

ZfS-Kurs „ \LaTeX für Naturwissenschaftler“

Karin Halupczok

Email: Karin.Halupczok@math.uni-freiburg.de

SoSe 2009

Erste Zusatzthemensitzung:
Eigene Strukturen

<http://home.mathematik.uni-freiburg.de/halupczok/latex.html>

Eigene Befehle

Eigene Umgebungen

Eigene Zähler

Eigene Längen

Eigene Befehle definieren

Mit

```
\newcommand{\meinBefehl}[n]{Definition}
```

wird ein neuer Befehl definiert. Dabei ist n die Anzahl der Parameter (zwischen 0 und 9);

Eigene Befehle definieren

Mit

```
\newcommand{\meinBefehl}[n]{Definition}
```

wird ein neuer Befehl definiert. Dabei ist n die Anzahl der Parameter (zwischen 0 und 9); [0] kann entfallen.

Eigene Befehle definieren

Mit

```
\newcommand{\meinBefehl}[n]{Definition}
```

wird ein neuer Befehl definiert. Dabei ist n die Anzahl der Parameter (zwischen 0 und 9); [0] kann entfallen.

Beispiel 1: für einen Befehl ohne Parameter:

```
\newcommand{\ZfS}{\mbox{Zf\hspace*{1pt}S}}
```

Eigene Befehle definieren

Mit

```
\newcommand{\meinBefehl}[n]{Definition}
```

wird ein neuer Befehl definiert. Dabei ist n die Anzahl der Parameter (zwischen 0 und 9); [0] kann entfallen.

Beispiel 1: für einen Befehl ohne Parameter:

```
\newcommand{\ZfS}{\mbox{Zf\hspace*{1pt}S}}
```

Statt `ZfS` liefert der Befehl `\ZfS` die Ausgabe

Eigene Befehle definieren

Mit

```
\newcommand{\meinBefehl}[n]{Definition}
```

wird ein neuer Befehl definiert. Dabei ist n die Anzahl der Parameter (zwischen 0 und 9); [0] kann entfallen.

Beispiel 1: für einen Befehl ohne Parameter:

```
\newcommand{\ZfS}{\mbox{Zf\hspace*{1pt}S}}
```

Statt `ZfS` liefert der Befehl `\ZfS` die Ausgabe `ZfS`.

Eigene Befehle definieren

Mit

```
\newcommand{\meinBefehl}[n]{Definition}
```

wird ein neuer Befehl definiert. Dabei ist n die Anzahl der Parameter (zwischen 0 und 9); [0] kann entfallen.

Beispiel 1: für einen Befehl ohne Parameter:

```
\newcommand{\ZfS}{\mbox{Zf\hspace*{1pt}S}}
```

Statt ZfS liefert der Befehl \ZfS die Ausgabe ZfS .

Beispiel 2: Abkürzung für den Folgepfeil:

```
\newcommand{\f}{\Rightarrow}
```

Eigene Befehle definieren

Mit

```
\newcommand{\meinBefehl}[n]{Definition}
```

wird ein neuer Befehl definiert. Dabei ist n die Anzahl der Parameter (zwischen 0 und 9); [0] kann entfallen.

Beispiel 1: für einen Befehl ohne Parameter:

```
\newcommand{\ZfS}{\mbox{Zf\hspace*{1pt}S}}
```

Statt ZfS liefert der Befehl \ZfS die Ausgabe ZfS .

Beispiel 2: Abkürzung für den Folgepfeil:

```
\newcommand{\f}{\rightarrow}
```

Der Befehl $\$a\f b\$$ ergibt

Eigene Befehle definieren

Mit

```
\newcommand{\meinBefehl}[n]{Definition}
```

wird ein neuer Befehl definiert. Dabei ist n die Anzahl der Parameter (zwischen 0 und 9); [0] kann entfallen.

Beispiel 1: für einen Befehl ohne Parameter:

```
\newcommand{\ZfS}{\mbox{Zf\hspace*{1pt}S}}
```

Statt `ZfS` liefert der Befehl `\ZfS` die Ausgabe `ZfS`.

Beispiel 2: Abkürzung für den Folgepfeil:

```
\newcommand{\f}{\Rightarrow}
```

Der Befehl `$a\f b$` ergibt $a \Rightarrow b$.

