

# 3. Zeichen und Schrift

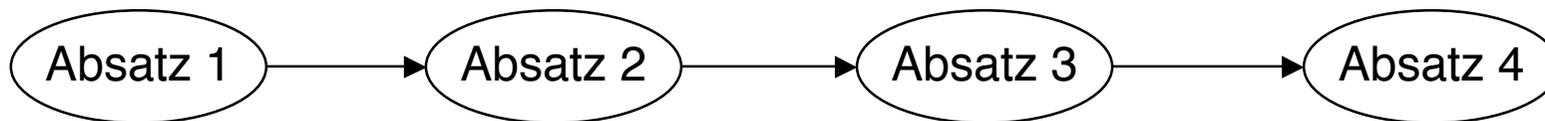
- 3.1 Medien Zeichen, Text, Schrift
- 3.2 Mikro-Typografie: Zeichensätze
- 3.3 Makro-Typografie: Gestalten mit Schrift
- 3.4 Hypertext und HTML 

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

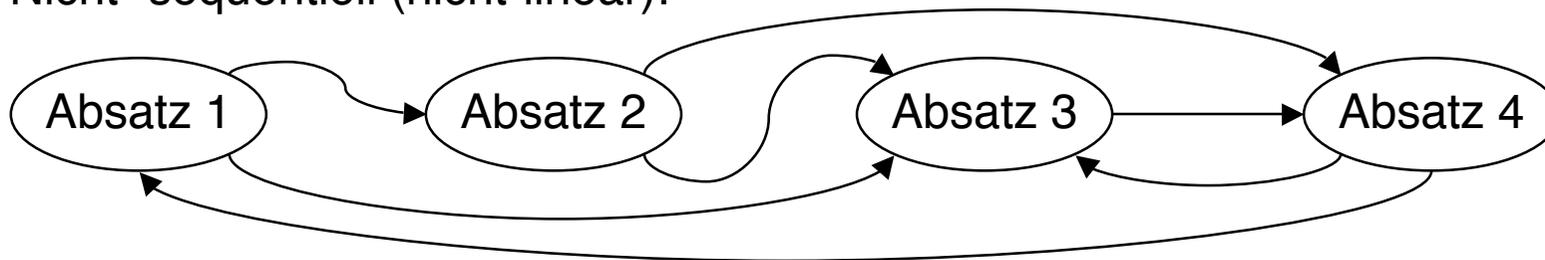
# Hypertext

- *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

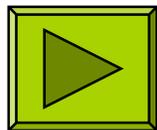
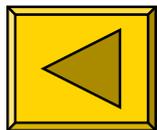


– Filmstücke

– Animationen

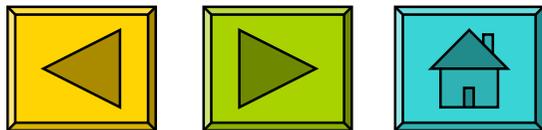


– ...



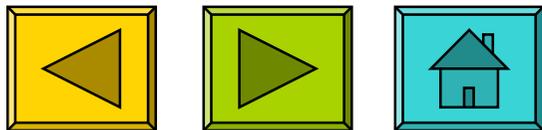
# Hypertext-Knoten

- *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*



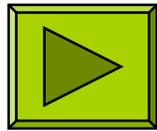
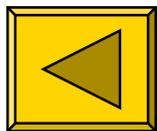
# Verbindung in Hypertext

- *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



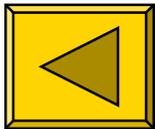
# Ted Nelson und Xanadu

- 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](http://www.xanadu.com))



