

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 zu Hypertext ←
- HTML
- Textstrukturierung
- Tabellen
- Cascading Style Sheets
- Strukturierte Seiten
- Medieneinbettung

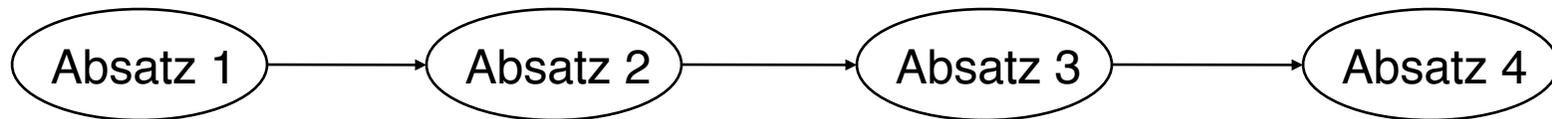
Literatur:
Medieninformatik-Buch:
Kapitel 10



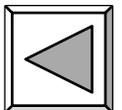
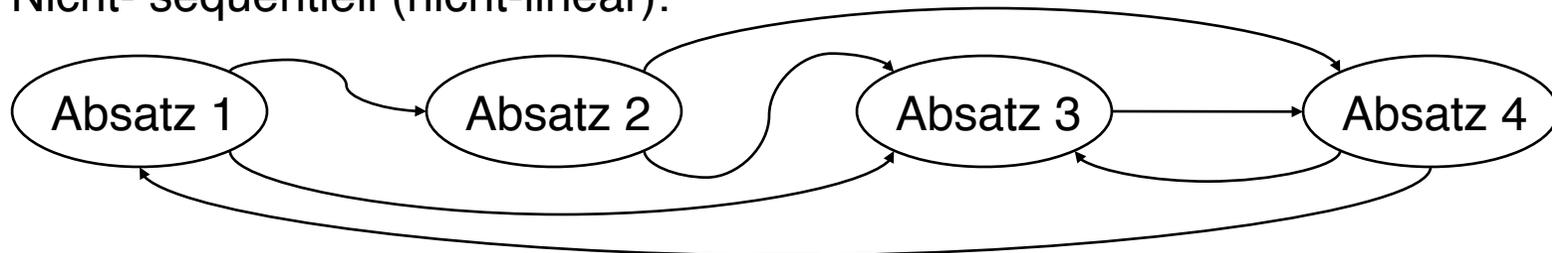
Hypertext

- *Hypertext*: Die Präsentation von Information als ein Netz von verbundenen Knoten, in dem der Leser frei, d.h. in nicht-linearer Reihenfolge navigieren kann.
- Der Begriff *hypertext* wurde von Ted Nelson geprägt, der es in seinem selbstverlegten Buch „Literary Machines“ als „nicht-sequentielles Schreiben (non-sequential writing)“ bezeichnet.

Sequentiell (linear):



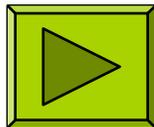
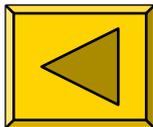
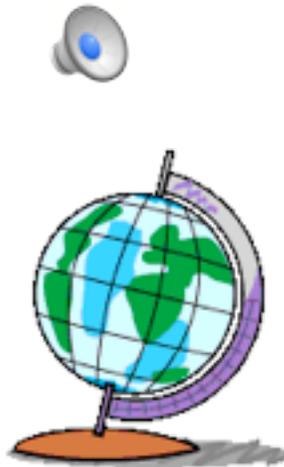
Nicht- sequentiell (nicht-linear):



Information in Hypertext-Knoten

- In strengem Wortsinn: Textstück, evtl. mit Abbildungen
 - Klein genug, um eigenständige Informationseinheit zu bieten
 - Meist auf eine Seite des Anzeigegeräts passend
- In erweiterten Definitionen („Hypermedia“):

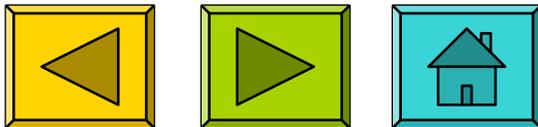
- Klänge
- Filmstücke
- Animationen
- ...



Hypertext-Knoten

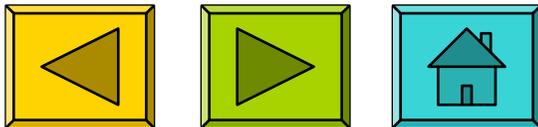
- *Knoten*: Datenstruktur
 - erlaubt es, einen in sich geschlossenen Informations-Inhalt abzulegen
 - ermöglicht Verbindungen zu weiteren Knoten

- Andere Bezeichnungen für das Konzept des Hypertext-Knotens:
 - *frame*
 - *work space*
 - *card*
 - *lexia*
 - *web page*



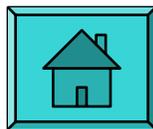
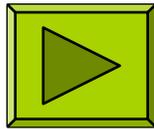
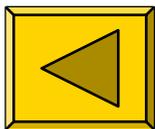
Verbindung in Hypertext

- *Verbindung (link)*: Durchlaufbare Assoziation zwischen zwei Knoten
- *Anker*: Sichtbare Region, die mit einem Eingabegerät ausgewählt werden muss, um die Verbindung zu aktivieren
 - In den meisten Systemen dürfen sich Anker nicht überlappen.
- Detaillierungsgrad des Verbindungsziels:
 - Einfache Verbindungen: Von Knoten zu Knoten
 - Zielgenaue Verbindungen: Auswahl eines bestimmten Teils der Information im Ziel-Knoten



