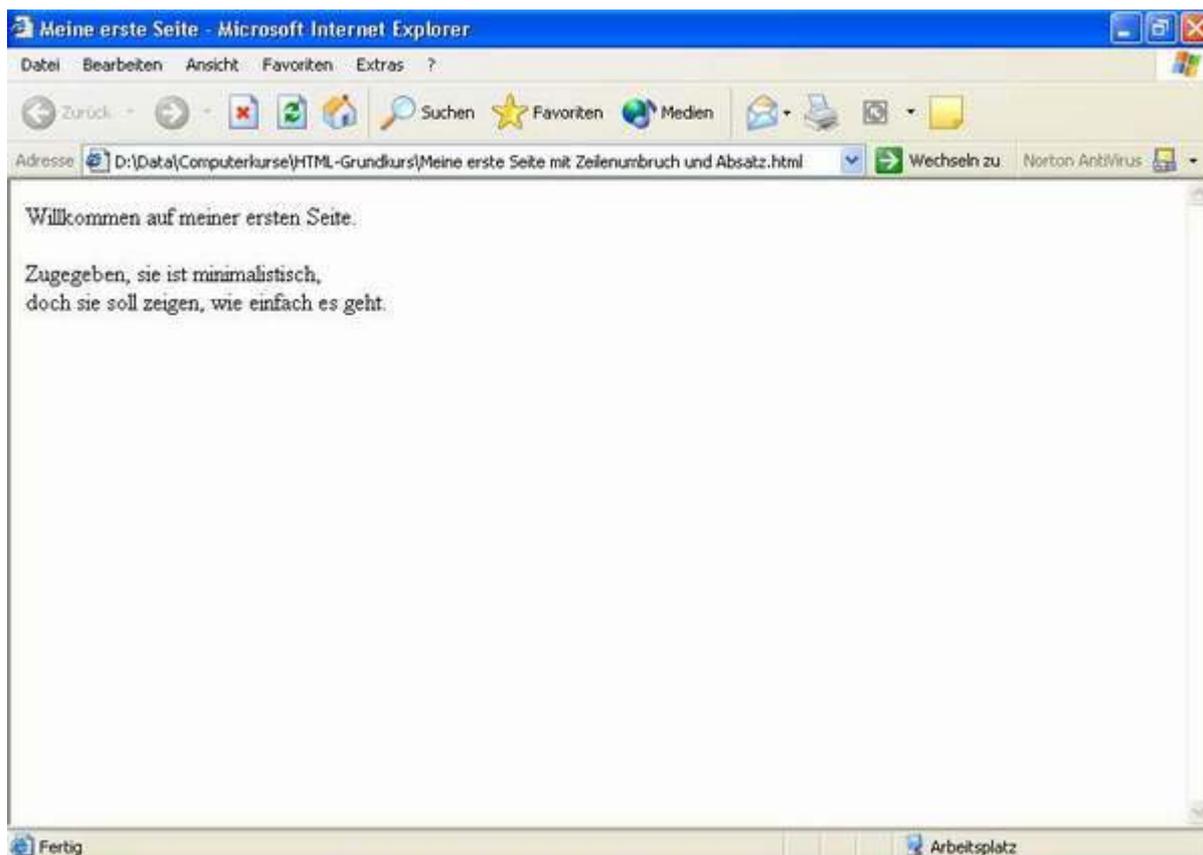
Die Wiedergabe einer Webseite ist dynamisch, passt sich also an Bildschirmgröße, Auflösung, Fenstergröße etc. an. (verkleinern Sie doch mal die Fenstergröße Ihrer ersten Webseite!) Möchte man z. B. einen Zeilenumbruch an einer bestimmten Stelle, muss man es dem Browser durch den *Tag* **
** (engl.: break) bekannt geben. Dieser *Tag* benötigt keinen *Endtag*!

Einen Absatz leite ich mit dem *Tag* **<p>** ein und wird mit **</p>** beendet.

In unserem Beispiel...

```
<!-- Quellcode Anfang -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN/">
<html>
<head>
<title>Meine erste Seite</title>
</head>
<body>
<p>Willkommen auf meiner ersten Seite. </p>
Zugegeben, sie ist minimalistisch,<br>
doch sie soll zeigen, wie einfach es geht
</body>
</html>
<!-- Quellcode Ende -->
```

...wird danach die Seite so angezeigt:



Ich möchte eine Überschrift? Das kann ich mit dem *Tag* `<h1></h1>` realisieren. Der zwischen den spitzen Klammern stehende Text wird nun als Überschrift1 interpretiert. (mögliche Varianten: `<h1></h1>` = sehr groß bis `<h6></h6>` = klein).

Danach vielleicht eine horizontale Linie? Der *Tag* `<hr>` erledigt es zuverlässig.

Doch den restlichen Text hätte ich gerne mittig dargestellt. Dies lässt sich mit dem *Attribut* eines *Tags* realisieren - hier die Ausrichtung. Füge ich nun dem *Tag* `<p>` das *Attribut* **align** (engl. = ausrichten) hinzu, kann man dieses Objekt rechts (**right**), mittig (**center**) oder links (**left**) ausrichten.

```
<p align="center">Zugegeben, sie ist minimalistisch,<br>
doch sie soll zeigen, wie einfach es geht </p>
```

Im Ganzen sieht das dann so aus:

```

<!-- Quellcode Anfang -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">
<html>
<head>
<title>Meine erste Seite</title>
</head>
<body>

<h1>Willkommen auf meiner ersten Seite.</h1>
<h2>Willkommen auf meiner ersten Seite.</h2>
<h3>Willkommen auf meiner ersten Seite.</h3>
<h4>Willkommen auf meiner ersten Seite.</h4>
<h5>Willkommen auf meiner ersten Seite.</h5>
<h6>Willkommen auf meiner ersten Seite.</h6>
<hr>
<p align="center">Zugegeben, sie ist minimalistisch,<br>
doch sie soll zeigen, wie einfach es geht </p>
</body>
</html>
<!-- Quellcode Ende -->

```

Nachdem ich diese Änderungen vorgenommen habe, sieht unsere Seite jetzt so aus:



Weitere Möglichkeiten der Formatierung sind u. a.:

- die Schrift (*Tag*: ``) in Größe (*Attribut*: **size**) und Art (*Attribut*: **face**),
- die Horizontale Linie (*Tag* `<hr>`) in einer festgelegten Breite (*Attribut*: **width** - Weite) und Größe (*Attribut*: **size**) und Farbe (*Attribut*: **color**) anzuzeigen oder
- den Text in ein Listing (*Tag* ``) zu fassen. In unserem Beispiel könnte das so aussehen:

```
<!-- Quellcode Anfang -->
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">
```

```
<html>
```

```
<head>
```

```
<title>Meine erste Seite</title>
```

```
</head>
```

```
<body>
```

```
<p align="center"><font size="5" face="Courier New, Courier, mono">Willkommen  
auf meiner ersten Seite.</font></p>
```

```
<hr width="80%" size="50" color="blue">
```

```
<li>Zugegeben, sie ist minimalistisch,</li>
```