# 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
  - 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

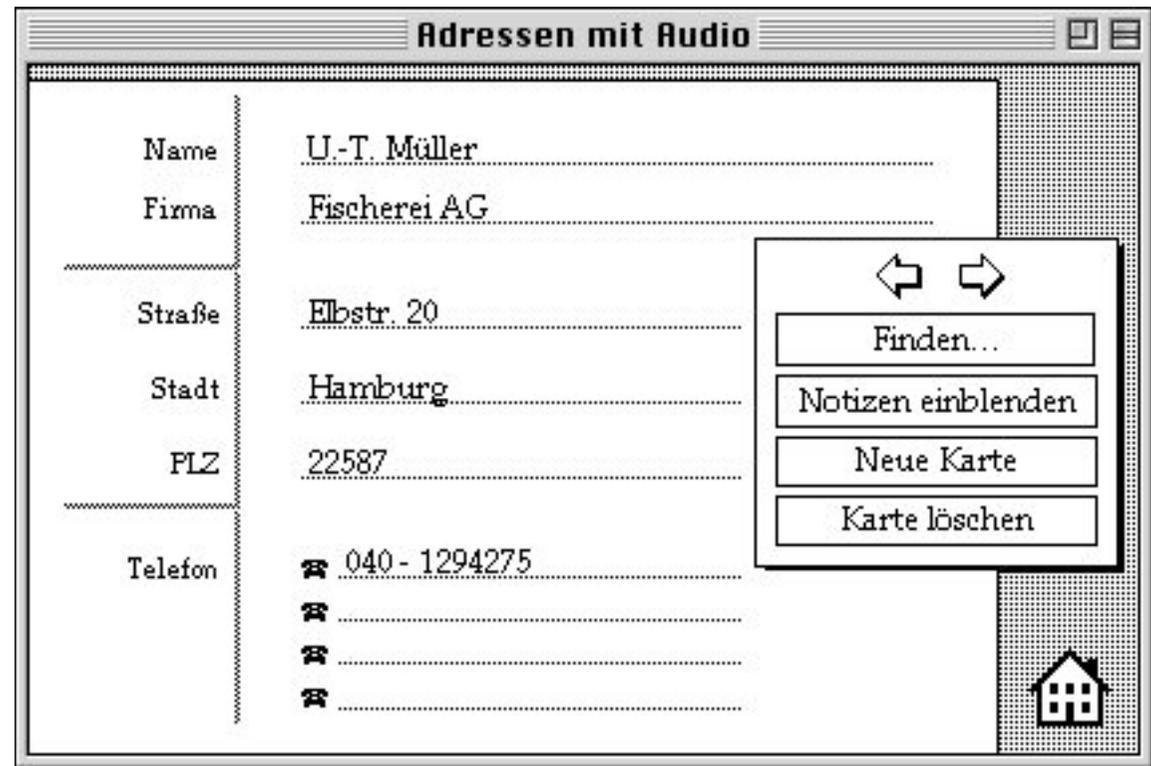


# Geschichte von Hypertext

- 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?

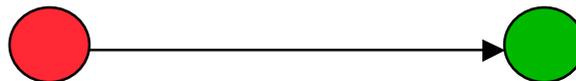
# Interaktivität

- 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“)

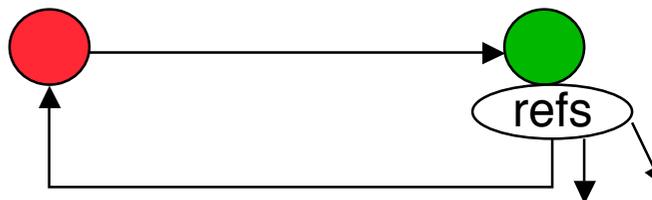
# Unidirektionale und bidirektionale Verbindungen

- 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 
- 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
    - » 2001 ca. 20 TeraByte (TByte =  $10^{12}$  Byte)
    - » 2004 ca. 10 PetaByte (PByte =  $10^{15}$  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 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 [ ATTRIBUT = "wert" ]* > Inhalt [</TAG>]`