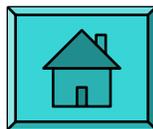
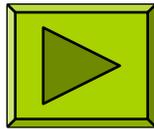
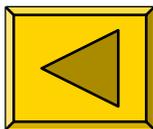
Geschichte von Hypertext

- ca. 1500, Erasmus von Rotterdam: Seitenzahlen für Querverweise in Büchern
- 1945, [Vannevar Bush: Memex](#)
- 1963, Doug Engelbart: NLS / Augment, Baumstruktur von Texten
- 1965, [Ted Nelson: Xanadu](#)
- 1975, Akscyn / McCracken (CMU): ZOG, später KMS (Knowledge Management System)
- 1976-1980, Allan Kay, Adele Goldberg, H.H. Ingalls (Xerox PARC): Objektorientierte Programmierung mit „Smalltalk“
- 1987, Bill Atkinson (Apple): [HyperCard](#)
- 1989, Tim Berners-Lee / Robert Cailleau (CERN): HTML / [WWW](#)



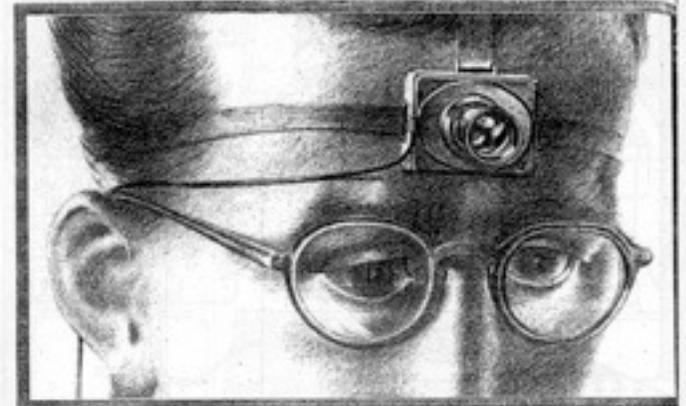
Ted Nelson und Xanadu

- Ted Nelson, geboren 1937, Soziologie-Student mit Nebenfach Informatik
- Ca. 1960, Studienarbeit: Idee für ein fortgeschrittenes interaktives Textverarbeitungssystem
- 1965: ACM-Jahrestagung, Papier mit dem Begriff „Hypertext“
- 1974: Buch „Dream Machines“ verweist klar auf frühere Visionen von Vannevar Bush
- Xanadu:
 - Benannt nach dem Gedicht „Kublai Khan“ von Coleridge, Palast in der Mongolei
 - » Coleridge sagt, Gedicht sei unvollständig wegen einer Unterbrechung
 - Idee: Magischer Ort von Freiheit und Gedächtnis, nichts wird vergessen
 - Xanadu-Software:
 - » Freigabe-Ankündigungen: 1974 -> 1976, 1987 -> 1988, 1988 -> 1991
 - » Ab 1992: Firmen XOC und Udanax
 - » Seit 1999 OpenSource (www.xanadu.com): 3D und bidirektionale Links



Vannevar Bush und Memex

- Vannevar Bush (1890 – 1974)
 - Direktor des „Office of Scientific Research and Development“ während des II. Weltkriegs
 - Visionär, viele Erfindungen, z.B. analoge Computer
- Memex (Memory Extension)
 - Artikel in *Atlantic Monthly* (1945) „As We May Think“
 - Memex: „a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility“
 - **Verbindung** (*join*) von Informationseinheiten



A SCENARIO OF THE FUTURE WHICH ENTAILS WITH A FEW CHANGES OTHER WITH UNDESIRABLE CONSEQUENCES. THE SMALL FIGURE IN THE STRIP ON THE LEFT SHOWS THE DESK.

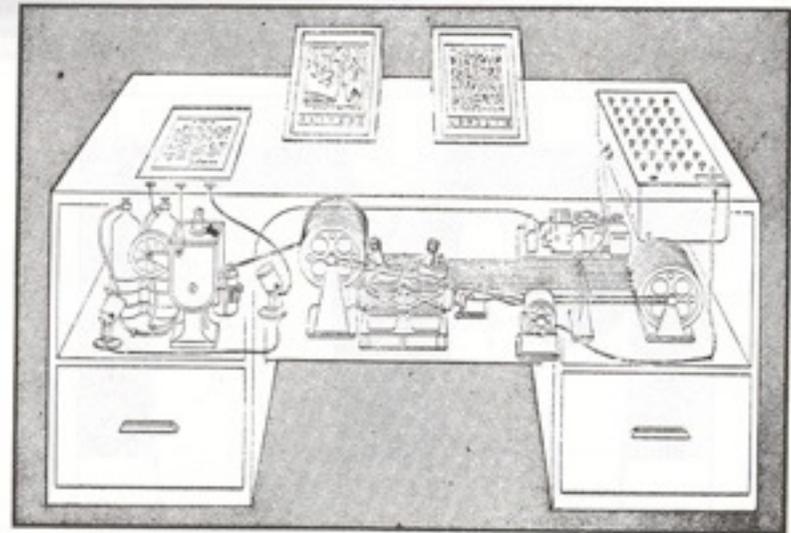
AS WE MAY THINK

A TOP U. S. SCIENTIST FORESEES A POSSIBLE FUTURE WORLD IN WHICH MAN-MADE MACHINES WILL START TO THINK

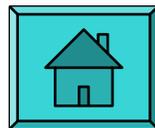
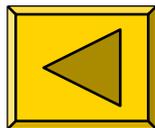
by VANNEVAR BUSH

