

# Digitale Medien

3. MARKUP LANGUAGES ALLGEMEIN, HTML, CSS

## Wo stehen wir?

2

- Vergangenen Vorlesung: Text und Typographie: Gestaltung eines Druckwerks nach ästhet. Gesichtspunkten, u.a. Wahl der Schrifttypen, Anordnung des Satzes und der Bebilderung
- Daten beinhalten auch eine Strukturierung,
  - Datensätze, die eine Adresse beschreiben bestehen aus Name, Vorname, Straße, Wohnort; die einzelnen Teile werden in Bitfolgen kodiert, wie aber soll die Struktur dargestellt werden?
  - Texte bestehen aus Kapiteln, Abschnitten, Sätzen, ...; wie die Struktur darstellen?
  - interne Kodierung in Bitfolgen vs. Kodierung bei der Eingabe? Menschen müssen Daten eingeben und auch lesen können. Maschinen müssen die Daten verarbeiten können.
- eine Möglichkeit: Markup-Languages (Auszeichnungssprachen)

## Webtechnologien

3

- **HTML:** Lingua Franca des Web, die Grundlage zum Darstellen von Hypertext und Medientypen. Gibt jedoch nur an was dargestellt wird und nicht wie!
- **CSS (Cascading Style Sheets):** Formatierungsangaben für beliebige HTML Elemente
- **HTML+CSS** sind vergleichbar mit einer festen Dokumentenvorlage (vgl. Word). Für bestimmte Anwendungsfälle ist es jedoch sinnvoll, beliebige Vorlagen definieren zu können. Dies erlaubt XML und XSL.
- **JavaScript:** Scriptsprache, die die dynamische Veränderung von Webseiten erlaubt.

## Markup-Language

4

- Markup: Textauszeichnung
- daher: Markup-Language = Auszeichnungssprache
- dient zur Beschreibung von Daten und teilweise des Verfahrens, das zur Bearbeitung dieser Daten nötig ist
- im Drucksatz: Anweisungen an den Schriftsetzer, bestimmte Teile hervorzuheben, sie also *auszuzeichnen*.
- Markup-Sprache:
  - Eine Sprache, die ein System von Markierungen definiert.
  - Bei einer Auszeichnungssprache werden die Eigenschaften, Zugehörigkeiten und Verfahren von bestimmten Wörtern, Sätzen und Abschnitten eines Textes beschrieben bzw. zugeteilt, meist in dem sie mit *Tags* markiert werden.
  - Strukturierung von Daten kann ebenfalls auf dieser Art und Weise beschrieben werden

## Markup-Language

5

- Unterteilung der Markup-Languages in:
  - Descriptive Markup Languages
    - beschreiben Daten und Struktur der Daten
    - HTML, DocBook, TEI, MathML, WML oder SVG, LaTeX
  - Procedural Markup Languages
    - beinhalten ebenfalls Beschreibungen zur Verarbeitung von Daten bzw. Darstellung von Dokumenten (Programme)
    - TeX, PDF, PostScript
- Betrachtung von Markup-Languages zunächst im Kontext von Dokumenten

## Standard Generalized Markup Language

6

- 1969 wurde GML (Generalized Markup Language) von Goldfarb, Mosher und Lorie bei IBM entwickelt
- weiterentwickelt und standardisiert → SGML = Standard Generalized Markup Language
- Dokument sollte plattform- und Anwendungs-unabhängig beschrieben werden
  - Trennung von Inhalt und Layout
  - Kennzeichnung der logische Struktur (Überschriften, Abschnitte, Kapitel)

## SGML – Beispiel

7

- Die Einheiten der Daten/des Textes werden durch *Tags* beschrieben  
Ein **<kursiv>hervorgehobenes</kursiv>** Wort.
- im Browser (dem zur Anzeige des Quelltextes bestimmten Programm) wird das Ganze dann so dargestellt:  
Ein **hervorgehobenes** Wort.
- Problem: Browser muss die entsprechenden SGML-Befehle (Tags) verstehen.
- SGML sehr allgemein anwendbar und daher sehr breit ausgelegt, daher speziellen Markup-Sprachen für spezielle Anwendungen
  - MathML zur Darstellung mathematischer Formeln
  - SVG als Format zur Beschreibung von Vektorgraphiken
  - HTML für verteilte und untereinander verlinkte Text-Dokumente – Hypertext Markup Language

