

## 1. Einführung und Grundgerüst

### 1.1. Was ist HTML

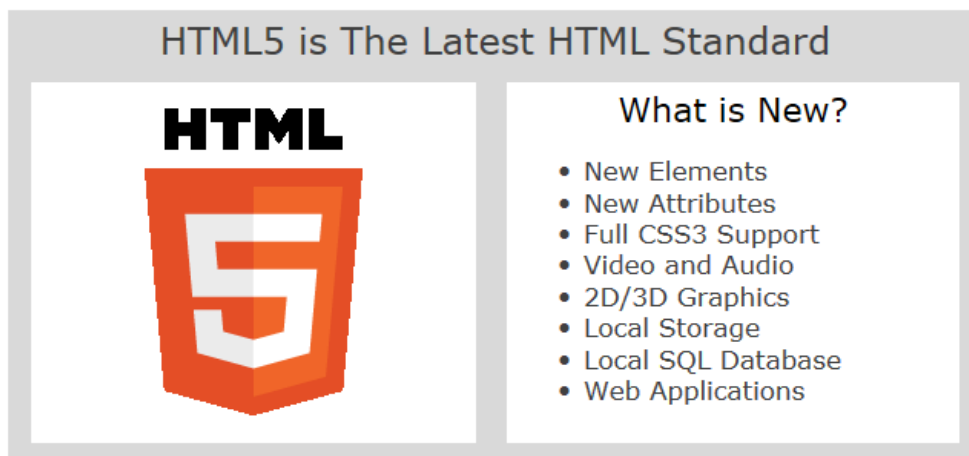
HTML – Hypertext Markup Language, Sprache zur Beschreibung von Inhalten im WWW  
(keine Programmiersprache)

Inhalt wird von Browsern (IE, FireFox, Opera) interpretiert und angezeigt

Entwickelt von: W3C - World Wide Web Consortium, [www.w3.org](http://www.w3.org), [validator.w3.org](http://validator.w3.org)

Aktuelle Version: 5.0, <http://www.w3.org/TR/html5-diff/> beschreibt die Unterschiede zu Version 4.0

eine gute Einführung (allerdings in Englisch) ist unter  
[http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)



### 1.2. TAGS

Befehle nennt man **tag**. Sie stehen in spitzen Klammern. So kann der Browser normalen Text und Befehle unterscheiden.

Tags sind die Markup Befehle von HTML, sie werden von spitzen Klammern eingeschlossen.  
Sie werden paarweise verwendet.

z.B.: Start-Tag: `<html>` ... End-Tag: `</html>`

Einfache Tags können direkt geschlossen werden

z.B.: `<br/>`

Kommentare:

Werden von `<!--` und `-->` eingeschlossen

z.B: `<!-- das ist ein Kommentar -->`

### 1.3. HTML-Grundgerüst

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Erste HTML Seite</title>
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    <p>Hier kommt der Text</p>
  </body>
</html>
```

- Die <!DOCTYPE> Deklaration muss in der ersten Zeile eines HTML5 Dokuments vor dem <html> tag sein.
- Die <!DOCTYPE> Deklaration ist kein HTML tag; sie ist eine Anweisung an den Web Browser mit welcher HTML Version er arbeiten soll.

### 1.4. DOCTYPE in Versionen vor HTML5

Bei ältere HTML-Versionen besteht der Doctype aus Versionsnummer (z.B. 4.01), Variante ( Transitional, Strict, Frameset) und Sprache der Tags (immer EN); spezifiziert von W3C.d

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01">

Es ist keine Variante angegeben, d.h: die Standard Variante 'Transitional' wird genommen. In Variante Transitional sind zum Beispiel die Regeln für die Elementverschachtelung etwas milder. Es ist nach dieser Variante erlaubt, zwischen <body> und </body> "nackten Text" außerhalb eines weiteren Elements zu notieren. Außerdem benötigen Sie diese Variante, wenn Sie in [Links](#) mit dem target-Attribut arbeiten wollen, beispielsweise um [Framesets](#) korrekt anzusteuern.

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">

Strengere Prüfung; Elemente und Attribute aus früheren HTML-Standards welche mittlerweile durch andere Möglichkeiten (vor allem durch [Stylesheets](#) ) ersetzbar sind, dürfen nicht mehr verwendet werden.

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">

Benötigt für Definition von Framesets und Frames

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">

Benötigt für Definition von Framesets und Frames

HTML5 basiert nicht mehr auf SGML (Standard Generalized Markup Language), deshalb wird kein DTD benötigt. SGML ist eine 1986 erstspezifizierte ISO-Norm.

## 1.5. HEADER

Beinhaltet Informationen über Verwaltung des HTML Files z.B. Title, Autor, Metadaten. Wichtige Subtags von Head sind:

- <title>Tag zum Festlegen des Titels z.B.:
- <meta> Metadaten; Steuerung durch HTTP Befehle;  
Anweisungen für Web-Server, Web-Browser und automatische Suchprogramme im Internet ("Robots"), Angaben zum Autor und zum Inhalt der Datei  
z.B.: <meta name="author" content="Max Mustermann">  
<meta name="keywords" content="Schlüsselwort1;Schlüsselwort2;Schlüsselwort3">  
<meta http-equiv="refresh" content="5; URL=http://www.google.at/">  
<base> Tag zum Festlegen der eindeutigen URL
- <link> Tag zum definieren logischer Beziehungen zu anderen Ressourcen (z.B. Stylesheets)^
- <base> definiert die Basis-URL für in der Seite verwendete relative URLs

Beispiel:

```
<head>
  <base href="http://www.spengergasse.at"/>
  
  <title>Textausgabe - Beispiel 1 a</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
```

## 1.6. HTML Body

- Beinhaltet Informationen die am Bildschirm dargestellt werden
- Zusammensetzung aus Tags der jeweiligen HTML Version

Tag: <body> ... </body>

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Hello World!</title>
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    <h1>Erstes Beispiel</h1>
    <p>
      Das ist ein Text für das erste Beispiel.<br>
      Hello World!
    </p>
  </body>
</html>
```

## 1.7. Prüfung des HTML Codes - Validierung

- Prüfung HTML Code auf korrekten Syntax, W3C Spezifikation
- <http://validator.w3.org/>  
Die HTML Datei muss auf die Webseite geladen werden (upload).  
Danach erfolgt Prüfung.
- [http://validator.w3.org/#validate\\_by\\_input](http://validator.w3.org/#validate_by_input)  
HTML Code kann in Fenster eingegeben und geprüft werden.

