



3. Zeichen und Schrift

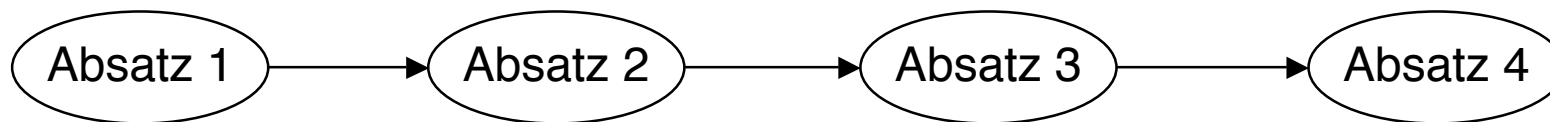
- 3.1 Medien Zeichen, Text, Schrift
- 3.2 Mikro-Typografie: Zeichensätze
- 3.3 Makro-Typografie: Gestalten mit Schrift
- 3.4 Hypertext und HTML 

- Allgemeines zu Hypertext 
- HTML
- Textstrukturierung
- Tabellen
- Cascading Style Sheets
- Framesets
- Medieneinbettung

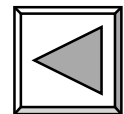
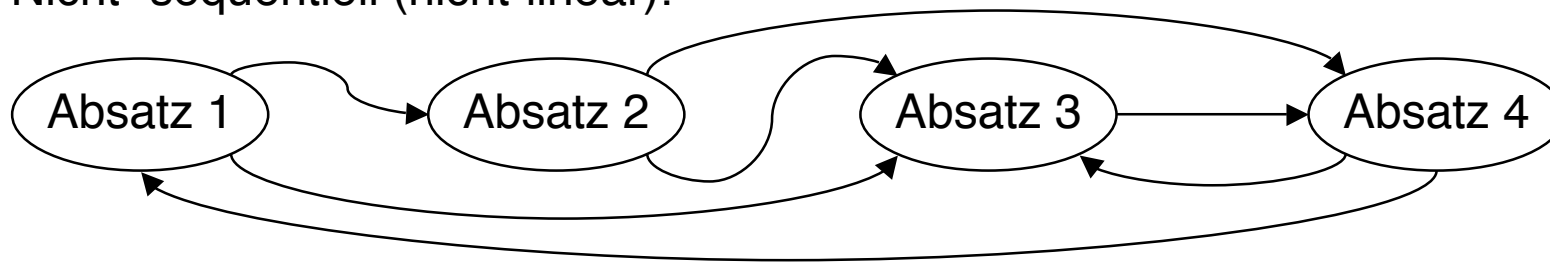
Hypertext

- *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):

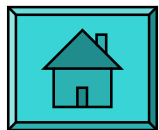
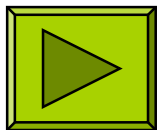
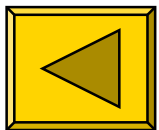


Nicht- sequentiell (nicht-linear):



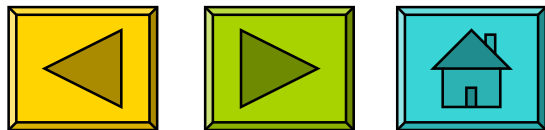
Information in Hypertext-Knoten

- 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 
 - ...



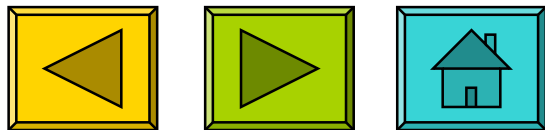
Hypertext-Knoten

- *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*



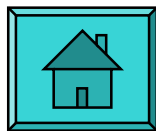
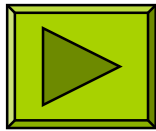
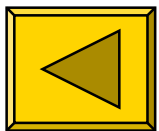
Verbindung in Hypertext

- *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



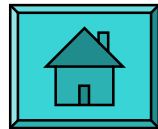
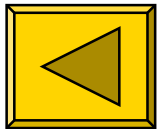
Ted Nelson und Xanadu

- 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“
 - Enthält das Konzept einer Verbindung (*join*) von Informationseinheiten

