


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

Weiterführende Literatur zu 3.3:

Christian Fries: Mediengestaltung, Fachbuchverlag Leipzig 2002

Textblöcke

- Augenführung auch auf der Seite wichtig:
 - Klare Gliederung (Überschriften und Absätze einheitlich)
 - Deutliche Trennung von Absätzen
 - » Abstand oder Einrückung

- Einzelzeilen und Absätze:
 - Niemals einzelne Zeile eines Absatzes durch Seitenumbruch abtrennen
 - „Hurenkind“ (letzte Zeile am Anfang einer Spalte oder Seite)
 - „Schusterjunge“ (Anfangszeile am Ende einer Spalte oder Seite)

Diese Absätze folgen ohne sichtbare Trennung aufeinander. Das erschwert das flüssige Lesen, vor allem das Überfliegen.

Diese Absätze benutzen als Trennung einen Einzug der ersten Zeile um 0,5 cm. Das erleichtert das flüssige Lesen, vor allem das Überfliegen – ohne zusätzlichen Platzverbrauch.

Textausrichtung

- Klassische Möglichkeiten der Textausrichtung:
 - Mittelachsensatz (zentriert)
 - Flattersatz, linksbündig
 - Flattersatz, rechtsbündig
 - Blocksatz

Ein kleiner Beispieltext
im Mittelachsensatz

Ein kleiner
Beispieltext im
linksbündigen
Flattersatz

Ein kleiner
Beispieltext im
rechtsbündigen
Flattersatz

Ein kleiner
Beispieltext
im Blocksatz

- Mittelachsensatz (Zentrierung) relativ schwer lesbar
 - Nur in Spezialsituationen (Bildunterschriften, Hauptüberschriften)
- Rechtsbündiger Flattersatz sehr schwer lesbar
 - Paßt nicht zu unserem kulturellen Hintergrund

Flattersatz

- Vorteile:
 - Lebendiges Erscheinungsbild
 - Keine Probleme bei kurzen Zeilen bzw. langen Worten
- Nachteile:
 - Unruhiges Erscheinungsbild
 - Optisch irreführende Erscheinungsbilder (Treppen und Bäuche)
- Regel:
 - Je kürzer die Zeilen, desto günstiger ist Flattersatz.

Bei links- oder
rechtsbündigem
Flattersatz gilt die Regel
„LANG - KURZ -
LANG“.
So ist das Ganze lesbar.
Vermeiden Sie Treppen
oder Bäuche.

Bei links- oder
rechtsbündigem
Flattersatz gilt
die Regel „LANG -
KURZ - LANG“.
So ist das Ganze lesbar.
Vermeiden Sie
Treppen oder Bäuche.

Blocksatz

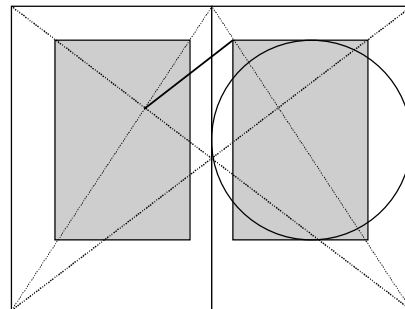
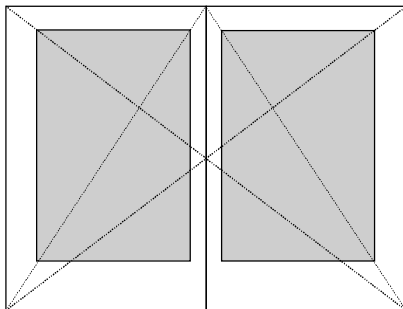
Ein kleiner
Beispieltext im
Blocksatz

Ein kleiner
Beispieltext
im
Blocksatz

- Vorteile:
 - Ruhiges Erscheinungsbild durch „Textflächen“
 - „Professioneller“ Eindruck
- Nachteile:
 - Sehr problematisch bei kleiner Spaltenbreite
 - » Große Abstände, Lücken, „Eselspfade“
 - Unregelmäßige Wortabstände können Lesbarkeit verschlechtern
- Regel:
 - Hohe Qualität nur bei sehr guter Information zur Worttrennung und mit guter Zeilenumbruch-Software (z.B. TeX, Desktop Publishing Systeme)

Satzspiegel

- *Satzspiegel* = Lage und Größe der bedruckten Fläche einer Seite
- Als harmonisch empfunden werden:
 - Gleiche Proportionen von Satzspiegel und Seite
 - Größerer Abstand nach unten als nach oben
- Diagonalkonstruktion für Doppelseiten:

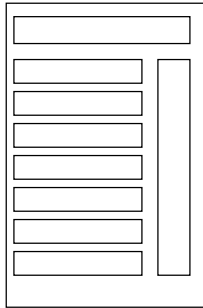


Rechts: „Goldener Schnitt“

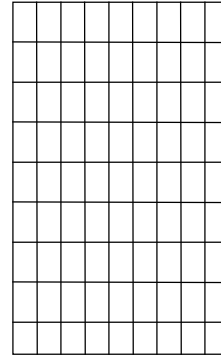
Zeilenregister, Seitenraster

- Zeilen einer Drucksache sollen „Register halten“:
 - Auf Vorder- und Rückseiten bzw. allen Seiten an der gleichen Position
- Idealerweise sollte das auch für die Zeilen einer Webpräsentation gelten
- Einheitliches Gestaltungsraster:

Module:

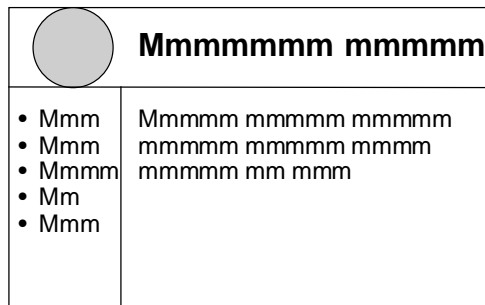


Gitter:
(Beispiel
Neuner-Teilung)



Mumble Screens

- „Murmel-Bildschirmseiten“
- Technik zur Überprüfung des optischen Eindrucks unabhängig vom Inhalt
- Text durch „Mmmmmm“ ersetzen.



Seitenlayout bei Online-Medien

- Festes Gestaltungsraster verwenden
- Texte klar gliedern und Struktur adäquat codieren:
 - Nominale Codierung: Reihenfolge und Ordnung nicht ersichtlich
 - Ordinale Codierung: Skala zur relativen Einordnung (z.B. Numerierung)
 - Relationale Codierung: Skala zur absoluten Einordnung incl. Abstandsmaß (z.B. Größe von Schrift oder Symbol)
- Übersichtliche Gestaltung:
 - Kurze Texte verwenden (Stichpunktstil)
 - Relativ kurze Zeilen verwenden
 - Auf Blocksatz kann oft verzichtet werden

3. Zeichen und Schrift

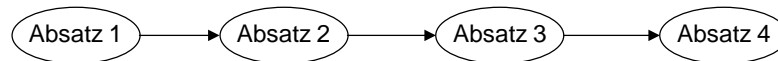
- 3.1 Medien Zeichen, Text, Schrift
- 3.2 Mikro-Typografie: Zeichensätze
- 3.3 Makro-Typografie: Gestalten mit Schrift
- 3.4 Hypertext und HTML



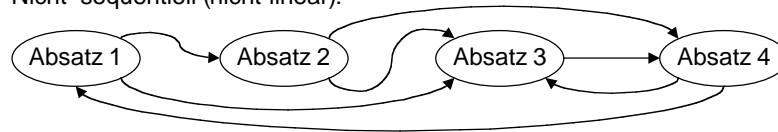
Hypertext

- *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 Anzeigergeräts 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
 - 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)



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



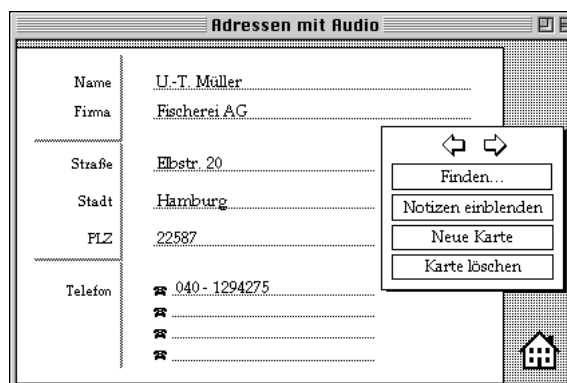
Geschichte von Hypertext

- 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?

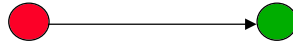
Interaktivität

- 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-Web“

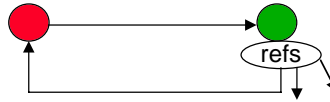
Unidirektionale und bidirektionale Verbindungen

- 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:



Hypertext im World Wide Web

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

WWW, HTML und HTTP

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

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

```
<P>\text{Überschrift}<BR>
\text{Text text <I>kursiver Text</I>
<UL>
<LI>Punkt in Aufz\text{ahl}ung
</UL>
```

Trennung Inhalt – Darstellung

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

Hypertext Markup Language HTML

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

HTML-Syntax

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

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

Einfaches HTML-Beispiel

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
"http://www.w3.org/TR/REC-html40/strict.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

