

LaTeX-Praxis

Dokumente gestalten

Jörn Clausen

joern@TechFak.Uni-Bielefeld.DE

Übersicht

- Schriften
- Umgebungen
- Makros und eigene Umgebungen

Schriften

- zu \TeX gehörende Schriftfamilie: Computer Modern
- verschiedene Schnitte: CM Roman, CM Sans, CM Typewriter, ...
- Variationen: **bold**, **bold extended**, *italic*, *slanted*, ...
- ebenfalls von Knuth entwickelt: METAFONT
- weitere Schriften: Linotype Syntax, Times, Helvetica, ...
- aber: viele mathematische Symbole nur in Computer Modern

Schriften, cont.

- *New Font Selection Scheme* (NFSS)
- vier Eigenschaften bestimmen Schrift:
 - Familie: Roman, Sans Serif, Typewriter
 - Serie: medium, **bold**
 - Form: upright, *italic*, *slanted*, SMALL CAPS
 - Größe: tiny, . . . , small, . . . , large, . . . , Huge
- Eigenschaften unabhängig voneinander
- aber nicht zu jeder Kombination Font vorhanden

Schriften auswählen

- zwei Varianten:

Befehl `\textsf{serifenlos}`
Deklaration `{\sffamily serifenlos}`

- NFSS-Anweisungen:

Familie `\textrm, \textsf, \texttt`
 `\rmfamily, \sffamily, \ttfamily`
Serie `\textmd, \textbf`
 `\mdseries, \bfseries`
Form `\textup, \textit, \textsl, \textsc`
 `\upshape, \itshape, \slshape, \scshape`

Schriften auswählen, cont.

- Größe nur per Deklaration: `{\small ganz klein}`

```
\tiny,...,\small,...,\large,...,\Huge
```

- manchmal noch zu finden: „alte“ Anweisungen

```
Das ist {\bf wichtig}!
```

- Kombinationen:

```
\textbf{das ist fett \textit{und kursiv}}  
\textsl{geneigt, \textbf{fett und \textsf{serifenlos}}}  
\textsl{geneigt und {\bfseries fett}}
```

Aufgaben

- Probiere die NFSS-Anweisungen aus. Versuche vor allem, mehrere Schrifteigenschaften gleichzeitig zu ändern.
- Nicht zu jeder Kombinationen von NFSS-Befehlen gibt es auch tatsächlich eine Schrift. Versuche, eine solche Kombination zu finden. In welcher Schrift wird der Text gesetzt? Wie weist \LaTeX auf das Problem hin?
- Was könnte der Grund dafür sein, daß man die Schriftgröße nur mit Deklarationen (`{\small ...}`) aber nicht mit Befehlen (`\textsmall{...}`) ändern kann?

- Die Anweisungen können leichter in Texten verwendet werden, während die Deklarationen besser für in Makro-Definitionen (siehe Folie 13) geeignet sind. Eine Änderung der Schriftgröße mitten im Satz kommt aber eher selten vor, daher hätten Anweisungen wie `\textsmall{...}` wenig Sinn gemacht.

```

cmr/bx/sc <- <- cmr/bx/n
cmr/bx/sl <- <- cmr/bx/n
cmr/m/sc <- <- cmr/m/sl
cmr/m/it <- <- cmr/m/sl
cmr/bx/it <- <- cmr/bx/n
cmr/bx/sl <- <- cmr/bx/n
cmr/bx/sc <- <- cmr/bx/n
cmr/bx/sl <- <- cmr/bx/n
cmr/bx/sc <- <- cmr/bx/n
cmr/bx/sl <- <- cmr/bx/n

```

- Ersetzungen:

roman	medium	<i>italic</i>	<i>slanted</i>	SMALL CAPS
Roman	medium	<i>italic</i>	<i>slanted</i>	small caps
Sans Serif	medium	<i>italic</i>	<i>slanted</i>	SMALL CAPS
Sans Serif	bold	upright	italic	slanted
Typewriter	medium	<i>italic</i>	<i>slanted</i>	SMALL CAPS
Typewriter	bold	upright	italic	small caps

- Die Computer Modern Familie:

Umgebungen

- Listen, Aufzählungen, Tabellen, Grafiken, ...
- Umgebungen „klammern“ Bereiche des Dokuments
- `\begin{umgebung} ... \end{umgebung}`
- bereits kennengelernt: `document`
- Umgebungen können geschachtelt werden
- Schachtelung muß passen:

```
\begin{ausser}          \begin{ausser}
  \begin{innen}        \begin{innen}
  ...                  ...
  \end{innen}          \end{ausser}
\end{ausser}           \end{innen}
```

Aufgaben

- Probiere die Umgebung `itemize` aus. Mit ihr kann man Listen erstellen:

```
\begin{itemize}
\item Huey
\item Dewie
\item Louie
\end{itemize}
```

- Versuche, mehrere `itemize`-Umgebungen zu schachteln, d.h. beginne innerhalb eines `\item`s eine neue Liste. Was ist zu beobachten? Wie tief lassen sich Listen auf diese Weise schachteln?

