

LMU München
Informatik

L^AT_EX - Befehle
Eine kleine Auswahl

Christian Schwabe
christian.schwabe@campus.lmu.de
25. Dezember 2010

Inhaltsverzeichnis

1 Installation

1.1 MikTeX + TeXnicCenter

In den Umgebungsvariablen (System -> Erweiterte Systemeinstellungen -> Erweitert -> Umgebungsvariablen -> Path) den Pfad zum MikTeX-bin-Ordner hinzufügen: **C:\Program Files\MikTeX\miktex\bin\;**

1.2 Verwendung von TikZ und TeXnicCenter

1. **gnuplot** downloaden und ins C-Programmverzeichnis entpacken: <http://www.gnuplot.info/>
2. In den Umgebungsvariablen (System -> Erweiterte Systemeinstellungen -> Erweitert -> Umgebungsvariablen -> Path) den Pfad zum gnuplot-binary-Ordner hinzufügen: **C:\Program Files\gnuplot\binary**
3. Im Programm TeXnicCenter eine Kommandozeilen Option hinzufügen:
`-interaction=nonstopmode` → `-src -interaction=nonstopmode -enable-write18 „%Wm“`
 unter Ausgabe -> Ausgabeprofile definieren -> Argumente die an den Compiler übergeben werden sollen

1.3 Verwendung von PDF-XChange Viewer und TeXnicCenter

Bei Verwendung des PDF-XChange Viewer als Standard-PDF-Viewer in TeXnicCenter, können die PDF-Dokumente vor dem Kompilieren automatisch geschlossen werden:

1. In TexnicCenter auf **Ausgabe** klicken
2. **Ausgabeprofil definieren**
3. **Viewer**
4. Beim **Pfad der Anwendung** den Pfad zum Viewer angeben, z.B: **D:\Programme\Tracker Software\PDF-XChange Viewer\pdf-viewer\PDFXCview.exe**
5. **Projektausgabe betrachten** → Kommandozeile auswählen und als Kommando **“%bm.pdf“** eintragen
6. Bei **Kompilierung vor Ausgabe schließen** auch Kommandozeile auswählen und **/close “%bm.pdf“** eintragen

1.4 Verwendung von SumatraPDF und TeXnicCenter (incl. SyncTeX)

Sumatra PDF ist ein extrem schlanker PDF-Viewer, der Änderungen an geöffneten PDF akzeptiert und sogar Vorwärts- und Rückwärtssuche unterstützt, d.h. das Springen zwischen entsprechenden Stellen in der PDF-Ausgabe und dem Quellcode. Nach dem Installieren von Sumatra PDF mit den Standardeinstellungen wird das Profil LaTeX ⇒ PDF mit dem neuen Namen **LaTeX ⇒ PDF (Sumatra)** kopiert. Folgende Einstellungen werden geändert:

- **(La)Tex**
 Bei **Argumente, die an den Compiler übergeben werden sollen** muss folgendes eingetragen werden:
`-synctex=-1 -interaction=nonstopmode “%pm“`
- **Viewer**
 Bei **Pfad der Anwendung** muss folgendes stehen:
`C:\Programme\SumatraPDF\SumatraPDF.exe -reuse-instance -inverse-search
 “C:\Programme\TeXnicCenter\TEXCNTR.EXE\“/ddecmd “[goto(‘%f’, ‘%l’)]\“`
 Hinweis: nach dem search kommt ein Leerzeichen!
- Bei **Projektausgabe betrachten** muss der Radiobutton bei **Kommandozeile** gesetzt werden.
 Außerdem ist das Kommando `[Open(“%bm.pdf“,0,0,1)]` einzugeben.

- Bei **Suche in Ausgabe** muss der Radiobutton bei **DDE-Kommando** gesetzt werden mit:
Server: SUMATRA
Thema: Control
Außerdem ist das Kommando `[ForwardSearch("%bm.pdf", "%Wc", %1, 0, 0, 0)]` einzugeben.
- Bei **Vor Kompilierung Ausgabe schließen** muss der Radiobutton bei **Nicht schließen** gesetzt werden.