Eigene Befehle definieren

Mit

```
\newcommand{\meinBefehl}[n]{Definition}
```

wird ein neuer Befehl definiert. Dabei ist n die Anzahl der Parameter (zwischen 0 und 9); [0] kann entfallen.

Beispiel 1: für einen Befehl ohne Parameter:

```
\newcommand{\ZfS}{\mbox{Zf\hspace*{1pt}S}}
```

Statt `ZfS` liefert der Befehl `\ZfS` die Ausgabe `ZfS`.

Beispiel 2: Abkürzung für den Folgepfeil:

```
\newcommand{\f}{\Rightarrow}
```

Der Befehl `$a\f b$` ergibt $a \Rightarrow b$.

Existiert der Befehl schon, kann man ihn mit `\renewcommand` statt `\newcommand` umdefinieren.

Eigene Befehle mit Parametern

Die Parameter werden in der Befehlsdefinition mit #1 bis #9 angesprochen.

Eigene Befehle mit Parametern

Die Parameter werden in der Befehlsdefinition mit #1 bis #9 angesprochen.

Beispiel für einen Befehl mit drei Parametern:

Eigene Befehle mit Parametern

Die Parameter werden in der Befehlsdefinition mit #1 bis #9 angesprochen.

Beispiel für einen Befehl mit drei Parametern:

```
\newcommand{\Vek}[3]{(#1_{#2}, \dots, #1_{#3})}
```

Eigene Befehle mit Parametern

Die Parameter werden in der Befehlsdefinition mit #1 bis #9 angesprochen.

Beispiel für einen Befehl mit drei Parametern:

```
\newcommand{\Vek}[3]{(#1_{#2}, \dots, #1_{#3})}
```

$\$ \Vek{a}{11}{1k} \$$ ergibt dann

Eigene Befehle mit Parametern

Die Parameter werden in der Befehlsdefinition mit #1 bis #9 angesprochen.

Beispiel für einen Befehl mit drei Parametern:

```
\newcommand{\Vek}[3]{(#1_{#2}, \dots, #1_{#3})}
```

$\$ \Vek{a}{11}{1k} \$$ ergibt dann (a_{11}, \dots, a_{1k}) .

Befehl mit optionalem Parameter

Der erste Parameter kann vordefiniert werden und wird dann zu einem optionalen Parameter.

Befehl mit optionalem Parameter

Der erste Parameter kann vordefiniert werden und wird dann zu einem optionalen Parameter.

Beispiel:

```
\newcommand{\VEK}[4][,]{(#2_{#3} #1 \dots #1 #2_{#4})}
```

Befehl mit optionalem Parameter

Der erste Parameter kann vordefiniert werden und wird dann zu einem optionalen Parameter.

Beispiel:

```
\newcommand{\VEK}[4] [,]{(#2_{#3} #1 \dots #1 #2_{#4})}
```

$\$ \backslash \text{VEK} \{a\} \{11\} \{1k\} \$$ ergibt dann

Befehl mit optionalem Parameter

Der erste Parameter kann vordefiniert werden und wird dann zu einem optionalen Parameter.

Beispiel:

```
\newcommand{\VEK}[4] [,]{(#2_{#3} #1 \dots #1 #2_{#4})}
```

$\$ \backslash \text{VEK} \{a\} \{11\} \{1k\} \$$ ergibt dann wieder (a_{11}, \dots, a_{1k}) .

Befehl mit optionalem Parameter

Der erste Parameter kann vordefiniert werden und wird dann zu einem optionalen Parameter.

Beispiel:

```
\newcommand{\VEK}[4][,]{(#2_{#3} #1 \dots #1 #2_{#4})}
```

$\$ \backslash \text{VEK} \{a\} \{11\} \{1k\} \$$ ergibt dann wieder (a_{11}, \dots, a_{1k}) .

$\$ \backslash \text{VEK} [\backslash \text{times}] \{a\} \{11\} \{1k\} \$$ ergibt aber

Befehl mit optionalem Parameter

Der erste Parameter kann vordefiniert werden und wird dann zu einem optionalen Parameter.

Beispiel:

```
\newcommand{\VEK}[4][,]{(#2_{#3} #1 \dots #1 #2_{#4})}
```

$\$ \backslash \text{VEK}\{a\}\{11\}\{1k\} \$$ ergibt dann wieder (a_{11}, \dots, a_{1k}) .