## 2. Texte

## 2.1. Container - Elemente zur Textstrukturierung

Texte müssen sich in einem Absatz oder einem Container befinden.

## Überschriften Container

# <hn> ... </hn>

n ... Entspricht der Ebene, 1 bis 6 möglich,

Ebene 1 entspricht der größten, Ebene 6 der kleinsten Überschrift

z.B: `<h1>Test&uuml;berschrift</h1>`

## Unter-&uuml;berschrift</h2>

## Abatz Container

<p> ... </p>

Absatz darf nur Text enthalten

## Allgemeine Container

<div> ... </div> Block Element, Container, Ebene

`<span>...</span>` allgemeines Objekt zur Formatierung

## Zeilenumbruch

<br> Zeilenumbruch

<nobr > ... </nobr >Umbruch verhindern

## 2.2. Whitespaces und Sonderzeichen

## Whitespaces

Whitespaces sind Sonderzeichen wie Leerzeichen, Tabs, Zeilenvorschub. Aufeinanderfolgende whitespaces werden zu einem Zeichen (whitespace) zusammengefasst.

z.B.: `<p>Das ist ein Text mit (einem) Leerzeichen.</p>`  
wird mit nur einem Leerzeichen dargestellt.



Um mehrere Leerzeichen darstellen zu können muss anstelle des Leerzeichens das Kürzel &nbsp; verwendet werden.

[illegible]

Das ist ein Text mit (fünf) Leerzeichen .

### weitere Sonderzeichen

Weitere Sonderzeichen wären unter anderem: Umlaute, scharfes S, ... Diese müssen für eine korrekte Darstellung als Unicode oder Kürzel dargestellt werden

#### **Unicode**

Muster: `&#XXX`; XXX entspricht dem Unicode-Wert

Beispiele:     ä ... `&#228`;  
              Ä ... `&#196`;

#### **Kürzel**

Muster: `&XXX`; XXX entspricht dem Kürzel

Beispiele:     ä ... `&auml`;  
              Ä ... `&Auml`;  
              ß ... `&szlig`;  
              Leerzeichen: `&nbsp`;

## **2.3. Übersicht über HTML-Tags**

**Block-Elemente**     Block-Elemente erzeugen einen eigenen Absatz im Textfluss. Je nach Elementtyp kann der Browser dabei automatisch einen größeren Abstand zum vorangehenden Inhalt einfügen. So erzeugen die Browser beispielsweise beim p-Element einen sichtbaren Abstand, damit die Absatzschaltung vom einfachen Zeilenumbruch unterscheidbar ist. Beim div-Element dagegen wird beispielsweise einfach eine neue Zeile begonnen.

Block-Elemente können in der Regel normalen Text und Inline-Elemente enthalten. Einige Block-Elemente können auch andere Block-Elemente enthalten. Ausnahmen von der Regel werden bei den Beschreibungen zu den einzelnen Elementen behandelt.

#### **Folgende Elemente gehören zu den Block-Elementen:**

address | blockquote | center | del | dir | div | dl | fieldset | form | h1-6 | hr | ins | isindex | menu |  
noframes | noscript | ol | p | pre | table | ul

(center, dir, isindex, menu und noframes nur bei HTML Transitional)

**Inline-Elemente**     Inline-Elemente erzeugen keinen eigenen Absatz im Textfluss. Inline-Elemente sind als untergeordnete, "innere" Elemente für Block-Elemente gedacht. Sie können in den meisten Fällen normalen Text und weitere Inline-Elemente enthalten, jedoch keine Block-Elemente. Ausnahmen oder Einschränkungen werden bei den Beschreibungen zu den einzelnen Elementen behandelt.

#### **Folgende Elemente gehören zu den Inline-Elementen:**

a | abbr | acronym | applet | b | basefont | bdo | big | br | button | cite | code | del | dfn | em | font | i |  
img | ins | input | iframe | kbd | label | map | object | q | samp | script | select | small | span | strong |  
sub | sup | textarea | tt | var

(applet, basefont, font, iframe, u, s und strike nur bei HTML Transitional)

#### **Begleitende Dokumente:**

*Zeichensatz.doc*

*Unicode Tabelle.doc*

### 3. Formatierung mit Stylesheets

CSS (Cascading Style Sheet) ist eine Sprache zur Formatierung von HTML-Dokumenten (1. Version 1996). Damit kann man Schriftarten, Farben, Rändern, Linien, Höhen, Breiten, Hintergrundbildern definieren. Weiters werden CSS Container für modernes Layout einer Web-Seite verwendet.

#### 3.1. Arten eine CSS-Formatierung zu plazieren:

##### direkt bei einem HTML Tag

```
<p style="font-weight: bold; font-style: italic; ">
```

##### im Header eines HTML Dokuments

```
<style type="text/css">
    /* Überschrift der 1. Ebene */
    h1 {color:#ffffff; font-weight:bold; font-size:12pt;}
</style>
```

Nur sinnvoll, wenn der Stil für einen einzigen bzw. einige wenige Tag(s) gebraucht wird.

##### als eigenständige CSS Datei

```
<link rel="stylesheet" type="text/css" href="test.css">
```

test.css          eigene Datei nur für Stylesheets

**Bitte IMMER eine eigenständige CSS-Datei anlegen un mit <link rel ... einbinden!**

#### 3.2. Vorteile von CSS (möglichst als eigene Datei)

- die Trennung von Inhalt und Gestaltung vereinfacht eine HTML-Seite, sie wird kleiner und übersichtlicher
- spätere Änderungen werden erleichtert
- mehr Gestaltungsmöglichkeiten: als reines HTML (z.B. Festlegung der Zeilenhöhe, überlappende Positionierung, usw.)
- mit CSS lassen sich die Anforderungen an zugleich ansprechende wie barrierefreie Webseiten realisieren
- CSS-Datei kann für mehrere Webseiten verwendet werden - einheitliches Layout

#### 3.3. Aufbau der Stylesheet-Angabe

Eine Stylesheet-Angabe besteht aus der Eigenschaft und dem Wert, getrennt durch einen Doppelpunkt. Mehrere Style-Definitionen werden durch Semikolon (;) getrennt.

