

3. Zeichen und Schrift

3.1 Medien Zeichen, Text, Schrift

3.2 Mikro-Typografie: Zeichensätze

3.3 Makro-Typografie: Gestalten mit Schrift

3.4 Hypertext und HTML 

- Allgemeines 
- Textstrukturierung
- Tabellen
- Cascading Style Sheets
- Framesets
- Medieneinbettung

Literatur: <http://de.selfhtml.org/>

Hypertext im World Wide Web

- Verteilter Hypertext
 - Knoten können auf verschiedensten Rechnern weltweit liegen
- Gute Integration von Grafik, mäßige Integration anderer Medientypen
- Seitenbeschreibung (HTML) orientiert an linearem Text statt Objektorientierung
 - Technologisch seit ca. 1985 überholt!
- Stark eingeschränkte Interaktivität
 - Umständliche Zusätze, z.B. Skriptsprachen
 - Kein Autorenmodus für verteilten Zugriff
- Extremer Verbreitungsgrad, extreme Informationsdichte:
 - Datenvolumen Anfang 2001 ca. 20 TeraByte (TByte = 10^{12} Byte)
 - 1999 zwei durchschnittliche WWW-Seiten nur 19 Hyperlinks voneinander „entfernt“!

WWW, HTML und HTTP

- Standardisierungsgremien:
 - IETF (Internet Engineering Task Force), z.B. HTTP
 - W3C (WWW Consortium), z.B. HTML
- Grundprinzip von HTTP:
 - Client (*Browser*) schickt Anfrage (*request*) über IP-Verbindung an Server
 - » GET: Liefere Inhalt zu URL
 - » HEAD: Wie GET, aber ohne echte Lieferung der Daten (nur „Header“)
 - » POST: Akzeptiere im Rumpf mitgelieferte Daten
 - » Diverse „Header Codes“ in der Anfrage, z.B. Browsertyp, Host, Zeichensatz-Encoding, Sprachen, ...
 - Server schickt Antwort (*response*)
 - » Hauptinhalt: HTML-Code
 - » Header-Codes auch in der Antwort

HTML-Request: Beispiel

- The following HTTP request was received from IP address 141.84.8.6 (port 50048) by IP address 195.60.17.253 (port 80):

```
sGET /dumprequest.html HTTP/1.1
Host: djce.org.uk
User-Agent: Mozilla/5.0 (Macintosh; U; PPC Mac OS X; de-de)
AppleWebKit/416.11 (KHTML, like Gecko) Safari/416.12
Accept: */*
Accept-Encoding: gzip, deflate
Accept-Language: de-de
Referer: http://www.google.com/search?client=safari&rls=de-
de&q=http+request+example&ie=UTF-8&oe=UTF-8
Connection: keep-alive
```

- <http://djce.org.uk/dumprequest.html>

Auszeichnungssprache (Markup Language)

- Auszeichnungssprache (*markup language*) für Text
 - Text ergänzt um Angaben für die Darstellung
 - Verbreitetes Konzept; Beispiele für andere Auszeichnungssprachen: LaTeX, RTF (Rich Text Format)
- Vergleich LaTeX/HTML
 - LaTeX-Beispiel:

```
\paragraph{"Überschrift}
Text text {\it kursiver Text}
\begin{itemize}
\item Punkt in Aufz"ahlung
\end{itemize}
```
 - HTML-Beispiel:

```
<P>&Uuml;berschrift<BR>
Text text <I>kursiver Text</I>
<UL>
<LI>Punkt in Aufz&auml;hlung
</UL>
```