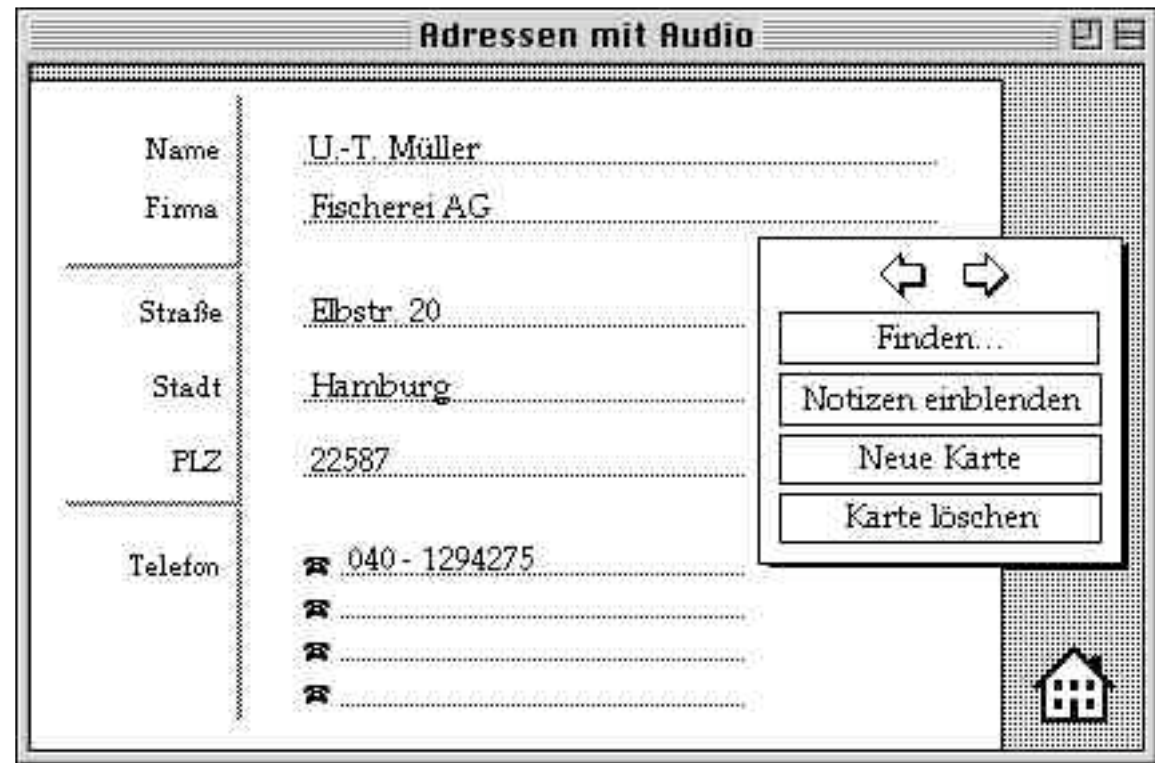


Geschichte von Hypertext

- 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

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



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?

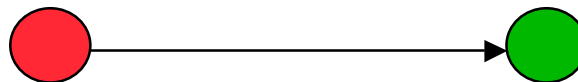
Interaktivität

- Grundelement *aller* historischen Hypertext-Visionen (Memex, Xanadu, HyperCard, 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“)

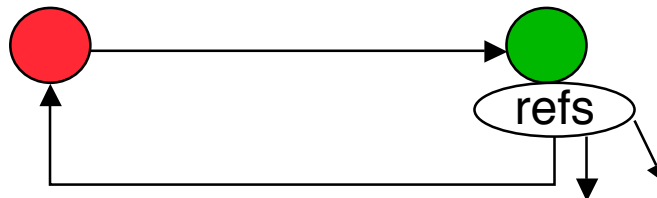
Unidirektionale und bidirektionale Verbindungen

- 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
- 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
- Extremes Verbreitungsgrad, extreme Informationsdichte:
 - 2001: ca. 550 Milliarden Dokumente (incl. „Deep Web“)
 - 2005: ca. 11,5 Milliarden Dokumente im Indexable Web
 - Juni 2008: ca. 63 Milliarden Dokumente im Indexable Web

