Programmierung mit (La)TEX

...und anderen Programmiersprachen

Wolfgang Dautermann

FH JOANNEUM

Grazer Linuxtage 2014

(Eigene) Kommandos (um)definieren

...ist vermutlich bekannt?

\newcommand und \renewcommand [LaTeX]

```
\newcommand{\meinbefehl}{Inhalt}
\renewcommand{\meinbefehl}{neuer Inhalt}

z.B.:
\newcommand{\mfg}{Mit freundlichen Grüßen,}
\mfg Wolfgang Dautermann
```

Mit freundlichen Grüßen,Wolfgang Dautermann

$\Rightarrow$ Leerzeichenfehler. Whitespace kennzeichnet das Befehlsende!

Leerzeichenproblembehebung(-sversuche)

...alles unschön

durch Kennzeichnung des Befehlsendes {} oder Umdefinition

```
\mfg{} Wolfgang Dautermann
\renewcommand{\mfg}{Mit freundlichen Grüßen}
\mfg, Wolfgang Dautermann
\renewcommand{\mfg}{Mit freundlichen Grüßen, }
\mfg Wolfgang Dautermann
```

Mit freundlichen Grüßen, Wolfgang Dautermann
Mit freundlichen Grüßen, Wolfgang Dautermann
Mit freundlichen Grüßen, Wolfgang Dautermann

Leerzeichenproblembehebung

Verwendung des Pakets `xspace`

`xspace` erkennt, wann ein Leerzeichen sinnvoll/notwendig ist

```
\newcommand{\gb}{Grossbritannien}
Ich fliege nach \gb.  \\ % kein Leerzeichen
Die Hauptstadt von \gb ist London. \\% Leerzeichen.
\renewcommand{\gb}{Grossbritannien\xspace}
Ich fliege nach \gb. \\ % kein Leerzeichen
Die Hauptstadt von \gb ist London. \\ % Leerzeichen.
```

Ich fliege nach Grossbritannien.
Die Hauptstadt von Grossbritannienist London.
Ich fliege nach Grossbritannien.
Die Hauptstadt von Grossbritannien ist London.

Kommandos mit Argumenten

1-9 notwendige Argumente

```
\renewcommand{\meinbefehl}[narg]{Inhalt #1 #2 ... #9}

z.B.:
\newcommand{\zugfahrt}[2]{Ich fahre von #1 nach #2.}
\zugfahrt{Graz}{Wien}
```

Ich fahre von Graz nach Wien.

Kommandos mit optionalem Argument

ein optionales Argument ist möglich

Defaultwert für #1 angeben.

```
\renewcommand{\meinbefehl}[narg][Defaultwert #1]{...}


z.B.:
\newcommand{\fahrt}[3][Zug]{#1fahrt von #2 nach #3.}
\fahrt{Graz}{Wien} \\
\fahrt[Auto]{Graz}{Wien}
```

Zugfahrt von Graz nach Wien.
Autofahrt von Graz nach Wien.

Eigene Umgebungen

newenvironment, renewenvironment

Analog zu \(re)newcommand:

```
\newenvironment{myenv}{<startbefehle>}{<endbefehle>}
```

stellt

```
\begin{myenv} ... \end{myenv}
```

zur Verfügung.

Variablen

Stringvariablen

...wurden grad behandelt. Makros.

Stringvariablen

```
$mystring = "Abc";

\newcommand{mystring}{Abc}
```

...und beim nächsten mal halt `\renewcommand`.

Variablen

„Integervariablen“ / Counter

Counter

```
\newcounter{MeinCounter}       % Deklaration

\stepcounter{MeinCounter}      % $i++

\addtocounter{MeinCounter}{n} % $i=$i+n (auch negativ)

\setcounter{MeinCounter}{n}    % $i=n

\value{MeinCounter}  % Wert (zum Rechnen, nicht zur Ausgabe)
```

Etliche Counter sind standardmässig definiert:

`page, section, subsection, enumi, enumii, equation, ...`

Variablen

„Integervariablen“ / Counter – Ausgabe

Counter

```
\theMeinCounter / \arabic{MeinCounter}   1, 2, 3, ...
\alph{MeinCounter}                        a, b, c, ...
\Alph {MeinCounter}                      A, B, C, ...
\roman{MeinCounter}                        i, ii, iii, ...
\Roman{MeinCounter}                        I, II, III, ...
```

Variablen

„Float-variablen“ / Längen

Counter

```
\newlength{MeineLaenge}       % Deklaration

\settolength{MeineLaenge}{f} % $i=f (mit Einheit! (mm, ...))

\addtolength{MeineLaenge}{f}  % $i=$i+f

0.6\Meinelaenge              % Multiplikation (0.6\textwidth)
```

Wahrheitswerte und `if`-Abfragen

Boolean-Variablen

```
\newboolean{myboolvar}          % Deklaration
\setboolean{myboolvar}{false} % Zuweisung

\boolean{myboolvar}             % Abfrage des Werts
```

Nutzung z.B. bei `if`-Abfragen mit dem Paket `ifthen`:

```
\ifthenelse{\boolean{boolvar}}{then-block}{else-block}
```

Weitere „Kontrollstrukturen“: `whiledo`

`whiledo` (Paket `ifthen`)

```
\newcounter{i}
\setcounter{i}{1}
\whiledo{\value{i} < 10}{
    \noindent = \arabic{i} \\
    \stepcounter {i}
}
```

Umfangreicheres Beispiel: 99 bottles of beer in LaTeX.