Trennung Inhalt – Darstellung

- Abstraktionsebene der Auszeichnung:
 - Entweder: „Fett 14pt“ (Mischung Inhalt-Darstellung)
 - Oder: „Überschrift Ebene 1“ (Trennung Inhalt-Darstellung) (mit separater Festlegung der Darstellung, z.B. Fett 14 pt)
- Vorteile einer starken Trennung Inhalt-Darstellung:
 - Leichtere Wartbarkeit (Regeln für die Darstellung einer Auszeichnungsklasse nur einmal definiert)
 - Bessere Plattformunabhängigkeit
 - » Konkrete optische Umsetzung („Rendering“) weitgehend der darstellenden Hardware/Software überlassen
 - Impliziter Zwang zur stilistischen Einheitlichkeit in der Darstellung
- Nachteile:
 - Verlust der Detailkontrolle über die Darstellung
 - Verlust von Flexibilität für Sonderfälle

Hypertext Markup Language HTML

- Geschichte:
 - 1969, Goldfarb, Mosher, Lorie (IBM): „Generic Markup Language“ (GML)
 - 1978, Standardisierung von GML durch ISO als „SGML“ (Standard Generic Markup Language“)
 - 1989, Tim Berners-Lee / Robert Cailleau: HTML
 - » Starke Einschränkung von SGML (spezieller Dokumententyp)
 - 1993, NCSA Mosaic Browser
 - 1999, Version 4 von HTML
 - 2000, XHTML (HTML 4.01 in XML, siehe später)
- Leistungsumfang von HTML:
 - Textattribute für die Darstellung festlegen
 - Spezielle Textformatierungen (z.B. Tabellen) definieren
 - Teile der Darstellungsoberfläche für interaktive Benutzereingaben vorsehen
 - Weitere Dokumente verschiedenster Art an beliebigen Stellen des Web einbinden

HTML-Syntax

`<TAG [ATTRIBUT = "wert"]* > Inhalt [</TAG>]`

- Tags:
 - Paarweise als Beginn-/Ende-Paar `<TAGX> ... </TAGX>`
 - Einzeln z.B. `
`
- Attribute:
 - Zulässige Attribute abhängig vom konkreten Tag
 - » Fast immer zulässig: CLASS, ID, LANG, STYLE
 - Attributwerte:
 - » In vielen Fällen ohne Anführungszeichen angebar (z.B. Zahlen)
 - » Stilistisch guter HTML-Code benutzt immer Anführungszeichen
- Zeilenumbrüche, mehrfache Leerzeichen, Tabulatoren i.A. ignoriert
- Kommentare: `<!-- ... -->`

Einfaches HTML-Beispiel

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/REC-html41/loose.dtd">
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Einfaches Beispieldokument HTML</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
Ein ganz einfacher Beispieltext. <BR>
```