Es sind maximal vier Ebenen möglich, sonst bricht L^AT_EX die Übersetzung mit einer Fehlermeldung ab.

```
* hier wieder dritte Ebene
. ebenfalls vierte Ebene
. vierte Ebene
* immer noch dritte Ebene
* dritte Ebene
– auch zweite Ebene
– zweite Ebene
```

- Das Symbol vor jedem Listeneintrag variiert mit der Schachtelungstiefe:

Aufgaben

- Ersetze in einer Listenumgebung das Wort `itemize` durch `enumerate`:

```
\begin{enumerate}  
\item ...  
\end{enumerate}
```

Was passiert?

- Probiere, `enumerate`-Umgebungen ineinander zu schachteln.
- Mische `itemize`- und `enumerate`-Umgebungen miteinander.

- Die Einträge in der Aufzählung werden fortlaufend nummeriert, je nach Schachtelungstiefe mit unterschiedlichen Zahlen oder Symbolen:
 1. erste Ebene
 2. auch hier erste Ebene
 - a) zweite Ebene
 - b) auch zweite Ebene
 - i. dritte Ebene
 - ii. immer noch dritte Ebene
 - A. vierte Ebene
 - B. ebenfalls vierte Ebene
 - iii. hier wieder dritte Ebene
- Listen und Aufzählungen können beliebig gemischt werden. Das Limit von maximal vier ineinandergeschachtelten Umgebungen gilt für die beiden Arten von Listen unabhängig.

Zitate

Von Pythagoras, der im alten Griechenland lebte, ist uns folgender Satz überliefert:

```
\begin{quote}
```

Die Fläche des Quadrats über der Hypotenuse ist gleich der Summe der Flächen der Quadrate über den Katheten.

```
\end{quote}
```

Von Pythagoras, der im alten Griechenland lebte, ist uns folgender Satz überliefert:

Die Fläche des Quadrats über der Hypotenuse ist gleich der Summe der Flächen der Quadrate über den Katheten.

Aufgabe

- Setze das folgende Gedicht:

```
\begin{verse}
  Wer reitet so sp"at durch Wind und Nacht? \\
  Es ist der Vater. Es ist gleich acht. \\
  Im Arm den Knaben er wohl h"alt, \\
  er h"alt ihn warm, denn er ist erk"alt'.

  Halb drei, halb f"unf. Es wird schon hell. \\
  Noch immer reitet der Vater schnell. \\
  Erreicht den Hof mit M"uh und Not -- \\
  der Knabe lebt, das Pferd ist tot!
\end{verse}
```

Welche Aufgabe haben die \\ am Zeilenende, was bewirkt die Leerzeile?

- Die doppelten backslashes am Zeilenende erzeugen einen Zeilenumbruch. Man kann sie auch in "normalem" Text verwenden, aber dort sollte man besser `\TeX` den Zeilenumbruch überlassen. In einem Gedicht muß man die Formatierung natürlich vorgeben, damit es sich hinten reimt. . . .
Die Leerzeile trennt einzelne Strophen voneinander, verhält sich also ähnlich wie die Leerzeile zum Trennen von Absätzen.

Makros

- direkte Verwendung von Schriftbefehlen vermeiden
- „hervorheben“ statt „kursiv“
- `\emph{sehr} gut`: *sehr gut*
- `\emph{auch \emph{das} noch}`: *auch das noch*
- Makros und Umgebungen selber definieren

einfache Makros definieren

- Floskeln, Abkürzungen, exakte Schreibweisen:

```
\newcommand{\TFDom}{TechFak.Uni-Bielefeld.DE}  
\newcommand{\MFG}{Mit freundlichen Gr\"u\ss{}}en}
```

```
Meine Adresse lautet \texttt{juser@\TFDom}.
```

```
\MFG, Joe User
```

- Pakete (z.B. babel) erst nach `\begin{document}` aktiv
- Makros flexibel definieren

Aufgaben

- Definiere und verwende ein Makro, das den Ausdruck „Hypertext Markup Language“ erzeugt. Schreibe den Satz

Die Hypertext Markup Language wurde von Tim Berners-Lee erfunden.

mit Hilfe dieses Makros.