(Zahlen aus en.wikipedia.org)

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 Aufzählung
\end{itemize}
```

- HTML-Beispiel:

```
<P>&Uuml;berschrift<BR>
Text text <I>kursiver Text</I>
<UL>
<LI>Punkt in Aufzählung
</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 Auszeichnungs-klasse 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 [ATTRIBUTE = "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 2007 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:

`ä` `ö` `ü`

ä ö ü

`Ä` `Ö` `Ü`

Ä Ö Ü

`ß`

ß

- Sonderzeichen der HTML-Syntax "maskieren":


`<` `>` `&` `"`;


< > & "

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
- 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
(*word 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

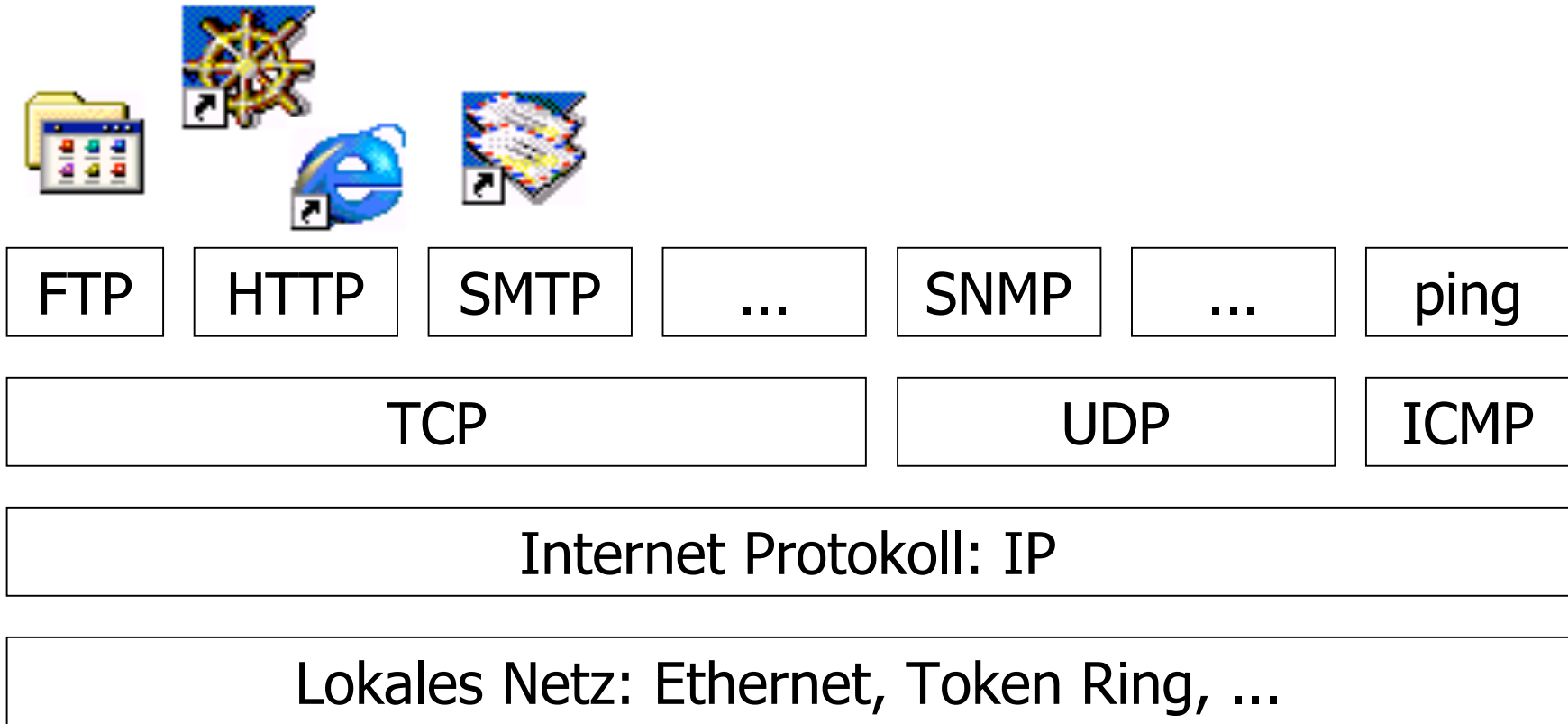
- Syntax:

Protokoll : / *lokalerNetzwerkname* / *Hostname* : *Port* / *Pfad*

`http://www.mimuc.de/`

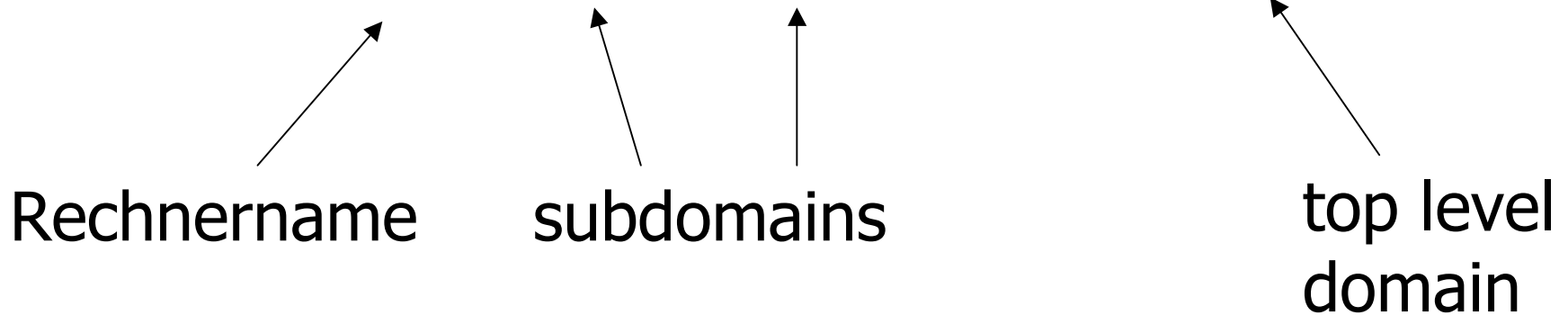
`http://Arbeitsgruppe/www.mimuc.de:8080
/usr/local/data/index.html`

IP: Protokollschichten



Domain Name System: DNS

www.informatik.uni-muenchen.de



Vergabe von domains:

www.internic.net

www.denic.de

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

Uniform Resource Locator: URL

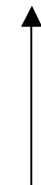
www.internic.net

telnet://yoda.cs.uni-sb.de/

mailto:hussmann@ifi.lmu.de

ftp://ftp.rz.uni-sb.de/pub/mirror/Linux

http://www.ifi.lmu.de/lehre/pruefungen.xhtml



Protokoll DNS-Hostname Ordner/Datei

Ext.

URL: Gegenbeispiele & Tips

<http://tight rope.hbks.uni-sb.de/pages/index.html>

<http://tightrope.hbks.uni-sb.de/neue datei.html>

<http://stop/go.hbks.uni-sb.de/pages/index.html>

Großschreibung egal

Großschreibung wichtig
Sonderzeichen gefährlich
keine Leerzeichen
richtige Extension

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
- 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 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 
- Framesets
- 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)
- In HTML 4.0 wird die Ablösung "alter" Konstrukte zugunsten einheitlicher CSS-beschriebener Styles forciert.
 - Universalattribut `style`
 - Alte Schreibweise (nicht mehr empfehlenswert):

```
<p><font size="7">Text</font></p>
```
 - Neue Schreibweise mit CSS-Syntax:

```
<p style="font-size:250%">Text</p>
```

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:
 - Universelles `style`-Attribut für alle HTML-Tags
 - z.B.
- Zentrale Stildefinitionen:
 - Festlegung der Style-Attribute für Standard-HTML-Elemente
 - z.B.

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

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

```
<html>
```

```
  <head>
```

```
    <title>Beispiel zu CSS</title>
```

```
    <style>
```

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

```
      h1     {font-family:Verdana; color:green}
```

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

```
<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