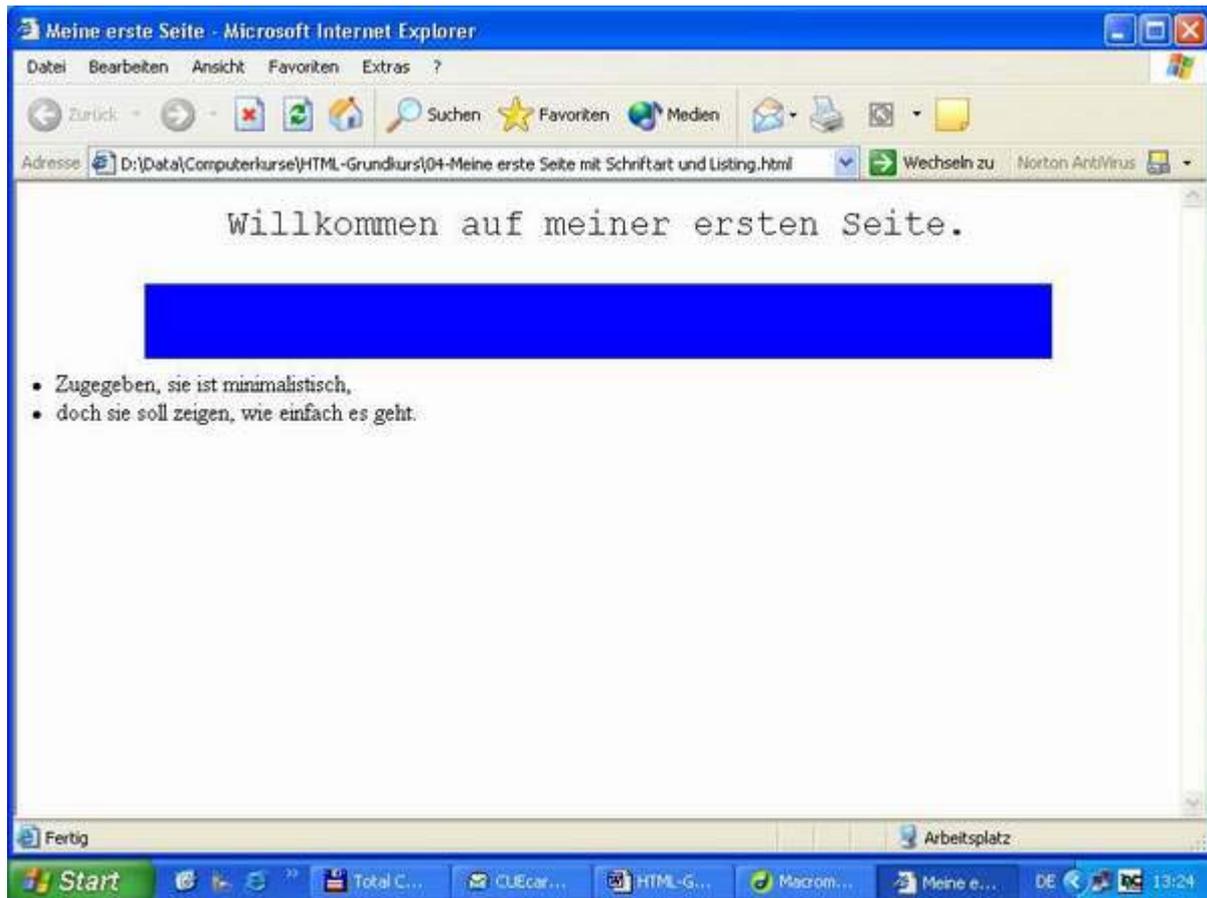
```
<li>doch sie soll zeigen, wie einfach es geht. </li>
```

```
</body>
```

```
</html>
```

```
<!-- Quellcode Ende -->
```

Wird im Internet Explorer so dargestellt:



Da wir später mit HTML-Editoren arbeiten werden, möchte ich nicht mit der Vielzahl an Tags Verwirrung stiften. Ich wollte nur das ziemlich einfache Grundprinzip der HTML-Programmierung erkennbar machen. Eine Auflistung der hier vorgestellten Tags finden Sie auf der Seite "Tags und Attribute"

Siehe auch [CSS – Stile](#).

Verzichten wir auf solche (und weitere) Formatierungen, können wir auch einen Text mit dem Tag `<pre></pre>` (preformatted) "vor-formatieren". Dabei werden Zeilenumbruch, Leerzeichen/Zeilen etc. wie im Quellcode interpretiert. Schriftgröße und Schriftart werden in Standardwerten des Browsers angezeigt, wie im nächsten Beispiel.
Also wieder zurück zum Minimalismus:

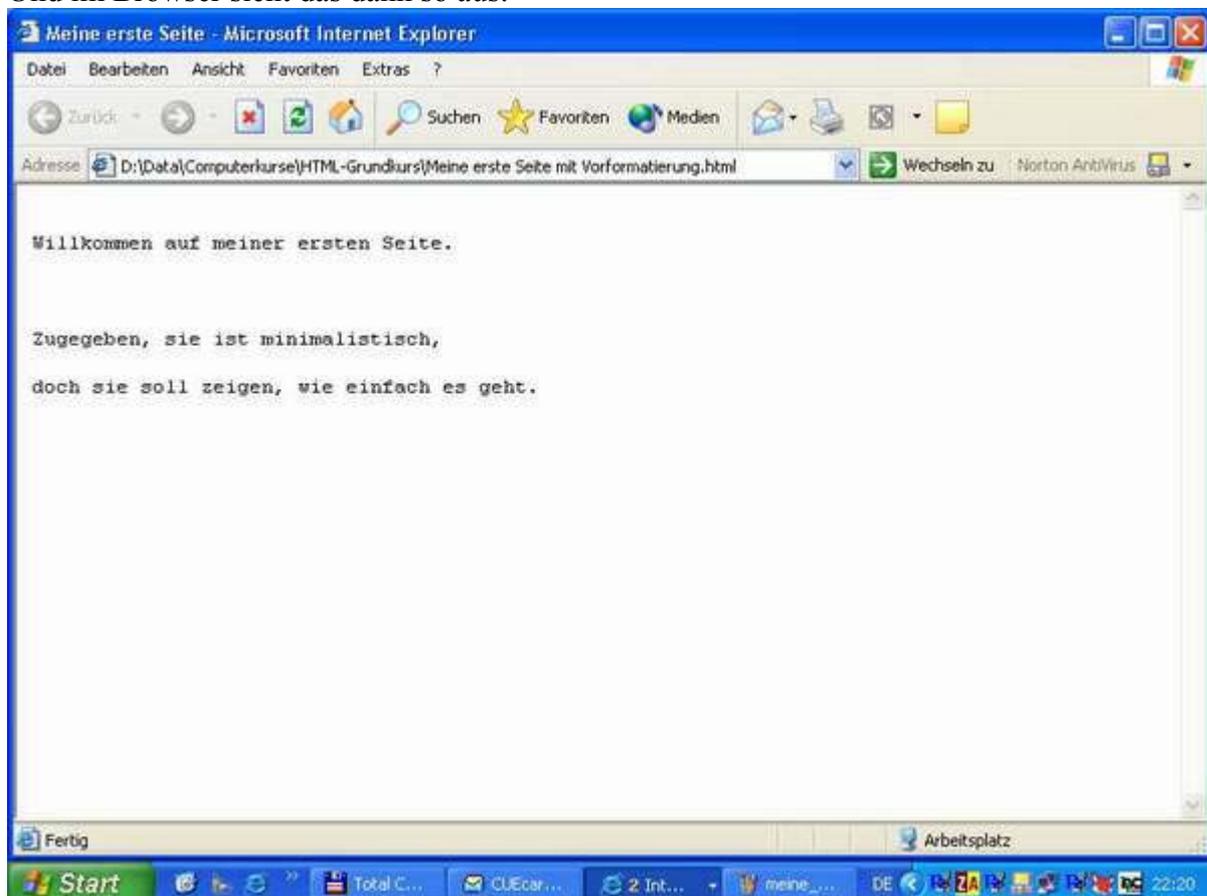
```
<!-- Quellcode Anfang -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">
<html>
<head>
<title>Meine erste Seite</title>
</head>
<body>
<pre>
Willkommen auf meiner ersten Seite.
```

Zugegeben, sie ist minimalistisch

doch sie soll zeigen, wie einfach es geht

```
</pre>
</body>
</html>
<!-- Quellcode Ende -->
```

Und im Browser sieht das dann so aus:



Bilder einfügen

Was wäre das Web ohne Bilder! Zwar kann man auch alles im Text beschreiben, doch ein Bild, eine Grafik kann es besser verdeutlichen. Ein Bild sagt mehr als tausend Worte!

