

### 13 Tabellen

eröffnen verschiedene Möglichkeiten in der Gestaltung einer Webseite. Die Ausrichtung von Text und Grafiken innerhalb einer Tabelle erlauben ebenso wie die Wahl verschiedener Hintergrundfarben einzelner Zellen vielfältige Designvarianten und Strukturierungen eines Dokuments.

Das folgende Beispiel zeigt eine einfache Tabelle; die Darstellung vereinigt die zeilenweise Programmierung der Tabelle mittels HTML-Code mit der Struktur im Raster (2,2) in einem gemeinsamen Schema.

```
<table>
<tr>
<td>
Akademische Titel können er-
dient werden, erdient oder
erdiniert.
</td>
<td>
Manche Studenten schaffen es, ihr Wissen
noch unter dem eigenen Niveau anzusie-
deln.
</td>
</tr>
<tr>
<td>
Experte ist einer, der nachher
alles vorher gewusst hat.
</td>
<td>
Bildung ist die Fähigkeit, falsche Schluss-
folgerungen ziehen zu können.
</td>
</tr>
</table>
```

<i>Akademische Titel können erdient werden, erdient oder erdiniert.</i>	<i>Manche Studenten schaffen es, ihr Wissen noch unter dem eigenen Niveau anzusiedeln.</i>
<i>Experte ist einer, der nachher alles vorher gewusst hat.</i>	<i>Bildung ist die Fähigkeit, falsche Schlussfolgerungen ziehen zu können.</i>

1. Die Markierungen `<table>...</table>` definieren eine Tabelle;
2. Tags `<caption>...</caption>` kennzeichnen *optional* eine Tabellenüberschrift.
3. Tabellen sind zeilenweise aufgebaut.  
Jede Zeile wird mit `<tr>` (table row) geöffnet und mit `</tr>` abgeschlossen.
4. Eine Zeile beinhaltet mehrere Zellen (Spalten).  
Jede Kombination `<td>...</td>` (table definition) generiert eine Zelle innerhalb der Zeile; **zwischen diese Tags sind alle Inhalte (Texte, Bilder,...) der aktuellen Zelle einzutragen.**  
Eine Zelle mit spezifischer Formatierung (fett, Ausrichtung zentrisch) ist der *optionale* Tabellenkopf `<th>...</th>` (table header).  
Damit eine Tabelle Inhalte überhaupt anzeigen kann, muss mindestens eine Zelle pro Zeile definiert sein.
  - Das *optionale* Attribut `style="text-align: left, center, right"` bestimmt die horizontale Ausrichtung von Zelleninhalten; die Kombination
  - `style="vertical-align: top, middle, bottom"` regelt deren vertikale Anordnung.
  - Auch Festlegungen von Höhe `height` und Breite `width` einzelner Zeilen oder Spalten mittels CSS-Formatierungen (s. Kapitel 19) sind möglich, soweit sie der Gesamtstruktur der Tabelle oder dem Platzbedarf von Inhalten nicht widersprechen.
  - Auf analoge Weise können Hintergrundfarben definiert werden.

Zeile 1		rowspan="2"
Zeile 2		
Zeile 3	colspan="2"	

```

<table>
<tr>
<td>...</td>
<td rowspan="2">...</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td colspan="2">...</td>
</tr>
</table>

```

- Das Attribut `rowspan="n"` kennzeichnet eine Tabellenzelle, die über **n** Zeilen reicht;
- `colspan="m"` definiert eine Zelle, die sich über **m** Spalten erstreckt.  
Achtung: Diese Attribute beziehen sich auf die Struktur der restlichen Tabellenzeilen oder -spalten.

Beachten Sie, dass im obigen Beispiel für die Zeile 2 in Ergänzung zur Struktur der Zeile 1 nur noch eine Zelle zu definieren ist!

Hinweis 1: Kommerzielle Web-Editoren bieten komfortable Werkzeuge zur Erstellung von Tabellen an; i. Allg. brauchen nur Anzahl der Spalten und Reihen vorgegeben werden – den Rest regelt das Programm.

Hinweis 2: Will man mögliche Textumbrüche und damit Verschiebungen innerhalb einer Tabelle bei Veränderungen des Browserfensters bzw. der Bildschirmauflösung vermeiden, sollte man der Tabelle mittels CSS-Formatierung eine feste Breite (in Pixeln) zuordnen.

## 14 Formulare

eröffnen die Möglichkeit, gezielt Informationen von Besuchern einer Seite abzufragen und Rückmeldungen zu erhalten. Die abgefragten Daten können an eine vorgegebene E-Mail-Adresse weitergeleitet und vom Empfänger ausgewertet werden; über ein entsprechendes Script ist sogar eine automatische Verarbeitung der übersandten Daten direkt auf Ihrem Server möglich.

```

<form action="mailto:myself@htwm.de" method="post" enctype="text/plain">
<p>Name: <input type="text" name="Besteller" size="25" maxlength="25">
</p>
Teesorte auswählen: <select name="Sorte">
<option>Assam</option>

```

```

<option selected>Darjeeling</option>
<option> Ceylon</option>
</select>
<p>
<input type="checkbox" name="Zugabe" value="Zucker" checked> gesüßt
</p>
<input type="submit" value="Bestellung auslösen!">
<input type="reset" value="Formular zurücksetzen">
</form>

```

Formulare werden mit dem Tag `<form>` eingeleitet und mit dem Tag `</form>` abgeschlossen.

**Achtung:** Positionieren Sie keine Formularelemente außerhalb dieses Bereiches.

- Das Attribut `action="mailto:..."` legt fest, dass die Daten an eine E-Mail-Adresse zu senden sind; alternativ wäre die URL-Adresse eines Script-Programms möglich, dass die Daten auswertet.
- Das Attribut `method="post"` gibt an, dass die Daten direkt übertragen werden sollen; die alternative Methode `get` schreibt alle Daten zunächst in Zwischenvariablen auf dem Server, um sie dann zur späteren Bearbeitung separat auslesen zu können.
- Der Parameter `enctype` schließlich gibt den Mime-Typ der übertragenen Daten an (im Beispiel: reiner Text).
- Das Attribut `name` legt die interne Bezeichnung eines Eingabefeldes fest:
  - Bei der Übertragung von Daten mittels E-Mail kennzeichnet der Feldname die Zugehörigkeit einer gesendeten Eintragung zum entsprechenden Eingabefeld.
  - Außerdem lassen sich damit mögliche, scriptbasierte Prüfungen oder Weiterverarbeitungen von Daten - vor Versendung an den Empfänger - einzelnen Formulareintragen zuordnen. Jedes Feld kann zur Auswertung der Daten über die vorgegebene Reihenfolge im Formular bzw. mit dem Attribut `name` identifiziert werden. Die Angabe des Feldnamens ist optional.
- Die Befehlsfolge `<select>...</select>` kennzeichnet eine Auswahlliste; sie bietet dem Anwender die Wahl zwischen mehreren vorgegebenen Begriffen.
  - Das Tag `<option>` identifiziert einzelne Auswahlmöglichkeiten; das zugehörige Attribut `selected` markiert die Option, die standardmäßig aktiviert ist.

```

<select name="Auswahlliste">
<option>Auswahl 1</option>
<option selected>Auswahl 2</option>
<option>Auswahl 3</option>
</select>

```

- Die Kombination `<textarea>...</textarea>` generiert ein mehrzeiliges Eingabefeld.

```
<textarea name="..." cols="60" rows="5">... </textarea>
```

- Der Befehl `<input>` weist generell auf eine Eingabe hin.

Mit dem Attribut `type` legen Sie fest, ob es sich dabei um eine der folgenden Varianten handelt:

- Textfeld, einzeilig (`text`, `password`)

```
<input type="text" size="10" ...>
```

Eingaben im Textfeld vom Typ `password` erfolgen verdeckt; der Browser zeigt eingetragene Buchstaben nur über einheitliche Ersatzsymbole an. Das Attribut `size` gibt an, wieviel Zeichen das Feld fasst.