```
<FONT FACE="Helvetica">Fontumschaltung Helvetica</FONT> <BR>
```

```
<FONT FACE="Times">Fontumschaltung Times</FONT> <BR>
```

```
<B>Fett</B> <I>Kursiv</I>
```

```
</BODY>
```

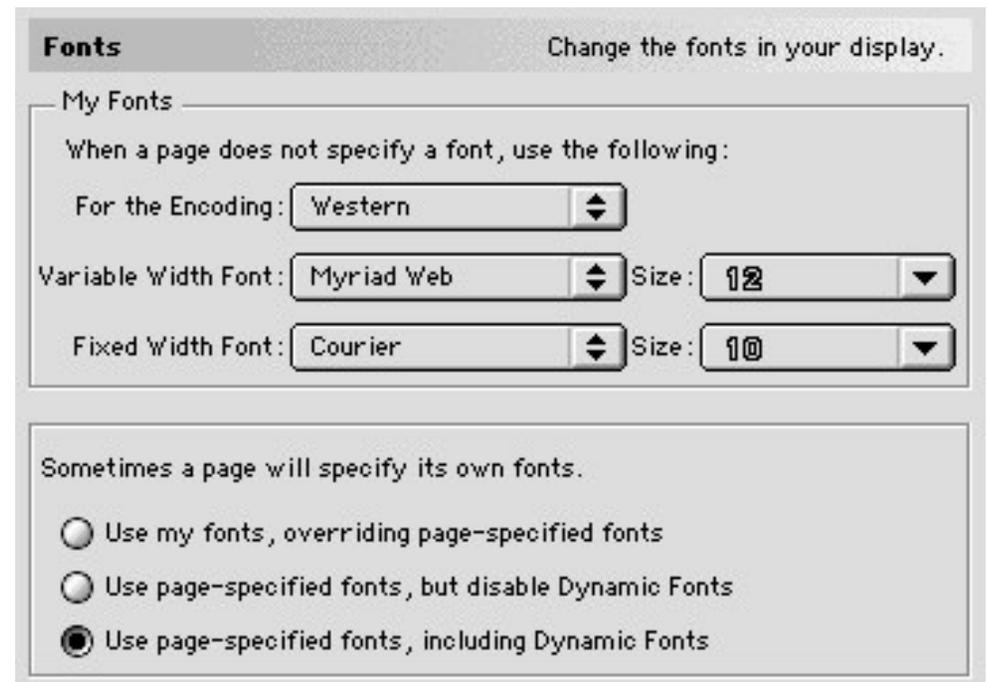
```
</HTML>
```

Datei: HTML1.HTML

Trennung Inhalt-Darstellung in HTML

- Starke Trennung:
 - bei Verwendung vordefinierter Textklassen
 - » z.B. <H1> für Überschriften, <ADDRESS> für Adressen
 - bei Verwendung von Cascading Style Sheets (sh. später)
- Schwache Trennung:
 - Bei expliziter Auszeichnung z.B. mit

- Prinzipiell ist in HTML keine vollständige Kontrolle über die Darstellung möglich.
 - Allerdings eine weitgehende Kontrolle für Standard-Plattformen und – Einstellungen



Dokumenttyp

- Verschiedene Versionen von HTML
 - Angabe benutzter Version mit DOCTYPE
 - In heutigen Browsern meist nicht überprüft!
 - Derzeit aktuelle Version: HTML 4.01 mit drei Varianten
- Strikt:
 - Modernes HTML, nicht mit alten Browsern kompatibel (vor Version 4.x)
 - Verwendung von Stylesheets und Style-Attributen

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```
- Transitional:
 - Auch ältere Konstrukte zulässig (z.B. zur Textausrichtung)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```
- Frameset:
 - Spezielle Angabe für Frameset-Dateien (sh. später)

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

Kopfeinträge

- Titel

```
<title> ... </title>
```

- Fenstertitel im Browser, Bezeichnung in Bookmarks, Anzeige bei Suchmaschinen

- Meta-Angaben für den Browser und Suchmaschinen:

```
<meta name="author" content="Heinrich Hussmann">
```

```
<meta name="description" content="Beispieldatei ...">
```

```
<meta name="keywords" content="HTML, HEAD">
```

```
<meta name="date" content="2003-04-30">
```

```
<meta name="robots" content="noindex"> (Auslesen verbieten)
```

- oder auch Angaben nach dem sog. "Dublin Core"-Schema (<http://dublincore.org>):

```
<meta name="DC.creator" content="Heinrich Hussmann">
```

- Meta-Angaben für den Web-Server und den Browser:

- Basis-Zeichensatz:

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

- Zeitpunkt für das Löschen aus "Proxy-Servern":