Ein besonderes Augenmerk liegt dabei auf zwei Grafikformate: das *.jpeg, bzw. *.jpg Dateiformat (Joint Photographic Experts Group) und das *.gif Dateiformat (Graphics Interface Format), da beide Formate eine hohe Kompression der Daten aufweisen. Dadurch werden Ladezeiten verkürzt und der Surfer muss nicht Minutenlang auf den Aufbau des Bildes warten!

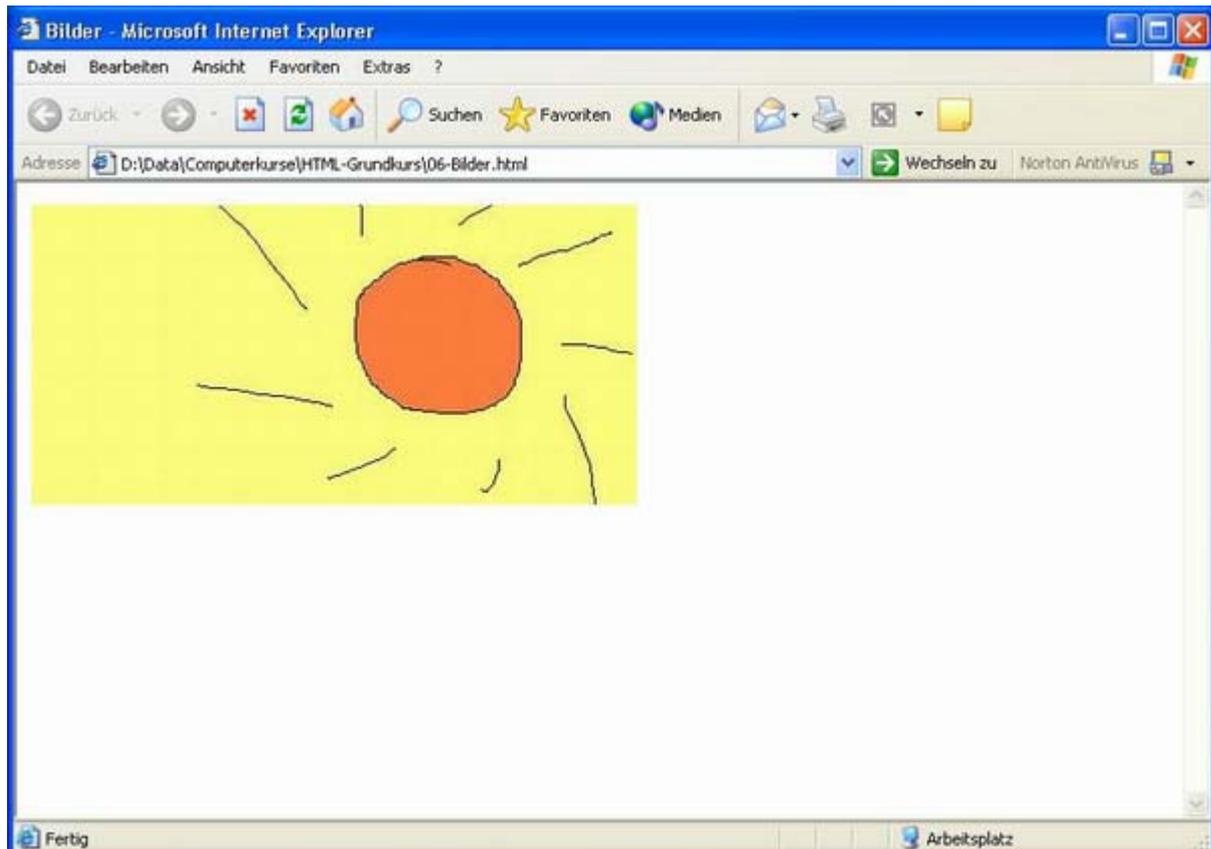
(Mehr über Grafikformate, Konvertierung, Animationen, etc... im Folgekurs).

Eingefügt wird das Bild mit dem *Tag* **** (engl.: Image = Bild), gefolgt von dem *Attribut* **src** (engl.: source = Quelle), welches auf den Speicherort und den Dateinamen verweist, z.B.: ****

```
<!-- Quellcode Anfang -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">
<html>
<head>
<title>Bilder</title>
</head>
<body>

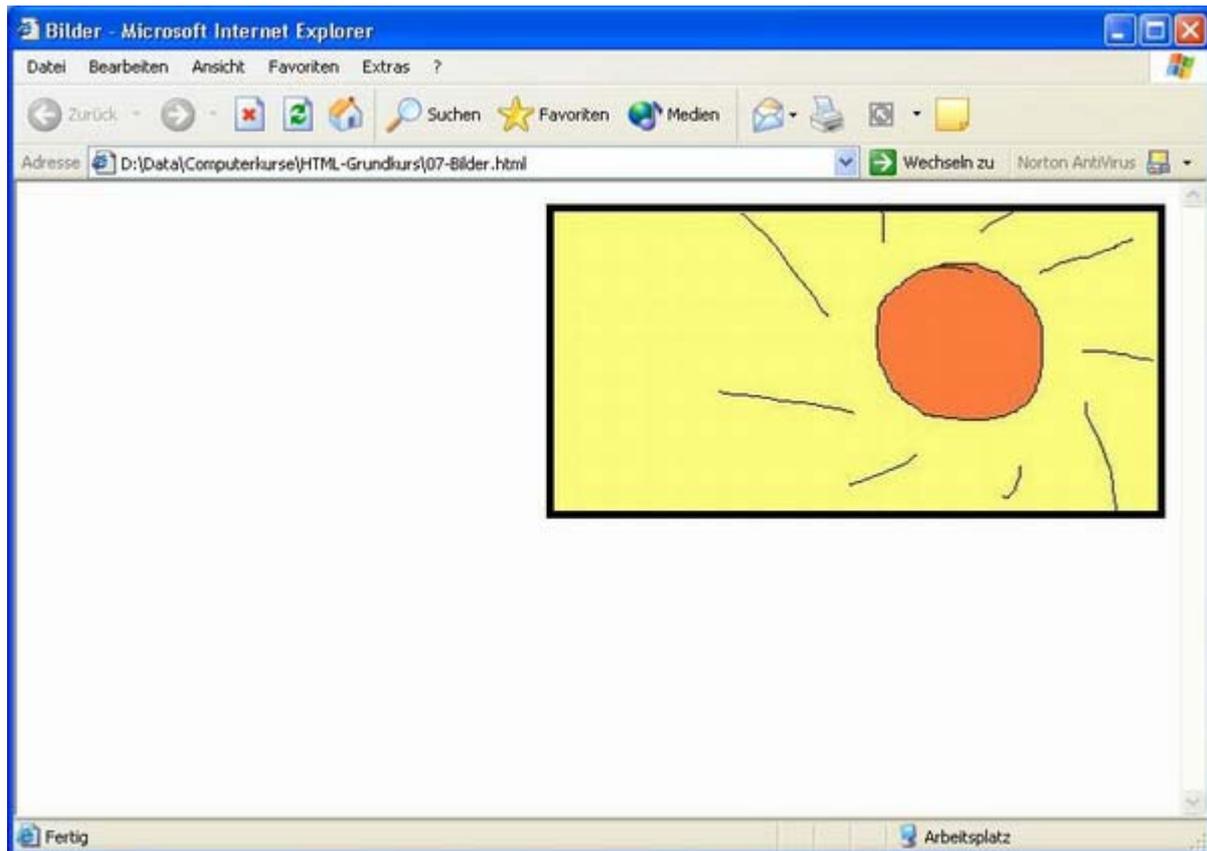
</body>
</html>
<!-- Quellcode Ende -->
```

Dies schreibe ich in die Datei bilder_01.html. Von dem Speicherplatz dieser Datei aus sucht der Browser nun nach einer Bilddatei Namens „sonne.jpg“ und fügt es im Browserfenster ein.