Selektor { Eigenschaft1:Wert1; Eigenschaft2:Wert2; }

Die Eigenschaft ist das, was du formatieren möchtest. z.B.: *color* für die Schriftfarbe, *border-width* für die Rahmenbreite oder *text-decoration* für Unterstreichungen. Als Wert sind Farbangaben, Einheiten oder Schlüsselworte möglich.

### 3.4. verschieden Selektoren

|           |  |
|-----------|--|
| . (class) | <b>Klassen-Selektor</b><br>unabhängig von TAG  |
| tag.class | <b>TAG - Klassen-Selektor</b><br>gilt nur für den TAG  |
| # (ID)    | <b>ID-Selektor</b><br>jede ID nur einmal in Dokument - eindeutige Identifizierung<br>dient auch als Zugriff von Skriptsprachen |
| tag:XXX   | <b>Link-Pseudoklassen</b><br>steuern bestimmte Zustände eines Links  |

### 3.5. Schriftfarbe

Der Farbwert wird im RGB-Farbraum angegeben:

color=#rrggbb bedeutet:

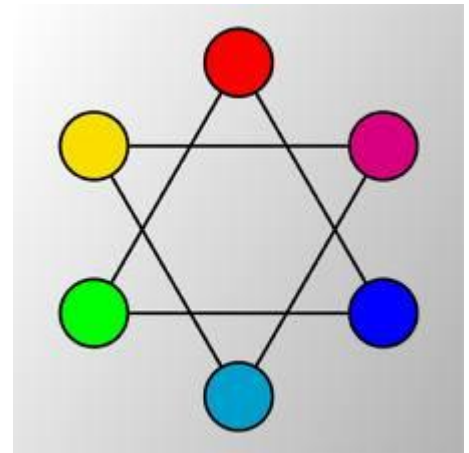
drei 2-stellige hexadezimale Zahlen 0-ffh, 0\_255d

rr     Rot-Anteil

gg     Grün-Anteil

bb     Blau-Anteil

|       |         |         |
|-------|---------|---------|
| z.B.: | #ff0000 | rot     |
|       | #000000 | schwarz |
|       | #ff00ff | magenta |



### 3.6. Beispiele

Text-Formatierung:

```
{    color: #0000ff;  
    font-style: italic;  
    font-weight: bold;  
    font-size:20px;  
    font-family:script,  
    text-align:center    }
```

---



## 4. Links

Links dienen zum Navigieren im www, sie ermöglichen komplexe "verlinkte", d.h. miteinander verknüpfte Webseiten. Es gibt vier verschiedene Arten von Links:

- Links mit relativer Adressierung (interne Links auf eigenem Server)
- Links mit absoluter Adressierung (eindeutiger www-Adresse auf externem Server)
- Textmarken
- Mail-Links

### 4.1. Aufbau des Link-Tags

<a href="XXX">YYY</a>

XXX: Entspricht der Adresse zu der gesprungen wird

YYY: Entspricht einem Text welcher bei dem Link erscheint

Attribut: target

Gibt an in welchem Fenster der Link geöffnet wird.

\_self            das neue Dokument wird im selben (aktuellen) Fenster geöffnet,  
                  default Wert

\_blank          das neue Dokument wird in neuem Fenster geöffnet

### 4.2. Anmerkungen

- Textmarken einer anderen Website können ebenfalls per Link angesprochen werden  
z.B.: <a href="http://www.seite.at/news.at#sport">
- **verschiedene Arten der relativen Adressierung**

|                    |   |
|--------------------|---|
| news.html          | Datei "news.html" ist im selben Verzeichnis   |
| ../sport/news.html | Datei "news.html" im Verzeichnis sport,<br>dieses ist ein Unterverzeichnis des aktuellen Verzeichnis  |
| ../sport/news.html | Datei "news.html" im Verzeichnis sport,<br>dieses ist ein Nachbar-Verzeichnis des aktuellen Verzeichnis<br>(eine Ebene höher, dort ins Verzeichnis sport) |
- **Attribute von mailto**

|                      |                 |
|----------------------|-----------------|
| &#38; subject= "XXX" | subject in Mail |
| &#38; body= "YYY"    | body in Mail    |

siehe *html5-Link1.html*  
*html5-Link2.html*

## 5. Bilder in HTML

``

Weitere Attribute siehe Beschreibung Stylesheets.

### Bilder als Link

```
<a href="http://www.graz.at">
  
  Grazer Uhrturn</a>
```

### Bilder als Hintergrund

```
<body background="hintergrund.jpg">
```

### Weitere Attribute von img sind (siehe css-Formatierung):

- width    Breite des Bildes  
z.B.: width= "300"    default: px- Pixel (width="300px")
- height    Höhe des Bildes  
z.B.: height= "300"
- padding – Innenabstand vom Bild  
default: px
- margin – Aussenabstand vom Bild  
default: px
- border-width= 0    kein Rahmen
- border-color - Rahmenfarbe

## 6. Tabellen

Tabellen bestehen aus Zellen, die in Zeilen und Spalten aufgeteilt sind. Sie werden zur Strukturierung der Webseite verwendet

- Tabelle wird mit `<table>` definiert
- Zeilen werden mit `<tr>` definiert
- Eine Zelle wird mit `<td>` definiert

### ■ Beispiel

```
<table>
  <tr>
    <td>ID</td> <td>Wert</td>
  </tr>
  <tr>
    <td>1</td> <td>ASDF</td>
  </tr>
</table>
...
```



| ID | Wert |
|----|------|
| 1  | ASDF |

## 7. weitere HTML-Tags

### 7.1. Link-Pseudoklassen

|           |   |
|-----------|---|
| a:hover   | Mauszeiger über Link                    |
| a:link    | Link im Normalzustand                   |
| a:visited | Link, wenn dieser bereits besucht wurde |
| a:focus   | Style, wenn Fokus auf dem Link ist      |
| a:active  | Gerade angeklickter Link                |

#### Schachtelungen

Tag1 Tag2 Tag3 {Attribut :Wert ; Attribut :Wert ; ...}

Tag 3 muss sich in Tag 2 befinden

Tag 2 muss sich in Tag 1 befinden

z.B.: a b

```
{color: #ff00ff}
```

#### Style für ganze Seite, Tabelle

Style für <body >

body , td

```
{ font-family: Arial;  
font-size: 14px;  
font-weight: bold; }
```