- Radiobutton (`radio`)

Radiobuttons erlauben die Entscheidung zwischen verschiedenen Alternativen. In einer Gruppe von Radiobuttons, die alle durch den gleichen Namen `name` gekennzeichnet sind, ist aber nur eine Entscheidung möglich. Das Attribut `checked` legt eine Vorauswahl fest, `value` definiert den zu übermittelnden Wert.

```
<input type="radio" name="grb" value="Auswahl1" checked> a)
<input type="radio" name="grb" value="Auswahl2"> b)
<input type="radio" name="grb" value="Auswahl3"> c)
```

- Checkbox (`checkbox`)

Der Auswahlprozess lässt mit der Markierung mehrerer Elemente einer Gruppe Mehrfachnennungen zu. Das Attribut `value` gibt den Wert an, der übermittelt werden soll; er kann sich von der Legende zur Checkbox unterscheiden.

- Schaltfläche zum Senden der Daten (`submit`)  
Ein Klick auf diesen Button bewirkt das Absenden der gesammelten Informationen gemäß den Festlegungen zum Attribut `action`. Per E-Mail wäre diese Aktion allerdings nur dann erfolgreich, wenn der E-Mail-Client korrekt eingerichtet ist.
- Schaltfläche zum Rücksetzen der Eintragungen (`reset`)

Das Attribut value definiert in den beiden, letzten Fällen die Buttonaufschrift.

- Seit HTML5 stehen weitere Formularelemente zum Tag <input> zur Verfügung, deren Eintragungen durch den Browser auf korrekte Syntax überprüft bzw. für die spezielle Eingabefelder bereitgestellt werden; dazu gehören die Attributwerte type="email, url, date, time, number"  
Browser, welche diese Formularelemente nicht verarbeiten können, stellen sie als Eingabefelder vom Typ text dar.

Zwischen öffnendem Tag <form> und schließendem Tag </form> können - zur gestalterischen Anordnung von Formularelementen in Rastern - Tabellen eingesetzt werden.

## 15 Framesets

können **parallel** die Wiedergabe **verschiedener** HTML-Dokumente auf einzelne Segmente (Frames) des Browserfensters verteilen. Jedes der Segmente vermag unabhängig von den anderen Frame-Bereichen eine dem Frame zugeordnete Datei anzuzeigen. Die Frames dienen als **Platzhalter** für beliebige HTML-Dateien. Das Konzept wird mit HTML5 nicht mehr unterstützt, ist aber noch so verbreitet, dass es im Detail erläutert werden soll:

1. Die Aufteilung des Anzeigefeldes in mehrere Bereiche, d. h. die Struktur des Framesets ist in einer dem Komplex **übergeordneten** Metadokument zu definieren (z. B.: 2-spaltiges Frameset):

```
<html>
<head>
<title>Frame</title>
</head>
<frameset cols="130,*" frameborder="yes" framespacing="2">
  <frame name="Links" src="startseite1.html" scrolling="auto" noresize>
  <frame name="Rechts" src="startseite2.html">
</frameset>
</html>
```

2. Die Datei, welche das Frameset definiert, hat eine gegenüber üblichen HTML-Dateien leicht veränderte Grundstruktur. Das Frameset legt nur die Aufteilung in unterschiedliche Anzeigebereiche fest und gibt Adressen der Dateien vor, die in den Frames angezeigt werden sollen. Die zugehörige HTML-Datei enthält neben diesen Definitionen keine sonstigen Inhalte und damit auch **kein** <body>-Tag (s. u.).

Beachten Sie diese Besonderheit bei der Arbeit mit HTML-Editoren, die i. Allg. mit der Erstellung einer neuen Seite automatisch die vollständige Grundstruktur einer HTML-Datei inklusive <body>-Tag bereitstellen.

- Mit den Attributen cols="..." (Spalten) bzw. rows="..."(Zeilen) wird die Aufteilung des Anzeigebereiches festgelegt. Dies geschieht über Pixelwerte (s. Bspl.) bzw. Prozentangaben, die das Größenverhältnis der einzelnen Segmente zueinander beschreiben. Der verbleibende Rest kann durch Angabe eines Sterns \* gekennzeichnet werden. (Testen Sie Ihren Entwurf bei verschiedenen Bildschirmauflösungen.)
  - Jedes auf ein Frameset folgendes Tag <frame> definiert mit der Angabe src="URL..." die vollständige Adresse der HTML-Datei, die im betreffenden Frame(bereich) beim Start des Framesets angezeigt werden soll.
  - Der Parameter name="..." gibt dem Frame einen internen Namen. **Benennen** Sie die einzelnen Frames, um Verweisen diese Anzeigebereiche zuordnen zu können; das ist immer dann wichtig, wenn z.B. ein Frame zur Navigation genutzt wird und andere die entsprechenden Inhalte anzeigen sollen.
  - Die Ergänzungen frameborder, framespacing variieren Abgrenzungen sowie Abstände zwischen den Frames.
  - Der zusätzliche Wert scrolling="auto, yes, no" fügt dem Frame Bildlaufleisten hinzu, falls dies erforderlich ist bzw. verhindert es.
  - noresize verhindert interaktive Veränderungen der Framegröße durch den Nutzer.
3. Informationen, die innerhalb der einzelnen Frames angezeigt werden sollen, wie Navigation, Texte und Bilder sind in untergeordneten, **eigenständigen** HTML-Dokumenten abzulegen, die auch für sich allein ins Netz gestellt werden könnten.
  4. Um ein verschachteltes Frameset zu erzeugen, fügt man anstelle des entsprechenden <frame>-Tags ein weiteres <frameset> mit den zugehörigen frames ein.

```

<frameset cols="80%,*">
  <frameset rows="25%,*">
    <frame name="LinksOben" src="...">
    <frame name="LinksUnten" src="...">
  </frameset>
  <frame name="Rechts" src="...">
</frameset>

```

5. Ein sinnvoller Einsatz von Framesets besteht darin, zur Navigation in einem Bereich Inhaltsverzeichnisse (Verweise), u. U. verteilt auf einzelne Hierarchieebenen, anzuzeigen und in einem anderen Bereich die Inhalte selbst - beides in Form eigenständiger HTML-Dateien. Bei der Erstellung dieser, dem Frameset untergeordneten HTML-Seiten ist festzulegen, in welchem Frame Dateien zu öffnen sind, die über eine Verknüpfung (Link) angesprochen werden.

- Der Befehl

```
<base target="interner FrameName">
```

welcher im Head-Bereich der untergeordneten HTML-Dateien anzugeben wäre, legt den Standardfall fest.

- Das Attribut target="..." als Ergänzung zu einem <a>-Tag verweist im Einzelfall, abweichend vom Standard, auf den internen Namen des Frames, in dem die HTML-Datei wiedergegeben werden soll, welche der Zieladresse des zugehörigen Links entspricht, z.B.:

```
<a href="Link" target="interner FrameName">
```

- Analoges gilt für das Attribut

```
onClick="parent.internerFrameName1.location='Link1' ;  
parent.internerFrameName2.location='Link2'; return false"
```

zum Tag <a>, mit dessen Werten der Inhalt verschiedener Frames per Klick gleichzeitig gewechselt werden kann.

- Beachten Sie, dass es nicht üblich ist, fremde Web-Seiten innerhalb eigener Frames anzuzeigen; vielmehr ist dazu ein neues Fenster mit dem Attribut target="\_blank" zu öffnen.

Achtung: Fälscher nutzen Sicherheitslücken, wie sie einige Browser aufweisen, um Frames selbst gesicherter https-Seiten zu manipulieren und eigene Inhalte einzufügen. Sicherheitsrelevante Web-Sites sollten deshalb ohne Framesets gestaltet werden.

## 16 Image-Maps

sind Grafiken, bei denen einzelne Bildbereiche als Verweise wirken; das Anklicken dieser sensitiven Elemente öffnet eine dem Element zugeordnete HTML-Seite.

Die Grafik, die als Image-Map vorgesehen ist, wird wie alle Bilder mit dem Tag <img...> auf eine Seite eingebunden - ergänzt durch das Attribut **usemap**.