DIRECTOR OF THE OFFICE OF SCIENTIFIC RESEARCH AND DEVELOPMENT

Condensed From the Atlantic Monthly, July 1945



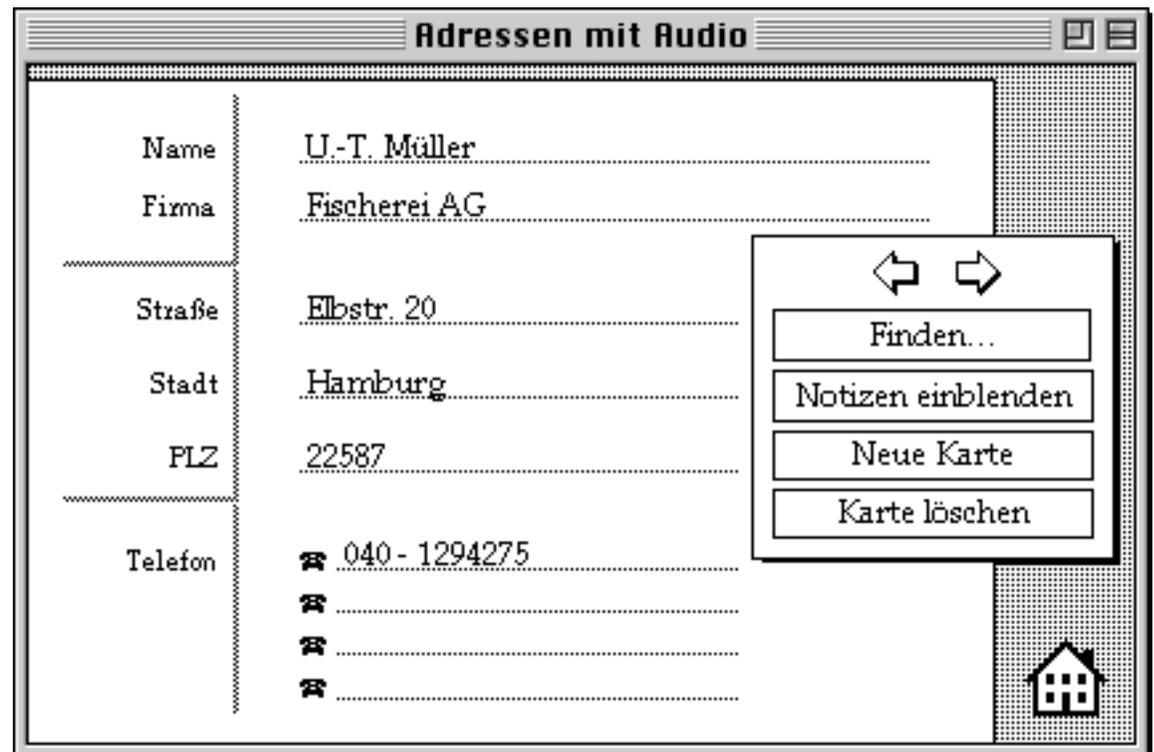
Memex in the form of a desk would instantly bring files and material on any subject to the operator's fingertips. Slanting translucent viewing screens magnify supermicrofilm filed by code numbers. At left is a mechanism which automatically photographs longhand notes, pictures and letters, then files them in the desk for future reference (*LIFE* 19(11), p. 123).



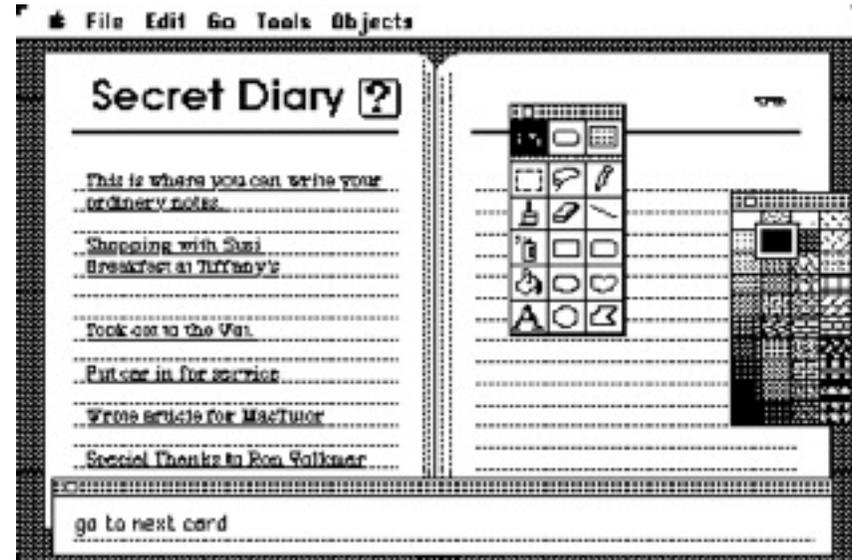
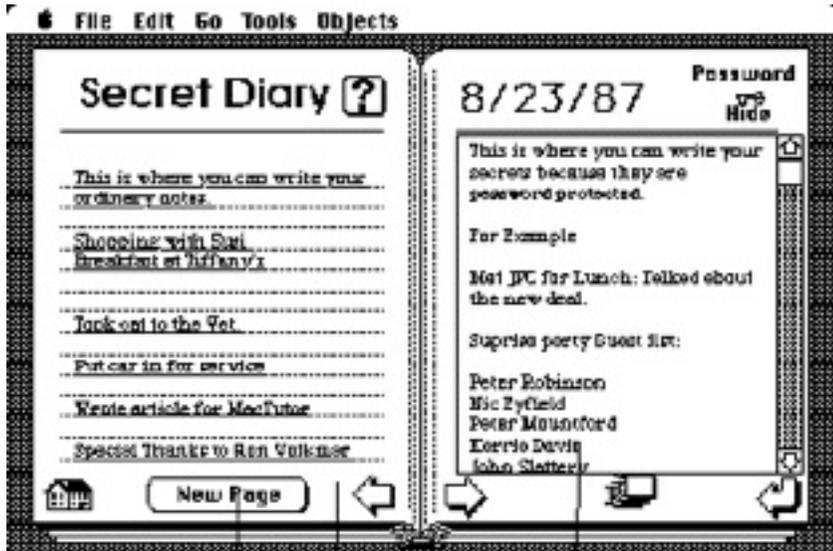
Hypertext-Autorensysteme