Nun könnten wir noch das wie und wo des Bildplatzes näher bestimmen, wie z. B. mit den *Attributen* Rahmen (**border**), rechtsbündig (**align="right"**) und einen alternativen Text (**alt="hier solltet Ihr eine Sonne sehen"**) eingeben. Eine saubere Programmierung verlangt auch die Attribute **widht** (Weite) und **height** (Höhe) des Bildes.

```
<!-- Quellcode Anfang -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">
<html>
<head>
<title>Bilder</title>
</head>
<body>
<div align="right"></div>
</body>
</html>
<!-- Quellcode Ende -->
```



Im Grunde genommen alles recht einfach. Noch einfacher ist es mit HTML-Editoren, die dem Anwender Hilfestellung geben. Noch einfacher geht es mit Tools wie Frontpage oder Dreamweaver. Doch das passende Grundwissen sollte man sich aneignen – schon alleine deswegen, um Fehler besser aus dem Weg räumen zu können.

Beim Bildeinfügen stößt man meist auf eins dieser Probleme:

1. Groß / Kleinschreibung nicht berücksichtigt
2. Umlaute
3. Fehler in der Pfadangabe (auch \ statt /).

1.) Würde ich im Quellcode auf die Datei Sonne.JPG statt sonne.jpg verweisen, so hätte ich auf einem Windows - Rechner den gleichen Erfolg: der Browser würde mit die Datei [sonne.jpg](#) anzeigen ([das ist der tatsächliche Name](#)), wie in der Datei 08-bilder.html. Veröffentliche ich jetzt diese Seite auf einem Linux - Server (die meisten Webserver im Netz laufen auf Linux ;-)) bekäme ich nur den Alternativtext angezeigt.

Auflösung: Linux - Rechner sind case sensitive (Unterscheiden zwischen Groß und Kleinschreibung), die Datei Sonne.JPG existiert nicht (die Datei heißt [sonne.jpg](#)), und kann folglich nicht angezeigt werden.

TIPP: Es ist egal, ob Sie die Dateinamen und Dateinamenserweiterungen groß oder klein schreiben, nur gewöhnen Sie sich an, die Dateinamen/serweiterungen IMMER GROß oder IMMER KLEIN zu schreiben.

2.) Auch Umlaute könnten zur Stolperfalle werden – am besten, Sie vermeiden Umlaute wie ä, ö und ü und verwenden stattdessen die Schreibweise ae, oe oder ue an.

3.) Fehlende oder falsche Pfadangabe. Dafür sollte man den Unterschied zwischen relativer und absoluter Pfadangabe kennen lernen und verstehen.

Da diese Problematik auch bei der Verlinkung auftritt, habe ich ihm ein extra Kapitel gewidmet: [relativ oder absolut](#).

Auch sollte man in der Pfadangabe immer einen Slash (/) statt ein Backslash (\) setzen. Der Microsoft Internet Explorer ignoriert zwar diesen Unterschied, Netscape besteht jedoch auf die korrekte Schreibweise mit / in der Pfadangabe, sonst kann er der Pfadangabe nicht folgen.

Hyperlinks - relativ oder absolut

Als Hyperlink, auch kurz Link (Verknüpfung, Verbindung, Verweis), bezeichnet man einen Verweis auf eine andere Datei (Bilder, Webseiten, Musikstücke, etc..) im Hypertext, der durch das HTML Protokoll automatisch verfolgt werden kann.

Einen Link kann man auch mit einem Querverweis in einem Buch vergleichen.

Typischerweise wird der Verweis auf eine andere Webseite im Quellcode "verankert" - Mit dem Tag `<a` (anker), gefolgt von `href` (Angabe der Hypertext-Referenz), dann der eigentliche Verweis = "**irgendeineseite.html**". Dann kommt der angezeigte Text wie **mehr...**, danach muss der geöffnete Tag nur noch mit `` geschlossen werden.

Eine Website benötigt eine Einstiegsseite (diese sollte immer `index.html` oder `default.html` heißen) mit Verweisen zu allen Themenseiten und in jeder Themenseite einen Rücksprungverweis zur Einstiegsseite. Hier ein Beispiel:

```
<a href="http://www.8ung.at/mcp.becker"> Hier geht es zu meiner Webseite</a>
```

Da nur ein Ordner (Die Website) und keine Datei angegeben wurde, wird der Browser nach der `http://www.8ung.at/mcp.becker/index.html` - Datei suchen und diese öffnen.

Egal, ob ich auf ein Bild verweise (``) oder einen Link einfüge (` andere Webseite `) dessen Ziel nicht im gleichen Verzeichnis liegt wie die ausgeführte html-Datei, muss ich den Weg zu dieser Datei benennen.

Beispiel 1: (**ohne Pfadangabe**, da im gleichen Verzeichnis)

Ich rufe die Datei `bilder.html` auf:

mcp.becker/homepage/html/bilder.html

Sie verweist auf die Datei `sonne.jpg`, die im *gleichen* Verzeichnis liegt

mcp.becker/homepage/html/sonne.jpg – eine Pfadangabe ist nicht nötig!

```

```

Beispiel 2: (**relative Pfadangabe**)

Bei einer umfangreichen Website mit mehreren Bilddateien habe ich das Verzeichnis „images“ angelegt, in dem dann alle Bilder liegen.

(mcp.becker/images/sonne.jpg)

Ich rufe wieder die Datei `bilder.html` auf

mcp.becker/homepage/html/bilder.html

– jetzt muß ich erst mal zwei Verzeichnisse höher (`homepage/html/`), dann ins Verzeichnis `/images`, wechseln, wo meine `sonne.jpg` liegt.

(``). Mit `../` wechsele ich ins jeweils höhere Verzeichnis!

Oder:

Ich rufe die Datei `beispiel.html` auf.

mcp.becker/kurse/homepage/html/beispiel.html

(sie liegt im Verzeichnis `/html`) und verweist auf das Bild

mcp.becker/kurse/homepage/html/beispiele/images/haus.jpg

so muß ich in die Unterverzeichnisse `beispiele/images` wechseln! Die relative Pfadangabe:

```

```

Beispiel 3: (**absolute Pfadangabe**)

Ich veröffentliche diese Website unter www.8ung.at. Dort liegt das Sonnenbild im Verzeichnis (www.8ung.at/mcp.becker/images/sonne.jpg)

Nun kann jeder dieses Bild mit der absoluten Pfadangabe verlinken.

(``).

Innerhalb einer Website sollte man immer die relative Pfadangabe bevorzugen. Stellen Sie sich vor, sie ziehen mit Ihrer Website um (z.B. von www.meinname.de nach www.unsername.de) – dann müssten sie ALLE absoluten Pfadangaben ändern.

Tabellen – Layout-Gestaltung

Tabellen werden meist verwendet, um Texte und Grafiken geordnet und übersichtlich am Bildschirm zu verteilen. Ursprünglich sind Tabellen zur Darstellung tabellarischer Daten geschaffen worden, doch durch Ihre individuelle Gestaltungsvielfalt sind sie in der heutigen Praxis überwiegend für Seiten-Layouts nicht mehr wegzudenken.

Somit sind Tabellen eine übliche Methode, eine Struktur zu verwirklichen.

Auch auf meiner Homepage habe ich mich der Tabellenform bedient, um die Navigation übersichtlich und strukturiert darzustellen.