Beispiel:

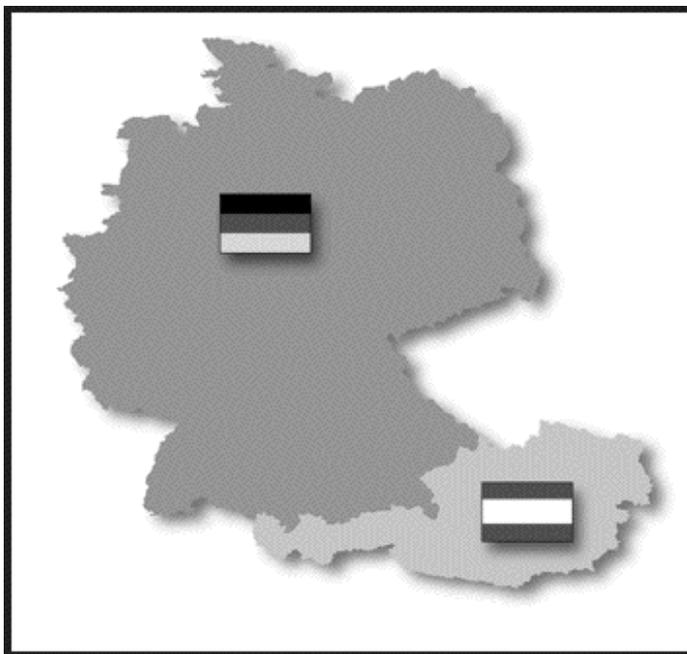
```

```

- usemap="..." Kennzeichnung der eingebundenen Grafik als Image-Map mit berührungssensitiven Bildbereichen

Die folgenden Programmzeilen, welche den <img>-Befehl ergänzen müssen, legen die sensitiven Bereiche der aktuellen Vorlage mit dem Namen "nachbarn" fest. Die anklickbaren Bereiche - im Beispiel beide Flaggen – öffnen zwei Seiten deutschland.html bzw. oesterreich.html :

```
<map name="nachbarn">
<area shape="rect" coords="51,42,84,70" href="deutschland.html">
<area shape="rect" coords="116,117,150,142" href="oesterreich.html">
</map>
```



Attribute, welche Form, Größe und Lage der anklickbaren Bereiche definieren, sind:

shape=rect: Rechteck	coords(linke, obere Ecke; rechte, untere Ecke)
shape=circle: Kreis	coords(Mittelpunkt; Radius)
shape=polygon: Vieleck	coords(Eckpunkte)

Die kartesischen Koordinaten (x,y) aller Punkte beziehen sich auf den Pixelabstand vom linken bzw. oberen Rand der Gesamtgrafik.

Hinweis: Eines von vielen Grafikprogrammen, das es ermöglicht, Position und Abmessungen einzelner Bereiche innerhalb eines Bildes festzustellen, ist die Applikation Microsoft

Paint. Aktuelle Cursorpositionen über einem geöffneten Bild werden dann im Anwendungsfenster links unten angezeigt.

## 17 Audio, Video, Animation

Multimediale Simulation ist eine moderne Form des Okkultismus.

Aktuelle Browser unterstützen die einfache Einbettung von HTML5 - <video> Elementen in HTML-Dokumente. Gegenwärtig ist diese Entwicklung noch auf drei Video-Containerformate beschränkt; letztere fassen Video- und Audio-Codec in einer Datei zusammen:

mp4: H264 video + AAC audio codec  
webm: VP8 video + Vorbis audio codec  
ogg: Theodora video + Vorbis audio codec

```
<video width="320" height="240" controls="controls">  
  <source src="___mp4" type="video/mp4" />  
  <source src="___ogg" type="video/ogg" />  
  Achtung: Ihr Browser unterstützt nicht das video -Tag!  
</video>
```

**Leider geben aus Lizenzgründen nicht alle Browser die oben genannten Dateitypen auch alle fehlerfrei wieder.**

---

In Webseiten lassen sich unabhängig von diesen Einschränkungen Multimediaelemente, welche mit der Grundausstattung eines Browsers nicht präsentiert werden können, **alternativ** zur obigen, einfachen Variante einbauen.

Sowohl Server als auch Browser verarbeiten Dateien, die nicht in HTML-Code geschrieben wurden, indem sie diese anhand ihrer MIME-Typen (Multipurpose Internet Mail Extensions) vorgegebenen Applikationen zuordnen. Erweiterungen professioneller Anbieter zum Browser - **Plug-ins** oder **Add-ons** - gewährleisten die Wiedergabe von Daten, welche nicht zu den Standardformaten einer HTML-Seite gehören. So ermöglichen Plug-ins Ihrem Browser das Abspielen von Multimedia-Dateien.

Der Standard-Befehl zum Einbinden von beliebigen Multimediaobjekten (Audio, Video ActiveX, Java-Applets, PDF und Flash) mittels Plug-ins in eine Webseite ist seit HTML4 das Tag <object>.

*ActiveX ist eine von Microsoft entwickelte Technik; sie soll spezifische Eigenschaften des MS-Windows-Betriebssystems für Web-Seiten nutzbar machen. ActiveX nutzt die OLE-Schnittstelle von Windows. Die Technologie ist für MS-Windows optimiert und daher nicht uneingeschränkt mit anderen Plattformen kompatibel.*

Das <object>-Tag bietet alternativ die Möglichkeit an, zwischen einleitendem und abschliessendem Tag weiteren HTML Code einzufügen, der nur dann interpretiert wird, wenn das eingebettete Objekt durch den aktuellen Browser nicht fehlerfrei angezeigt werden kann.

Seit HTML5 ist in diesem Kontext auch das Container-Tag <embed> für die Einbindung interaktiver Inhalte auf eine Web-Seite optional nutzbar.

Im Folgenden wird die Einbindung von Multimediaobjekten auf einer Web-Seite simultan mit beiden Tags <object> und <embed> formuliert.

- Beide Tags, <object> und <embed>, werden simultan programmiert um verschiedenen Browsern (Firefox, MSIE, Safari) und Sicherheitseinstellungen zu genügen.
- Der Suffix des Dateinamens oder die Angabe des MIME-Typs bestimmen das Plug-in bzw. die Applikation zum Start des eingebetteten Objekts.

```
<object width="Höhe in Pixeln" height="Breite in Pixeln" data="Datei (relative oder absolute Adresse)" type="MIME-Typ">
```

```
<embed height="Höhe in Pixeln" width="Breite in Pixeln" src="Datei (relative oder absolute Adresse)" type="MIME-Typ"...>
```

```
</object>
```

**Beachten Sie, dass Dateisuffix und MIME-Typ miteinander korrelieren müssen; letzterer muss nur angegeben werden, wenn aus dem Dateinamen nicht auf den Dateityp geschlossen werden kann.**

**Achtung:**

*Falls Sie die obigen oder folgende Quelltexte direkt als Kopie übernehmen möchten, so entfernen Sie alle harten Zeilenumbrüche innerhalb einzelner Tags, da sonst der entsprechende Code nicht funktioniert. Die Umbrüche wurden aus Platzgründen eingefügt.*

Weitere, teilweise optionale Angaben sind:

- autoplay - selbständiger Start von Video- bzw. Audiodateien (true, false)
- controller - (Standard)Steuerleiste (true, false)
- loop - Schleife (true, false)
- volume - Lautstärke (0...100)
- playeveryframe - Wiedergabe aller Bilder eines Films (true, false)  
u. U. bedingt diese Festlegung eine langsamere Abspielrate
- name - Name zur Nutzung des Plug-ins innerhalb eines JavaScript-Programms

### 17.1 Quicktime (.mov)

Der Befehlscode zur Einbindung eines Quicktimefilms lautet im simultanen Aufruf beider Befehle :

```
<object width="320" height="240" data="____.mov">
  <param name="type" value="video/quicktime">
  <param name="bgcolor" value="silver">
  <param name="cache" value="false">
  <param name="playeveryframe" value="true">
  <param name="loop" value="false">
  <param name="volume" value="100">
  <param name="autoplay" value="false">
  <param name="controller" value="true">

  <embed width="320" height="240" src="____.mov" type="video/quicktime">

</object>
```

Das <param>-Tag dient der Übergabe von Parametern an das Plug-in, welches mit dem <object>-Tag eingebettet wurde.

**Beachten Sie, dass Dateisuffix und MIME-Typ miteinander korrelieren müssen:** Die Dateierendung .mov erfordert den Typ video/quicktime.

### 17.2 Flashobjekt (.swf)

Die Einbindung von Flashobjekten auf einer Web-Seite sollte simultan mit den beiden Tags <object> und <embed> programmiert werden, um verschiedene Browserversionen und Sicherheitseinstellungen zu bedienen. Der Befehlscode zur Einbindung eines Flash-Objektes (\_\_\_\_.swf) lautet:

```
<object width="320" height="240" data="____.swf">
  <param name="type" value="application/x-shockwave-flash">
  <param name="cache" value="false">
  <param name="volume" value="100">

  <embed width="320" height="240" src="____.swf"
  type="application/x-shockwave-flash">
</embed>

</object>
```

**Beachten Sie, dass Dateisuffix und MIME-Typ miteinander korrelieren müssen:** Die Dateiendung .swf erfordert den Typ application/x-shockwave-flash.