$\$ \backslash \text{VEK}[\backslash \text{times}]\{a\}\{11\}\{1k\} \$$ ergibt aber $(a_{11} \times \dots \times a_{1k})$.

Befehl mit optionalem Parameter

Der erste Parameter kann vordefiniert werden und wird dann zu einem optionalen Parameter.

Beispiel:

```
\newcommand{\VEK}[4][,]{(#2_{#3} #1 \dots #1 #2_{#4})}
```

$\$ \backslash \text{VEK}\{a\}\{11\}\{1k\} \$$ ergibt dann wieder (a_{11}, \dots, a_{1k}) .

$\$ \backslash \text{VEK}[\backslash \text{times}]\{a\}\{11\}\{1k\} \$$ ergibt aber $(a_{11} \times \dots \times a_{1k})$.

Bemerkung: Soll innerhalb der Befehlsdefinition eine Matheumgebung stehen, nimmt man dafür besser

$\backslash \text{ensuremath}\{ \textit{Formel} \}$ anstatt die Formel darin mit $\$ \textit{Formel} \$$ zu umschließen.

Eigene Befehle

Eigene Umgebungen

Eigene Zähler

Eigene Längen

Eigene Umgebungen definieren

```
\newenvironment{meineUmgebung}[n]{AnfangDef}{EndeDef}
```

definiert eine neue Umgebung. Dabei ist n die Anzahl der Parameter (zwischen 0 und 9);

Eigene Umgebungen definieren

`\newenvironment{meineUmgebung}[n]{AnfangDef}{EndeDef}`
definiert eine neue Umgebung. Dabei ist n die Anzahl der Parameter (zwischen 0 und 9); [0] kann entfallen.

Eigene Umgebungen definieren

`\newenvironment{meineUmgebung}[n]{AnfangDef}{EndeDef}`
definiert eine neue Umgebung. Dabei ist n die Anzahl der Parameter (zwischen 0 und 9); [0] kann entfallen.

Üblicherweise greift man bei der Definition auf eine vorprogrammierte Umgebung zurück.

Eigene Umgebungen definieren

`\newenvironment{meineUmgebung}[n]{AnfangDef}{EndeDef}`
definiert eine neue Umgebung. Dabei ist n die Anzahl der Parameter (zwischen 0 und 9); [0] kann entfallen.

Üblicherweise greift man bei der Definition auf eine vorprogrammierte Umgebung zurück.

Parameter dürfen nur im {AnfangDef}-Teil der Definition auftreten.

Beispiel für eine eigene Umgebungen

Beispiel:

```
\newenvironment{xxx} [1]  
  {\hspace*{#1cm}\begin{minipage}[b]{5cm}\itshape}  
  {\end{minipage}ENDE}
```

Beispiel für eine eigene Umgebungen

Beispiel:

```
\newenvironment{xxx}[1]
  {\hspace*{#1cm}\begin{minipage}[b]{5cm}\itshape}
  {\end{minipage}ENDE}
```

```
\begin{xxx}{2}
hier kommt Text, viel Text, und noch mehr Text
\end{xxx}
```

ergibt

Beispiel für eine eigene Umgebungen

Beispiel:

```
\newenvironment{xxx}[1]
  {\hspace*{#1cm}\begin{minipage}[b]{5cm}\itshape}
  {\end{minipage}ENDE}
```

```
\begin{xxx}{2}
hier kommt Text, viel Text, und noch mehr Text
\end{xxx}
```

ergibt

*hier kommt Text, viel Text, und
noch mehr Text* ENDE

Eigene Umgebungen: Übung

Übung:

Definieren Sie eine Listenumgebung `Liste` mit einem Parameter. Diese soll wie die `itemize`-Umgebung funktionieren, der Parameter bestimmt dabei den Abstand (in mm) zwischen den einzelnen Punkten.

In der Anwendung soll der Quellcode dann so aussehen:

```
\begin{Liste}{4}  
\item Text  
\item Text  
\end{Liste}
```

Es soll also innerhalb der Liste die Länge `\itemsep` auf #1 gestellt werden.