#### Container

<span> ohne Zeilenumbruch

<div> mit Zeilenumbruch

Container dienen zur Aufnahme von Formatierungen

## 7.2. Listen

```
<h1>
<ul>
  <li>Montag </li>
  <li>Dienstag</li>
  <li>Mittwoch </li>
</ul>
```

```
<h1>
<ul>
  <li>Montag </li>
  <li>Dienstag</li>
  <li>Mittwoch </li>
</ul>
```

```
<h1> Verschachtelte Listen</h1>
<ul>
  <li>Suchmaschinen
    <ul>
      <li>Google</li>
      <li>AltaVista</li>
      <li>Fireball</li>
    </ul>
  </li>
  <li>Verzeichnisse
    <ul>
      <li>Yahoo</li>
      <li>Web.de</li>
      <li>Dino-Online</li>
    </ul>
  </li>
  <li>Was anderes</li>
  <li>Noch was anderes</li>
</ul>
```

```
<h1> Nummerierte Liste definieren</h1>
<ol>
  <li>bei Anette vorbeischauen</li>
  <li>bei Bianca vorbeischauen</li>
  <li>bei Christine vorbeischauen</li>
</ol>
```

### Liste

- Montag
- Dienstag
- Mittwoch

### Verschachtelte Listen

- Suchmaschinen
  - Google
  - AltaVista
  - Fireball
- Verzeichnisse
  - Yahoo
  - Web.de
  - Dino-Online
- Was anderes
- Noch was anderes

### Nummerierte Liste definieren

1. bei Anette vorbeischauen
2. bei Bianca vorbeischauen
3. bei Christine vorbeischauen

## 8. neue Elemente in HTML 5

**<time>** Auszeichnung von Zeitangaben wie Datum und Uhrzeit

**<summary>** die Zusammenfassung für ein *details*-Element

**<section>** Abschnitt/Bereich innerhalb eines Dokuments

**<rp>, <rt>, <ruby>** Annotationen für Ruby

**<progress>** Darstellung eines Fortschrittsbalkens mit verschiedenen Einstellungsmöglichkeiten

**<output>** Ergebnis einer Berechnung

**<nav>** Auszeichnung von Navigationen einer Webseite

**<meter>** Auszeichnung von Maßangaben in einem definiertem Bereich; z.B. *3/4 Prozent* oder *Notendurchschnitt 1,8*

**<menu>** Wiederauferstehung dieses Tags in HTML5: für auklappbare Menüleisten

**<mark>** Hervorhebung von Text

**<keygen>** Generator zur automatischen Erstellung eines Schlüsselpaars

**<hgroup>** mehrere Überschriften lassen sich zusammenfassen

**<header>** Kopfbereich eines Abschnitts oder eines kompletten Dokuments

**<footer>** Fußbereich eines Abschnitts oder eines kompletten Dokuments

**<figcaption>** *figure*-Element erhält eine Legende

**<figure>** eingebundene Bilder und Videos erhalten eine Beschreibung

**<embed>** Plugin-Inhalt

**<details>** Steuerungselemente oder zusätzliche Informationen (im Gegensatz zu *<aside>* ein- und ausblendbar)

**<datalist>** eine Liste von Optionen für andere Input-Elemente

**<command>** Nutzerbefehle werden nun über diese Feld bestätigt; kennzeichnet somit Buttons die mit JavaScript hinterlegt werden

**<canvas>** Bitmap-Malfläche durch programmierbaren Code steuerbar

**<source>** Inhalt

**<audio>** Audio-Inhalt

**<video>** Video-Inhalt

**<aside>** Inhalt der den Hauptinhalt der Seite ergänzt, z.B. Begriffserklärung

**<article>** Inhalt der in sich geschlossen ist, z.B. Kommentar im Blog oder Artikel einer Zeitschrift

## 9. Spezielle CSS-Formatierungen

### 9.1. Boxmodell

position:static

Default-Wert, Element wird an der aktuellen Position gesetzt.

position:relative

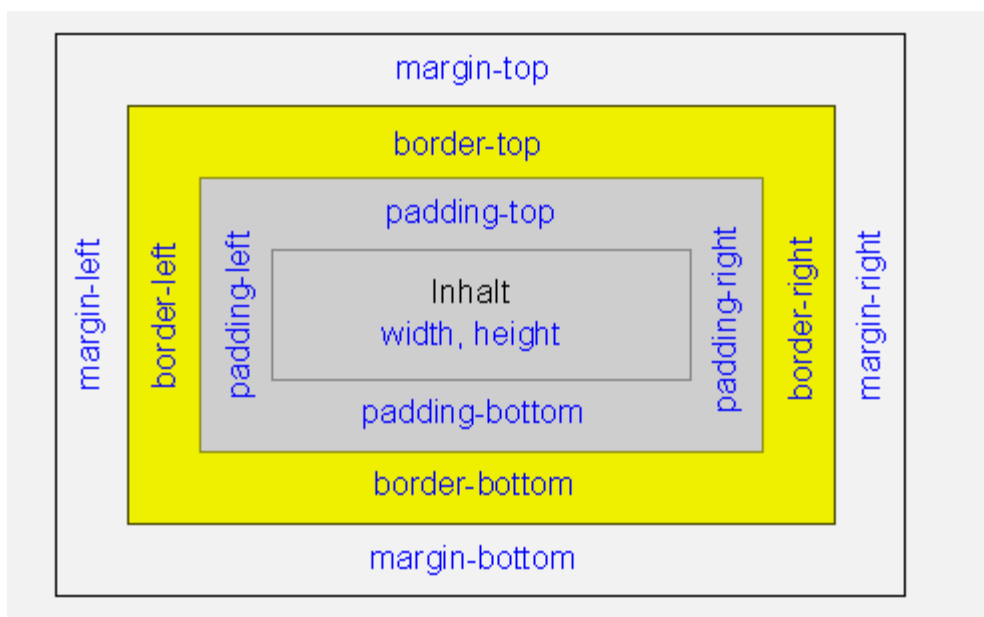
Top, bottom, left, right sind relativ zur derzeit aktuellen Position.

position:absolute

Top, bottom, left, right beziehen sich auf die linke obere Bildschirm-Ecke.

