Einführung in das Textsatzsystem (LA)TEX

Vorlesung über (LA)TEX im Sommersemester 2009

universitatis studii heydelbergensis

Arno Trautmann

Heidelberg

Vorlesung 12, 3. Juli 2009

Teil XII

Typographische Feinheiten
Pakete selbst schreiben

„normale“ Typographie – Nachtrag

- Setzen echter Anführungszeichen: „ “ statt
- mit TEX: `\glqq \grqq`
- mit LATEX und `babel`: `` `" '" ``
  (" ist active char in babel und wird auch für andere Zwecke verwendet)
- mit XELATEX: „ “

Zweispaltiger Satz – Nachtrag

- Dokumentoption `twocolumn` setzt zweispaltig

Zweispaltiger Satz – Nachtrag

- Dokumentoption `twocolumn` setzt zweispaltig
- Umgebungen `figure*` und `table*` gehen über beide Spalten
- Vorsicht mit Reihenfolge bei Verwendung mit und ohne *!

Mailinglisten – Nachtrag

- tex-d-l – deutschsprachige Mailingliste
- texhax – internationale Liste
- newsgroups: (d.)c.t.t.
  (Benötigt Zugang zum Usenet, z. B. über Uni Heidelberg)
- Umgangsformen beachten!

Typographie

- Makro- und Mikrotypographie
- Makrotypographie: Anordnung von Text auf einer Seite
- Textumbruch und Absatzausrichtung
- Anordnung von Bildern, konsistente Abstände
- passende Wahl von Schriften
- (Kerning)

Mikrotypographie

- typographische Feinheiten auf Buchstaben- oder „Subbuchstabenniveau“:
- character protrusion
- font expansion
- the adjustment of interword spacing
- additional kerning
- hyphenatable letterspacing (tracking)
- possibility to disable all or selected ligatures.

(Aus der `microtype`-Dokumentation)

Mikrotypographie

- typographische Feinheiten auf Buchstaben- oder „Subbuchstabenniveau“:
- character protrusion
- font expansion
- the adjustment of interword spacing
- additional kerning
- hyphenatable letterspacing (tracking)
- possibility to disable all or selected ligatures.

(Aus der `microtype`-Dokumentation)
Das typolexikon bietet eine ausführlichere Definition

typokurz

- Dokument `typokurz.pdf` gibt kurze, übersichtliche Anleitung zu typographischen Tips
- im Internet unter dem Namen zu finden
- Textauszeichnungen, Striche, Abkürzen etc.

microtype

- Paket microtype kümmert sich um alle mikrotypographischen Effekte
- benötigt pdfTEX als Engine
- für alle Features: aktuelle Engine (pdfTEX $>$ 1.40)
- LuaTEX

Die minimale Klasse

minimal.cls

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{minimal}[2001/05/25 Standard LaTeX minimal class]

\renewcommand\normalsize{\fontsize{10pt}{12pt}\selectfont}

\setlength{\textwidth}{6.5in}
\setlength{\textheight}{8in}

\pagenumbering{arabic}  % but no page numbers are printed because:
\pagestyle{empty}       % this is actually already in the kernel
```

Eine absolut minimale Klasse

```
\renewcommand\normalsize{\fontsize{10}{12}\selectfont}
```

⇒ Definition von `\normalsize` entscheidend!

Klassen Laden

- normalerweise Klassen nicht „from scratch“ schreiben

⇒ `\LoadClass{scrartcl}` lädt `scrartcl` mit allen Einstellungen der Klasse

- danach weitere Funktionalität/Einstellung möglich

Klassen Laden

- normalerweise Klassen nicht „from scratch“ schreiben

⇒ `\LoadClass{scrartcl}` lädt `scrartcl` mit allen Einstellungen der Klasse

- danach weitere Funktionalität/Einstellung möglich
- Pakete einbinden: `\RequirePackage{paket}` statt `\usepackage{paket}`
- Befehle definieren: `\DeclareRobustCommand` statt `\newcommand`
- restlicher Code wie im normalen LaTeX-Dokument
- nützlich: `\AtBeginDocument{}`, `\AtEndDocument{}`

literate programming

- Programmcode sollte mit Dokumentation zusammen geschrieben werden

$\Rightarrow$ möglichst viele Kommentare zum Code schreiben!

literate programming

- Programmcode sollte mit Dokumentation zusammen geschrieben werden

$\Rightarrow$ möglichst viele Kommentare zum Code schreiben!

- viele Kommentare bremsen TEX aus
- Klasse schreiben: Code+Dokumentation
  Klasse verwenden: reinen Code
- dafür: DocStrip

DocStrip

- Programm DocStrip entfert alle Kommentare aus einem TEX-Dokument
- selbst in TEX implementiert

DocStrip

```
\documentclass[english]{ltxdoc}
\begin{document}
\DocInput{mathphysletter.dtx}
\end{document}

% Programmcode \protect \char "2026\relax
```

Code

Code muss immer in der folgenden Umgebung eingeschlossen sein:

```
%␣␣␣␣\begin{macrocode}
\DeclareRobustCommand\Kuchen{\kuchen@ist@lecker}
%␣␣␣␣\end{macrocode}
```

dtx und ins

- .dtx-Datei enthält Code und Dokumentation
- .ins-Datei enthält Anweisungen zur Codeerzeugung
- Kompilieren der .dtx liefer Dokumentation, Kompilieren der .ins Klasse/Paket

zukünftige Entwicklung

- nähere Zukunft: LuaTEX
- Paketerstellen auf Basis der Skriptsprache lua
- Standardprogrammieraufgaben wesentlich einfacher als mit TEX
- Einbinden externer Bibliotheken einfach möglich

zukünftige Entwicklung

- nähere Zukunft: LuaTEX
- Paketerstellen auf Basis der Skriptsprache lua
- Standardprogrammieraufgaben wesentlich einfacher als mit TEX
- Einbinden externer Bibliotheken einfach möglich
- ferne Zukunft: LATEX3

LaTeX3

- geplanter, neuer LaTeX-Kernel
- Sinn: Abstraktion von TeX
- Vereinfachung oft verwendeter Strukturen zum „Programmieren“
- Lernen aus Fehlern/Problemen von LaTeX 2_ε

LaTeX3
Befehlsstruktur

- Stelle des @ wird vom _ eingenommen
- alle Befehle bekommen strukturiert vergebene Namen
- Namenskollisionen ausgeschlossen, da Benennung nach Modulen
- Angabe von Argumenten, getrennt durch Doppelpunkt

LaTeX3
Befehlsstruktur

- Syntax für Parameter:
  `\access_module_description_type`
- Syntax für Funktionen:
  `\hmodule_description:arg-spec`
- Bsp.: `\seq_push:Nn`

⇒ siehe `texdoc expl3`

LaTeX3
Sinn

- „wrapper“ für alle TeX-Primitiven
- leichteres Programmieren durch strukturierte Befehle
- nachvollziehbare Befehlsnamen
- „Überladen“ von Funktionsnamen

Was man mitnehmen sollte

- wofür ist LaTeX gut, wann sollte man es verwenden und warum ist es oft besser als andere Systeme?
- wie ist der prinzipielle Aufbau eines LaTeX-Dokumentes?
- wo kann es Probleme geben bei unterschiedlichen Systemen?
- wo findet man Hilfe?
- was ist typographisch schön und was ist „schön“?
- was verbessert die Lesbarkeit?

„I hope to die before I *have* to use Microsoft Word.“
Donald E. Knuth, 02. 10. 2001 in Tübingen.

Happy TEXing!