## von SGML zu HTML

8

- HTML – HyperText Markup Language
- stellt eine Anwendung von SGML dar
- benutzt teilweise gleiche Befehle und gleiche Syntax  
→Marken (Tags) in spitzen Klammern <tag>
- zwei Arten von Tags zur Kennzeichnung von :
  - Stellen im Text: <br>
  - Bereichen im Text: <em>Bereich</em>

## HTML Grundlagen Trennung von Inhalt und Layout bzw. Struktur

9

- Verwendung von Tags zur Beschreibung der Dokumentteile und der Strukturierung
- Aussehen kann von Browser zu Browser variieren, je nach Interpretation der Tags durch den Browser
- Beispiel: <html></html>, <em></em>, <i></i>

## Elementare Dokumentgestaltung

10

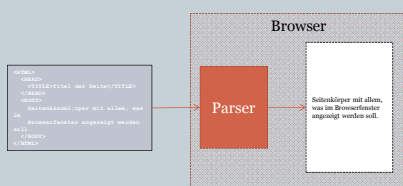
- Grundgerüst eines HTML-Dokumentes

```
<HTML>
  <HEAD>
    <TITLE>Titel der Seite</TITLE>
  </HEAD>
  <BODY>
    Seiten&ouml;rper mit allem, was im
    Browserfenster angezeigt werden soll.
  </BODY>
</HTML>
```

## Darstellung von HTML-Dokumenten

11

- Browser parst HTML-Dokument, jeder
- HTML-Parser: Software, die HTML-Tags kennt und interne Datenstruktur aufbaut.
  - Führt auch Fehlerkontrolle durch



## Zeichenkodierung

12

- Vorlesung „Informationen und ihre Darstellung“: ASCII, Unicode
- HTML-Editor: Zeichen → Byte
- HTML-Browser: Byte → Zeichen
- Problem: Codierung kann unterschiedlich sein
- ausser/vor Unicode, regionenspezifische Standards
  - ISO-8859 alle europäischen Sprachen
  - ISO-8859-1 (westeuropäisch, amerikanisch)
  - ISO-8859-2 (mitteleuropäisch, slawisch)
  - ISO-8859-3 (Esperanto, Galizisch, Maltesisch und Türkisch)
- dem Browser muss mitgeteilt werden, welche Kodierung verwendet wird
- fehlt die Angabe, wird ein Standard verwendet

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

## Erstellen und Editieren von HTML-Dokumenten

- Zeilenumbrüche werden nicht berücksichtigt
  - HTML-Element `<br>` verwenden
- Tabulatoren werden nicht berücksichtigt.
  - Vorteil: Tabulatoren und Zeilenumbrüche können für die Formatierung des Quelltextes verwendet werden.
- Zeilenumbruch, Tabulator und mehrere Leerzeichen sind sog. Weißraum-Zeichen
  - Werden zu einem Leerzeichen zusammengefasst
- Wenn mehrere Leerzeichen gewünscht -> Sonderzeichen (geschütztes Leerzeichen) einsetzen

## Erstellen und Editieren von HTML-Dokumenten

- Jeder beliebige Texteditor, der Unicode erzeugt, ist nutzbar:
  - Windows: PSPad, jEdit, Notepad
  - Mac: BBEdit, TextWrangler, jEdit
- Vorteil: Syntaxhervorhebung

## Tags

16

- Markierungen, die Layout oder Strukturierung des Dokumentes darstellen
- Tags stehen in spitzen Klammern
- bei HTML: Groß-/Kleinschreibung spielt keine Rolle
- Bereiche spezifizierende Tags
  - Start: `<tag_name>`
  - Ende: `</tag_name>`
  - Beispiel: `<html>...</html>` oder `<body>...</body>`
  - Ende-Kennzeichnungen können bei einigen Tags wegfallen
- Positionen spezifizierende Tags
  - wie Start-Tags
  - Beispiele: `<hr>` oder `<br>`

## HTML-Elemente

17

- Markierungen, die Layout oder Strukturierung des Dokumentes darstellen
- Ein HTML-Element ist der Teil eines HTML Dokuments der durch einen Start- und EndTag begrenzt wird.
  - Incl. Start- und EndTag
  - Inhalt des Elements: Alles zwischen Start- und EndTag

```
<tag>  
  Inhalt des HTML-Elements  
</tag>
```

