

L^AT_EX-Kurs: Nützliche Pakete

Carl Georg Heise

Technische Universität München

17. April 2009

- 1 Das hyperref-Paket
 - Überblick
 - Befehle
 - Paketoptionen
- 2 Das listings-Paket
 - Überblick
 - Befehle
 - Istset-Optionen
 - Sprachen
- 3 Kleine Pakete
 - upgreek
 - csquotes
 - textcomp
 - color

Übersicht

- 1 Das hyperref-Paket
 - Überblick
 - Befehle
 - Paketoptionen
- 2 Das listings-Paket
 - Überblick
 - Befehle
 - Istset-Optionen
 - Sprachen
- 3 Kleine Pakete
 - upgreek
 - csquotes
 - textcomp
 - color

Das *hyperref*-Paket

Funktionen

- Erzeugt Links innerhalb des Dokuments um schnell hin und her zu navigieren.
- Kann automatisch Links auf erwähnte Webseiten erzeugen.
- Fügt der pdf-Datei Informationen zu Autor, Titel... hinzu.

Einbinden

```
\usepackage{hyperref}
```

Sollte als eines der *letzten* Pakete eingebunden werden!

- Das Paket erzeugt weitgehend automatisch Links innerhalb des Dokuments.
- Diese sind in der pdf-Datei standardmäßig mit einem farbigen Rahmen hinterlegt.

Achtung

Die Dokumentklasse sollte folgendermaßen aussehen, sonst kann *hyperref* Probleme bereiten.

```
\documentclass[pdfTeX,paper=<Größe>,...]{scr...}
```

Links

Hyperlinks

```
\href{URL}{Text}
```

Verlinkt Text auf eine URL.

Beispiel

```
\href{http://www.google.com/search?q=LaTeX}%  
{Google -- \LaTeX}
```

Google – LaTeX

Links

Dokumentinterne Links

```
\hypertarget{Name}{Text}  
\hyperlink{Name}{Text}
```

Erstellt einen dokumentinternen Link auf den Text in *hypertarget*.

Beispiel

```
Integer venenatis tincidunt. \hypertarget{name}{Pellentesque}  
ac velit. Proin volutpat feugiat justo? Aliquam cursus  
pharetra justo! Aenean interdum accumsan erat. Ut venenatis  
libero et tellus. Fusce dui erat, rhoncus pellentesque,  
lobortis ac, pretium eu, nisi? \hyperlink{name}{Etiam} aliquam  
neque vitae ante...
```

Paketooptionen

hypersetup

```
\hypersetup{...}
```

Ermöglicht viele Einstellungen beim hyperref-Paket. Muss vor dem Anfang des Dokuments stehen.

Allgemeine Struktur

```
\hypersetup{  
Option1=Wert1,  
Option2=Wert2,  
...  
}
```

hypersetup-Optionen

- `breaklinks=true` – Links dürfen über mehrere Zeilen gehen.
- `pdftex=true` – Wenn nicht beim Laden der Klasse deklariert...
- `colorlinks=true` – Färbt den *Text* von Links
- `linkcolor=red, urlcolor=cyan1...`
- `bookmarksnumbered=true` – Nummeriert Lesezeichen in Acrobat

¹Benötigt das *color*-Paket

hypersetup-Optionen

- `linkbordercolor={1 0 0}` – Farbe von Linkrahmen als RGB-Color
- `urlbordercolor={0 1 1}` – Analog für URLs
- `pagebordercolor={1 1 0}` – Analog für Links auf Seiten
- `pdfborder={0 0 0}` – Entfernt Rahmen um Links
- `pdftitle={Titel}` – Der Titel des pdf-Dokuments
- `pdfauthor, pdfsubject, pdfcreator, pdfstartpage...` – Analog

Übersicht

1 Das hyperref-Paket

- Überblick
- Befehle
- Paketooptionen

2 Das listings-Paket

- Überblick
- Befehle
- Istset-Optionen
- Sprachen

3 Kleine Pakete

- `upgreek`
- `csquotes`
- `textcomp`
- `color`

Das listings-Paket

Funktionen

- Ermöglicht es sehr einfach, Quelltext in L^AT_EX darzustellen
- *listings* ist äußerst praktisch, wenn dieser viele Sonderzeichen enthält.
- Bringt einheitliche Formatierung des Quelltexts und (evtl.) Syntaxhighlighting.