### 17.3 RealPlayer (.rm)

Die Einbindung von RealMedia Objekten ist auf einer Web-Seite simultan mit den beiden Tags <object> und <embed> zu programmieren, um verschiedene Plattformen, Browser-Versionen und Sicherheitseinstellungen zu bedienen. Der Befehlscode zur Einbindung eines Trailers lautet:

```
<object width="320" height="240" data="____.rm">
  <param name="type" value="application/vnd.rn-realmedia">
  <param name="controls" value="ImageWindow">
  <param name="console" value="one">
  <param name="autostart" value="true">

  <embed width="320" height="240" src="____.rm"
  type="application/vnd.rn-realmedia">

</object>
```

**Beachten Sie, dass Dateisuffix und MIME-Typ miteinander korrelieren müssen:** Die Dateiendung .rm erfordert den Typ application/vnd.rn-realmedia.

Eine Besonderheit des RealPlayers besteht darin, dass sämtliche Kontrollelemente, wie ImageWindow, ControlPanel, InfoPanel, ein eigenes <object>-Tag benötigen. Letztere sind simultan mit dem Parameter console zu ergänzen, um über dessen (einheitlichen) Wert die einzelnen Tags zu verketten.

## 17.4 Media Player (.wmv)

Der HTML-Code zur Einbindung eines Windows Media-Videos unter Nutzung des Media Players lautet:

```
<object width="320" height="240" data="___wmv">
  <param name="type" value="application/x-mplayer2">
  <param name="ShowControls" value="true"
  <param name="ShowStatusBar" value="true">
  <param name="autoStart" value="true">

  <embed width="320" height="240" src="___wmv"
  type="application/x-mplayer2">
</object>
```

**Beachten Sie, dass Dateisuffix und MIME-Typ miteinander korrelieren müssen:** Die Dateierendung .wmv erfordert den Typ application/x-mplayer2.

- Der Player wird nicht mehr als Gesamtobjekt auf die Seite eingefügt, sondern hierarchisch aus einzelnen Elementen (Statusbar, Controls, ...) zusammengestellt. Das Playerobjekt selbst ist nur der Grundbaustein.

***Das Flip4Mac WMV Plugin ist das offizielle Plug-in, um WMV-Inhalte im Web mittels Mac OSX unter Verwendung von Quicktime anzuzeigen.***

Die Entwicklung der Rechentechnik ist bei der Wehrlosigkeit vor dem Computer angelangt.

---

Das Internet ist ein hervorragendes Instrument zur Lösung von Problemen, die es ohne es nicht gäbe.

## 17.5 Audio (.mp3)

Seit HTML5 steht das Tag <audio> zur Verfügung:

```
<audio controls="controls">
  <source src="___ogg" type="audio/ogg">
  <source src="___mp3" type="audio/mp3">
  <source src="___wav" type="audio/wav">
  Achtung: Ihr Browser unterstützt nicht das audio-Tag!
</audio>
```

Audiodateien lassen sich alternativ in der simultanen Kombination der Tags <object> und <embed> innerhalb einer HTML-Datei einfügen und mit einem Browser abspielen. Dies geschieht über Audio Plug-ins (z. B. QuickTime, RealOne Player, Media Player), welche diese Formate verarbeiten können und die auf dem Rechner installiert sein müssen.

Die folgende Befehlskette ist kompatibel zu den genannten Playern.

```
<object height="24" width="240" data="___mp3">
  <param name="type" value="audio/mpeg">
  <param name="controller" value="true">
  <param name="autoplay" value="false">

  <embed height="24" width="240" src="___mp3" type="audio/mpeg">
</object>
```

- Die Angabe height="24" steht für die übliche Höhe einer Standardsteuerleiste.
- Den selbstständigen Start des eingebetteten Objekts verhindert die Ergänzung autoplay="false".
- Im aktuellen Beispiel besitzt die Audio-Datei, welche wiedergegeben werden soll, den MIME-Typ audio/mpeg.

### 17.6 Java-Applet (.class)

Mit der objektorientierten Programmiersprache Java lassen sich Anwendungen (Applets) erstellen, welche als Element einer Web-Seite unter der Regie des Browsers ausgeführt werden. Voraussetzung ist eine aktuelle Java-Umgebung (JRE) auf dem Rechner des Nutzers.

Die einfachste Möglichkeit, ein Java-Applet in eine HTML-Seite einzufügen, basiert direkt auf der Verwendung des object-Tags: Der folgende Code ist mit aktuellen Browsergenerationen kompatibel.

```
<object height="240" width="320" type="application/x-java-applet">
  <param name="code" value="___class" />
  <param name="archive" value="___jar" />
  Applet failed to run. No Java plug-in was found.
</object>
```

- Das Attribut code definiert die Quelle des Java-Applets (Namenserweiterung class), das gestartet werden soll. Ein Applet wird nur dann aktiviert, wenn die geforderte Java-Umgebung (inklusive Plug-in) auf dem Rechner installiert ist.
- Umfasst das Applet mehr als eine Datei, so empfiehlt es sich, eine Archiv-Datei anzulegen, die alle Einzelkomponenten bündelt. Download-Zeiten lassen sich damit reduzieren. Standard Java Archive (Suffix: jar) basieren auf dem Format

von ZIP-Dateien und werden mit entsprechenden Tools angelegt, die zu Java Entwicklungsumgebungen gehören. Die Spezifizierung dieser Archive erfolgt über das Attribut archive im <param>-Tag.

- Achten Sie darauf, einen vollständigen oder relativen Pfad zum Archiv immer dann anzugeben, wenn sich letzteres in einem anderen Ordner oder Verzeichnis befindet als die aufrufende HTML-Datei.

## 18 Metatags

Im Cyberspace weiß keiner, wohin er will, ist aber umso schneller dort.

Die Informationsflut im Internet macht eine Orientierung nicht leicht; Hilfe bieten Suchmaschinen oder Kataloge.

- Suchmaschinen untersuchen mit sogenannten "Robots" oder "Spidern" Ihre Webseite entweder nach Anmeldung oder sogar eigenständig. Stoßen sie auf eine Internetseite, die noch nicht gelistet ist, werden deren Inhalte ausgewertet und nach gefundenen Schlüsselworten in eine Liste eingetragen. Wer schnell gefunden werden will, muss seine Website für Suchmaschinen optimieren; dies geschieht mittels Meta Tags.  
Diese Befehle enthalten Informationen, die nicht auf dem Bildschirm angezeigt werden, aber von Suchmaschinen abgefragt werden können.
- Teilweise werden die zugehörigen Webseiten begutachtet und als redaktionell betreuter Eintrag in einen Katalog (Verzeichnis) aufgenommen.

Im Allg. setzen Suchmaschinen (z. B. Google, Yahoo, Bing, Lycos, metacrawler, Ixquick, Ask, AOL Suche, Fireball, Acoon) verschiedene **Spider** ein und die gefundenen Seiten werden je nach Suchdienst unterschiedlich bewertet. Häufig werden nur die Inhalte der ersten Textzeilen mit der Suchmaschine gelistet (textbasierte Stichwortlisten). Eine Information, die für Suchmaschinen sehr wichtig ist, wird vom Nutzer oft nur oberflächlich formuliert, nämlich der im Header eingetragene **Titel** der Webseite.

Sie erhöhen Ihre Chancen, schnell gemäß eigener Vorgaben gefunden zu werden, wenn sie Meta Tags im Head-Bereich des Quelltextes sinnvoll anwenden.

Da Robots weder Framesets, Grafiken oder Javascripts inhaltlich auswerten können, helfen in diesen Fällen Meta Tags, Informationen darüber an die Suchmaschine zu übermitteln. Platziert im Head-Bereich einer Seite könnte das so aussehen:

```
<meta name="description" content="Beschreibung Ihrer Site...">
```

Die von Ihnen verfasste Beschreibung wird bei eventuellen Treffern aufgrund von Suchabfragen als Information ausgegeben. Die Beschreibung sollte den tatsächlichen Inhalt der Seite in normaler Sprache wiedergeben.

Zusätzlich haben Sie die Möglichkeit, Stichwörter aufzulisten. Stichwörter können sich aus mehreren Begriffen zusammensetzen und werden durch Komma oder Semikolon voneinander getrennt. Auch die Stichwörter merkt sich der Spider.