- Tags:
  - Paarweise als Beginn-/Ende-Paar      `<TAGX> ... </TAGX>`
  - Einzeln      z.B. `<BR>`
- Attribute:
  - Zulässige Attribute abhängig vom konkreten Tag
    - » Fast immer zulässig: CLASS, ID, LANG, STYLE
  - Attributwerte:
    - » In vielen Fällen ohne Anführungszeichen angebbbar (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>
```

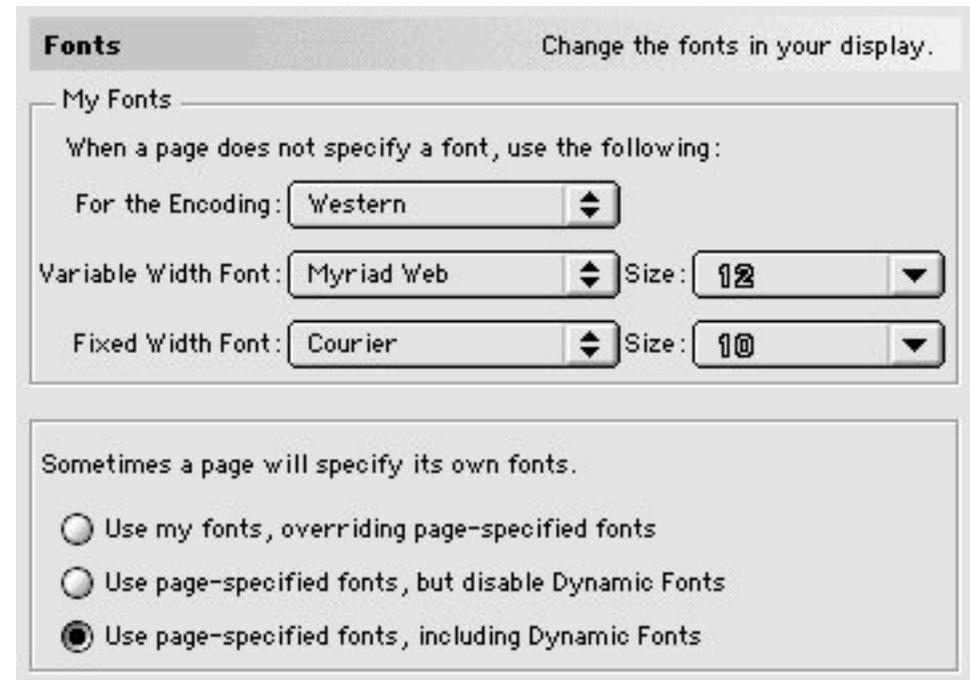
```
</HTML>
```

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 <FONT>

- 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. `&#174` (®), `&#8364` (€)
  - Explizite Namen, z.B. `&reg;`, `&euro;`

- Wichtige Namen für deutsche Sonderzeichen:

`&auml;` `&ouml;` `&uuml;`

ä ö ü

`&Auml;` `&Ouml;` `&Uuml;`

Ä Ö Ü

`&szlig;`

ß

- Sonderzeichen der HTML-Syntax "maskieren":

`&lt;` `&gt;` `&amp;` `&quot;`

< > & "

# 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  
(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 `<ul> <li> list item 1 </li> </ul>`
- Nummerierte Listen `<ol> <li> list item 1 </li> </ol>`
- 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 `<br>`
- Zeilenumbruch verhindern `<nobr>`
- "Geschütztes" Leerzeichen  
(*non-breaking space*) `&nbsp;`
- Zeilenumbruch im Wort erlauben  
(*word break*) `<wbr>`

# Logische Auszeichnungen im Text

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

- Auszeichnungen:

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

# Physische Auszeichnungen im Text

- Beschreibung der konkreten Formatierung des Textstücks

- Auszeichnungen:

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