Bereiche des Box-Modells

margin: Aussenabstand (margin, oder margin-left, margin-right, margin-top, margin-bottom)  
border: Umrandung  
padding: Innenabstand  
Inhalt: width,height



**ACHTUNG:** Unterscheidung altes Box-Modell (box-sizing:content-box) und alternatives Modell von CSS 3 (box-sizing:border-box):

**box-sizing:content-box**      Angabe für width, height bezieht sich auf den Bereich OHNE margin, border, padding

**box-sizing:border-box**      width, height ist der Bereich MIT margin, border, padding

## 9.2. display

Legt fest, wie ein Element angezeigt werden soll

|                       |   |
|-----------------------|---|
| <u>display:block</u>  | erzwingt einen Block - das Element erzeugt eine neue Zeile. |
| <u>display:inline</u> | das Element wird im laufenden Textfluss angezeigt.          |
| <u>display:none</u>   | Element wird nicht angezeigt                                |

## 9.3. float

Über das Float-Element kann eine Ausrichtung von Elementen geschehen.

Float (engl. = fließen, Hin- und Herbewegung) bestimmt wie ein Blockelement reagieren soll - im Klartext, neben dem Blockelement dürfen sich weitere Elementen befinden, die ansonsten unterhalb platziert würden

```
#navi {  
    float:left;  
    background-color:thistle;  
    width: 150px;  
}  
  
#inhalte {  
    background-color:lightblue;  
    margin-left: 200px;  
    width: 300px;  
    height: 100px;  
}
```

Startseite  
Impressum

Hier **HTML** und CSS lernen und die eigene  
Website erstellen - selbst ist der Mensch

```
<div id="navi">  
<ul>    <li>Startseite</li>  
        <li>Impressum</li>  
</ul>  
</div>  
  
<div id="inhalte">  
    Hier HTML und CSS lernen und die  
    eigene Website erstellen - selbst ist der Mensch  
</div>
```

Jedes Element kann "floaten": Absätze, <div>s, Listen, Tabellen, Grafiken und selbst Inline-Elemente wie <span> oder <strong>.

Mit float: können Sie bestimmen, dass nachfolgende Elemente das aktuelle Element bzw. den aktuellen Bereich umfließen. Folgende Angaben sind möglich:

|                    |  |
|--------------------|--|
| <u>float:left</u>  | Element steht links und wird rechts davon von nachfolgenden Elementen umflossen. |
| <u>float:right</u> | Element steht rechts und wird links davon von nachfolgenden Elementen umflossen. |
| <u>float:none</u>  | kein Umfluss (Normaleinstellung).  |



## 9.4. clear

Die Angabe clear beendet das Umfließen andere Elemente. Ein mit clear formatiertes Element ist das Erste, dass nicht mehr neben anderen Elementen steht. Es kann aber trotzdem mit der float-Eigenschaft formatiert werden, damit nachfolgende Elemente um dieses Element wieder herum fließen.

|                    |                                      |
|--------------------|--------------------------------------|
| <u>clear:left</u>  | beendet einen Umfluß mit float:left  |
| <u>clear:right</u> | beendet einen Umfluß mit float:right |
| <u>clear:both</u>  | beendet den Umfluß um beide Seiten   |

## 9.5. position

|                          |   |
|--------------------------|---|
| <u>position:static</u>   | keine spezielle Positionierung, normaler Elementfluss (Normaleinstellung)   |
| <u>position:relative</u> | relative Positionierung (Verschiebung), gemessen an der Normalposition oder Anfangsposition des Elements selbst   |
| <u>position:absolute</u> | absolute Positionierung, gemessen am Rand des nächsthöheren Vorfahrenelements, das nicht die Normaleinstellung (position:static) aufweist, scrollt mit. |
| <u>position:fixed</u>    | absolute Positionierung, gemessen am "Viewport", d.h. am Browserfenster, bleibt beim Scrollen stehen.   |

## 9.6. z-index

z.B.: z-Index:1

Ein Layer mit der höheren Zahl liegt über dem Layer mit der niedrigeren Zahl. Ist kein z\_index festgelegt erfolgt die Darstellung nach der Reihenfolge im HTML-File.

## 9.7. Kommentare in der CSS-Datei

Wie beim Programmieren /\* beliebiger Kommentar \*/. ACHTUNG: // ist nicht möglich.  
Gute Referenzen im Internet: z.B. <http://www.css4you.de/>

## 10. Neuerungen in CSS3

### 10.1.1. Mehrspaltige Layouts

#### box-sizing – ein alternatives Box-Modell

Nach Aktivierung durch **box-sizing: border-box** werden die Angaben für width als Gesamtbreite interpretiert werden. Innenabstände und Rahmen werden von dieser Gesamtbreite subtrahiert. Damit kann die Gesamtbreite direkt vorgegeben werden und die effektiv verbleibende Nutzbreite im Innenraum der Box wird vom Browser dynamisch festgelegt.

Zwischen dem alten Box-Modell von CSS 1 (**box-sizing: content-box**) und dem alternativen Modell von CSS 3 (**box-sizing: border-box**) kann beliebig umgeschaltet werden.

#### calc() – Rechnen innerhalb von CSS

Für Größenangaben kann auch eine Rechenvorschrift übergeben werden, damit können Boxen Boxen mit flexibler Breite gestaltet werden.

```
#left {
  float:left;
  padding: 0 1em;
  border-right: 1px #888 dotted
  width: calc( 30% - 2*1em - 1px );
}

#right {
  float:right;
  padding: 0 1em;
  width: calc( 70% - 2*1em );
}
```

#### Column – Spaltensatz mit CSS

```
#3col_container {
  column-count: 3;
  column-gap: 20px;
}
```

**column-count:** legt die Anzahl der Spalten innerhalb eines Containers fest  
**column-gap:** regelt den horizontalen Abstand zwischen den einzelnen Spalten

Auf diese Weise wird die effektive Breite der Spalte durch den Browser ermittelt. Der Designer gibt lediglich die Anzahl der gewünschten Spalten vor. In erster Linie eignet sich diese Eigenschaft zur Aufteilung von Texten in mehrere nebeneinander liegende Spalten.

Mit den [CSS Tables](#) ist es möglich, die [Vorteile von Tabellenlayout mit der Flexibilität von CSS](#) zu kombinieren.