2 Abstände

2.1 Horizontale Abstände

<code>\!</code>	<code>xx xx</code>	
normal ohne LZ	<code>xx xx</code>	
<code>\,</code>	<code>xx xx</code>	
<code>\:</code>	<code>xx xx</code>	
normal mit LZ	<code>xx xx</code>	
<code>\;</code>	<code>xx xx</code>	(Leerzeichen)
<code>\enspace</code>	<code>xx xx</code>	
<code>\quad</code>	<code>xx xx</code>	
<code>\qquad</code>	<code>xx xx</code>	
<code>\hspace{1.0cm}</code>	<code>xx xx</code>	

2.2 Vertikale Abstände

<code>\smallskip</code>	etwa 1/4 Zeile
<code>\medskip</code>	etwa 1/2 Zeile
<code>\bigskip</code>	etwa 1 Zeile
<code>\vfill</code>	Abstand zwischen 0 und unendlich
<code>\vspace{n}</code>	Ein n hoher Abstand
<code>\vspace*{n}</code>	Erzwingen eines n hohen Abstandes
<code>\addvspace{}</code>	zusätzlicher Abstand zwischen Absätzen
<code>\newpage</code>	Seitenwechsel
<code>\\</code>	Zeilenwechsel
<code>Leerzeile</code>	neuer Absatz (ggf. mit Einrückung) und Zeilenwechsel

3 Mathematische Symbole

3.1 Indizes und Potenzen

x^2	x^2
y_{n+1}	y_{n+1}

3.2 Mathematische Akzente

\vec{a}	\vec{a}
\dot{a}	\dot{a}
\ddot{a}	\ddot{a}
$\overline{\text{coin}}$	$\overline{\text{coin}}$

3.3 kleine griechische Buchstaben

α	α	λ	λ	ϕ	ϕ
β	β	μ	μ	φ	φ
γ	γ	ξ	ξ	ω	ω
δ	δ	π	π		
ϵ	ϵ	σ	σ		
ε	ε	τ	τ		

3.4 große griechische Buchstaben

Δ	Δ	Φ	Φ	Ω	Ω
Θ	Θ	Ψ	Ψ	Γ	Γ

3.5 Zahlenbereiche

\mathbb{C}	\mathbb{C}	\mathbb{Q}	\mathbb{Q}	\mathbb{Z}	\mathbb{Z}
\mathbb{N}	\mathbb{N}	\mathbb{P}	\mathbb{P}		

3.6 sonstige Symbole

\imath	\imath	\emptyset	\emptyset
j	j	∇	∇
\wp	\wp	\triangle	\triangle
\Re	\Re		
\Im	\Im	\forall	\forall
∂	∂	\exists	\exists
∞	∞	\neg	\neg
\checkmark	\checkmark	\Join	\Join
$\sqrt{}$	$\sqrt{}$	\Box	\Box
\parallel	\parallel	\intercal	\intercal
$*$	$*$	\prime	\prime

4 Operatoren

4.1 Binäre Operatoren

\times	<code>\times</code>	\cdot	<code>\cdot</code>	\circ	<code>\circ</code>
\mp	<code>\mp</code>	\pm	<code>\pm</code>	\oplus	<code>\oplus</code>
\cap	<code>\cap</code>	\cup	<code>\cup</code>	\odot	<code>\odot</code>
\vee	<code>\vee</code>	\wedge	<code>\wedge</code>	\div	<code>\div</code>

4.2 „große“ Operatoren

Σ	<code>\sum</code>	\prod	<code>\prod</code>	\int	<code>\int</code>
\bigwedge	<code>\bigwedge</code>	\bigcup	<code>\bigcup</code>		