- Klassisches Vorbild: HyperCard (1987)
 - Viele Nachbildungen, z.B. SuperCard, MetaCard
 - Ideen eingegangen in kommerzielle Produkte: Asymetrix ToolBook, Microsoft PowerPoint

- Grundkonzepte:
 - Karteikarten-Metapher
 - Autorenmodus und Anzeigemodus
 - Grafischer Editor
 - **Objektorientierte Sprache** zur Ereignisbehandlung (bei HyperCard: HyperTalk)
 - Medienintegration



HyperCard Autorensystem



Probleme beim Hypertext-Design

- Navigationspfad vs. Ordnung der Knoten
 - Was heißt „Zur nächsten Karte“?
(HyperTalk: `on mouseUp go to next card end mouseUp`)
 - Lösung z.B. in WWW-Browsern: Navigation im dynamischen Zugangspfad
- Orientierung im „Labyrinth“
 - Grundlegende Vision von Hypertext nicht für alle Informationsbedürfnisse angemessen
 - Lösungen z.B.
 - » Suchmaschinen (analog im Buch: Register)
 - » strenge Baumstruktur (analog im Buch: Inhaltsverzeichnis)
 - » Navigationsanzeigen (analog im Buch: relative Position)
 - » Lesezeichen (*bookmarks*) (analog im Buch: Lesezeichen)
- Informationsbereitstellung für verschiedene Lesergruppen:
 - Findet jede(r) alles, was er/sie braucht?

Interaktivität

- Grundelement *aller* historischen Hypertext-Visionen (Memex, Xanadu, HyperCard, **auch** WWW):
 - Lesemodus und Autorenmodus
- Verändern von Hypertext-Dokumenten sollte ähnlich intuitiv sein wie das Lesen
- Hypertext-Systeme sollten Rechteverwaltung und Versionsverwaltung integrieren
- Derzeit im WWW höchstens ansatzweise realisiert:
 - Online-Foren, interaktive Linksammlungen
 - Beurteilungssysteme im E-Business (z.B. bei Amazon)
 - „Wiki“ („Wiki-wiki“, „Wiki-Web“)
 - Blogging Software
 - Social Networks

"The idea was that anybody who used the web would have a space where they could write and so the first browser was an editor, it was a writer as well as a reader. "

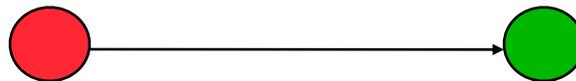
Tim Berners-Lee 2005

<http://news.bbc.co.uk/2/hi/technology/4132752.stm>

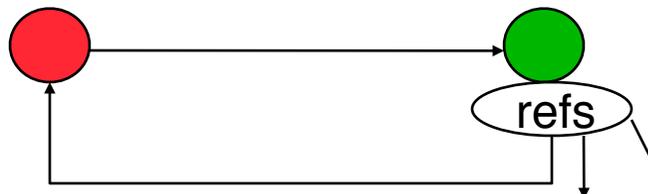
Unidirektionale und bidirektionale Verbindungen

- Xanadu-Vision:
 - Verbindungen sind bidirektional
 - Quell- und Zielobjekt können beliebig bewegt werden, ohne die Verbindung zu verletzen
- Praxis in HyperCard, PowerPoint, WWW etc.:
 - Unidirektionale Links
 - Viele Links zeigen „ins Leere“

Unidirektional:



Bidirektional:



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 zu Hypertext
- HTML 
- Textstrukturierung
- Tabellen
- Cascading Style Sheets
- Strukturierte Seiten
- Medieneinbettung

Literatur:

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

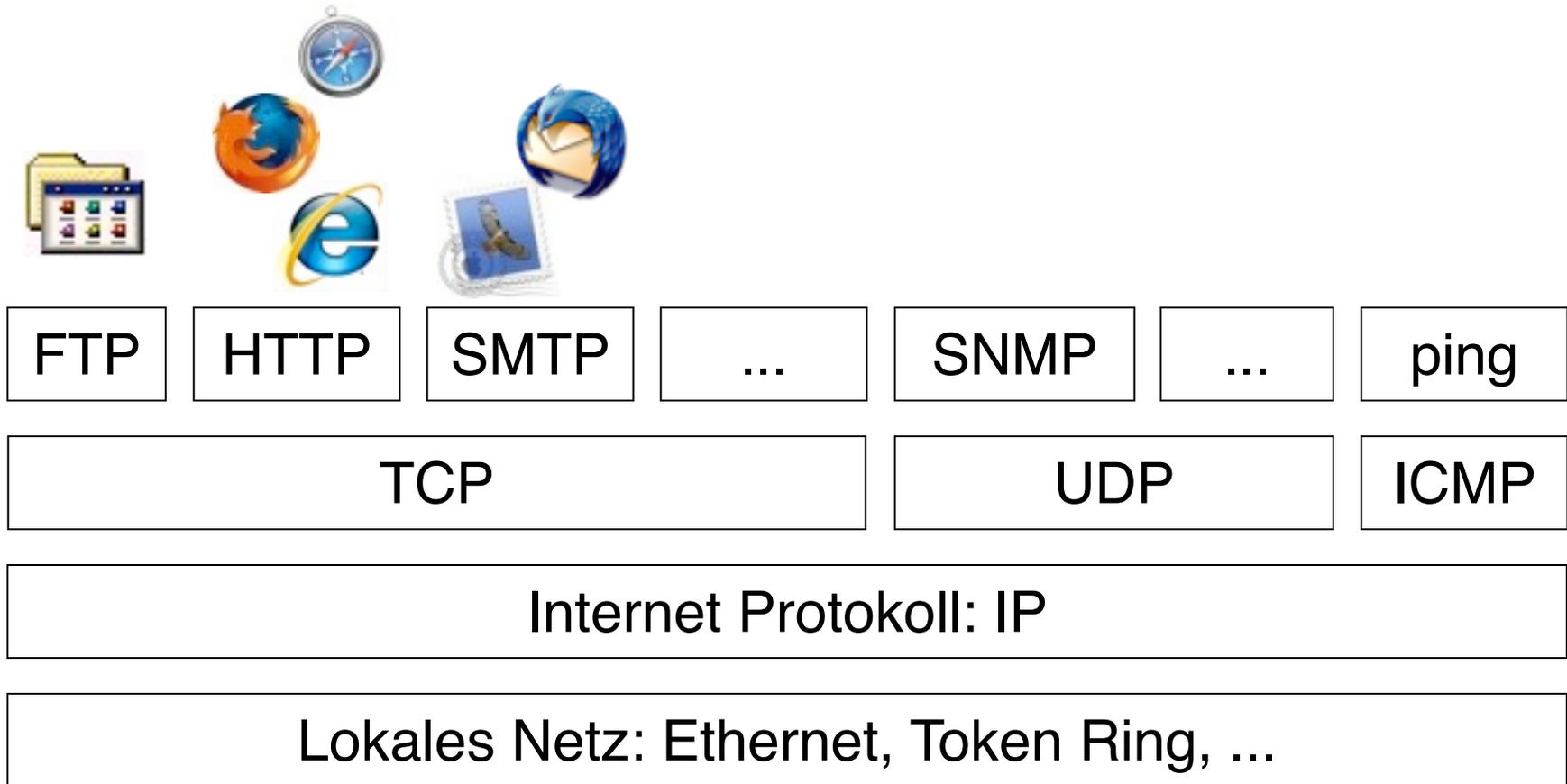
Peter Kröner: HTML5,
Open Source Press 2010

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:
 - 2001: ca. 550 Milliarden Dokumente (incl. „Deep Web“)
 - 2005: ca. 11,5 Milliarden Dokumente im Indexable Web
 - "The Indexed Web contains **at least 2.91 billion pages** (Saturday, 06 November, 2010)."
 - "The Indexed Web contains **at least 11.45 billion pages** (Thursday, 10 November, 2011)."
(www.worldwidewebsite.com)

(Andere Zahlen aus en.wikipedia.org)

Internet: Protokollschichten



Domain Name System: DNS

www.informatik.uni-muenchen.de

Rechnername

Subdomains

Top-Level
Domain

Vergabe von Domains:

www.internic.net

www.denic.de

.com .org .net .edu ...
.de .fr .uk .jp ...

URL: Gegenbeispiele & Tips

`http://tight rope.test.lmu.de/pages/index.html`

`http://tightrope.test.lmu.de/neue datei.html`

`http://stop/go.test.lmu.de/pages/index.html`

- Großschreibung egal
- Leerzeichen und manche Sonderzeichen verboten

- Großschreibung wichtig
- Sonderzeichen und Leerzeichen gefährlich
- richtige Extension

WWW, HTML und HTTP

- Standardisierungsgremien:
 - IETF (Internet Engineering Task Force), z.B. HTTP
 - W3C (WWW Consortium), z.B. HTML
- Grundprinzip von HTTP (HyperText Transfer Protocol):
 - 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