Wenn Sie "Medienmanager" als **Keyword** festlegen und ein User als Stichwort in eine Suchmaschine ebenfalls Medienmanager eintippt, sollte Ihre Site in der Liste der gefunden Websites erscheinen.

```
<meta name="keywords" content="Medienmanager, Medien, ...">
```

Wiederholen sich Ihre Stichworte im Textanfang einer Seite, so haben Sie gute Chancen bei einer textbasierten Stichwortabfrage gefunden zu werden. Passen diese Informationen nicht in das vorgesehene Design der Seite, empfiehlt es sich, die Stichwörter in einer ersten Textzeile, für den Betrachter unsichtbar, in der Hintergrundfarbe zu wiederholen. Sie können auch versuchen Tippfehler zu berücksichtigen; so ist es durchaus denkbar, dass nach einem "Medienmanger" gesucht wird.

Weisen Sie den **Robot** an, sämtlichen Links zu folgen; sukzessive werden dann alle Ihre Seiten in den Suchindex aufgenommen.

```
<meta name="robots" content="index, follow">
```

Der folgende Tag sorgt dafür, dass nur die angemeldete Seite indiziert wird. Der Robot folgt nicht den Links, um weitere Seiten zu untersuchen.

```
<meta name="robots" content="nofollow">
```

Mit Hilfe einer Variante können Sie dem Spider mitteilen, dass er bestimmte Seiten NICHT untersuchen soll. Zum Beispiel Muster- oder Testseiten bzw. **Navigationsseiten**, die Bestandteil eines Framesets sind.

```
<meta name="robots" content="noindex">
```

Der nächste Tag verhindert, dass Seiten via Proxy-Server zwischengespeichert werden. Mit "no-cache" anstelle von "private" verhindern Sie auch das Speichern im Browsercache, vorausgesetzt der Browser reagiert auf diese Forderung.

```
<meta http-EQUIV="pragma" content="private">
```

"Revisit" bewirkt, dass der untersuchende Robot nach einer anzugebenden Anzahl von Tagen zurückkehrt. Dies ist jedoch wenig praktikabel, da die Spider eigenen **Regeln** folgen und nicht selten überlastet sind.

```
<meta name="revisit-after" content="10 days">
```

Für den Fall, dass Sie Ihren Provider wechseln wollen, ist der folgende Meta-Tag nützlich. Mit ihm werden Besucher automatisch nach einer vorgegebenen Anzahl von Sekunden auf die neue URL weitergeleitet. Setzen Sie den Wert auf "0" erfolgt die Weiterleitung sofort. Auch die Präsentation von Bildfolgen oder wechselnden Fensterinhalten ist damit leicht zu programmieren.

<meta http-equiv="refresh" content="20; URL="Folgeseite...">

Ihren Namen und Ihr Copyright verewigen Sie so:

Herausgeber:

<meta name="publisher" Content="Fachbereich Medien">

Urheberrecht:

<meta name="copyright" Content="P. Will">

Teilen Sie den Suchmaschinen mit, in welcher Sprache Ihre Seiten gelesen werden können. Moderne Suchmaschinen führen eine Sprachprüfung durch und ignorieren dieses Tag.

<meta name="Content-Language" Content="de">

Der Einsatz von Meta-Tags kann sich durchaus lohnen. Mit diesen Tags verleihen Sie Ihren Seiten mehr Gewicht.

Formulieren Sie ein Thema: z.B. Bildung, Wissenschaft, Forschung, Technik, Gesellschaft, Soziales, Kultur, Medien,

<meta name="page-topic" Content="Thema...">

Charakterisieren Sie die Art Ihrer Seite: z.B. Anleitung, Anzeige, Bericht, Reportage, Bild, Foto, Email-Archiv, FAQ, Forschungsbericht, Aufgabensammlung, Formular, Karte, Plan, Katalog, Verzeichnis, Adressen, Produktinfo, Download, Software, Audio, Video

<meta name="page-type" Content="Art...">

An wen richtet sich die Seite? Anfänger, Fortgeschrittene, Profis, Erwachsene, Jugendliche, Kinder, Frauen, Männer, Schüler, Azubis, Studenten

<meta name="audience" Content="Publikum...">

Weitere Meta Befehle dienen der Archivierung und sind für normale Web-Sites ohne Belang.

Eine ausführliche Sammlung aller Meta Tags finden Sie unter:

<http://wdvl.com/Authoring/HTML/Head/Meta/>

Wer Metainformationen nicht selbst codieren will, kann einen Meta-Tag-Generator nutzen, z.B.:

<http://www.sitesubmission.de/metatag.htm>

Verschiedene Dienstleister bieten den automatisierten Eintrag einer Webseite in unzählige Suchmaschinen an. Automatisierte Eintragungen in verschiedene Suchmaschinen erfolgen z. B. unter:

<http://www.suchmaschineneintragung.de>  
<http://www.submit-it.de>

**Achtung:** Es gibt auch unseriöse Anbieter.

Die Suchmaschine Google gilt als eine der besten Anbieter mit sehr guten Trefferquoten; war selbst aber zeitweise dem Vorwurf (Googleating) ausgesetzt, von professionell arbeitenden Spammern manipuliert zu werden.

*Erwarten Sie nicht zu viel; manche Suchmaschinen arbeiten nur mit simplen Algorithmen und nutzen keine der Metainformationen .*

---

## CSS

Web-Seiten sind nicht nur ein informelles Medium; auch die Inszenierung spielt eine entscheidende Rolle bei der Entscheidung potenzieller Interessenten, sich näher mit Inhalten zu beschäftigen.

HTML strukturiert eine Seite, CSS gestaltet sie.

## 19 Cascading Style Sheets (CSS)

haben sich in diesem Kontext als **Stilelement** inzwischen einen festen Platz im Web-Design erobert.

StyleSheets werden aus den gleichen Gründen verwendet wie Formatvorlagen in Programmen zur Textverarbeitung. Sie definieren das Erscheinungsbild gekennzeichnete Bereiche oder Elemente mittels Vorgaben zum Layout einzelner Komponenten bzw. einer ganzen Seite. Dabei geht der Befehlssatz von CSS weit über die Leistungsfähigkeit üblicher HTML-Tags hinaus; selbst interaktive Aktionen sind möglich. Zunehmend ersetzen Style Sheets in ihrer Funktionenvielfalt auch Skripte.

**Achtung:** StyleSheets werden kontinuierlich weiterentwickelt und ergänzt; deshalb setzen selbst aktuelle Browser die neuesten Entwicklungen nicht immer im gewünschten Stil um.

Folgende Formatierungen kann ein StyleSheet u. a. vorgeben:

- Schrifttyp, -größe und -farbe
- Textauszeichnungen (kursiv/ fett/ unterstrichen und andere)
- Textausrichtungen (links/ rechts/ zentrisch/ Blocksatz)
- mehrspaltige Texte
- Zeilen-, Textabstände, Einrückungen
- Rahmen, Schatten, Abmessungen
- Hintergrundfarben oder Hintergrundbilder
- Randabstände
- Positionierungen (relativ und absolut für beliebige Objekte)
- Ebenen
- Farben, Farbverläufe
- modifizierte Formen von Mauszeigern

Auch das genaue **Positionieren** von Komponenten innerhalb einer Webseite ist unter Verwendung von StyleSheets sehr einfach zu realisieren. Es lassen sich absolute, pixelgenaue Werte definieren; selbst verschiedene, übereinander liegende Ebenen sind möglich. Schriftgrößen können in der typografischen Maßeinheit Punkt (pt) oder in Millimetern festgelegt werden. Gleiches gilt für Rand- oder Wortabstände.

Rotationen und Transformationen einzelner Elemente sind ebenso möglich wie die Unterscheidung verschiedener Anzeigemedien.

Es genügt, Inhalte und Elemente eines Dokuments hierarchisch zu gliedern. **Layout** und **Formatierung** der Seite übernimmt dann der Browser nach den vorgegebenen Stil-Anweisungen des Webdesigners.

StyleSheets können über verschiedene Anwendungsformen (lokal, intern, extern) in eine HTML-Seite eingebunden werden..

Prinzipiell lassen sich alle Tags mit dem globalen Attribut style ergänzen.