4.3 Relationen

\leq	<code>\leq</code>	\geq	<code>\geq</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>
\in	<code>\in</code>	\ni	<code>\ni</code>
\notin	<code>\notin</code>	$\not\equiv$	<code>\not\equiv</code>
\equiv	<code>\equiv</code>	\sim	<code>\sim</code>
\approx	<code>\approx</code>	\cong	<code>\cong</code>
\neq	<code>\neq</code>	$\not\equiv$	<code>\not\equiv</code>
$:=$	<code>\coloneqq</code>	\equiv	<code>\eqqcolon</code>
\triangleq	<code>\triangleq</code>	$\hat{=}$	<code>\hat{=}</code>

4.4 Pfeile

\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>
\rightarrow	<code>\rightarrow</code>	\longrightarrow	<code>\longrightarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\longleftarrow	<code>\longleftarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>
\uparrow	<code>\uparrow</code>	\Uparrow	<code>\Uparrow</code>
\downarrow	<code>\downarrow</code>	\Downarrow	<code>\Downarrow</code>
\nearrow	<code>\nearrow</code>	\searrow	<code>\searrow</code>
\nwarrow	<code>\nwarrow</code>	\swarrow	<code>\swarrow</code>

4.5 beschriftete Pfeile

$$B \xrightarrow[n]{n \pm i - 1} C$$

```

1 \begin{center}
2 $B \xrightarrow[n]{n \pm i - 1} C$
3 \end{center}

```

4.6 Anordnung über-/untereinander

$$\begin{array}{c} yz \\ x \\ a \\ bcd \\ \vdots \\ = \end{array}$$

```
1 \begin{center}
2 $ \underset{x}{yz} $\ll
3 $ \overset{a}{bcd} $\ll
4 $ \stackrel{!}{=} $
5 \end{center}
```

4.7 Klammern

{	\{	}	\}
	\		
⌊	\lfloor	⌋	\rfloor
⌈	\lceil	⌉	\rceil
<	\langle	>	\rangle
⌌	\llfloor	⌍	\rrfloor

Um mathematische Ausdrücke mit der richtigen Klammergröße zu versehen, `\left` oder `\right` davor schreiben, z.B.: `\left(` oder `\right\)`.

4.8 Fallunterscheidungen

$$f(x) = \begin{cases} x/2 & , x \text{ gerade} \\ 3x+1 & , x \text{ ungerade} \end{cases}$$

```
1 $f(x) = \begin{cases}
2 x/2 \quad \& , x \text{ \text{ gerade }} \\
3 3x+1 \quad \& , x \text{ \text{ ungerade }} \\
4 \end{cases} $
```


5 Mathematische Elemente

5.1 Wurzeln

$$\sqrt{x} \quad \backslash sqrt{x}$$

$$\sqrt[3]{x} \quad \backslash sqrt[3]{x}$$

5.2 Brüche und Binomial-Koeffizienten

$$\frac{x^2}{y} \quad \backslash frac{x^2}{y}$$

$$x^{\frac{1}{2}} \quad x^{\backslash frac{1}{2}}$$

$$\binom{n}{k} \quad \{n \backslash choose k\}$$

$$\frac{x}{y+2} \quad \{x \ atop y + 2\}$$

5.3 Waagerechte Striche und Klammern

$$\overline{m+n} \quad \backslash overline{m+n}$$

$$\underbrace{a+b+\cdots+z}_{26} \quad \backslash underbrace{a+b+\cdots+z}_{26}$$

5.4 Funktionsnamen

arg	$\backslash arg$	ln	$\backslash ln$	sin	$\backslash sin$
cos	$\backslash cos$	log	$\backslash log$	tan	$\backslash tan$
exp	$\backslash exp$	max	$\backslash max$		
lg	$\backslash lg$	min	$\backslash min$		