Mit dem Tag `<table></table>` (table = Tabelle) wird eine Tabelle erstellt (Endtag erforderlich)

Eine Zeile wird mit dem Tag `<tr></tr>` (Table Row = Tabellen Reihe) wird eine Zeile eingefügt.

Das Tag `<td></td>` (Table Data = Tabellen Daten) wird eine Zelle erstellt, in welcher später der Inhalt eingefügt wird.

In der Regel sind diese Tabellen "unsichtbar", d.h. nicht als solche erkennbar. Zum besseren Verständnis werden die Tabellen erst einmal mit dem Attribut `border="1"` gekennzeichnet.

Und hier unsere erste Tabelle - sie soll aus einer Zeile und zwei Spalten (also Inhalte) bestehen:

```
<!-- Quellcode Anfang -->
<!DOCTYPE HTML PUBLIC „-//W3C//DTD HTML 4.0//EN//“>
<html>
<head>
<title>Meine erste Seite mit einer Tabelle </title>
</head>
<body>

<table border="1">
<tr>
<td>Inhalt 1</td>
<td>Inhalt 2</td>
</tr>
</table>

</body>
</html>
<!-- Quellcode Ende -->
```

Das sieht dann so aus:

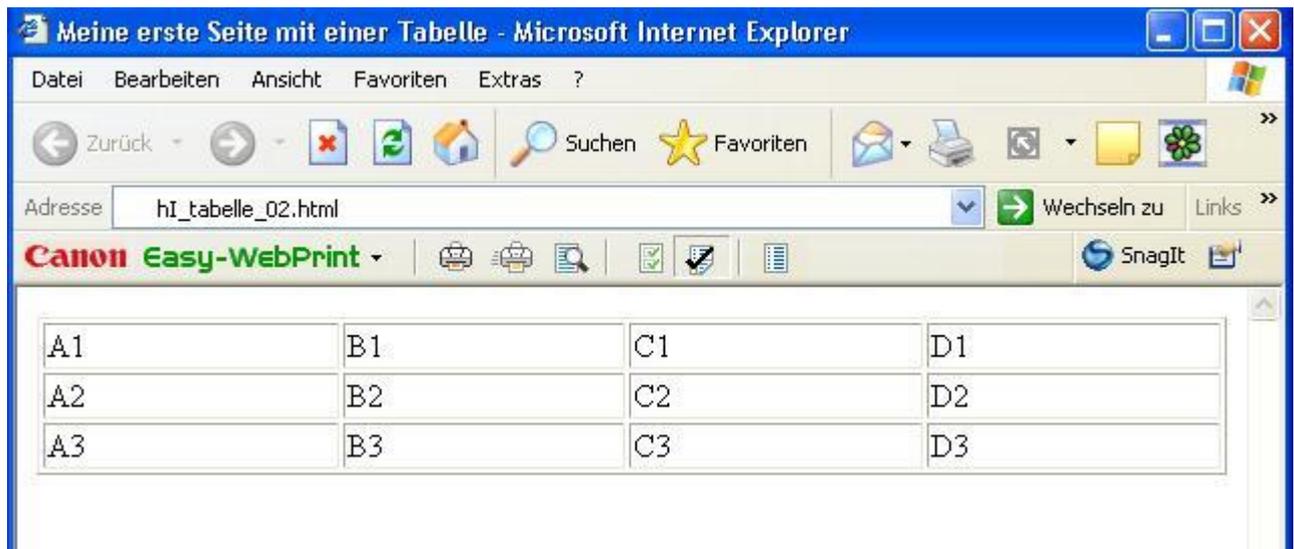


Im nächsten Beispiel möchten wir eine Tabelle mit vier Spalten (A - D) und drei Zeilen (1 - 3) erstellen, die zusätzlich noch 100% der verfügbaren Breite einnimmt (Attribut **weight="100%"** im innerhalb des **<table>** Tags. Der Quellcode sieht dann so aus:

```
<!-- Quellcode Anfang -->
<!DOCTYPE HTML PUBLIC „-//W3C//DTD HTML 4.0//EN//“>
<html>
<head>
<title>Meine erste Seite mit einer Tabelle </title>
</head>
<body>

<table border="1" weight="100%">
<tr>
<td> A1</td>
<td> B1</td>
<td> C1</td>
<td> D1</td>
</tr>
<tr>
<td> A2</td>
<td> B2</td>
<td> C2</td>
<td> D2</td>
</tr>
<tr>
<td> A3</td>
<td> B3</td>
<td> C3</td>
<td> D3</td>
</tr>
</table>
</body>
</html>
<!-- Quellcode Ende -->
```

Und im Browser sehen wir dann dieses Ergebnis:



Im nächsten Beispiel möchte ich die Zellen A1 - D1 verbinden - was mit dem Attribut **colspan="4"** (**cols** = columns = Spalten und **span** = überspannen) innerhalb des ersten **<th>** Tags erreicht wird.

Dann möchte ich die Zellen A2 und A3 verbinden - dies wird mit dem Attribut **rowspan="2"** (**rows** = Reihen und **span** = überspannen) innerhalb des ersten **<th>** Tags erreicht wird.

Und um etwas Farbe in die Tabelle zu bekommen, fügen wir den Zellen noch das Attribut **bgcolor="farbe"** (**bgcolor** = Background Color = Hintergrundfarbe) hinzu. (bei "farbe" können bestimmte englische Farbnamen oder auch Hexadezimale Werte, z.B. #C50B3A eingesetzt werden). Hier der Quellcode der Tabelle:

```
<!-- Quellcode Anfang -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">
<html>
<head>
<title>Meine erste Seite mit einer Tabelle - Zellen verbinden und einfärben</title>
</head>

<body>

<table border="1" width="100%">
<tr bgcolor="orange">
<th colspan="4"> 4 Spalten verbinden f&uuml;r die Tabellen&uuml;berschrift</th>
</tr>

<tr>
<td width="25%" bgcolor="#C50B3A" rowspan="2"> 2 Reihen verbinden </td>
<td width="25%" bgcolor="lime"> B2</td>
<td width="25%" bgcolor="yellow"> C2</td>
<td width="25%" bgcolor="red"> D2</td>
</tr>
```

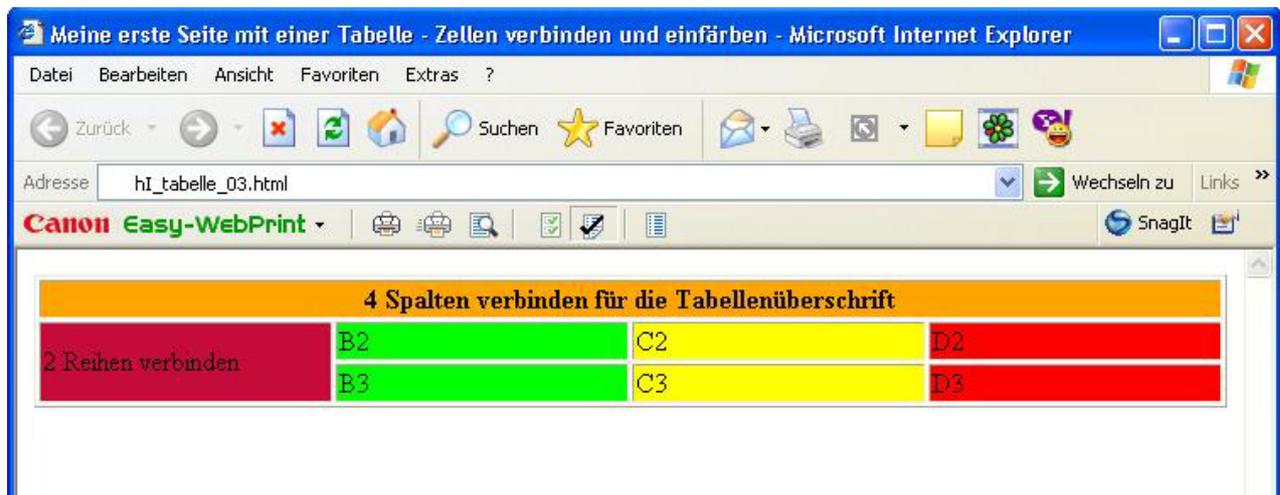
```

<tr>
<td width="25%" bgcolor="#00FF00"> B3</td>
<td width="25%" bgcolor="#FFFF00"> C3</td>
<td width="25%" bgcolor="#FF0000"> D3</td>
</tr>
</table>

</body>
</html>
<!-- Quellcode Ende -->

```

Dies wird im Browser dann so dargestellt:



So, und zum Schluss noch ein Beispiel eines typischen Tabellen-Layouts.