Eigene Befehle

Eigene Umgebungen

Eigene Zähler

Eigene Längen

Definition eigener Zähler

Mit

```
\newcounter{meinZaehler}[Ruecksetzzaehler]
```

wird ein eigener Zähler namens *meinZaehler* eingerichtet.

Definition eigener Zähler

Mit

```
\newcounter{meinZaehler}[Ruecksetzzaehler]
```

wird ein eigener Zähler namens *meinZaehler* eingerichtet.

Die eigenen Zähler können so wie die in \LaTeX vordefinierten Zähler verändert werden: mit

```
\setcounter, \addtocounter, \stepcounter...
```

Definition eigener Zähler

Mit

```
\newcounter{meinZaehler}[Ruecksetzzaehler]
```

wird ein eigener Zähler namens *meinZaehler* eingerichtet.

Die eigenen Zähler können so wie die in \LaTeX vordefinierten Zähler verändert werden: mit

```
\setcounter, \addtocounter, \stepcounter...
```

Ein Zähler mit dem Namen *meinZaehler* wird ausgegeben mit `\themeinZaehler`.

Eigene Zähler

Der Befehl `\stepcounter{meinZaehler}` setzt dabei alle Zähler auf Null zurück, für die *meinZaehler* als Rücksetzer definiert wurde.

Eigene Zähler

Der Befehl `\stepcounter{meinZaehler}` setzt dabei alle Zähler auf Null zurück, für die *meinZaehler* als Rücksetzer definiert wurde.

Sofern also

`\newcounter{Zaehler}[meinZaehler]` definiert wurde, wird der Zähler *Zaehler* nach dem `\stepcounter{meinZaehler}`-Befehl auf Null stehen.

Eigene Zähler

Der Befehl `\stepcounter{meinZaehler}` setzt dabei alle Zähler auf Null zurück, für die *meinZaehler* als Rücksetzer definiert wurde.

Sofern also

`\newcounter{Zaehler}[meinZaehler]` definiert wurde, wird der Zähler *Zaehler* nach dem `\stepcounter{meinZaehler}`-Befehl auf Null stehen.

Ebenso funktioniert

`\refstepcounter{meinZaehler}` wobei dieser Befehl darüberhinaus es ermöglicht, daß auf den Zählerstand mit Verweisen Bezug genommen werden kann (also mit `\ref{Marke}`, falls nach diesem Befehl die Marke `\label{Marke}` gesetzt wurde).

Ein Beispiel

Beispiel:

Dieser⁽¹⁾ Satz⁽²⁾ hat⁽³⁾ fünf⁽⁴⁾ Wörter⁽⁵⁾.

Das Verb steht in diesem Satz an 3. Stelle.

Ein Beispiel

Beispiel:

Dieser⁽¹⁾ Satz⁽²⁾ hat⁽³⁾ fünf⁽⁴⁾ Wörter⁽⁵⁾.

Das Verb steht in diesem Satz an 3. Stelle.

Dies wird so gemacht:

```
\newcommand{\wortnummer}%  
{\refstepcounter{Wortzaehler}$^{\(\theWortzaehler)}$}
```

```
Dieser\wortnummer\  
Satz\wortnummer\  
hat\wortnummer\  
\label{Verb}  
fünf\wortnummer\  
Wörter\wortnummer.
```

Das Verb steht in diesem Satz an `\ref{Verb}.` Stelle.

Eigene Befehle

Eigene Umgebungen

Eigene Zähler

Eigene Längen

Eigene Längenmaße

Mit

```
\newlength{\meineLaenge}
```

wird ein neuer Längenbefehl definiert. Standardmäßig bekommt er so die Länge Null zugewiesen.

Eigene Längenmaße

Mit

```
\newlength{\meineLaenge}
```

wird ein neuer Längenbefehl definiert. Standardmäßig bekommt er so die Länge Null zugewiesen.

Auch für selbstdefinierte Längen gibt es die Befehle: `settowidth`, `setlength`, `addtolength`

Eigene Längenmaße

Mit

```
\newlength{\meineLaenge}
```

wird ein neuer Längenbefehl definiert. Standardmäßig bekommt er so die Länge Null zugewiesen.

Auch für selbstdefinierte Längen gibt es die Befehle: `settowidth`, `setlength`, `addtolength`

Den Betrag einer Länge kann man mit `\the\meineLaenge` in der Längeneinheit `pt` ausgeben lassen.