5.5 Komma als Dezimaltrennzeichen

Das Komma ist in LaTeX standardmäßig ein Aufzählungszeichen. Soll ein Komma als Dezimaltrennzeichen verwendet werden, so kann dies durch geschweifte Klammern bewerkstelligt werden.

$$3,14 \quad 3\{, \}14$$

$$3,14 \quad 3,14$$

5.6 Matrizen

Es gibt für jede verschiedene Matrizenart eine eigene Umgebung.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

```
1 $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$
```

$$\begin{bmatrix} 0 & \cdots & 1 \\ 2 & \cdots & 3 \end{bmatrix}$$

```
1 $\begin{bmatrix} 0 & \cdots & 1 \\ 2 & \cdots & 3 \end{bmatrix}$
```

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

```
1 $\begin{vmatrix}
2 a & b \\
3 c & d \\
4 \end{vmatrix}$
```

$$\begin{vmatrix} 0 & 1 \\ 2 & 3 \end{vmatrix}$$

```
1 $\begin{Vmatrix}
2 0 & 1 \\
3 2 & 3 \\
4 \end{Vmatrix}$
```

5.7 Mehrzeilige Gleichungen

$$\begin{aligned} V_{Max} &= 12345 \\ &= 2345 \\ &= 345 \end{aligned}$$

```
1 \begin{align*}
2 V_{Max} &= 12345 \\
3 &= 2345 \\
4 &= 345 \\
5 \end{align*}
```

$$\begin{aligned} V_{Max} &= 12345 && \text{Axiom 1} \\ &= 2345 && \text{Axiom 2} \\ &= 345 && \text{Axiom 3} \end{aligned}$$

```
1 \begin{alignat*}{2}
2 V_{Max} &= 12345 \quad \text{Axiom 1} \\
3 &= 2345 \quad \text{Axiom 2} \\
4 &= 345 \quad \text{Axiom 3} \\
5 \end{alignat*}
```

$$\begin{array}{lll} 1221 & test & 1221 \\ 23332 & testtext & 23332 \end{array}$$

```
1 $\begin{array}{lcl}
2 1221 & test & 1221 \\
3 23332 & testtext & 23332 \\
4 \end{array}$
```

5.8 Beispiele

$$\begin{array}{ll} a \bmod b & a \bmod b \\ x \equiv a \pmod{b} & x \equiv a \pmod{b} \\ \lim_{x \rightarrow 0} \frac{\sin x}{x} & \lim_{x \rightarrow 0} \frac{\sin x}{x} \\ \sum_{n=1}^{\infty} (T \geq n) & \sum_{n=1}^{\infty} (T \geq n) \end{array}$$

6 Programmcode einbinden

6.1 Java Code

```

1 public static int iterativ (int n) {
2     int letzte = 1;
3     // for-Schleife
4     for (int i = 2; i <= n; i++) {
5         int temp = i;
6     }
7     return temp;
8 }

```

```

1 \lstset {language=Java, basicstyle =\small\mdseries,
      tabsize=3, frame=shadowbox, framexleftmargin=8
      mm, xleftmargin=10mm, rulesepcolor=\color{blue},
      numbers=left, numberstyle=\tiny, stepnumber=1,
      numbersep=12pt, backgroundcolor=\color{white},
      commentstyle=\itshape\color{darkgreen},
      keywordstyle=\color{darkblue}, stringstyle =\color
      {darkred}}
2 \begin{lstlisting}
3 public static int iterativ (int n) {
4     int letzte = 1;
5     // for-Schleife
6     for (int i = 2; i <= n; i++) {
7         int temp = i;
8     }
9     return temp;
10 }
11 \end{lstlisting}

```

6.2 Pseudocode

7 Formatierung von Text

7.1 Fett, kursiv,...

fetter Testtext
kursiver Testtext
 unterstrichener Testtext
2-fach unterstrichener Testtext
 Schreibmaschinen-Testtext
 KAPITÄLCHEN TESTTEXT

```
1 \textbf{fetter Testtext }\\
2 \textit{kursiver Testtext }\\
3 \underline{unterstrichener Testtext }\\
4 \underline{\underline{2-fach unterstrichener Testtext }}\\
5 \texttt{Schreibmaschinen-Testtext }\\
6 \textsc{KAPITÄLCHEN TESTTEXT }
```