Damit man keine Rahmen sieht, geben wir dem Attribut **border** den Wert **"0"** mit (**border="0"**) und bestimmen mit den Attributen **cellpadding="0"** (cellpadding = Zelleninnenabstand, der Abstand zwischen Zellenrand und Zelleninhalt in Pixeln) und **cellspacing="0"** (cellspacing = Zellenabstand, der Abstand der Zellen untereinander in Pixeln), dass keine Lücken (Abstände) entstehen.

Im Quellcode stellt sich das so dar:

```

<!-- Quellcode Anfang -->
<!DOCTYPE HTML PUBLIC „-//W3C//DTD HTML 4.0//EN//“>
<html>
<head>
<title>Meine erste Seite mit einer Tabelle - Seitenlayout mit Tabellen</title>
</head>
<body text="#000000" bgcolor="#FFFFFF" link="#006633" alink="#FF0000"
vlink="#006633">

<table border="0" cellpadding="0" cellspacing="0" width="100%">
<tr bgcolor="orange">
<th> </th>
<th><h1>HTML-Kurs I </h1></th>
</tr>
<tr>
<td width="25%" height="100%" valign="top" bgcolor="orange">
<h1>Navigation</h1>
<p><a href="../hI_syntax.html" target="_blank">Syntax</a><br>
<a href="../hI_formatierungen.html" target="_blank">Formatierung</a> <br>
<a href="../hI_bilder.html" target="_blank">Bilder</a><br>
<a href="../hI_relativ.html" target="_blank">Hyperlinks</a><br>
<a href="../hI_tabellen.html" target="_blank">Tabellen</a><br>
<a href="../hI_frames.html" target="_blank">Frames</a></p>
<h1>Aktuell </h1>
<p>Heute im Angebot:<br>
Ganz neue, frische Webseiten!!!</p>

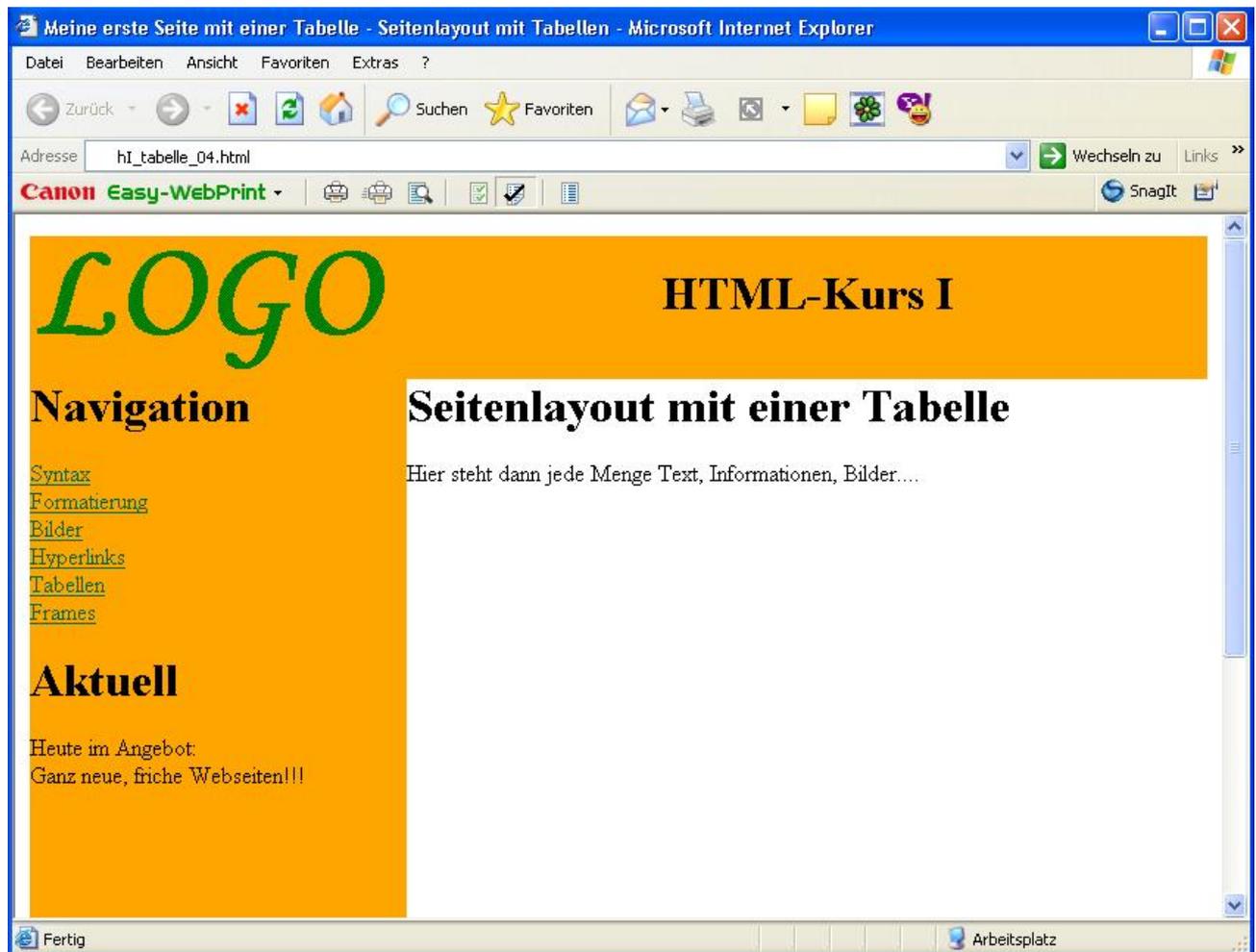
<!-- Hier ein paar leere Umbrüche als Lückenfüller -->
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<!-- Ende leere Umbrüche-->

</td>
<td valign="top">
<h1>Seitenlayout mit einer Tabelle</h1>
<p>Hier steht dann jede Menge Text, Informationen, Bilder....</p>
<p>&nbsp;</p></td>
</tr>
</table>

</body>
</html>
<!-- Quellcode Ende -->

```

Und hier das Ergebnis....



....kann sich doch schon sehen lassen!

Frames

Mit Hilfe von Frames (Felder) können Sie den Anzeigebereich des Browsers in verschiedene, frei definierbare Segmente aufteilen. In jedem Feld wird ein eigener Inhalt (eine .html Datei) angezeigt.

So wird häufig die Navigation in einem Frame (meist auf der linken Seite) und die Seiteninformation in einem anderen Frame dargestellt.

Ein Frame wird in der Startseite (index.html) mit dem Tag `<frameset></frameset>` definiert, und zwar vor dem `<body></body>` Tag, denn in dieser Datei wird kein Inhalt gespeichert, nur die Information, wo die Inhalte liegen! Dabei wird in unserem Beispiel mit dem Attribut `cols` die Spaltenbreite definiert (von Links nach Rechts).

Das kann mit einer festen Pixelzahl (z.B. 50) oder einer Prozentangabe erfolgen (z.B. 20%).