Einbinden

```
\usepackage{listings}
```

Festlegen der Sprache

Außerdem wird folgender Befehl benötigt

```
...
\begin{document}
\lstset{language=Java}
%Oder eine andere Sprache
...
\end{document}
...
```

Wieso nicht *Verbatim*?

Standard TeX

```
\verb=public void main(){ bla }=
public void main(){ bla }
```

- Innerhalb von Verbatim können alle(!) Sonderzeichen beliebig verwendet werden.
- Verbatim endet mit dem Zeichen, mit dem es eingeleitet wurde
- Verbatim ist grundsätzlich unformatiert und sehr unflexibel
- Das listings-Paket bietet deutlich mehr Optionen und ist außerdem kein „Low-Level-TeX“.

Wieso nicht *Verbatim*?

Analoger Befehl

linline erzeugt kurze „Code-Schnipsel“

```
Lorem \linline=public void main(){ bla }= ipsum
```

```
Lorem public void main(){ bla } ipsum
```

Für längeren Code

Die *lstlisting*-Umgebung

```
Lorem ipsum
\begin{lstlisting}
private logPrint(String s) {
    Class cal=Class();
    ...}
\end{lstlisting}
dolor et amet
```

Ausgabe

```
Lorem ipsum
private logPrint(String s) {
    Class cal=Class();
    ...}
dolor et amet
```

Der *lstinputlisting*-Befehl

```
\lstinputlisting{demo.java}
```

Teile der Datei

```
\lstinputlisting[lastline=10]{demo.java}
```

Gibt die Datei nur bis Zeile 10 aus.

Allgemeine Struktur

```
\lstset{  
Option1=Wert1,  
Option2=Wert2,  
...  
}
```

Die neuen Optionen werden übernommen, ab dem Punkt wo `\lstset` steht.

Optionen für *lstset* I

- `inputencoding=latin1` – Inputencoding (auch UTF8...)
- `basicstyle=\small` – Stil des Codes (hier kleiner)
- `keywordstyle=\bfseries` – Stil von Keywords (hier fett)
- `commentstyle=\itshape` – Stil von Kommentaren (hier kursiv)
- `stringstyle=\ttfamily` – Stil der Strings (hier im Typewriterstil)

Optionen für *lstset* II

- `showstringspaces=true` – Zeigt □ für Leerzeichen
- `tabsize=3` – Größe eines Tabstopps
- `extendedchars=true` – Benötigt für ä, ö, ü...
- `backgroundcolor=\color{yellow}` – Hintergrundfarbe (benötigt *color*-Paket)

Oftmals ist es nützlich, Quellcode mit Zeilennummern zu versehen.

Zeilennummern

- `numbers=left` – Zeilennummern am linken Rand
- `numbers=none` – Deaktiviert Zeilennummern
- `numberstyle=\tiny` – Stil der Zeilennummern
- `stepnumber=5` – Schrittweite der Nummerierung

Zeilennummern – Weitere Optionen

```
\begin{lstlisting}[firstnumber=1000]
...
```

Beginnt die Zeilennummern bei 1000

```
\begin{lstlisting}[firstnumber=last]
...
```

Beginnt bei der letzten Zeilennummer

Sprache

Achtung

Damit *listings* die Programmiersprache (und damit die Keywords) erkennt, muss unbedingt die verwendete Sprache angegeben werden.

```
language=[<Dialekt>]<Sprache>
```

Sprache

Die folgenden Sprachen sind bereits im *listings*-Paket implementiert.

Unterstützte Sprachen

- [ANSI]C, [Objective]C, [Sharp]C...
- [ISO]C++, [ANSI]C++...
- Cobol, Delphi, [95]Fortran, Pascal
- Java, Python, Perl, R, Ruby
- HTML, PHP, XML
- Mathematica, Matlab
- und viele mehr

Mehr Keywords

Falls ihr mehr Keywords benötigt (z. B. eigene Funktionen) gibt es folgende *lstset*-Option

morekeywords

```
morekeywords={myFunction,myClass}
```

Übersicht

- 1 Das hyperref-Paket
 - Überblick
 - Befehle
 - Paketooptionen
- 2 Das listings-Paket
 - Überblick
 - Befehle
 - Istset-Optionen
 - Sprachen
- 3 Kleine Pakete
 - upgreek
 - csquotes
 - textcomp
 - color

