

# L<sup>A</sup>T<sub>E</sub>X

Programmieren - Eigene Definitionen

# LaTeX-Programmierung

- Das zu Grunde liegende TeX-System ist eine Programmiersprache.
- LaTeX ist eine Sammlung von Makros die in TeX programmiert sind.
- Es gibt entsprechend für den Anwender viele Elemente einer Programmiersprache um den Textsatz zu gestalten.
  - ▶ Variablen: alle Format-Eigenschaften sind in Variablen abgelegt, die programmatische geändert werden können.
  - ▶ Diese Variablen können über viele LaTeX-Befehle verändert werden.
  - ▶ Es gibt verschiedene Variablen-Typen (vor allem Längen und Zähler)
  - ▶ Auf TeX-Ebene: Kontrollstrukturen (if then, Schleifen, ...)

# Eigene Längen definieren

# Eigene Längen definieren

- Eigene Längen können mit dem Befehl `\newlength` definiert werden.
- Es gibt viele Situationen in denen das nützlich ist:
  - ▶ Spaltenbreiten von Tabellen für ein ganzes Dokument
  - ▶ Einheitliche Breite von Bildern (automatisch skaliert!)
  - ▶ ...

```
\newlength{\imgWidth}  
\setlength{\imgWidth}{0.8\textwidth}
```

# Eigene Längen definieren

- Erst wird die Länge mit `\newlength` erzeugt, dann mit den Längenänderungsbefehlen gesetzt.
- Im Beispiel wird eine neue Länge `\imgWidth` definiert.
- Alle Bilder im Text können mit dieser Variable auf 80% der Textbreite skaliert werden.
- Gefällt Ihnen das nicht mehr müssen Sie nur die 0.8 ändern, und alle Bilder werden neu skaliert.

```
\newlength{\imgWidth}  
\setlength{\imgWidth}{0.8\textwidth}
```

# Befehle

# `\newcommand` I

- Mit `\newcommand` lassen sich eigene Befehle definieren.
- In der gezeigten Form dienen Sie als Abkürzung für häufig auftretende Befehle, die sonst umständlich einzutippen wären.

Der Name des neuen Befehls.  
Mit `\` eingeben!



```
\newcommand{\name}{definition}
```



Die Definition des Befehls

## Beispiele

```
\newcommand{\zB}{z.\,B.\,@}  
\newcommand{\abk}{Abkürzung}
```

# `\newcommand` II

- Die fortgeschrittene Variante von `\newcommand` gebraucht **Argumente**.
- So wird der selbst-definierte Befehl zur Funktion.
- In der Definition wird über `#1` auf das erste Argument zugegriffen, `#2` das zweite, etc.
- Das geht für maximal 9 Argumente.

Der Name des neuen Befehls.  
Mit `\` eingeben!

Anzahl Argumente

```
\newcommand{\name}[1]{definition}
```

The diagram shows the LaTeX command `\newcommand{\name}[1]{definition}` enclosed in a box. Two arrows point from the text above to parts of the code: one from 'Der Name des neuen Befehls. Mit \ eingeben!' to `\name`, and another from 'Anzahl Argumente' to `[1]`. A third arrow points from 'Die Definition des Befehls' to `{definition}`.

Die Definition des Befehls

## Beispiel

```
\newcommand{\fett}[1]{\textbf{#1}}
```



# `\newcommand` II

Der Name des neuen Befehls.

Mit `\` eingeben!

Anzahl Argumente

Maximal 9 möglich



```
\newcommand{\name} [2] {definition}
```



Die Definition des Befehls. Innerhalb der Definition stellen `#1` und `#2` das jeweilige Argument dar.

## Beispiel

```
\newcommand{\einheit} [2] {  
  #1 \, #2 \@  
}
```

# Aufgabe

- Definieren Sie sich eine eigene Betonung, die Sie `\zB` für Fremdwörter oder Fachvokabeln gebrauchen wollen.
- Wie wäre es mit `\franz` für französische Vokabeln? Und `\sissi` für österreichische?

# Aufgabe

- Definieren Sie einen Befehl

`\anschrift` der drei

Argumente entgegen

nimmt:

- ▶ Name
- ▶ Straße
- ▶ Ort

```
\newcommand{\anschrift}[3]{  
    ...  
}
```

# Aufgabe

- Definieren Sie einen Befehl

`\anschrift` der drei

Argumente entgegen

nimmt:

- ▶ Name
- ▶ Straße
- ▶ Ort

Mein Beispiel

```
\newcommand{\fett}[1]{\textbf{#1}}
\newcommand{\anschrift}[3]{
  \newline
  \fbox{\textsc{Name}: \textbf{#1},
  \textsc{Straße}: \textbf{#2},
  \textsc{Ort}: \textbf{#3}}
}
```

# Weitere `\new...` Befehle

- `\newenvironment`
- `\newcolumnntype`
- Die lernen wir in Programmieren II kennen.

# `\renewcommand`

- Bestehende Befehle werden mit `\renewcommand` verändert.
- Die Syntax ist die gleiche wie bei `\newcommand`
- Auch Argumente können übergeben werden.
- LaTeX verhindert so, dass bestehende Befehle aus Versehen überschrieben werden.