# Verweise (Links)

- Klassischer Hypertext-Verweis
  - Markierter *Anker* im Text
  - *Referenz* auf andere HTML-Datei
- Syntax:  
`<a href=" ...Ziel..." > Text </a>`
- Beschreibung des Ziels
  - Vollständige URI (sh. nächste Folie)
  - Absolute Adressierung auf gleichem Rechner  
`<a href="/users/hussmann/xyz.html">`
  - Relative Adressierung auf gleichem Rechner  
`<a href=" ../xyz.html">`
  - 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

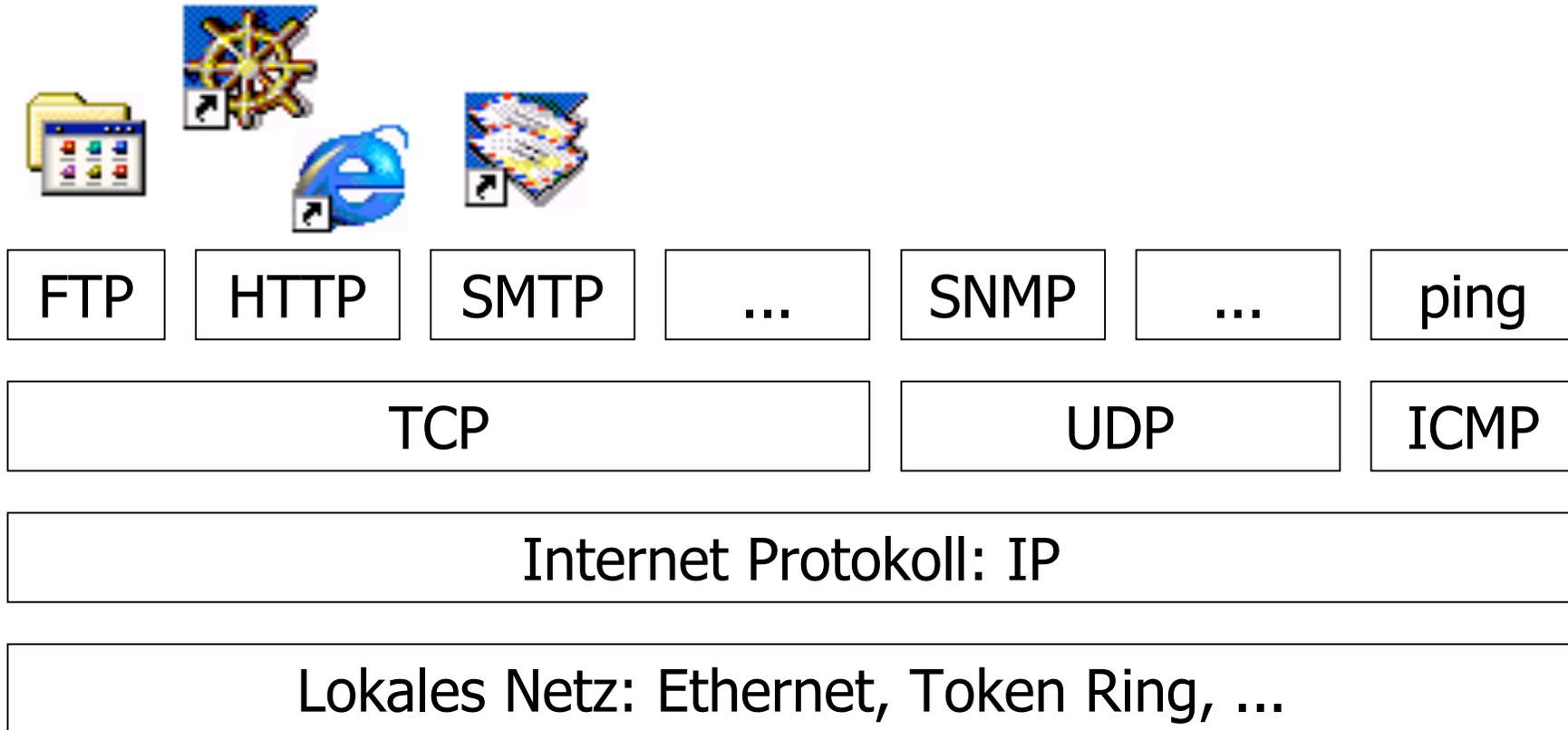
- Syntax:

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

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

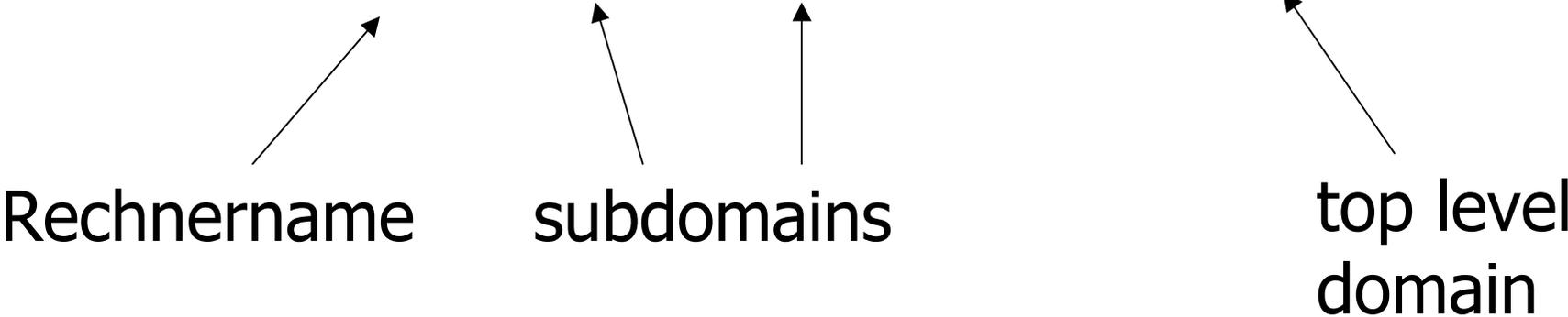
`http://Arbeitsgruppe/www.mimuc.de:8080  
/usr/local/data/index.html`

# 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  
(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>&lt;table&gt;</code>				
<code>&lt;tr&gt;</code>	<code>&lt;th&gt;</code> <code>&lt;/th&gt;</code>	<code>&lt;th&gt;</code> <code>&lt;/th&gt;</code>	<code>&lt;th&gt;</code> <code>&lt;/th&gt;</code>	<code>&lt;/tr&gt;</code>
<code>&lt;tr&gt;</code>	<code>&lt;td&gt;</code> <code>&lt;/td&gt;</code>	<code>&lt;td&gt;</code> <code>&lt;/td&gt;</code>	<code>&lt;td&gt;</code> <code>&lt;/td&gt;</code>	<code>&lt;/tr&gt;</code>
<code>&lt;tr&gt;</code>	<code>&lt;td&gt;</code> <code>&lt;/td&gt;</code>	<code>&lt;td&gt;</code> <code>&lt;/td&gt;</code>	<code>&lt;td&gt;</code> <code>&lt;/td&gt;</code>	<code>&lt;/tr&gt;</code>
				<code>&lt;/table&gt;</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)