Das upgreek-Paket

Funktionen

- Normalerweise sind griechische Buchstaben in Formeln immer kursiv ($\alpha, \varepsilon \dots$)
- `\usepackage{upgreek}` liefert Befehle für aufrechte griechische Buchstaben

Das upgreek-Paket

Beispiel

```
\[\upalpha, \upvarepsilon, \Upphi, \Upomega\]
```

$\alpha, \varepsilon, \Phi, \Omega$

Schöne Anführungszeichen – Das *csquotes*-Paket

Funktionen

- Einfache doppelte Gänsefüßchen " sind im Allgemeinen nicht sehr schön anzusehen.
- In \LaTeX lässt sich dies zwar bewerkstelligen, ist aber mühsam.
- Außerdem erfordern bestimmte Sprachen (z. B. Schweizerdeutsch) andere Anführungszeichen.
- Das *csquotes*-Paket löst diese Probleme.
- *csquotes* kann außerdem verschachtelte Anführungszeichen.

Wichtig

csquotes benötigt T1-Fontencoding für die Anführungszeichen.

Das *csquotes*-Paket

Einbinden

```
\usepackage[style=<Stil>]{csquotes}  
\enquote{Text}
```

Stil bestimmt die Art der Anführungszeichen.

Stil ändern

```
\setquotestyle{<Stil>}
```

Ändert den Stil der Anführungszeichen im Dokument.

Das *csquotes*-Paket

Beispiel

```
\usepackage[style=german]{csquotes}
```

...

Dieser `\enquote{Text im \enquote{Text}}` steht
in Anführungszeichen.

Dieser „Text im ‚Text‘“ steht in Anführungszeichen.

Stile der Anführungszeichen

Übersicht

- „german“ – Normales Deutsch
- «swiss» – Schweizerdeutsch
- “american” – Englisch (USA)
- ‘british’ – Englisch (UK)
- « french » – Französisch
- «spanish» – Spanisch

Längere Texte

textquote

```
\textquote[<Quelle>]{<Text>}
```

Zitiert einen längeren Text

Beispiel

```
\textquote[Cicero]{Lorem Ipsum...}
```

„Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus leo mauris, tincidunt nec, fringilla in, consectetur sit amet, sapien. Integer lectus justo, mattis sit amet, iaculis convallis, vestibulum nec, odio. Morbi dapibus adipiscing ligula. Aenean aliquam erat id urna.“ (Cicero)

Weitere Symbole mit dem *textcomp*-Paket

Funktionen

- Verschönert ein paar vorhandene Symbole (z. B. © `\textcopyright`)
- Fügt neue Symbole (u. a. Währungen) hinzu

Das *textcomp*-Paket

Währungen

£	<code>\textsterling</code>	₫	<code>\textdong</code>
\$	<code>\textdollar</code>	¥	<code>\textyen</code>
₱	<code>\textpeso</code>	₩	<code>\textwon</code>

Andere Symbole

©	<code>\textcopyright</code>	®	<code>\textregistered</code>
™	<code>\texttrademark</code>	№	<code>\textnumero</code>
‰	<code>\textperthousand</code>	¶	<code>\textparagraph</code>
∞	<code>\textmarried</code>		

Im Internet gibt es noch viele weitere Pakete mit Symbolen.

Bringe Farbe ins Spiel mit dem *color*-Paket!

Funktionen

- Erlaubt **farbigen** Text.
- Teilweise für bestimmte Optionen anderer Pakete benötigt (z. B. `hyperref`)

Einbinden

```
\usepackage{color}
```

Funktionen

```
\color{orange} Text \rule{.2cm}{.2cm} orange  
bis zum nächsten Wechsel.  
\textcolor{purple}{Lila Text}  
\pagecolor{yellow} %Gelbe Seite
```

Text ■ orange bis zum nächsten Wechsel. Lila Text

Color-Boxen

```
\colorbox{cyan}{Cyane Box}  
\fcolorbox{orange}{blue}{Blaue Box orangem Rand}
```

Cyane Box Blaue Box orangem Rand

Eigene Farben

Definition

```
\definecolor{salmon}{rgb}{1,0.8,0.7}
```

Alternativ auch `cmymk` oder `gray`

Dieser Text ist lachsfarben.

Bei einmaliger Verwendung

```
\color[rgb]{1,0.5,0.7}  
Lorem ipsum dolor...
```

Lorem ipsum dolor...