### 10.1.2. Animationen

Bisher waren Animationen das unangefochtene Territorium von Javascript. Das könnte sich mit CSS 3 ändern: Hier ist es möglich, Änderungen zwischen verschiedenen CSS-Einstellungen nicht abrupt sondern mit einer Animation auszuführen.

**Ein paar Beispiele:** (bisher nur in Webkit-Browsern: Safari und Chrome)

<http://www.the-art-of-web.com/css/css-animation/>

<http://24ways.org/2009/css-animations>

<http://www.1stwebdesigner.com/development/50-awesome-css3-animations/>

<http://webkit.org/blog/324/css-animation-2/>

Es bleibt zu hoffen, dass die Autoren es mit den CSS Transitions nicht übertreiben — die verhassten `<blink>` und `<marquee>` Tags sind nichts gegen missbrauchtes CSS3!

### 10.1.3. Verwaltung von Schriftarten

Mit der Einführung von `@font-face` ist es möglich, in Webseiten Schriftarten zu benutzen, die nicht auf dem Rechner des Betrachters installiert sind. Die Schriftart sollte im "TrueType"-Format auf den Server hochgeladen werden.

Kostenloses Angebot von Google zur Einbindung von Fonts

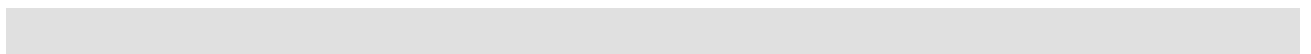
<http://code.google.com/webfonts>

Nähere Infos, wie das `@font-face`-Modul verwendet werden kann

<http://tobias-otte.de/essays/web-fonts/>

### 10.1.4. Weitere Funktionen

- Schlagschatten: `box-shadow`
- Farbverläufe: `linear-gradient`
- Halbtransparente Hintergrund- und Rahmenfarben
- Rotation



## 11. CSS-Menü

```
<!DOCTYPE HTML>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>div Beispiel</title>
  <link href="6-MenuLI.css" rel="stylesheet" type="text/css">
</head>

<body>

<!-- NAVI -->
<div id="ENavi">
  <ul><li id="id1"><span>Programme</span>
    <ul>
      <li class="subM"><a href="pageA1.html">Winword</a></li>
      <li class="subM"><a href="pageA2.html">Excel</a></li>
      <li class="subM"><a href="pageA3.html">Powerpoint</a></li>
    </ul></li>
    <ul><li id="id2"><span>Wetter</span>
      <ul>
        <li class="subM"><a href="pageA1.html">XXXXX</a></li>
      </ul></li>
      <ul><li id="id3"><a href="stephansdom.jpg"><span>Sport</span></a></li>
    </ul>
  </div>

<!-- Block in welchen Text, usw. kommt
      wenn er vor dem Menü steht überdeckt das aufgeklappte Menü den Text -->
<div id="iblock">
TEXT!!!!!!!!!!!!!!!!!!!!
</div>

</body>
</html>
```

| Programme        | Wetter | Sport |
|------------------|--------|-------|
| TEXT!!!!!!!!!!!! | XXXXX  |       |

```
/*
    Vorlage für Menü mit PopUp Untermenü
    ZENTRIERT
    DI. Susanne Kalliany
*/
body {
    font-family:Arial,Helvetica,sans-serif;
}

/* Farbe für den Link ohne visited, aktiv, ... */
a:link { color:#ff0000; text-decoration:bold; font-family:Arial,sans-serif; }

/* Farbe für den Link wenn er angeklickt wurde = selbe wie a:link
    sonst ändert sich die Frabel nach dem Klicken */
a:visited { color:#ff0000; text-decoration:none; font-family:Arial,sans-serif; }

/* Farbe wenn man mit der Maus "drüberfährt" */
a:hover { color:#0000ff; }

/* ENavi: Menü
*****/
#ENavi {
    position: absolute;
    top: 10px;
    width: 100%;
    height:50px;
    text-align:center;
}

#ENavi ul {
    list-style: none;
    margin: 0px;
    padding: 0px;
    text-align:center;
}

#id1 {
    position:absolute;
    width:33%;
}
#id2 {
    position:absolute;
    width:33%;
    left:33%;
}
#id3 {
    position:absolute;
    width:33%;
    left:66%;
}

/* Menüleiste */
#ENavi span {
    color: #ffffff;
    background: #006400;
    padding-top:4px;
    padding-bottom:4px;

    border-width: 1px;
    border-style: solid;
    border-color: #ccc #888 #555 #bbb;
    margin: 0;
    cursor: pointer;
    display: block;
}

/* pull-Down Menü */
#ENavi .subM {
    font-weight:normal;
    text-align:center;
    font-size: 90%;
}
```

```
padding-top:4px;
padding-bottom:4px;
height:20px;
width:100%;

border-width: 1px;
border-style: solid;
border-color: #ccc #888 #555 #bbb;
}

#ENavi li {
    height:28px;
    width:25%;
    z-index:1;
}

/* Breite einzelner Menüpunkt */
/* damit Menü den nachfolgenden Text überdeckt */

/* hover
----- */
#ENavi div:hover {
    height:28px;
    cursor: pointer;
}

#ENavi span:hover {
    color: #ffff00;
}

/* sichtbar/unsichtbar
----- */
#ENavi ul ul,
#ENavi ul li:hover ul ul,
#ENavi ul ul li:hover ul ul
{
    display: none;
}

#ENavi ul li:hover ul,
#ENavi ul ul li:hover ul,
#ENavi ul ul ul li:hover ul
{
    display: block;
    background: #cccccc;
}

/* Hintergrund für Menü */

#ENavi ul ul ul {
    position:relative;
    right:100%;
    top:-28px;
    margin:0px;
}

/* Position von Block unter Menü
----- */
#iblock {
    position: absolute;
    top: 100px;
    background: #888888;
    z-index:0;
    width:100%;
}

/* Abstand vom oberen Fensterrand */
/* damit Menü den nachfolgenden Text überdeckt */
```

## 12. Video in HTML 5 einbinden

```
<video controls width=640 height=426>
<source src="musikvideo.ogv" type='video/ogg; codecs="theora, vorbis"' />
<source src="musikvideo.webm" type='video/webm' >
<source src="musikvideo.mp4" type='video/mp4'>
<p>Video is not visible, most likely your browser does not support HTML5
video</p>
</video>
```

**width:** die angezeigt Breite der Videos  
**height:** die angezeigt Höhe des Videos  
**poster:** Bild angezeigt, wenn das Video noch nicht gestartet wurde.  
(Das Attribut poster wird von nicht allen Browsern unterstützt.)  
**autoplay:** das Video wird sofort gestartet, wenn ein Besucher die HTML-Seite aufruft  
**controls:** Steuerelemente zum Stoppen, Starten ...  
**autobuffer:** Zwischenspeicherung aktivieren, Video soll möglichst ruckelfrei wiedergegeben werden

Es können mehrere Video-Formate angegeben werden; der Browser verwendet das erste von ihm unterstützte Format. Wenn width und height angegeben werden Wenn Höhe und Breite festgelegt sind, wird beim Laden der Seite der Raum für das Video reserviert, das Seitenlayout wird während des Ladens nicht geändert.