Direkt im Quellcode wirken CSS-Befehle nur an ausgewählten Stellen einer Seite. Beispielsweise innerhalb einer Überschrift oder eines Abschnitts.

```
<h1 style="font-size:12pt; color:red">Überschrift</h1>
<p style="background-color:silver; text-align:right">Abschnitt</p>
```

- *Einzelne Komponenten einer Stilinformation werden durch Semikola voneinander getrennt.*
- *Eigenschaften und ihre Werte werden mittels Doppelpunkt verknüpft.*

Die Kombination `<span>...</span>` formatiert einzelne Textpassagen oder -elemente. Innerhalb dieser Markierung wird die Anzeige dem gewählte Stil angepasst.

```
<span style="line-height:150%">Text</span>
```

Mit der Befehlskette `<div>...</div>` erstellen Sie einen Block, der mehrere unterschiedliche Elemente enthalten kann. Innerhalb dieses Bereiches können verschiedene CSS-Formate für alle enthaltenen Elemente festgelegt werden.

```
<div id="..." style="font-size:40pt; position:absolute; left:50px; top:100px; z-index:2">
...
</div>
```

*Schriftgröße 40 Points, Position (50,100) Pixel, z-Ebene 2*

Komponenten einer HTML Seite können formatiert werden, indem man Style Anweisungen im Dateikopf unterbringt.

Das Layout von Formularelementen ließe sich z. B. mit folgenden Vorgaben verändern, die ausschließlich die Tags `<input>`, `<textarea>` und `<select>` betreffen:

```
<html>
<head>
<title>Seitenname</title>
<style type="text/css">
input, textarea, select {
border-color : #5a7394;
border-width : 2;
background-color : #ccddee;
font-family : times, serif;
font-size: 12pt;
color: red;
padding: 4px; }
</style>
</head>
<body>
...
</body>
</html>
```

- *Einzelne Komponenten einer Stilinformation werden durch Semikola voneinander getrennt.*
- *Eigenschaften und ihre Werte werden mittels Doppelpunkt verknüpft.*

Besonders effektiv kann man StyleSheets dann einsetzen, wenn sie als **Formatvorlagen** in eine externe Datei (Endung .css) ausgelagert werden. Die Vorlage wirkt auf alle Seiten einer Web-Präsentation, in denen sie aktiviert ist. Das spart Code; der Datenumfang wird reduziert. Spätere Änderungen lassen sich effizient für die **gesamte Webseite** realisieren. Die Formatvorlagen sind in einer externen Textdatei (z. B. style.css) gespeichert, die ausschließlich Style Anweisungen (z.B. grau markierten Code oben) enthält. Die Aktivierung der entsprechenden Anweisungen erreichen Sie mit folgender Befehlszeile im Dateikopf (head).

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

---

Das Attribut class ermöglicht es, benutzereigene Formate zu definieren:  
Die Bezeichnung der Klasse ist frei wählbar, ihr wird lediglich ein Punkt vorangestellt, z.B.

```
tr.pw1 { font-family : sans-serif; font-size : 9pt; text-align : right; }  
(Text mit einer Größe von 9pt in serifenloser Schrift und nach rechts ausgerichtet)
```

**Tags** innerhalb des HTML-Körpers aktivieren die vorgegebenen Formatierungen der Klasse und weisen damit den markierten Elementen (z. B. in einer Tabellenzeile):

```
<tr class="pw1">
```

diese Formatierungen zu.

Lokale Methoden überschreiben bzw. ergänzen interne oder externe StyleSheets. Bedarf dafür kann dann bestehen, wenn durch ein externe Datei bereits Stilinformationen vorgegeben sind, die für das gesamte Dokument gelten.

Die pixelgenaue Positionierung von Elementen in verschiedenen übereinanderliegenden Ebenen unter Verwendung von CSS-Befehlen ermöglicht die Befehlskette <div>...</div> (Blocklevel-Tag).

- Die Stapelung der Objekte geschieht mittels aufsteigender Nummern des z-index.
- Horizontale und vertikale Lage eines Elements regelt der Stil position mit den Koordinaten left und top.
- Die absolute Positionierung position:absolute richtet das Objekt unabhängig von anderen Seitenelementen im Browserfenster aus;
- alternativ ist die Angabe position:relative, bei der die Positionierung relativ zu gleichwertigen Blöcken oder Komponenten der Seite erfolgt.

```
<div id="E1" style="position:relative; left:70px; top:10px; z-index:1">  
<span style="font-size:60pt; color:#...">Ebene 1</span></div>
```

```
<div id="E2" style="position:absolute; left:90px; top:400px; z-index:2">
<span style="font-size:80pt; color:#...">Ebene 2</span></div>
```

Beachten Sie, dass sich das Erscheinungsbild bei Angabe von Koordinaten in der Maßeinheit px (Pixel) auch mit der Bildschirmauflösung ändern kann.

Einzelne Attribute eines Styles können mittels Selektoren je nach Ereignissen noch modifiziert werden, so dass sich das Erscheinungsbild des betreffenden Bereiches ändert – z.B. in Abhängigkeit davon, ob sich der Mauszeiger über der entsprechenden Komponente befindet oder das Element per Mausklick aktiviert wurde; Verweise in Navigationsleisten ließen sich mit folgendem Style interaktiv variieren:

```
<style type="text/css">
a:link { color: black;
font: menu;
font-weight: bold;
text-decoration: none;
padding: 4pt;
border: 1pt solid white;
border-radius: 20pt; }
a:visited { color: gray;
font: menu;
font-weight: bold;
text-decoration: none;
padding: 4pt;
border: 1pt solid white;
border-radius: 20pt; }
a:hover { color: white;
font: menu;
font-weight: bold;
text-decoration: none;
background-color: #5A7394;
padding: 4pt;
border-bottom: 1pt solid black;
border-right: 1pt solid black;
border-top: 1pt solid silver;
border-left: 1pt solid silver;
border-radius:20pt; }
a:active { color: maroon;
text-decoration: none; }
</style>
```

Stylesheets werden lokal für Bereiche einzelner Tags festgelegt oder intern im Head-Abschnitt einer Web-Seite bzw. extern in einer separaten Datei definiert, auf die sich die Web-Seite bezieht.

Verwenden Sie Stylesheets, um z.B. typographische Eigenschaften von Schriften festzulegen.

Regeln Sie Positionierung, Rahmung, Färbung und Transparenz ausgewählter Elemente.

Die Darstellung einer Seite lässt sich auf unterschiedliche Ausgabemedien anpassen.

#### Übersicht: Referenz CSS2

Die Bedeutung der meisten Stilelemente erschließt sich, grundlegende Erfahrungen im Webdesign vorausgesetzt, aus deren Bezeichnungen. Systematisches Probieren mit dem Ziel, verschiedenste Komponenten vorhandener Web-Seiten mittels Stylesheets zu modifizieren, führt in diesem Zusammenhang schnell zu praktisch verwertbaren Erfahrungen. In den folgenden Tabellen wird eine relevante Auswahl vorgestellt.

#### Schrift

<b>Eigenschaft</b>	<b>Beschreibung</b>	<b>Wert</b>
font	Schrift	font-style font-variant font-weight font-size/line-height font-family caption icon menu message-box small-caption status-bar
font-family	Schriftart	family-name generic-family
font-size	Schriftgröße	xx-small x-small small medium large x-large xx-large smaller larger pt, px, %
font-size-adjust		none number
font-stretch	Schriftbreite	normal wider narrower ultra-condensed extra-condensed condensed

		semi-condensed semi-expanded expanded extra-expanded ultra-expanded
font-style	Schriftstil	normal italic oblique
font-variant	Buchstabenart	normal small-caps
font-weight	Schriftdicke	normal bold bolder lighter 100,200,300,400,500 600,700,800,900

#### Text

Eigenschaft	Beschreibung	Wert
color	Textfarbe	color
direction	Text(lese)richtung	ltr rtl
letter-spacing	Abstand zwischen Buchstaben	normal pt, px
text-align	Textausrichtung	left right center justify
text-decoration	Textverzierung	none underline overline line-through blink
text-indent	Einrückung	pt, px, %
text-shadow	Schatten	none color pt, px
text-transform	Art der Letter	none capitalize uppercase lowercase

unicode-bidi	Textrichtung	normal embed bidi-override
white-space	Leerzeichen	normal pre nowrap
word-spacing	Wortabstand	normal pt, px