7.2 Textsatz

Standardmäßig wird Text im Blocksatz, also links- und rechtsbündig gesetzt. Es gibt jedoch auch für die anderen Satze, bestimmte Umgebungen:

<code>\begin{center}</code>	<code>\begin{flushleft}</code>	<code>\begin{flushright}</code>
Text	Text	Text
<code>\end{center}</code>	<code>\end{flushleft}</code>	<code>\end{flushright}</code>

7.3 Schriftgröße

Die Schriftgröße ändert man mit einem Befehl ohne Parameter. Die neue Größe gilt bis zur nächsten Änderung, dem Ende der aktuellen Umgebung oder bei z.B. `{\large ... }` bis zur schließenden Klammer.

Testtext (tiny)

Testtext (scriptsize)

Testtext (footnotesize)

Testtext (small)

default

Testtext (large)

Testtext (Large)

Testtext (LARGE)

Testtext (huge)

Testtext (Huge)

```
1 \tiny{ Testtext ( tiny ) }\\
2 \scriptsize{ Testtext ( scriptsize ) }\\
3 \footnotesize{ Testtext ( footnotesize ) }\\
4 \small{ Testtext ( small ) }\\
5 \normalsize{ default }\\
6 \large{ Testtext ( large ) }\\
7 \Large{ Testtext ( Large ) }\\
8 \LARGE{ Testtext ( LARGE ) }\\
9 \huge{ Testtext ( huge ) }\\
10 \Huge{ Testtext ( Huge ) }
```

7.4 Kommentare

Kommentare werden durch `%` eingeleitet und reichen bis zum Zeilenumbruch der aktuellen Zeile.
Eine Möglichkeit mehrzeilige Kommentare zu erzeugen gibt es nicht!

7.5 Bereiche (Scope)

Wenn gewisse Formatierungen (Schriftfarbe, Zeilenhöhe, Schriftgröße) nur in einem bestimmten Bereich wirken sollen (nicht im ganzen weiteren Text), dann kann der Bereich mit **geschweiften Klammern** umschlossen werden. Die Formatierungen werden dann innerhalb dieser Klammern definiert und gelten dann nur in diesem Scope.

7.6 Mehrspaltiger Text

Dieser ziemlich dümmliche Text soll als Demonstration dienen, wie das Mehr-Spalten-Layout funktioniert und wann und wo die Zeilen und Wörter umgebrochen werden und wie der Textfluss ist.

```
1 \begin{ multicols }{2}
2 Dieser ziemlich dümmliche Text soll als Demonstration
   dienen, wie das Mehr-Spalten-Layout funktioniert
   und wann und wo die Zeilen und Wörter
   umgebrochen werden und wie der Textfluss ist .
3 \end{ multicols }
```

7.7 Besondere Zeichen in L^AT_EX

<code>_</code>	<code>_</code>	<code>\</code>	<code>\backslash</code>	<code>~</code>	<code>\textasciitilde</code>
<code>\$</code>	<code>\\$</code>	<code>\$</code>	<code>\\$</code>	<code>&</code>	<code>\&</code>
<code>#</code>	<code>\#</code>	<code>%</code>	<code>\%</code>	<code>^</code>	<code>\hat</code>