---

### 12.1. Videoformate und Browser-Unterstützung

Derzeit gibt es 3 unterstützten Video-Formate für die <video> Element: MP4, WebM und Ogg:

| <b>Browser</b>      | <b>MP4</b> | <b>WebM</b> | <b>Ogg</b> |
|---------------------|------------|-------------|------------|
| Internet Explorer 9 | JA         | NO          | NO         |
| Firefox 4.0         | NO         | JA          | JA         |
| Google Chrome 6     | JA         | JA          | JA         |
| Apple Safari 5      | JA         | NO          | NO         |
| Opera 10,6          | NO         | JA          | JA         |

- MP4 = MPEG 4-Dateien mit **H264-Codec** und AAC-Audio-Codec
- WebM = WebM-Dateien mit VP8 Video-Codec und Vorbis-Audio-Codec
- Ogg = Ogg-Dateien mit Theora-Video-Codec und Vorbis-Audio-Codec

## 13. Zeichensatz, HTML Parser

Unter einem **Zeichensatz** versteht man einen Vorrat an Elementen ([Zeichen](#)) zur Darstellung von Sachverhalten. Solche Elemente können unter anderem die [Buchstaben](#) eines [Alphabetes](#), [Ziffern](#), aber auch andere [Symbole](#) sein, etwa die [Sonderzeichen](#), die Zeichen der Lautumschrift des [IPA](#)-Codes oder der [Brailleschrift](#), [Piktogramme](#) verschiedenster Art oder [Steuerzeichen](#) (unsichtbare Zeichen). Speziell in der [Informationstechnik](#) versteht man unter einem Zeichensatz die Gesamtheit der Zeichen einer bestimmten [Zeichenkodierung](#) und deren Zuordnung.

| Name                    | Einführung | Bit | Zeichen   | darstellbare Zeichen          | Normen                    | erster Einsatz                  |
|-------------------------|------------|-----|-----------|-------------------------------|---------------------------|---------------------------------|
| <a href="#">ASCII</a>   | 1963       | 7   | 128       | 95                            | ANSI X3.4-1968            | Fernschreiber Teletype ASR33    |
| <a href="#">EBCDIC</a>  | 1964       | 8   | 256       | 93 bis 192                    |                           | <a href="#">IBM-Großrechner</a> |
| <a href="#">Unicode</a> | 1991       | 21  | 1.114.112 | derzeit 100.000 (Unicode 5.1) | <a href="#">ISO 10646</a> | Xerox, Apple                    |

Die Zahl der darstellbaren Zeichen ist geringer als die Zahl der definierbaren Zeichen, da ein Teil des Zeichenvorrates als nicht-druckbare Steuerzeichen verwendet wird.

### 13.1. 7-Bit Zeichensatz (ASCII-Zeichensatz)

Der ASCII-Zeichensatz ist der "Urvater" aller anderen Zeichensätze und enthält  $2^7 = 128$  Zeichen. Das sind die lateinischen Grundbuchstaben, Zahlen und einige sonstige Zeichen, aber keine durch Diakritika (Umlaut, Zeichen mit Akzent,..) oder sonstige Zeichen erweiterten Buchstaben, also auch keine deutschen Umlaute. Mit ihm lässt sich also praktisch nur Englisch schreiben, andere Schriftsysteme müssen transliteriert werden. Dafür ist er der universellste aller Zeichensätze, da er praktisch von jedem Computersystem und jedem Programm verstanden wird, deshalb wird er vor allem im Internet bis heute verwendet. Er ist sozusagen der kleinste gemeinsame Nenner aller anderen Zeichensätze, d. h., die in ihm enthaltenen Zeichen sind in (fast) allen anderen Zeichensätzen identisch angeordnet.



## 13.2. Erweiterter ASCII Zeichensatz (8-bit)

([Übersicht über die verschiedenen osteuropäischen Zeichensätze und Links zu Schriften hier.](#))

Diese Zeichensätze enthalten  $2 \text{ hoch } 8 = 256$  Zeichen. Sie stellen Erweiterungen des [ASCII-Zeichensatzes](#) dar: in der ersten Hälfte bis zum 128. Zeichen sind sie mit ihm identisch, und in der zweiten Hälfte enthalten sie je nach Sprache und Computersystem verschiedene weitere Zeichen. So enthalten Zeichensätze für westeuropäische Sprachen z. B. die deutschen Umlaute und französischen akzentuierten Vokale, osteuropäische Zeichensätze enthalten für Tschechisch, Polnisch, Ungarisch notwendige Sonderzeichen, kyrillische Zeichensätze die kyrillischen Buchstaben. Daher rührt auch die Erscheinung, die wohl jeder, der z. B. eine russische Webpage ansehen wollte, schon mal gesehen hat, kennt. Ein russischer Text sieht plötzlich so aus: "Ååååîëÿ, äîáðûé îîé ïöÿðääü...". In diesem Falle ist nun eine Schrift mit westeuropäischem Zeichensatz für Text mit dem Zeichensatz Windows-Cyrillisch ausgewählt worden; die Verwendung einer entsprechenden Schrift schafft Abhilfe.