HTTP-Request: Beispiel

`http://djce.org.uk/dumprequest`

The following HTTP request was received from IP address 141.xx.y.z (port 54820) by IP address 91.84.196.2 (port 80):

```
GET /dumprequest HTTP/1.1
Host: djce.org.uk
User-Agent: Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_4; en-us) AppleWebKit/533.18.1 (KHTML, like Gecko) Version/5.0.2 Safari/533.18.5
Accept: application/xml,application/xhtml+xml,
text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Referer: http://www.google.de/search?
client=safari&rls=en&q=dumprequest&ie=UTF-8&oe=UTF-8&redir_esc=&ei
=p6HVTI7-Bs3Oswav5t3bCA
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

Manuelle Auszeichnung von Text



Fig. 18.1. Type proof of three paragraphs from an earlier chapter in this manual, marked for indexing (for explanation of marking see text, 18.34–36).

- Hinzufügen von Anmerkungen
 - zur Korrektur
 - zur Definition des Satzes oder
 - für die Erstellung von Zusatzinformation (Index etc.)

Beispiel:
Chicago Manual of Style 1982,
Nach: www.tecnotertulia.com

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{"Uberschrift}
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:
 - Bessere 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
 - Spezieller Dokumenttyp von SGML
- 1993, NCSA Mosaic Browser
- 1999, Version 4 von HTML, lange Zeit stabil
- 2000, XHTML 1.0 (HTML 4.01 in XML, siehe später)
- 2004, Browserhersteller gründen *Web Hypertext Application Technology Working Group* (WHATWG) – "Rebellion gegen W3C"
- 2007, W3C gründet HTML5 Working Group
- 2009, W3C löst XHTML-2.0-Arbeitsgruppe auf
- HTML5 in Entwurfsstadium parallel bei WHATWG und W3C
 - stabile Version für 2014 erwartet: <http://www.w3.org/2011/02/htmlwg-pr.html>

Hypertext Markup Language HTML: Leistung

- Leistungsumfang von HTML:
 - Text und Mediendokumente zu Seiten zusammenfassen
 - Logische Struktur von Seiten definieren
 - Hyperlinks auf andere Dokument im Web einbinden
 - Teile der Darstellungsfläche für Interaktion und Animation vorsehen
- Früher im Leistungsumfang von HTML, schrittweise in "Style Sheets" ausgelagert (und mit HTML5 vollständig verlagert)
 - Textattribute für die Darstellung festlegen
 - Spezielle Textformatierungen (z.B. Tabellen) definieren
 - Position von Dokumentteilen auf der Seite festlegen

HTML-Syntax

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

- Elemente (*tags*):
 - Paarweise als Beginn-/Ende-Paar z.B. `<p> ... </p>`
 - Einzeln z.B. `
`
- Attribute:
 - Zulässige Attribute abhängig vom konkreten Tag
 - » *Immer* zulässig (in HTML5): **class, id, lang, title, style, hidden**
 - Attributwerte:
 - » In vielen Fällen ohne Anführungszeichen angebar (z.B. Zahlen)
 - » Stilistisch guter HTML-Code benutzt immer Anführungszeichen
- Zeilenumbrüche, mehrfache Leerzeichen, Tabulatoren i.A. ignoriert
- Kleinschreibung empfohlen
(in HTML5 aber Groß- und Kleinschreibung äquivalent)
- Kommentare: `<!-- ... -->`

Einfaches HTML-Beispiel

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Einfaches Beispieldokument HTML</title>
```

```
  </head>
```

```
  <body>
```

```
    <h2>Hello World!</h2>
```

```
    <br>
```

```
    <h1>Ueberschrift auf erster Ebene</h1>
```

```
    <p>Ganz normaler Text</p>
```

```
  </body>
```

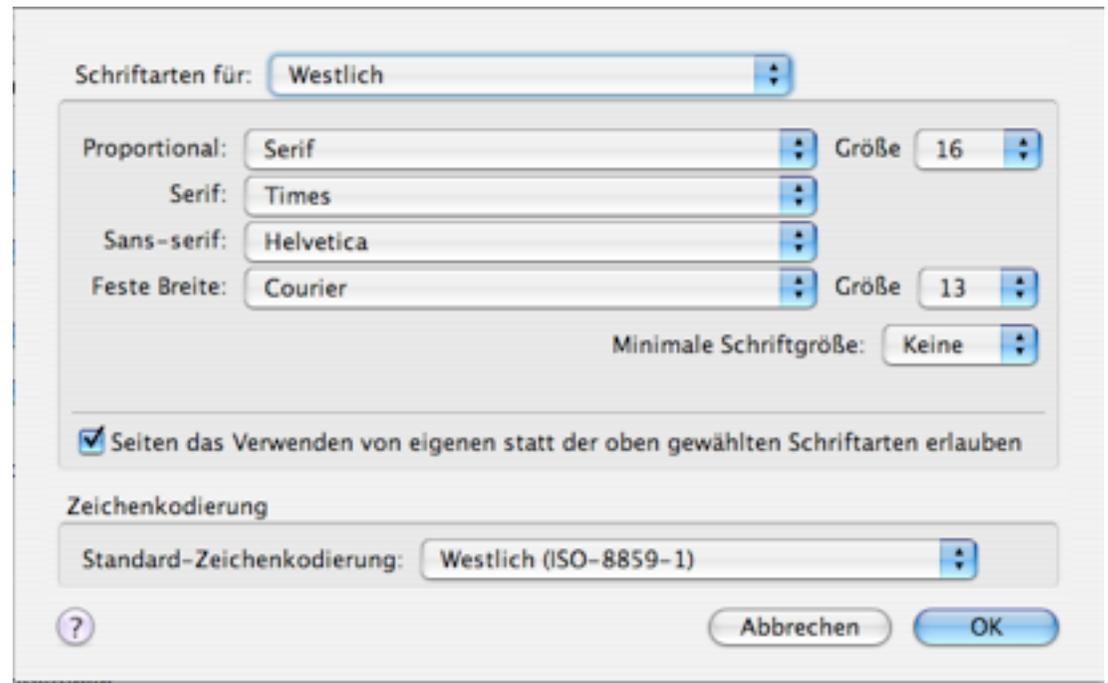
```
</html>
```

Datei: html1.html

Trennung Inhalt-Darstellung in HTML

- Starke Trennung:
 - bei Verwendung vordefinierter Textklassen
 - » z.B. <h1> für Überschriften
 - bei Verwendung von Cascading Style Sheets (siehe später)
- Schwache Trennung (nur in HTML-Versionen bis 4):
 - Bei expliziter Auszeichnung z.B. mit

- In HTML alleine ist keine vollständige Kontrolle über die Darstellung möglich.
- Siehe "Style Sheets"!