7.8 Das EURO-Symbol

```
\usepackage{eurosym}
...
\euro
```

7.9 Verlinkungen im Text

Mit dem Paket `hyperref` werden automatisch Hyperlinks vom Verweis zur Marke gesetzt. Das Paket sollte am besten als letztes geladen werden, da sehr viele Einstellungen von diesem überschrieben werden. Das Laden und das Setup der Einstellungen erfolgt z.B. mit folgendem Code:

```
\usepackage{hyperref}
\hypersetup{
pdfpagemode=FullScreen,
pdftitle={TeX Befehle},% Titel des PDF-Dokuments
pdfauthor={Christian Schwabe},% Autor(Innen) des PDF-Dokuments
pdfpagelayout=SinglePage,
bookmarks=true,      % Lesezeichen erzeugen
bookmarksopen=true,  % Lesezeichen ausgeklappt
bookmarksnumbered=true, % Anzeige der Kapitelzahlen am Anfang der Namen der Lesezeichen
pdfstartpage=Zahl,   % Seite, welche automatisch geöffnet werden soll
baseurl=http://www.server.de/dateiname.pdf, % URL des PDF-Dokuments (oder Hintergrundinformationen)
pdfsubject={Kurzbeschreibung als ein Satz}, % Inhaltsbeschreibung des PDF-Dokuments
```

```
pdfkeywords={Stichwörter},    % Stichwortangabe zum PDF-Dokument
breaklinks=true,             % ermöglicht einen Umbruch von URLs
colorlinks=true,             % Einfärbung von Links
linkcolor=black,             % Linkfarbe: schwarz
anchorcolor=black,           % Ankerfarbe: schwarz
citecolor=black,             % Literaturlinks: schwarz
filecolor=black,             % Links zu lokalen Dateien: schwarz
menucolor=black,             % Acrobat Menü Einträge: schwarz
pagecolor=black,             % Links zu anderen Seiten im Text: schwarz
urlcolor=black               % URL-Farbe: schwarz
```

Ein internes Sprung-/ Linkziel wird definiert durch:

```
\hypertarget{name}{text}
```

Ein klickbarer Link wird definiert durch:

```
\hyperlink{name(=HypertargetName)}{text}
```

URL's werden folgendermaßen dargestellt:

```
\href{http://www.caipiranha.de}{Caipi Homepage}
```

8 Listen und Aufzählungen

8.1 Aufzählung mit Punkten

- Punkt 1
 - Punkt 1
 - Punkt 2
 - * Punkt 1
 - * Punkt 2
- Punkt 2
- blablabla

```

1 \begin{itemize}
2   \item Punkt 1
3     \begin{itemize}
4       \item Punkt 1
5       \item Punkt 2
6         \begin{itemize}
7           \item Punkt 1
8           \item Punkt 2
9         \end{itemize}
10    \end{itemize}
11  \item Punkt 2
12  \item blablabla
13 \end{itemize}

```

8.2 Aufzählung mit Nummern

1. Punkt 1
 - a) Punkt 1
 - b) Punkt 2
 - i. Punkt 1
 - ii. Punkt 2
2. Punkt 2
3. blablabla

```

1 \begin{enumerate}
2   \item Punkt 1
3     \begin{enumerate}
4       \item Punkt 1
5       \item Punkt 2
6         \begin{enumerate}
7           \item Punkt 1
8           \item Punkt 2
9         \end{enumerate}
10    \end{enumerate}
11  \item Punkt 2
12  \item blablabla
13 \end{enumerate}

```

8.3 Aufzählung mit benutzerdefinierten Zeichen

Diese Zeichen benötigen das Package „bbding“.

- ☞ Punkt 1
- ☞ Punkt 2
- ✕ Punkt 3
- ◆ Punkt 4
- ❖ Punkt 5
- ▮ Punkt 6
- ➡ Punkt 7
- Punkt 8
- Punkt 9
- Punkt 10

```

1 \begin{itemize}
2   \item [\HandRight] Punkt 1
3   \item [\PencilRight] Punkt 2
4   \item [\XSolidBrush] Punkt 3
5   \item [\DiamondSolid] Punkt 4
6   \item [\OrnamentDiamondSolid] Punkt 5
7   \item [\ArrowBoldRightStrobe] Punkt 6
8   \item [\ArrowBoldDownRight] Punkt 7
9   \item [\CircleSolid] Punkt 8
10  \item [\Square] Punkt 9
11  \item [\SquareSolid] Punkt 10
12 \end{itemize}