- Was ist an dieser Makrodefinition problematisch?

```
\newcommand{\ecoli}{\itshape E.~coli}
```

Wie kann man sie verbessern?

- Was passiert, wenn Du versuchst, dieses Makro zu definieren?

```
\newcommand{\dots}{...}
```

- Das Makro `\dots` ist bereits definiert, daher bricht L^AT_EX die Übersetzung mit einer Fehlermeldung ab. Man kann Makros mit dem Befehl `\renewcommand` umdefinieren, allerdings ist das in diesem Fall nicht ratsam.

```
\newcommand{\ecoli}{\itshape E.~coli}
\newcommand{\ecoli}{\textit{E.~coli}}
```

Die beiden folgenden Definition verhindern dies:

```
Das Bakterium \ecoli{} wird auch als das Haustier der Genetiker ...
```

- Der Font-Wechsel kann sich auf nachfolgenden Text auswirken:

```
Die \HTML{} wurde von Tim Berners-Lee erfunden.
\newcommand{\HTML}{Hypertext Markup Language}
```

- einfache Makrodefinition:

Makros mit Argumenten

- aus einem Reiseführer:

```
\newcommand{\City}[1]{\textsc{#1}}  
\newcommand{\CityAt}[2]{\textsc{#1 (#2)}}
```

Die Reise f"uhrt von \City{K"oln} und \City{Bonn} den Rhein entlang "uber \City{Koblenz} nach \City{Wiesbaden} und von dort weiter nach \CityAt{Frankfurt}{am Main}.

- [1], [2]: Anzahl Parameter
- #1 erster Parameter, #2 zweiter Parameter, ...
- maximal 9 Parameter

Aufgaben

- Überzeuge Dich davon, daß die gezeigten Makros funktionieren:

```
\newcommand{\City}[1]{\textsc{#1}}  
\newcommand{\CityAt}[2]{\textsc{#1 (#2)}}
```

... von `\City{K"oln}` nach `\CityAt{Frankfurt}{am Main}`.

- Was passiert, wenn Du die Makros mit der falschen Anzahl Parameter verwendest:

... von `\CityAt{K"oln}` nach `\City{Frankfurt}{am Main}`.

- Bei der "falschen" Verwendung gibt es keine Warnungen oder Fehlermeldungen. Das Makro `\CityAt` nimmt sich den nächsten Buchstaben (das "n" von "nach") als zweites Argument, das Makro `\City` ignoriert das zweite Paar Klammern einfach, der Text "am Main" wird als ganz normaler Text gesetzt.

Aufgaben

- Definiere ein Makro `\email`, um EMail-Adressen zu setzen.

```
\email{webmaster}{Uni-Bielefeld.DE}
```

soll die Ausgabe `webmaster@Uni-Bielefeld.DE` erzeugen.

- Definiere ein Makro `\tfemail`, um EMail-Adressen innerhalb der Technischen Fakultät zu setzen.

```
\tfemail{juser}
```

soll die Ausgabe `juser@TechFak.Uni-Bielefeld.DE` erzeugen. Verwende das Makro `\email` bei der Definition von `\tfemail`.

```
\newcommand{\email}[2][\textt{#1@#2}]{\textt{#1@#2}}
\newcommand{\tfemail}[1][\email{#1}{TechFak.Uni-Bielefeld.DE}]
```

Makros, cont.

- sinnvolle, sprechende Namen verwenden
- Verwendung beschreiben, nicht Formatierung
- `\City{...}` statt `\Smallcaps{...}`
- Namenskollisionen vermeiden
- z.B. durch gemischte Groß-/Kleinschreibung

Umgebungen selber definieren

- ähnlich zu Makros
- gleiche Regeln für Namenswahl
- Was passiert bei `\begin{...}`, was bei `\end{...}`?
- % Kommentarzeichen von TeX
- hier: Unterdrückung unbeabsichtigter Zeilenumbrüche

einfache Umgebung

```
\newenvironment{Merksatz}
{\begin{quote}%
 \large\bfseries}
{\end{quote}}
```

```
Chemiker wissen:
\begin{Merksatz}
  Nie das Wasser in die S"auere,
  sonst geschieht das Ungeheure.
\end{Merksatz}
```

Chemiker wissen:

**Nie das Wasser in
die Säure, sonst ge-
schieht das Unge-
heure.**

Hausaufgabe

- Sieh Dir (z.B. in der „[L^AT_EX 2_ε-Kurzbeschreibung](#)“) die folgenden Umgebungen an:
 - `description`
 - `center`, `flushleft`, `flushright`
 - `tabbing`