## Definition

```
\renewcommand{\name}[1]{definition}
```

## Beispiel

```
\renewcommand{\em}{\bfseries}
```

Anmerkung: Im Beispiel geht allerdings die Schalter-Logik verloren.

# Zähler

# Zähler

## Liste vordefinierter Zähler

- Zähler sind wie Längen ein Variablentyp mit denen in LaTeX programmiert werden kann.
- Mit Zählern (engl. *counter*) regelt LaTeX (fast) alle Verweise im Dokument.
- Wie bei Längen
  - können eigene Zähler definiert werden.
  - werden die Zähler mit diversen Befehlen bearbeitet.

- part
- chapter
- section
- subsection
- subsubsection
- paragraph
- subparagraph
- page
- equation
- figure
- table
- footnote
- mpfootnote

Für Listen:

- enumi
- enumii
- enumiii
- enumiv



# Zähler-Befehle

- Die Befehle sind einfach und selbsterklärend.
- Seltsam: Zähler werden mit `\the` angezeigt, aber ohne `,` !
- Es können eigene Zähler für beliebige Sachen definiert werden.

```
\stepcounter{enumi}
\addtocounter{section}{zahl}
\setcounter{equation}{zahl}

\newcounter{numDoener}

% zeigt formatierten Text an
\thenumDoener % nicht \the\numDoener

% gibt den Wert unformatiert aus
% zum Gebrauch in Rechnungen
\value{numDoener}
```

# Aufgabe

- Lassen Sie sich im Text den Abschnitt, Paragraphen und Seitenzahl anzeigen.
- Erzeugen Sie mehrere Unterabschnitte und lassen diese bei 10 starten.
- Definieren Sie einen eigenen Zähler und den Sie für eine Beispieltext nach jedem ‚und‘ hochzählen.
- Fortgeschritten: erzeugen Sie einen neuen Befehl `\seitenund` der die Seitenzahl und den aktuellen Stand von `und` kombiniert und darstellt.

```
\stepcounter{enumi}
\addtocounter{section}{zahl}
\setcounter{equation}{zahl}

\newcounter{numDoener}

% zeigt formatierten Text an
\thenumDoener % nicht \the\numDoener

% gibt den Wert unformatiert aus
% zum Gebrauch in Rechnungen
\value{numDoener}
```

# Zähler-Darstellung

Bezeichnung	Befehl	Darstellung
Arabisch	<code>\arabic</code>	1. 2. 3. ...
Römisch klein	<code>\roman</code>	i. ii. iii. ...
Römisch groß	<code>\Roman</code>	I. II. III. ...
Alphabetisch klein	<code>\alph</code>	a. b. c. ...
Alphabetisch groß	<code>\Alph</code>	A. B. C. ...

# Zähler-Darstellung

- Zur Änderung *aller* Aufzählungen des Dokumentes wird `\theenumi` (und `\theenumii`, ...) in der Präambel neu definiert.

```
\documentclass{article}

\renewcommand{\theenumi}{\Roman{enumi}}

\begin{document}

\begin{enumerate}
\item Räuber Hotzenplotz
\item Kasperl
\item Seppel
\item Großmutter
\item Wachtmeister Dimpfelmoser
\end{enumerate}

\end{document}
```

# Zähler-Darstellung

- Zur Änderung *einer einzigen* Aufzählung wird `\theenumi` (und `\theenumii`, ...) direkt in der entsprechenden Aufzählung neu definiert.

```
\documentclass{article}

\begin{document}

\begin{enumerate}
\renewcommand{\theenumi}{\Roman{enumi}}
\item Räuber Hotzenplotz
\item Kasperl
\item Seppel
\item Großmutter
\item Wachtmeister Dimpfelmoser
\end{enumerate}

\end{document}
```

# Aufgabe

- Ändern Sie die Nummerierung der Umgebung `enumerate` einmal
  - ▶ lokal
  - ▶ global

# Zähler-Darstellung

Das geht natürlich  
auch für die  
anderen Zähler.

## A Räuber Hotzenplotz

...

## B Großmutter Kaffemühle

Wie schon einleitend in Abschnitt A erwähnt ist der Räuber Hotzenplotz ein schlimmer Halunke...

I. Räuber Hotzenplotz

II. Kasperl

III. Seppel

IV. Großmutter

V. Wachtmeister Dimpfelmoser

Der wichtigste Protagonist ist in der Liste unter Punkt II zu finden.

Section:B

Paragraph B.0.0.0

Seite i

```
\documentclass{article}

\renewcommand{\thepage}{\roman{page}}
\renewcommand{\thesection}{\Alph{section}}

\begin{document}

\begin{enumerate}
\renewcommand{\theenumi}{\Roman{enumi}}
\item Räuber Hotzenplotz
\item Kasperl
\item Seppel
\item Großmutter
\item Wachtmeister Dimpfelmoser
\end{enumerate}

\end{document}
```

# Aufgabe

- Ändern Sie die Zählung der Seitenzahlen auf römisch (klein).
- Ändern Sie die Zählung der Abschnitte auf alphabetisch (Groß).