```

8.4 Veränderung der Standard-Zeichen

- Punkt xx
 - Punkt yy
 - Punkt yyy
 - ◇ Punkt 1
 - ◇ Punkt 2
- Punkt 2
- blablabla

```

1 \renewcommand{\labelitemi}{\circ$}
2 \renewcommand{\labelitemii}{\bullet$}
3 \renewcommand{\labelitemiii}{\diamond$}
4 \begin{itemize}
5   \item Punkt xx
6   \begin{itemize}
7     \item Punkt yy
8     \item Punkt yyy
9     \begin{itemize}
10      \item Punkt 1
11      \item Punkt 2
12    \end{itemize}
13   \end{itemize}
14   \item Punkt 2
15   \item blablabla
16 \end{itemize}

```

- a) Punkt 1
 - a.I Punkt 1
 - a.II Punkt 2
 - 1.) Punkt 1
 - 2.) Punkt 2
- b) Punkt 2
- c) blablabla

```

1 \renewcommand{\labelenumi}{\alph{enumi}}
2 \renewcommand{\labelenumii}{\alph{enumi}.\Roman{enumii}}
3 \renewcommand{\labelenumiii}{\arabic{enumiii}.)}
4 \begin{enumerate}
5   \item Punkt 1
6   \begin{enumerate}
7     \item Punkt 1
8     \item Punkt 2
9     \begin{enumerate}
10      \item Punkt 1
11      \item Punkt 2
12    \end{enumerate}
13   \end{enumerate}
14   \item Punkt 2
15   \item blablabla
16 \end{enumerate}

```


9 Eigene Kommandos definieren

9.1 Allgemein

Syntax zum definieren von eigenen Kommandos:

`\newcommand{\neuerName}{Definition}`

Man kann den neuen Kommandos auch Argumente übergeben (maximal 9 Argumente!):

`\newcommand{\neuerName}[Anzahl der Argumente]{Definition}`

9.2 Eigene Kommandos



Aufgabe 3

Lösung Aufgabe 4

6

$$x^2$$

=

für Text in Math-Umgebungen

```
1 \FALSCH \\
2 \RICHTIG \\
3 \Aufg{3} \\
4 \Loes{4} \\
5 \UAufg{6} \\
6 \mathee{x^2} \\
7 \igl \\
8 \txt {für Text in Math-Umgebungen} \\
```

10 Tabellen und Tabulatoren

10.1 Tabellen

Tabellen werden mit der `tabular`-Umgebung definiert.

```
\begin{tabular}[<Position>]{<Spaltendefinition>}
...
<Eintrag> & <Eintrag> \\
...
\end{tabular}
```

- Jede Zeile wird mit dem Zeilenumbruch `\\` abgeschlossen.
- Spalten werden mit Hilfe des `&`-Zeichens voneinander getrennt.
- Bei `<Spaltendefinition>` gibt man die Anzahl der Spalten in der Form `|l r c|` an. Über die Zeichen `right`, `left` und `center` wird die Textausrichtung in der Spalte definiert.
- Weitere mögliche Formateinträge:
 - `p{breite}` Absatz der Breite *breite*
 - `*{anzahl}{format}` das *format* wird *anzahl* mal wiederholt, z.B. `*{3}{c|} \triangleq {c|c|c|}`
 - `@{text}` der Text *text* wird anstatt des normal verwendeten Zwischenraumes zwischen den benachbarten Spalten angebracht.
- **Die Anzahl der Ausrichtungszeichen legt die Spaltenzahl fest!**
- Mithilfe des Befehls `\multicolumn{<Spaltenzahl>}{Ausrichtung}{Text}` kann ein Eintrag über mehrere Spalten gehen.
- `\hline` für eine horizontale Linie über die ganze Tabellenbreite (anzugeben direkt nach einem `\\`).
- `\cline{von-bis}` für eine horizontale Linie über einzelne Spalten (wie `\hline` anzugeben direkt nach einem `\\`).
- `\vline` für eine vertikale Linie innerhalb einer Spalte.