```
<meta http-equiv="expires"
```

```
content="Sat, 15 Dec 2001 12:00:00 GMT">
```

```
<meta http-equiv="expires" content="43200"> (Zeit in Sekunden)
```

Sonderzeichen

- Zeichen können auf drei Weisen angegeben werden:
 - Direkter Zeichencode (Zeichensatz des Editors)
 - Unicode-Angabe, z.B. `®` (®), `€` (€)
 - Explizite Namen, z.B. `®`, `€`

- Wichtige Namen für deutsche Sonderzeichen:

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

- Sonderzeichen der HTML-Syntax "maskieren":

<code>&lt;</code>	<code>&gt;</code>	<code>&amp;</code>	<code>&quot;</code>	<	>	&	"
-----------------------	-----------------------	------------------------	-------------------------	---	---	---	---

HTML-Editoren

- Software-Produkte zum bequemen Erstellen von HTML-Seiten ohne direkte Nutzung von HTML:
 - z.B. Adobe GoLive, Macromedia DreamWeaver, Microsoft FrontPage
- Vorteile:
 - Erlauben direktere Beurteilung des grafischen Effekts
 - Ersparen viele Unannehmlichkeiten von HTML
- Nachteile:
 - Gefahr der Vernachlässigung des entstehenden HTML-Codes
 - "Verunreinigen" manchmal den Code durch Editor-Artefakte
- Empfehlung:
 - Nur verwenden, wenn HTML und entstehender Code voll verstanden
 - HTML-Code-Ansicht der Werkzeuge benutzen

3. Zeichen und Schrift

- 3.1 Medien Zeichen, Text, Schrift
- 3.2 Mikro-Typografie: Zeichensätze
- 3.3 Makro-Typografie: Gestalten mit Schrift
- 3.4 Hypertext und HTML
(Fortsetzung) 

- Allgemeines
- Textstrukturierung 
- Tabellen
- Cascading Style Sheets
- Framesets
- Medieneinbettung

Weitere Informationen: <http://de.selfhtml.org/>

Elemente zur Strukturierung des Texts

- Überschriften `<h1>...</h1>` ... `<h6>...</h6>`
- Absätze `<p>...</p>`
- Unnummerierte Listen ` list item 1 `
- Nummerierte Listen ` list item 1 `
- Definitionslisten `<dl> <dt>term</dt> <dd>defn</dd> </dl>`
- Zitate `<blockquote> Zitattext </blockquote>`
- Adressen `<address> Adreßtext </address>`
- Vorformatierter Text `<pre> z.B. Programmtext </pre>`
 - Dicktengleiche (Nicht-Proportional-)Schrift
 - Umbruch und Leerzeichen wie in der HTML-Datei
- Trennlinie `<hr>`

Zeilenumbruch

- Zeilenumbruch erzwingen `
`
- Zeilenumbruch verhindern `<nobr>`
- "Geschütztes" Leerzeichen
(*non-breaking space*) ` `
- Zeilenumbruch im Wort erlauben
(*work break*) `<wbr>`

Logische Auszeichnungen im Text

- Inhaltliche Beschreibung der Art des Textstücks
 - Konkrete Formatierung separat festgelegt

- Auszeichnungen:

– Betont	<code> ...</code>	<code></code>
– Stark betont	<code> ...</code>	<code></code>
– Quelltext	<code><code> ...</code>	<code></code></code>
– Beispiel	<code><samp> ...</code>	<code></samp></code>
– Tastatureingabe	<code><kbd> ...</code>	<code></kbd></code>
– Variable	<code><var> ...</code>	<code></var></code>
– Zitat	<code><cite> ...</code>	<code></cite></code>
	<code><q cite="quelle"> ...</code>	<code></q></code>
– Definition	<code><defn> ...</code>	<code></defn></code>
– Akronym	<code><acronym> ...</code>	<code></acronym></code>
– Abkürzung	<code><abbr> ...</code>	<code></abbr></code>

Physische Auszeichnungen im Text

- Beschreibung der konkreten Formatierung des Textstücks

- Auszeichnungen:

– Fett	<code> ...</code>	<code></code>
– Kursiv	<code><i> ...</code>	<code></i></code>
– Schreibmaschine	<code><tt> ...</code>	<code></tt></code>
– Unterstrichen	<code><u> ...</code>	<code></u></code>