Bei der Angabe der "Übrigen Spalte" wird nach einer Angabe in Pixeln meist ein Platzhalter, das Asterix (*) benutzt, um anzuzeigen, dass der gesamte Rest der verfügbaren Fläche dem Übrigen Segment zugeteilt wird. (`<frameset cols="200,*">`)

Im Anschluss der Aufteilung werden die zu ladenden Quellen (Source - `src`) angegeben. Hier:

```
<frame src="navigation.html">
```

```
<frame src="start.html">
```

Nun bekommt jeder Frame noch das Attribut `name` mit auf dem Weg. Das ist besonders wichtig, wenn später ein Link vom `"navigationsframe"` im `"inhaltsframe"` geöffnet werden soll. In unserem Beispiel also:

```
<frame src="navigation.html" name="navigationsframe">
```

```
<frame src="start.html" name="inhaltsframe">
```

Für den Fall, dass der Browser keine frames unterstützt, wird der `<body></body>` Tag mit dem `<noframes></noframes>` Tag eingeschlossen, dort kann eine Nachricht eingebunden werden, wie z. B., dass man bitte eine neue Browser-Version installieren muss, um diese Webseite zu sehen. Hier der Quellcode der Datei index.html:

```

<!-- Quellcode Anfang -->
<!DOCTYPE HTML PUBLIC „-//W3C//DTD HTML 4.0//EN//“>
<html>
<head>
<title>Meine erste Seite mit Frames</title>
</head>
<frameset cols="200,*">
<frame src="navigation.html" name="navigationsframe">
<frame src="start.html" name="inhaltsframe">
</frameset>
<noframes>
<body>
Ihr Browser unterstützt keine Frames - bitte installieren Sie eine neuere Version.
</body>
</noframes>
</html>
<!-- Quellcode Ende -->

```

Damit wäre der Rahmen definiert, wir benötigen jetzt noch die beiden html-Dateien navigation.html und index.html. In der Datei navigation.html werden die Links mit dem Attribut **target="inhaltsframe"** angewiesen, das Ziel im Hauptframe anzuzeigen, das ganze sieht dann so aus:

```

<!-- Quellcode Anfang -->
<!DOCTYPE HTML PUBLIC „-//W3C//DTD HTML 4.0//EN//“>
<html>
<head>
<title>Navigation</title>
</head>
<body text="#000000" bgcolor="#FFFFFF" link="#006633" alink="#FF0000"
vlink="#006633">

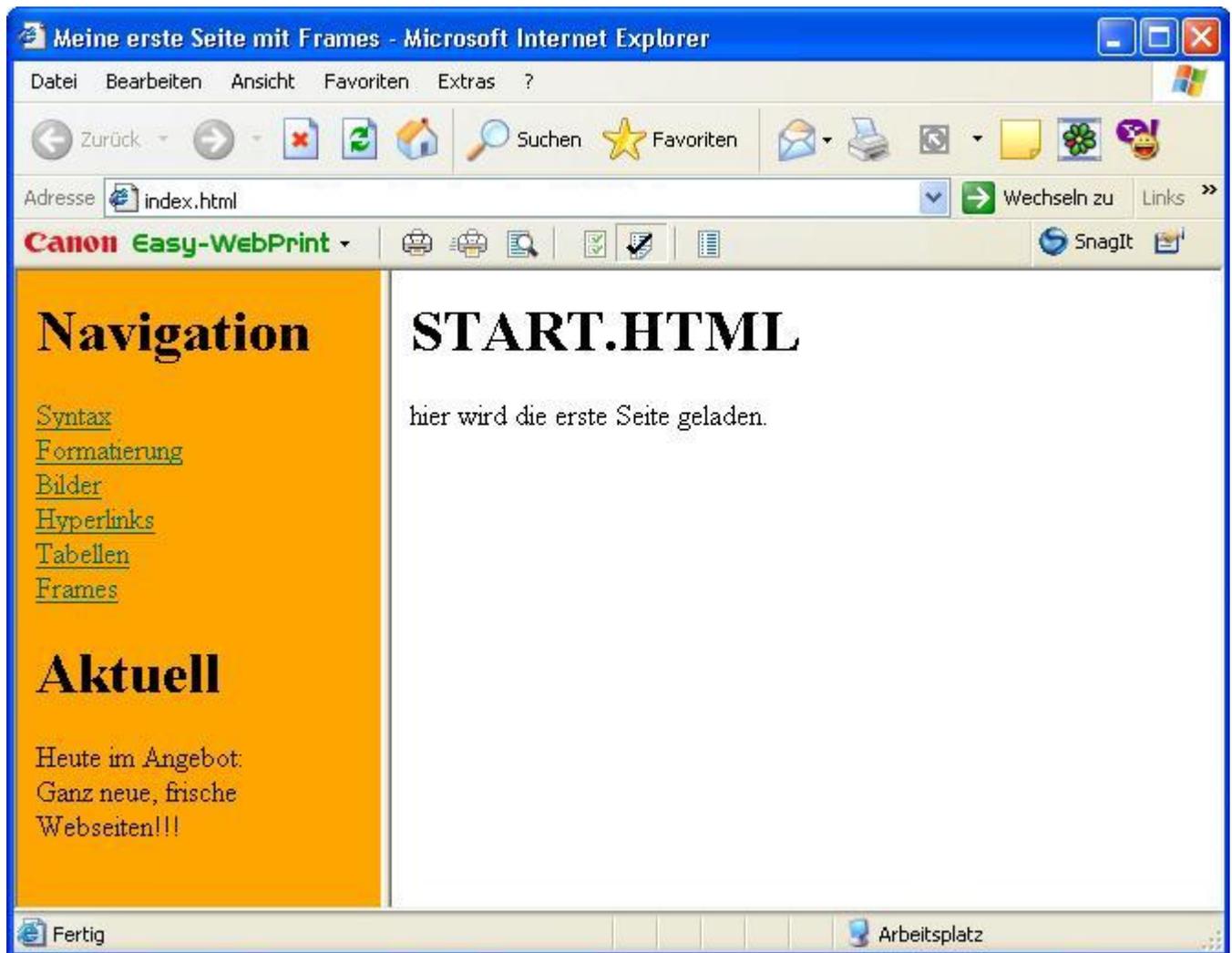
<h1>Navigation</h1>
<p>

<a href="../hI_syntax.html" target="inhaltsframe">Syntax</a><br>
<a href="../hI_formatierungen.html" target="inhaltsframe">Formatierung</a> <br>
<a href="../hI_bilder.html" target="inhaltsframe">Bilder</a><br>
<a href="../hI_hyperlinks.html" target="inhaltsframe">Hyperlinks</a><br>
<a href="../hI_tabellen.html" target="inhaltsframe">Tabellen</a><br>
<a href="../hI_frames.html" target="inhaltsframe">Frames</a></p>
<h1>Aktuell </h1>
<p>Heute im Angebot:<br>
Ganz neue, frische Webseiten!!!</p>

</body>
</html>
<!-- Quellcode Ende -->

```

Und das ganze sieht dann so aus:



FTP

Seit den Anfängen des Internet ist auch das File Transfer Protocol - kurz FTP - mit von der Partie. (Siehe auch Internet-Kurs, Dienste, FTP). Hier betrachten wird lediglich die Nutzung zur Synchronisation der Homepage mit dem Webserver.

FTP-Programme giebt's wie Sand am Meer, mittels Suchmaschinen findet man auch schnell ein paar kostenlose Programme, die ihre Arbeit ebenso verrichten wie die kostenpflichtigen. Das Windows hauseigene FTP-Programm heist schlicht ftp.exe (im Systemordner) - ein rein Befehlszeilenorientiertes Programm (DOS) und dürfte kaum noch eingesetzt werden. Heute sind grafische Programme Standard und machen die Arbeit Kinderleicht.