#### Hintergründe

<b>Eigenschaft</b>	<b>Beschreibung</b>	<b>Werte</b>
background	Hintergrund	background-color background-image background-repeat background-attachment background-position
background-attachment	Hintergrundgrafik fixieren Bildlauf ermöglichen	fixed scroll
background-color	Hintergrundfarbe	color-rgb color-hex color-name transparent
background-image	Hintergrundbild	url none
background-position	Position eines Hintergrundbildes	top left top center top right center left center center center right bottom left bottom center bottom right x-% y-% x-pos y-pos
background-repeat	Wiederholung eines Hintergrundbildes	repeat repeat-x repeat-y no-repeat

## Tabellen

<b>Eigenschaft</b>	<b>Beschreibung</b>	<b>Wert</b>
border-collapse	Rahmenart	collapse separate
border-spacing	Rahmenabstand	pt, px
caption-side	Position der Überschrift	top bottom left, right
empty-cells	Behandlung von leeren Zellen	show hide
table-layout	Tabellenstruktur	auto fixed

## Rahmen

<b>Eigenschaft</b>	<b>Beschreibung</b>	<b>Wert</b>
border	Rahmen	border-width border-style border-color
border-bottom	Rahmen unten	border-bottom-width border-style border-color
border-bottom-color	Rahmenfarbe unten	border-color
border-bottom-style	Art des Rahmens	border-style
border-bottom-width	Rahmenbreite unten	thin medium thick pt, px
border-color	Rahmenfarbe	color
border-left	Rahmen links	border-left-width border-style border-color
border-left-color	Rahmenfarbe links	border-color
border-left-style	Rahmenart links	border-style

border-left-width	Rahmenbreite links	thin medium thick pt, px
border-right	Rahmen rechts	border-right-width border-style border-color
border-right-color	Rahmenfarbe rechts	border-color
border-right-style	Rahmenart rechts	border-style
border-right-width	Rahmenbreite rechts	thin medium thick pt, px
border-style	Rahmenart	none hidden dotted dashed solid double groove ridge inset outset
border-top	Rahmen oben	border-top-width border-style border-color
border-top-color	Rahmenfarbe oben	border-color
border-top-style	Rahmenart oben	border-style
border-top-width	Rahmenbreite oben	thin medium thick pt, px
border-width	Rahmenbreite	thin medium thick pt, px

### Abmessungen

<b>Eigenschaft</b>	<b>Beschreibung</b>	<b>Wert</b>
height	Höhe	auto pt, px, %
line-height	Zeilenabstand	normal number pt, px, %
max-height	Maximum Höhe	none pt, px, %
max-width	Maximum Breite	none pt, px, %
min-height	Minimum Höhe	pt, px, %
min-width	Minimum Breite	pt, px, %
width	Breite	auto pt, px, %

### Listen

<b>Eigenschaft</b>	<b>Beschreibung</b>	<b>Wert</b>
list-style	Listenart	list-style-type list-style-position list-style-image
list-style-image	Bild als Listenzeichen	none url
list-style-position	Position des Listenzeichens	inside outside
list-style-type	Listentyp	none disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-alpha upper-alpha lower-greek

		lower-latin upper-latin
marker-offset		auto pt, px

#### Abstand

Eigenschaft	Beschreibung	Wert
margin	Abstand	margin-top margin-right margin-bottom margin-left
margin-bottom	Abstand nach unten	auto pt, px, %
margin-left	Abstand nach links	auto pt, px, %
margin-right	Abstand nach rechts	auto pt, px, %
margin-top	Abstand nach oben	auto pt, px, %

#### Randabstände

Eigenschaft	Beschreibung	Wert
padding	Randabstand	padding-top padding-right padding-bottom padding-left
padding-bottom	Abstand nach unten	pt, px, %
padding-left	Abstand nach links	pt, px, %
padding-right	Abstand nach rechts	pt, px, %
padding-top	Abstand nach oben	pt, px, %

### Positionierung

<b>Eigenschaft</b>	<b>Beschreibung</b>	<b>Wert</b>
bottom	Abstand nach unten	auto pt, px, %
clip	Ausschnitt	shape auto
left	Abstand nach links	auto pt, px, %
overflow	overflow	visible hidden scroll auto
right	Abstand nach rechts	auto % pt, px
top	Abstand nach oben	auto pt, px, %
vertical-align	Vertikale Ausrichtung	baseline sub super top text-top middle bottom text-bottom pt, px, %
z-index	Stapelreihenfolge	auto number

### Outlines

<b>Eigenschaft</b>	<b>Beschreibung</b>	<b>Wert</b>
outline	outline	outline-color outline-style outline-width
outline-color	Outlinefarbe	color

		invert
outline-style	Outlineart	none dotted dashed solid double groove ridge inset outset
outline-width	Outlinedicke	thin medium thick pt, px

#### Klassifizierung

Eigenschaft	Beschreibung	Wert
cursor	Mauszeiger	url auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help
display	Anzeige	none inline block list-item run-in compact marker

		table inline-table table-row-group table-header-group table-footer-group table-row table-column-group table-column table-cell table-caption
float	Position innerhalb eines anderen Elements	left right none
position	Art der Positionierung	static relative absolute fixed
visibility	Sichtbarkeit	visible hidden collapse

## 20 Responsive Webdesign

Die Verschiedenheit von Geräten (Computerbildschirm, Tablet, Smartphone, eBook-Reader, Drucker, ...) zur Wiedergabe von HTML-Seiten macht es erforderlich, dass letztere im grafischen Erscheinungsbild auf die aktuell genutzte Hardware reagieren müssen.

Medienabfragen (Media Queries) prüfen Kriterien (device-aspect-ratio, orientation, color, monochrome, resolution, ...), die entscheiden, welche Stylesheets dem aktuellen Ausgabe-medium (screen, print, projection, ...) zugeordnet werden sollen.

Die CSS-Frameworks **Bootstrap** oder **iWebKit** beinhalten in diesem Kontext jeweils ein Paket von Dateien, welche die einfache Erstellung von Webseiten oder Webapplikationen kompatibel zu Smartphones oder Tablets ermöglichen. Mittels vorgefertigtem CSS-Code lassen sich in kurzer Zeit, relativ einfach Webseiten erstellen, deren Oberflächen den Erscheinungsbildern und Abmessungen der genannten mobilen Geräte entsprechen.

Alle Dateien der beiden Pakete finden Sie unter:

<http://holdirbootstrap.de/>

<http://snippetspace.com/portfolio/iwebkit/>

## 21 Veraltete Tags

Führung bedeutet, alle anderen vor sich herzutreiben.

Die ständige Weiterentwicklung von HTML und seine Ergänzung durch CSS führte dazu, dass einige Tags (z.B.: font, center) und Attribute (z.B.: border, bgcolor, align, valign, cellspacing, cellpadding, size) seit HTML5 als veraltet gelten. Aktuelle Browser zeigen sie noch an; zukünftige Browsergenerationen müssen diese Anweisungen nicht mehr akzeptieren. Sie sind deshalb nicht erwünscht (deprecated). Aus diesem Grund empfiehlt es sich, entsprechenden HTML-Code durch Stylesheets zu ersetzen.

Die folgende Tabelle zeigt sieben relevante Beispiele:

deprecated	CSS
<pre>&lt;table width="480" cellspacing="5" cellpadding="10"&gt;   &lt;tr&gt;     &lt;td&gt;...&lt;/td&gt;     ...   &lt;/tr&gt; &lt;/table&gt;</pre>	<pre>&lt;table style="width:480; border-collapse: separate; border-spacing:5"&gt;   &lt;tr&gt;     &lt;td style="padding:10"&gt;...&lt;/td&gt;     ...   &lt;/tr&gt; &lt;/table&gt;</pre>
<pre>&lt;img src="logo.png" border="2"&gt;</pre>	<pre>&lt;img src="logo.png" style="border-width:2; border-style:solid"&gt;</pre>
<pre>&lt;center&gt; zentrisch &lt;/center&gt;</pre>	<pre>&lt;div style="text-align: center"&gt; zentrisch &lt;/div&gt;</pre>
<pre>&lt;font face="Helvetica"&gt; Textstelle: Schrifttyp &lt;/font&gt;</pre>	<pre>&lt;span style="font-family: Helvetica"&gt; Textstelle: Schrifttyp &lt;/span&gt;</pre>