## 13.3. UNICODE

Um dem angeführten "Zeichensalat" der verschiedenen [8-bit-Zeichensätze](#) ein Ende zu schaffen, und da besonders in fernöstlichen Sprachen mehr als 256 Zeichen gebraucht werden, ist der 2-byte-Zeichensatz UNICODE geschaffen worden, der  $2 \text{ hoch } 16 = 65536$  Zeichen enthalten kann und damit für jeden Buchstaben in jeder Sprache der Welt eine eindeutige Codenummer bereithält. Über die Zuordnung zu den Codenummern entscheidet das [UNICODE-Konsortium](#), das den UNICODE-Standard veröffentlicht. Es werden immer noch neue Sprachen und Zeichen hinzugefügt. Man informiere sich auf den [UNICODE-Webseiten](#) über Details.

**Schriften (Fonts) für UNICODE:** Natürlich wird es keine Schrift geben, die alle 65536 Zeichen enthält, sondern jede UNICODE-kompatible Schrift wird immer nur einen sinnvollen Teilbereich von Zeichen enthalten.

## 13.4. UTF-8

UTF-8 (Abk. für 8-bit [UCS Transformation Format](#)) ist die am weitesten verbreitete [Kodierung](#) für [Unicode](#)-Zeichen.

Bei der UTF-8 Kodierung wird jedem Unicode-Zeichen eine speziell kodierte [Bytekette](#) variabler Länge zugeordnet. UTF-8 unterstützt bis zu vier Byte, auf die sich wie bei allen [UTF](#)-Formaten alle Unicode-Zeichen abbilden lassen. Auch bei dem in [Webbrowsern](#) verwendeten [HTML](#) setzt sich UTF-8 zur Darstellung sprachspezifischer Zeichen zunehmend durch und ersetzt die vorher benutzten [HTML-Entities](#)<sup>3</sup>

Mit UTF-8 darf auf die Codierung von Umlauten, Sonderzeichen und kaufmännischen Zeichen im HTML-Dokument verzichtet werden: „ä, Ä, ö, Ö, ü, Ü, ß, ©, è, à“. Deutsch und Französisch, Spanisch und Schwedisch (und mehr ...) können in einem Dokument geschrieben werden.

**Einstellung in den Meta-Tags:**

```
<meta charset="utf-8" />
```

## 13.5. Weitere 8-Bit Zeichensätze

ISO 8859-1 (Latin1): möglichst viele Sonderzeichen westeuropäischer Sprachen ab.

ISO/IEC 8859-15: zusätzlich Eurosymbol und einige französische Zeichen

Windows-1252 (Western European): Microsoft-Betriebssystem Windows

Bei der Eingabe von Text sendet jede Taste eine bestimmte Codenummer an den Computer. Die Zuordnung der Codenummern zu den Tasten erfolgt durch eine "**Tastaturtreiber**" genannte Datei. Um den Text am Bildschirm darzustellen oder auf dem Drucker auszugeben, muss nun die **Schriftart** (engl. **Font**) für jede Codenummer das richtige Zeichen darstellen. Daraus ergibt sich, dass es für verschiedene Zeichensätze verschiedene Schriften (Fonts) gibt, und auch der Tastaturtreiber dem verwendeten Zeichensatz entsprechen muss.

## 13.6. HTML entities

Auszeichnungssprachen (*Markup Languages*) verwenden Entitäten (engl. *entity* bzw. *entities*, deutsch *Einheiten*, *Gebilde*), um wiederkehrende Informationseinheiten zu standardisieren.

Eine HTML Entity beginnt stets mit & und endet mit einem Semikolon (;). Dazwischen befindet sich eine Codierung mit maximal 6 Buchstaben. Für die Umlaute ä, ü, ö und ß sind die Entities **&auml;**, **&uuml;**, **&ouml;** und **&szlig;**.

## 13.7. Unicode für HTML

`&# [x] [Nummer] ;`

Beispiele einer Notation in HTML:

`&#945;`

`&#x3B1;`

Das `x` notieren Sie dann, wenn Sie die Nummer hexadezimal angeben. Bei dezimaler Angabe lassen Sie das `x` weg. Die Unicode-Nummer eines gewünschten Zeichens können Sie aus den Codetabellen ermitteln (siehe Abschnitt über [das Unicode-System](#)).

### 13.8. Darstellung eines Textes bei Verwendung unterschiedlicher Zeichensätze in der HTML-Datei.

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
```

Hier kommt der Text ÄÖÜ &&& %%% \$\$\$ §§§

Output bei Verwendung einer „falschen“ Codepage.

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-5">
```

---

Hier kommt der Text ФЖМ &&& %%% \$\$\$ III

**ACHTUNG:** Bei der Erstellung einer Webseite sollte im verwendeten Editor dieselbe Codierung wie in den Meta-Tag eingestellt sein!

Meist wird UTF-8 verwendet; dh.

```
<meta charset="utf-8" />
```

### 13.9. HTML-Parser

Unter einem HTML-Parser versteht man eine Software, die HTML-Auszeichnungen erkennt und in strukturierten Text umsetzt. Jeder Web-Browser verfügt über einen HTML-Parser, um überhaupt mit HTML klarzukommen.

Solche HTML-Parser werden nun leider auf den meisten Web-Seiten mit Syntaxfehlern in der Textauszeichnung konfrontiert. Oft sind es kleinere, nicht allzu tragische Fehler, doch es gibt auch viele Web-Seiten, deren HTML-Quelltext nur das Prädikat "ungenügend" verdient, weil darin übelste Verunstaltungen der HTML-Regeln vorkommen. Strenge Parser, die genau auf Einhaltung der HTML-Regeln prüfen, müssten die Anzeige solcher Web-Seiten eigentlich abbrechen, und anstelle der Seite würden die Browser dann nur eine lapidare Fehlermeldung anzeigen. Da ein solcher Browser am breiten Markt jedoch keine Chance hätte, weil er kaum eine bekannte Web-Seite anzeigen würde, sind die HTML-Parser der heute verbreiteten Browser ziemlich gutmütige Wesen, die so ziemlich alles fressen, was ihnen vorgesetzt wird und irgendetwas daraus machen, meistens sogar durchaus das, was der Autor der Web-Seite erreichen wollte.

Am weitesten in dieser Kunst hat es der HTML-Parser des Internet Explorers gebracht. Das hat dem Internet Explorer einerseits den Ruf beschert, "am besten" HTML zu beherrschen, aber Fachleute rümpfen aus dem gleichen Grund gerne die Nase über diesen Browser mit dem Argument, dass er durch sein Verhalten das schlampige und fehlerhafte Kodieren von HTML nur fördere.