Dokumenttyp

- Verschiedene Versionen von HTML
 - Angabe benutzter Version mit DOCTYPE
 - In heutigen Browsern meist nicht überprüft!
 - Derzeit aktuelle Version: HTML 4.01(mit Varianten) und HTML5
- 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 (derzeit am weitesten verbreitet):
 - 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">
```
- HTML5-Stil:
 - Kurz und knapp:

```
<!DOCTYPE html>
```

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 charset="ISO-8859-1">` (HTML5)

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

 - `<meta http-equiv="expires" content="Sat, 15 Dec 2010 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 müssen "maskiert" werden:

<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, Adobe DreamWeaver, Microsoft FrontPage, NVU
- 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 

- Allgemeines
- Textstrukturierung 
- Tabellen
- Cascading Style Sheets
- Strukturierte Seiten
- Medieneinbettung

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

Elemente zur Strukturierung des Texts

- Überschriften `<h1>...</h1>` ... `<h6>...</h6>`
- Abschnitte `<section> </section>` (HTML5)
- Absätze `<p>...</p>`
- Unnummerierte Listen ` list item 1 `
- Nummerierte Listen ` list item 1 `
- Definitionslisten `<dl> <dt>term</dt> <dd>defn</dd> </dl>`
- Externe Beiträge `<article> </article>` (HTML5)
- Zitate `<blockquote> Zitattext </blockquote>`
- Adressen `<address> Kontaktinformation </address>`
- Vorformatierter Text `<pre> z.B. Programmtext </pre>`
 - Dicktengleiche (Nicht-Proportional-)Schrift
 - Umbruch und Leerzeichen wie in der HTML-Datei
- Thematischer Wechsel (Trennlinie) `<hr>`

Zeilenumbruch

- Zeilenumbruch erzwingen `
`
- "Geschütztes" Leerzeichen
(*non-breaking space*) ` `

Logische Auszeichnungen im Text

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

- Auszeichnungen:

- Betont ` ... `
- Steigerung durch Verschachteln
- Wichtig ` ... `
- Quelltext `<code> ... </code>`
- Beispiel `<samp> ... </samp>`
- Tastatureingabe `<kbd> ... </kbd>`
- Variable `<var> ... </var>`
- Zitat `<cite> ... </cite>`
`<q cite="quelle"> ... </q>`
- Definition `<defn> ... </defn>`
- Abkürzung `<abbr> ... </abbr>`

Physische Auszeichnungen im Text

- Historische Funktion:
Beschreibung der konkreten Formatierung des Textstücks (bis HTML 4)
- HTML soll, spätestens ab HTML5, nur noch logisch auszeichnen!
- Sinnvolle Auszeichnungen, die nahe an physischer Auszeichnung sind:
 - Abgesetzt, eher fett ` ... `
 - Abgesetzt, eher kursiv `<i> ... </i>`
- Beispiele für alte Auszeichnungen, nicht mehr zu verwenden!:
 - Schreibmaschine `<tt> ... </tt>`
 - Unterstrichen `<u> ... </u>`
 - Größer `<big> ... </big>`
 - Hochgestellt `^{...}`
- Für mathematische Formeln verwendet man heutzutage MathML!

Verweise (Links)

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

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
- Beispiele:
`http://www.mimuc.de/`
`http://Arbeitsgruppe/www.mimuc.de:8080/usr/local/data/index.html`
`mailto:hussmann@ifi.lmu.de`
`sftp://heinrich.hussmann:@www.medien.ifi.lmu.de/public_html/
dm0910/dm1.m4b`

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):

```
<a name="hierher">Text</a>
```

- Ansprechen des Zielankers:

```
<a href="xyz.html#hierher">Text</a>
```

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 

- Allgemeines
- Textstrukturierung
- Tabellen 
- Cascading Style Sheets
- Strukturierte Seiten
- Medieneinbettung

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

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:

```
<table>  
  <tr>  
    <th>  
    </th>    <th>  
    </th>    <th>  
    </th>    </tr>  
  <tr>  
    <td>  
    </td>    <td>  
    </td>    <td>  
    </td>    </tr>  
  <tr>  
    <td>  
    </td>    <td>  
    </td>    <td>  
    </td>    </tr>  
</table>
```


Mit `<thead>`, `<tbody>` und `<tfoot>` kann man logische Bereiche definieren.

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 zwischen Zellen : Attribut `cellspacing="n"` in `<table>`
 - Abstand Rahmen-Zellen : Attribut `cellpadding="n"` in `<table>`
- Textformatierung, Ausrichtung etc.
 - Spezielle Attribute (z.B. `align`)
 - Cascading Style Sheets (sh. unten)

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 
- Strukturierte Seiten
- Medieneinbettung

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

Cascading Style Sheets (CSS)

- Von HTML prinzipiell unabhängige Sprache zur Beschreibung von Formatierungsinformation
 - Standardisierung durch W3C
 - Besonders für HTML geeignet
- Entstehungsgeschichte:
 - Vielzahl von "Standard-Attributen" in vielen HTML-Elementen (align, pos, color, font, ...)
 - Vereinheitlichung in CSS (aktuelle Version 2.0, CSS 3 in Entwicklung)
- Ablösung "alter" Konstrukte zugunsten CSS-beschriebener Styles: Empfehlung in HTML bis Version 4, verpflichtend ab HTML5
 - Universalattribut `style`
 - Alte Schreibweise (nicht mehr empfehlenswert):
`<p>Text</p>`
 - Schreibweise mit CSS-Syntax:
`<p style="font-size:7">Text</p>`
 - Empfehlenswert für modernes HTML: Rein logische Auszeichnung
 - » Formatierung separat festgelegt (in CSS)

CSS-Eigenschaften, Beispiel Schriftformatierung

- CSS-Syntax: Eigenschaft-Wert-Paare
 - Beispiel: `font-size:250%`
- Umfangreiche Liste an Eigenschaften und Maßeinheiten
- Eigenschaften zur Schriftformatierung:
 - `font` Zusammenfassung anderer Eigenschaften
 - `font-family` Gewünschte Schrift(en) mit Priorisierung
 - `font-style` Kursiv / normal
 - `font-variant` Kapitälchen (*small caps*) / normal
 - `font-size` Größe (numerisch oder ungenau)
 - `font-weight` Strichstärke (fett / mager)
 - `font-stretch` Laufweite
 - `word-spacing` Wortabstand
 - `letter-spacing` Zeichenabstand
 - `color` Farbe
 - ...

CSS-Syntax

- Eigenschaft-Wert-Paar

Eigenschaft : *Wert* z.B. `font-style:italic`

– Wenn als Wert eines HTML-Attributs: Anführungszeichen "" empfehlenswert

- Mehrere Eigenschaft-Wert-Paare

– Abtrennen mit Strichpunkt

z.B. `font-style:italic; font-size:large;`

- Anführungszeichen für Werte (z.B. bei Leerzeichen im Wert)

– Einfache Anführungszeichen ''

z.B. `font-family:'Times New Roman'`

- Mehrere Werte (Sequenz) für eine Eigenschaft

– Abtrennen mit Komma

z.B. `font-family:'Times New Roman', 'Times', serif`

Weitere CSS-Eigenschaften

- Schriftformatierung (auch mit Schriftartendatei)
- Ausrichtung und Absatzkontrolle
- Außenrand und Abstand
- Innenabstand
- Rahmen
- Hintergrundfarben und -bilder
- Listenformatierung
- Tabellenformatierung
- Pseudoformate
 - z.B. `link`, `visited`, `focus`
- Positionierung und Anzeige von Elementen
- Layouts für Printmedien
- Sound-Kontrolle für Sprachausgabe
- Anzeigefenster

Einbindung von CSS in HTML (1)

- Individuell formatieren:

- `style`-Attribut bei HTML-Tags benutzen

- z.B.

- ```
<p style="font-weight:bold; font-size:200%">
```

- ```
Beispieltext</p>
```

- Zentrale Stildefinitionen:

- Festlegung der Style-Attribute für Standard-HTML-Elemente

- z.B.

- ```
body {margin-left:100px; }
```

- ```
h1 { font-size:48pt;
```

- ```
font-style:italic;
```

- ```
border-bottom:solid thin black; }
```

- ```
p,li { font-size:12pt;
```

- ```
line-height:14pt;
```

- ```
font-family:Helvetica,Arial,sans-serif;
```

- ```
letter-spacing:0.2mm;
```

- ```
word-spacing:0.8mm;
```

- ```
color:blue; }
```

Einbindung von CSS in HTML (2)

- Ablage von zentralen Stildefinitionen im Kopfbereich der HTML-Datei

```
<style type="text/css">  
... Stildefinitionen ...  
</style>
```

 - Wegen Problemen älterer Browser oft Stildefinitionen als Kommentar
- Ablage von zentralen Stildefinitionen in separater CSS-Datei (.css)
 - Enthält nur Stildefinitionen, kein HTML
 - Einbindung in HTML-Dateien:

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

Beispiel zu CSS (Variante 1)

```
<!DOCTYPE html>

<html>
  <head>
    <title>Beispiel zu CSS</title>
    <style>
      p      {font-family:Times; font-size:20pt}
      h1     {font-family:Verdana; color:red}
    </style>
  </head>
  <body>
    <h1>&Uuml;berschrift 1</h1>
    <p>Absatz 1</p>
    <h1>&Uuml;berschrift 2</h1>
    <p>Absatz 2</p>
    <h1>&Uuml;berschrift 3</h1>
    <p>Absatz 3</p>
  </body>
</html>
```

styles.html

Beispiel zu CSS (Variante 2)

```
<!DOCTYPE html>

<html>
  <head>
    <title>Beispiel zu CSS</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
  </head>
  <body>
    <h1>&Uuml;berschrift 1</h1>
    <p>Absatz 1</p>
    <h1>&Uuml;berschrift 2</h1>
    <p>Absatz 2</p>
    <h1>&Uuml;berschrift 3</h1>
    <p>Absatz 3</p>
  </body>
</html>
```

Datei `styles.css` (im gleichen Verzeichnis):

```
p      {font-family:Verdana; font-size:16pt}
h1     {font-family:Verdana; color:green}
```

stylesfile.html