<pre>&lt;p&gt;&lt;font size="4"&gt; Abschnitt: Schriftgröße &lt;/font&gt;&lt;/p&gt;</pre>	<pre>&lt;p style="font-size: 14pt;"&gt; Abschnitt: Schriftgröße &lt;/p&gt;</pre>
<pre>&lt;table align="right" bgcolor="..."&gt; &lt;tr&gt; &lt;td align="..."&gt; ... &lt;/td&gt; &lt;td valign="..."&gt; ... &lt;/td&gt; &lt;/tr&gt; &lt;/table&gt;</pre>	<pre>&lt;table style="float: right; background-color: ..."&gt; &lt;tr&gt; &lt;td style="text-align: ..."&gt; ... &lt;/td&gt; &lt;td style="vertical-align: ..."&gt; ... &lt;/td&gt; &lt;/tr&gt; &lt;/table&gt;</pre>
<pre>&lt;hr size="12" width="40%" align="left" color="red"&gt;  &lt;hr size="8" width="50%" align="center" color="blue"&gt;</pre>	<pre>&lt;hr style="height: 12pt; width: 40%; float: left; background-color: red"&gt;  &lt;hr style="height: 8pt; width: 50%; clear: left; background-color: blue"&gt;</pre>

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

- left : Element steht links und wird rechtsseitig von nachfolgenden Elementen umflossen
- right : Element steht rechts und wird linksseitig von nachfolgenden Elementen umflossen
- none : Kein Umfluss (Standardeinstellung)

Mittels clear: lässt sich ein Umfluss abbrechen und die Fortsetzung unterhalb des umflossenen Elements oder Bereichs erzwingen. Folgende Angaben sind möglich:

- left : Fortsetzung unterhalb nach float: left
- right : Fortsetzung unterhalb nach float: right
- both : Fortsetzung unterhalb
- none : Keine Fortsetzung unterhalb (Standardeinstellung)

## 22 Publizieren im Netz

Das Problem der modernen Informationsgesellschaft ist nicht die Bereitstellung von Daten sondern ihre Entsorgung.

Ist die Erarbeitung eines Web-Projekts abgeschlossen und soll das Ergebnis der Öffentlichkeit präsentiert werden, steht noch die Übertragung (upload) aller verwendeten Dateien auf einen Server aus, der das WWW bedient. Internetprovider z. B.:

- 1&1 Internet AG
- Strato
- Host Europe GmbH

bieten Ihnen dazu kostenpflichtig eine eigene Adresse (Domain) sowie Speicherplatz an. Die Übertragung der zu Ihrem Web-Projekt gehörigen Dateien geschieht am zweckmäßigsten über eine FTP-Verbindung und mit entsprechender Software, z.B.:

Windows	MacOS
WS-FTP	Cyberduck
FTP Commander	Captain FTP

Die vorliegenden Programme, neben denen es noch eine Vielzahl weiterer, professioneller Angebote gibt, übertragen Dateien einfach per drop and drag (anklicken und ziehen). Dazu benötigen Sie Angaben zum gewählten Server, über einen möglichen Pfad unter dem Ihre Dateien auf diesem Server abgelegt sind, den User-Namen, der vom Betreiber des Servers vergeben wird, sowie ein Passwort, das die Zugangsberechtigung regelt.

Im folgenden sind zwei Server der Hochschule Mittweida aufgeführt:

ftp-home.hs-mittweida.de	Unix-Homebereich
ftp-www.hs-mittweida.de	Webpräsenz-Bereich

Sie bedienen Professoren, Mitarbeiter und Studenten der Hochschule; der Zugang erfolgt über den NCC-Username und das zugehörige Passwort. Jedem Nutzernamen, der vom Hochschulrechenzentrum vergeben wird, ist ein persönlicher, geschützter Speicherbereich des Hochschulservers zugeordnet.

**Beachten Sie:** Für Zugänge von Rechnern außerhalb des Hochschulnetzes sind nur die verschlüsselten Protokolle SFTP und FTPS freigegeben.

Erfolgt der Zugriff von einem Rechner außerhalb des Hochschulnetzes nicht über die erwähnten, sicheren Protokolle, so ist vor der Übertragung von Daten mittels FTP-Programm eine VPN-Verbindung (Virtual Private Network) aufzubauen.

**Achtung:** Die HTML-Datei, welche das gesamte Projekt öffnet muss die Bezeichnung:

**index.html** bzw. **index.htm**

haben, um als Hauptseite erkannt zu werden.

Web-Präsentationen, die auf dem Hochschulserver (s. o.) abgelegt sind, haben folgende Internetadresse:

<http://www.hs-mittweida.de/ordner/nutzername>

Die Bezeichnung ordner steht in diesem Zusammenhang beispielhaft für den konkreten Ordner, in dem die Datei index.html bzw. index.htm abgelegt ist.

Beachten Sie folgende Regeln bei der Übertragung Ihrer Daten auf den Server:

- Erzeugen Sie keine gebrochenen Links; übertragen Sie alle zum Projekt gehörenden Dateien auf den Webserver.
- Verwenden Sie nur relative Adressen. Behalten Sie unbedingt eine zum Entwurf **identische** Verzeichnisstruktur bei und verändern Sie in diesem Kontext nicht die Pfade zu Bildern oder Verweisen.
- Achten Sie bei Dateinamen auf Groß- bzw. Kleinschreibungen und **vermeiden** Sie **Leerzeichen** innerhalb eines Namens sowie **Sonderzeichen** wie z.B. deutsche Umlaute.
- Webdesigner stehen neuesten Browserversionen aufgeschlossener gegenüber als der konventionelle Surfer; überprüfen Sie Ihre Web-Präsentation vor einer Veröffentlichung im Netz mit Browsern, für die eine große Verbreitung angenommen werden kann.
- Testen Sie bei verschiedenen, gängigen Bildschirmauflösungen.
- Verwenden Sie niemals exotische Schriften, wenn zu erwarten ist, dass nur wenige Anwender diese auf ihrem Rechner installiert haben.
- Vermeiden Sie sehr kleine Schriftgrößen, die oft schwer zu lesen sind.
- Weniger ist mehr: Bieten Sie durchgängig ein klares, reduziertes Design an. Gestaltung allein ist noch keine Botschaft; erliegen Sie nicht der Versuchung von Farborgien, aufwendigen Grafiken oder unbegründeten Flashobjekten. Bedenken Sie, dass mögliche Nutzer Web-Seiten häufig nur dann akzeptieren, wenn dafür kurze Ladezeiten eingehalten werden können. Verzichten Sie also auf überflüssigen Schnick-Schnack. Nichts ist z. B. so lästig wie einführende Animationen, deren Sinn es ist, keinen zu haben.
- Verwenden Sie, wenn nicht unbedingt erforderlich, keine Komponenten wie Grafiken, Audio- oder Videodateien, die einen Speicherbedarf von deutlich mehr als 1 MB beanspruchen.
- Achten Sie bei der Übernahme fremder Inhalte auf Urheberrechte.

## 23 Literatur

Was die Professoren verdauen, fressen ihre Studenten.

Bücher HTML / CSS, JavaScript/JScript:

- Duckett J.: HTML & CSS - Erfolgreich Websites gestalten & programmieren, Pearson Design
- Hogan B.P., Fröhlich S.: HTML5 & CSS3 Webentwicklung mit den Standards von morgen, O'Reilly Verlag
- Prevezanos C.: Jetzt lerne ich HTML5, Markt+Technik Verlag
- Münz S., Gull C.: HTML5 Handbuch, Franzis-Verlag
- Potschien Denis: Pure HTML5 und CSS3, Franzis-Verlag
- Maurice F.: CSS3: Die neuen Features für fortgeschrittene Webdesigner, dpunkt-Verlag
- Agular R.R.: HTML und CSS, Praxisrezepte für Einsteiger, mitp-Verlag
- Haiges S., Spiering M.: HTML5-Apps für iPhone und Android, Francis
- Zillgens C.: Responsive Webdesign, Hanser Fachbuch
- Niederst R.J., Lichtenberg K.: Webdesign mit (X)HTML und CSS: Ein Praxisbuch zum Auffrischen und Vertiefen, O'Reilly Verlag
- Hauser T., Maurice F., Wenz C.: Das Website Handbuch: Programmierung und Design, Markt+Technik Verlag
- Chung D., Agular R. R.: HTML Ge-Packt, mitp-Verlag
- Meyer E.A.: CSS – kurz & gut, O'Reilly Verlag
- Steyer R.: Erfolgreich JavaScript lernen, Addison-Wesley
- Seeboerger-Weichselbaum M.: JavaScript Ge-Packt, mitp-Verlag

iPhone Apps:

- HTML5 Reference Guide (Evolonix)
- Mobile CSS Reference (Mike Murry)