– Durchgestrichen	<code><s> ...</code>	<code></s></code>
– Größer	<code><big> ...</code>	<code></big></code>
– Kleiner	<code><small> ...</code>	<code></small></code>
– Hochgestellt	<code><sup> ...</code>	<code></sup></code>
– Tiefgestellt	<code><sub> ...</code>	<code></sub></code>

Verweise (Links)

- Klassischer Hypertext-Verweis
 - Markierter *Anker* im Text
 - *Referenz* auf andere HTML-Datei
- Syntax:
` Text `
- Beschreibung des Ziels
 - Vollständige URI (sh. nächste Folie)
 - Absolute Adressierung auf gleichem Rechner
``
 - Relative Adressierung auf gleichem Rechner
``
 - Adressierung spezieller Stellen in der Zielseite (sh. übernächste Folie)

Uniform Resource Identifier (URI)

- Offiziell: Oberbegriff von *Uniform Resource Locator (URL)* und *Uniform Resource Name (URN)*
- In der Praxis:
 - URN kaum benutzt
(obwohl hilfreiche Trennung zwischen logischer und physischer Adresse)
 - URI = URL
- Syntax:
Protokoll : / lokalerNetzwerkname / Hostname : Port / Pfad
`http://www.mimuc.de/`
`http://Arbeitsgruppe/www.mimuc.de:8080`
`/usr/local/data/index.html`

Zielgenaue Verweise: Dokumentinterne Anker

- Hinter jeder Verweisadresse kann (mit # abgetrennt) eine Stelle in dem adressierten Dokument spezifiziert werden.
- Deklaration des Zielankers (z.B. in xyz.html):
`Text`
- Ansprechen des Zielankers:
`Text`

Stilistische Anmerkungen zu Verweisen

- Guter Stil:
 - Ankertext hat inhaltliche Bedeutung
- Beispiele:
 - Gut:
"Es steht auch vertiefende Information für Sie bereit."
 - Schlecht:
"Für vertiefende Information klicken Sie hier."

 - Gut:
"Zurück zur Institutsseite"
 - Schlecht:
"back"

3. Zeichen und Schrift

- 3.1 Medien Zeichen, Text, Schrift
- 3.2 Mikro-Typografie: Zeichensätze
- 3.3 Makro-Typografie: Gestalten mit Schrift
- 3.4 Hypertext und HTML
(Fortsetzung) 

- Allgemeines
- Textstrukturierung
- Tabellen 
- Cascading Style Sheets
- Framesets
- Medieneinbettung

Weitere Informationen: <http://selfhtml.teamone.de>

Tabellen (1)

- Aufteilen der Fläche in Zeilen und Spalten in flexibler Weise
 - Klassische Tabellen, Matrizen
 - Allgemeines Hilfsmittel zum Layout (bei unsichtbar gemachten Trennlinien)
 - Achtung: Tabellen werden meist erst nach vollständigem Laden angezeigt
- Allgemeine Tabellenform:

<code><table></code>				
<code><tr></code>	<code><th></code> <code></th></code>	<code><th></code> <code></th></code>	<code><th></code> <code></th></code>	<code></tr></code>
<code><tr></code>	<code><td></code> <code></td></code>	<code><td></code> <code></td></code>	<code><td></code> <code></td></code>	<code></tr></code>
<code><tr></code>	<code><td></code> <code></td></code>	<code><td></code> <code></td></code>	<code><td></code> <code></td></code>	<code></tr></code>
				<code></table></code>

Tabellen (2)

- Vordefinition der Spaltenbreite (schnellere Anzeige!)
 - `<colgroup> <col width=...> ... </colgroup>`
- Unregelmässige Zellen einer Tabelle
 - Zelle über mehrere Spalten: Attribut `colspan="n"` in `<th>` und `<td>`
 - Zelle über mehrere Zeilen: Attribut `rowspan="n"` in `<th>` und `<td>`
- Rahmen
 - Attribut `border="n"` in `<table>`
- Abstände
 - Abstand Rahmen-Zellen: Attribut `cellspacing="n"` in `<table>`
 - Abstand zwischen Zellen: Attribut `cellpadding="n"` in `<table>`
- Textformatierung, Ausrichtung etc.
 - Spezielle Attribute (z.B. `align`)
 - Cascading Style Sheets (sh. unten)