Die wohl bekanntesten sind:

Der Total Commander - ein ausgereiftes Universalwerkzeug für lokale Dateiverwaltung, Packen/Entpacken und auch für FTP-Verbindungen mit dem Look+Feel des legendären Norton Commander. Shareware;

Und der WS_FTP - Wohl die älteste grafische FTP-Oberfläche der Windows-Welt und das am weitesten verbreitete FTP-Programm. Uploaden und Downloaden und Fernarbeiten auf Server-Rechnern. Shareware; eine ältere eingeschränkte Version ist für Privatanwender als Freeware verfügbar.

Es ist jedoch bei allen Programmen gleich, man benötigt immer die gleichen Daten, um sich mit dem Server zu verbinden und Dateien auszutauschen, hier am Beispiel des Total Commanders beschrieben:

Ein Klick auf das FTP-Symbol  öffnet ein Auswahlfenster verfügbarer gespeicherter Verbindungsdaten.



Ist noch keine gespeichert, kann man mit einem Klick auf "Neue Verbindung eine hinzufügen:

Hier sollten die Verbindungsdaten, die man vom Provider (Anbieter) bekommt eintragen. Im Grunde reichen Servername (meist gleich der Web-Adresse, Benutzername und Passwort.

FTP: Verbindungsdetails

Titel:

Servername[:Port]:

Benutzername:

Passwort:

Warnung: Das Speichern des Passworts ist ein Sicherheitsrisiko!

Entferntes Verz.:

Lokales Verz.: >>

Sende Befehle:

Server-Typ:

Benutze Firewall-Server (Proxy)

Passiven Modus für Transfers verwenden (wie WWW-Browser)

Sende Befehl zum Aufrechterhalten der Verbindung

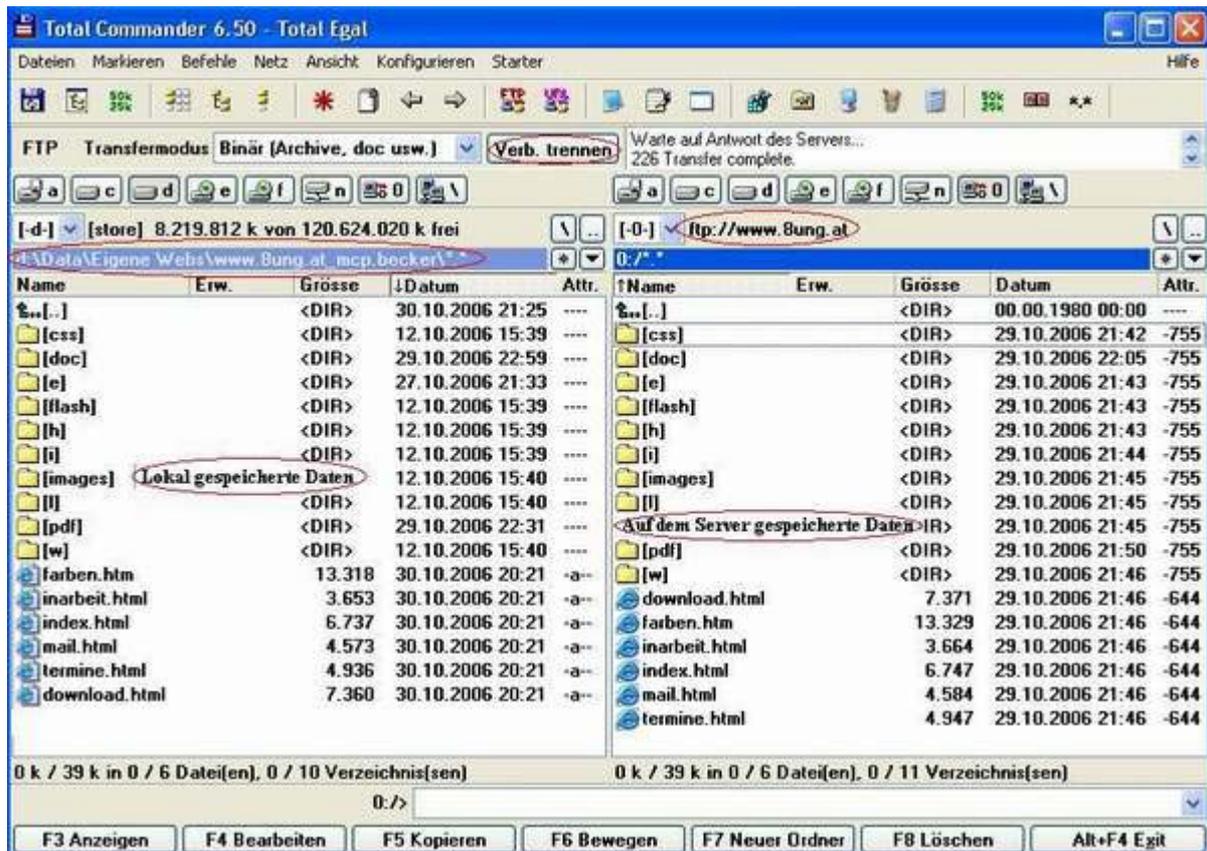
Befehl: Sendintervall: Alle s

Mit OK die Daten speichern, und um Fenster Mit FTP-Server verbinden auf Verbinden klicken.

Mit FTP-Server verbinden

Verbinden mit

Nun wird im Aktiven Fenster die Ordner und Dateien auf dem Server angezeigt, auf der anderen Seite sollte man seine lokal gespeicherte Webseite öffnen.



Hier kann man wie gewohnt von einer Seite zur anderen kopieren und verschieben, Dateien löschen, etc...

Damit sind sie dann veröffentlicht und im World Wide Web verfügbar.

Tags und Attribute

Zum Schluss eine Zusammenfassung der wichtigsten Tags und deren Attribute:
Dabei ist es völlig egal, ob Sie die Tags groß oder klein schreiben!

Tag	Engl.	Beschreibung	Beispiel/Kommentar
<code><body></body></code>	body	Textkörper	<i>Inhalte werden im Browser dargestellt</i>
<code>
</code>	break	Zeilenumbruch	
<code><h1></h1></code>	heading	Überschrift 1	
<code><head></head></code>	head	Kopf	<i>innerhalb dieses Tags stehen Seiteninformationen, wie der Titel (<title>) und die <meta> - Tags</i>
<code></code>	image source	Bild einfügen (Angabe der Quelle)	<code></code>
<code><hr></code>	horizontal rule	Horizontale Linie	
<code><html></html></code>	Hypertext markup Language	Dateiformat	
<code></code>	listing	Auflistung	
<code><p></p></code>		Absatz	
<code><pre></pre></code>	preformatted	Vorformatiert	
<code><table></table></code>	Table	Tabelle	
<code><tr></tr></code>	Table Row	Tabellen Reihe	Fügt eine Zeile in die Tabelle
<code><td></td></code>	Table Data	Tabellen Daten	Dort wird der Inhalt eingegeben
			<pre> <table> <tr> <td> Tabellen Inhalt Zelle 1 </td> <td> Tabellen Inhalt Zelle 2 </td> </tr> </table> </pre>
<code><title></title></code>	title	Seitenname	
Attribut	Engl.	mögliche Werte	Beispiel
<code><...align...></code>	ausrichten		
	right	rechts	<code><p align="right"></p></code>

	center	mittig	<code><p align="center"></p></code>
	left	links	<code><p align="left"></p></code>
<code><font...></code>	Schrift		
	face	Art	<code></code>
	size	Größe	
	color	Farbe	
<code></code>	image source	Bild einfügen	<code></code>
	width	Weite	
	height	Höhe	
	border	Rahmen	
	alt	Alternativer Text	<i>Sollte das Bild nicht verfügbar sein, wird dieser Text angezeigt</i>