Fragile Befehle

`\protect` hilft dagegen...

- Befehle mit „moving Arguments“
- z.B. `\footnote`
- werden zum Glück weniger...

Live Demo...

Umständlich. Gehts einfacher?

Ja. z.B. mit:

- Lua
- Shell & Co.
- Gnuplot, Sage
- Lisp
- Python
- Perl
- ...

Lua: LuaLaTeX

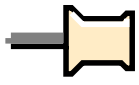
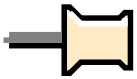
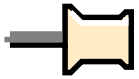
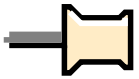
...die Zukunft von LaTeX

- Übersetzen mit `lualatex`
- Lua-code direkt einbinden mit: `\directlua`
- Achtung mit Lua-Kommentaren!
- → Lua-Files mit der Lua-Funktion `dofile()` einbinden

Live Demo...

LuaLaTeX-Beispiele

abspeichern und selbst ausprobieren..

1. Wert von π mit Lua ausgeben: 
2. mehrere Lua-Befehle hintereinander. Problem-Demo: Lua-Kommentar (--) einfügen: 
3. Beispiel 2 mit eigener Lua-Datei:  , 
4. Kommandos mit Lua definieren:  , 

Live Demo...

Shell und Co

aus Sicherheitsgründen[1] normalerweise sehr eingeschränkt bzw. deaktiviert...

Ausführen von Programmen

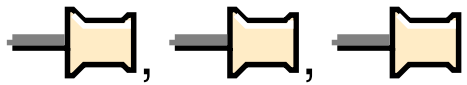
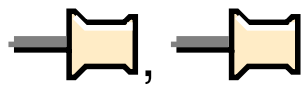
- Liste der zulässigen Programme: `shell_escape_commands` in `/usr/share/texmf/web2c/texmf.cnf`
- Alles erlauben mit der Option `--shell-escape`
- `\input{|"./meinprogramm.sh"}`
- (oder umständlicher:
  `\write18{./meinprogramm.sh > scriptoutput.tex}`
  `\input{scriptoutput.tex}`)

Live Demo...

[1] Are Text-Only Data Formats Safe? Or, Use This LaTeX Class File to Pwn Your Computer: http://cseweb.ucsd.edu/~hovav/dist/texhack.pdf

LaTeX-Beispiele mit externen Programmen

abspeichern und selbst ausprobieren..

1. Shell- und Perlskript aufrufen: 
   Übersetzen mit `pdflatex --shell-escape helloworld.tex`
2. Gleichungen und Integrale mit Maxima lösen und das Ergebnis ins Output-File automatisch übernehmen[2]: 
   Übersetzen mit `pdflatex --shell-escape helloworld.tex`
   Maxima muss logischerweise installiert sein.

[2] Warum soll ich selber rechnen? Dafür habe ich einen Rechner!

Lisp

LISP on TeX — A LISP interpreter on TeX

http://www.ctan.org/tex-archive/macros/latex/contrib/lisp-on-tex

Introduction

LISP on TeX is a LISP interpreter written only with TeX macros. It works as a style file of LaTeX.
LISP on TeX adopts static scoping, dynamic typing, and eager evaluation. We can program easily with LISP on TeX.

```
\usepackage{lisp-on-tex}
\lispinterp{
   LISP-CODE
}
```

Beispiel: Fakultätsberechnung in Lisp: 

Viel umfangreicheres Beispiel in der Doku: Mandelbrot-Berechung in Lisp.

Python

pythontex als MIDDLEPROCESSOR

Verwendung

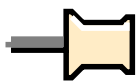
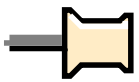
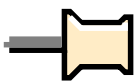
```
\usepackage{pythontex}
...
\py{PYTHON-EXPRESSION}
\pyc{PYTHON-CODE}
```

Aufruf

```
pdflatex ...
pythontex ...
pdflatex ...
```

Pythontex

abspeichern und selbst ausprobieren..

1. Python für math. Berechnungen nutzen[3]: 
2. Kommandos mit Python definieren: 
3. Graphiken generieren und verwenden: 

[3] wozu selber rechnen?

Perl

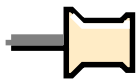
Verwendung

```
\usepackage{perltex}
...
\perlnewcommand
\perlrenewcommand
\perlnewenvironment
\perlrenewenvironment
\perldo
...
perltex [--latex=...] [--nosafe] [--makesty] xyz.tex
```

Aufruf standardmässig in SANDBOX (keine Module möglich, ...).
`--nosafe` ermöglicht vollen Zugriff. Oder: `--permit=*feature*`

Perltex

abspeichern und selbst ausprobieren..

Beispiel: eigene Kommandos: substr und SHA1-Berechnung(!) als LaTeX-Befehle 

Übersetzen mit:

```
perltex --nosafe --latex=pdflatex perltex1.tex
```

Weitere Möglichkeiten

Es gibt noch viele weitere interessante Pakete auf CTAN

- `arrayjobx`: Array-Datenstrukturen
- `datatool`: CSV-Dateien verarbeiten
- `sagetex`: Mathematikpaket SAGE verwenden.
- `gnuplottex`: Gnuplot verwenden.
- `stringstrings`: Stringmanipulation
- ...

Einfach mal auf CTAN suchen – es gibt Zusatzpakete für (fast) jedes Problem!

Vielen Dank

Fragen? (hoffentlich richtige...) Antworten!

Vielen Dank für Ihre Aufmerksamkeit

Wolfgang Dautermann

wolfgang.dautermann [AT] fh-joanneum.at