} HTML-Element

## Attribute

18

- Oft haben Elemente bestimmte Eigenschaften, Beispiele:
  - Bild hat eine Quelle
  - Bilder haben eine alternative Beschreibung
  - Wichtig für die Barrierefreiheit
  - Listen sollen mit einem bestimmten Zahlenwert beginnen
  - Links haben ein Ziel auf das sie verweisen
  - Der Hintergrund einer Seite hat eine Farbe (nicht Weiß)
- Eigenschaften werden als Attribut angegeben:
  - `<tagname attribut=„Wert“>`

## Strukturierung von Text

19

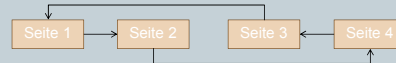
- Überschriften (heading):  
`<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` und `<h6>`
- Absätze: `<p>`
- Zeilenumbrüche: `<br>`

## Beispiel

```
<HTML>
<HEAD>
<TITLE>Vorlesung Digitale Medien</TITLE>
</HEAD>
<BODY>
<H1>Digitale Medien</H1>
<p>
Für die Vorlesung Digitale Medien ist
keine Anmeldung erforderlich, sie steht
allen offen.
</p>
</BODY>
</HTML>
```

## Hypertext

- Hypertext ist die Präsentation von Dokumenten als ein Netz von Knoten, welche durch Links miteinander verbunden sind.
- NutzerInnen können diese Dokumente frei (nicht-linear) durch folgen von Links besuchen



Hußmann, Uni München: <http://www.medien.fh-lmu.de/lehre/ws0708/dmi/dm3c.pdf>, [Henning, 2003]

## Hypertext

- Begriff wurde von Ted Nelson 1965 geprägt:
  - ACM Paper mit dem Begriff „Hypertext“
- Ursprüngliche Idee geht auf Vannevar Bush zurück:
  - Artikel: „As we may think“ (The Atlantic Monthly), 1945
  - Zukünftiges System Memex (Memory Extender)
    - Analoges Computersystem zur Speicherung, Ansicht und Verknüpfung von Dokumenten

Hußmann, Uni München: <http://www.medien.fh-lmu.de/lehre/ws0708/dmi/dm3c.pdf>,  
<http://www.ps.uni-eb.de/~duchier/pub/vbush/vbush-all.shtml>

## Hypertext Geschichtliches

- 1945 Vannevar Bush konzipiert Memex
- 1963 Douglas Engelbart stellt NLS, ein System zur Strukturierung von Texten vor,
  - betont Baumstruktur
  - erlaubt Hyperlinks
- 1965 Ted Nelson erfindet Xanadu eine HTML-ähnliche Sprache und verwendet den Begriff Hypertext  
<http://www.xanadu.com.au/ted/XUsurvey/xuDation.html>
- 1969 Hypertext Editiersystem von Nelson & Van Dam
  - Kennt Multimediale Dokumente
- 1975 ZOG verteiltes Hypermedia System für militärische Zwecke
  - Später KMS: Knowledge Management System
- 1987 Apple: HyperCard
- 1989 Tim Berners-Lee und Robert Cailleau erfinden WWW mit Beschreibung durch HTML

## Hypertext - Knoten

- Dokument, das entweder eine Kombination mehrerer Multimedia-Objekte oder ein einzelnes Medien-Objekt enthält
- → Medienobjekte sind entweder in Knoten integriert oder stellen selbst Knoten dar
- Innerhalb eines Knotens können wiederum Verweise auf andere Knoten enthalten sein
- Andere Bezeichnungen für Knoten:
  - Seite, Page, Frame, Karte, Screen

Hußmann, Uni München: <http://www.medien.fh-lmu.de/lehre/ws0708/dmi/dm3c.pdf>

## Hypertext – Verweise

- Verweise (Links) sind gerichtet: Quellknoten → Zielknoten
- Assoziation zwischen zwei Knoten
- Region, innerhalb des Knotens, der ausgewählt werden kann: Anker
- Arten von Verbindungen:
  - zu einem Knoten
  - zu einem Punkt innerhalb eines Knotens

## Verweise und Bilder

26

- Stärke von HTML sind Verweise:
- Markierung des Bereiches der als Verweis dient
  - `<A></A>`, A steht für Anker (im Text)
  - Zieladresse des Links als Parameter:
    - `<A HREF=„http://www.ovgu.de“>Link im Text</A>`
- Bilder einbinden:
  - Markierung der Stelle, an der das Bild eingefügt werden soll: `<IMG>`
  - Adresse des Bildes als Parameter
  - `<IMG SRC=„http://...“>`
  - Alt-Attribut für alternative Beschreibung



```
<img src=„http://upload.wikimedia.org/wikipedia/de/3/3d/FuBK-Testbild.png“ alt=„Testbild“>
```

## Tabellen

27

- Aufbau einer Tabelle:

```
<table border="1">
<tr>
<th>Berlin</th>
<th>Hamburg</th>
<th>München</th>
</tr>
<tr>
<td>Miljö</td>
<td>Kiez</td>
<td>Bierdampf</td>
</tr>
<tr>
<td>Buletten</td>
<td>Frikadellen</td>
<td>Fleischpflanzerl</td>
</tr>
</table>
```

Berlin	Hamburg	München
Miljö	Kiez	Bierdampf
Buletten	Frikadellen	Fleischpflanzerl

- Tabellen sind hinsichtlich der Barrierefreiheit problematisch,
- Tabellen haben eine Semantik

## Listen

28

- geordnete Listen:

```
<ol>
<li>erster Punkt</li>
<li>zweiter Punkt</li>
</ol>
```

1. erster Punkt
2. zweiter Punkt

- ungeordnete Listen:

```
<ul>
<li>erster Punkt</li>
<li>zweiter Punkt</li>
</ul>
```

- erster Punkt
- zweiter Punkt

## Sonderzeichen in HTML

29

- Seit HTML 4.0, jedes Zeichen aus Zeichenvorrat nach ISO 10646 (Unicode) benutzbar.
- Erzeugung durch spezielle numerische Notation nach Schema:
  - `&#[x][Nummer];`
- Beispiele einer Notation in HTML: `&#945;` & `&#x3B1;`
- Alternativ: `&[Name];`
- Beispiele: `&Auml;` & `&Omega;` & `&euro;`

<http://de.seithtml.org/html/referenzzeichen.htm>

## Beispiele Attribute

32

- Beispiel geordnete Listen Startwert setzen:

```
<ol start="3">
<li>erster Punkt</li>
<li>zweiter Punkt</li>
</ol>
```

3. erster Punkt
4. zweiter Punkt

- Beispiel Hintergrundfarbe setzen:

```
<body bgcolor="#FF9933">
```



## iFrames

33

- InlineFrame, erlaubt, Seiten in Seiten zu integrieren

```
<HTML>
<HEAD>
<TITLE>Titel</TITLE>
</HEAD>
<iframe src =„HTML-Seite.html“ width="100%" height="300">
<p>Your browser does not support iframes.</p>
</iframe>
</HTML>
```

## Weitere interessante HTML-Elemente

- **Strukturierung von Text:**
  - Zitate: `<blockquote> </blockquote>`
  - Adressen: `<address> </address>`
  - Vorformatierter Text: `<pre></pre>`
  - Trennlinie: `<hr>`
- **Zeilenumbrüche:**
  - erzwingen: `<br>`
  - verhindern: `<nobr>`
  - geschütztes Leerzeichen: `&nbsp;`

## Editoren für HTML

- <http://www.online-html-editor.de/>
- Amaya: <http://www.wssexpert.de/amaya/>
- Dreamweaver:  
<http://www.adobe.com/de/products/dreamweaver>
- Homepage: <http://www.adobe.com/products/homepage/>
- PSPad: <http://www.pspad.com/de>

## Zusammenfassung HTML

36

- Markup Language zur Beschreibung von Dokumenten üblicherweise für WWW
- Beschreibt Struktur des Dokumentes, Layout wird vom Browser umgesetzt:
  - z.B. können Einrückungen, Absatzabstände oder ähnliches nicht spezifiziert werden
  - HTML auch nicht dafür gedacht! Keine Layoutsprache!
- Stärke: Links, daher auch *Hypertext* Markup Language
- heute: fast universell einsetzbar, enthält viele Elemente, um Dokumente umfassend zu beschreiben.
- Layout erfolgt über Stylesheets (siehe später)
- Syntax relativ locker, daher weitere Entwicklung: XHTML

## Literatur

- Selfhtml: <http://de.selfhtml.org>
- HTML3.2-Einführung:  
<http://www.boku.ac.at/html3.2/einf/hein.html>
- <http://de.html.net/tutorials/html/>