links	zentriert	rechts	Dies ist ein Text, der wie ein Absatz der Breite 2,5 cm formatiert wird.	DM 2.50
eins und zwei	drei	1 2	Absatz	\$ 20.50

```

1 \begin{ tabular } { l | c | r || p { 2.5cm } | r @ { . } l }
2 links & zentriert & rechts & Dies ist ein Text,
   der wie ein Absatz der Breite 2,5 cm
   formatiert wird. & DM 2 & 50 \\
3 \hline
4 \multicolumn{ 2 } { c | } { eins und zwei } & drei & 1 \
   \vline { } 2 & Absatz & \$ 20 & 50 \\
5 \cline { 1-2 }
6 \end{ tabular }
```

10.1.1 Zeilenabstand

Falls `\hline` verwendet wird, ist die Spaltenhöhe etwas zu klein. Dazu gibt es 3 Möglichkeiten dies zu korrigieren:

10.1.1.1 Arraystretch

Kann auch verwendet werden, um generell die Zeilenhöhe zu verringern oder erhöhen.

Fügt Platz über und unter dem Zeilentext ein!

```
\renewcommand{\arraystretch}{1.2}
```

10.1.1.2 Extrarowheight

Fügt nur Platz über dem Zeilentext ein! Fügt aber auch Platz ein, falls gar kein \hline in der Zeile vorhanden ist!

```
\usepackage{array}
...
\setlength{\extrarowheight}{1.5pt}
```

10.1.1.3 Bigstruts

10.1.2 Spaltenabstand

10.2 Tabulatoren

Hier muss man selbst für die Breite der Spalten sorgen!

- `\=` setzt einen Tabulator an der aktuellen Spaltenposition.
- `\>` springt um eine Tabulatorposition nach rechts.
- `\<` springt um eine Tabulatorposition nach links.
- `\+` verschiebt den linken Rand um eine Tabulatorposition nach rechts (muß vor dem Zeilenumbruch mit `\\` erfolgen!).
- `\-` verschiebt den linken Rand um eine Tabulatorposition nach links (muß vor dem Zeilenumbruch mit `\\` erfolgen!).
- `\pushtabs` speichert die aktuellen Tabulatorpositionen auf dem Stapel und löscht dann die aktuellen (verschachtelbar).
- `\poptabs` löscht die aktuellen Tabulatorpositionen und lädt die gespeicherten vom Stapel an deren Stelle (verschachtelbar).
- `\kill` entfernt die aktuelle Zeile (meist eine Musterzeile).
- `\\` beendet die aktuelle Zeile.

erste Spalte	zweite Spalte	breite	dritte Spalte
vorne	mitte	hinten	
	mitte	hinten	
		hinten	
vorne	mitte	hinten	

```
1 \begin{tabbing}
2 erste Spalte breit \= zweite Spalte \=
3 dritte Spalte \kill
4 erste Spalte \> zweite Spalte \> dritte Spalte \\
5 vorne \> mitte \> hinten \+ \\
6 mitte \> hinten \+ \\
7 hinten \- \- \\
8 vorne \> mitte \> hinten \\
9 diese Zeile erscheint nicht \kill
10 \end{tabbing}
```

10.3 listliketab-Package

Ermöglicht es Listen mit Tabellen zu kombinieren und so Tabulatoren innerhalb von Listen zu verwenden.

Anstelle der `tabular`-Umgebung innerhalb von `listliketab`, können auch `tabularx`, `longtable` und evtl andere verwendet werden.

`\usepackage{listliketab}`

- OneOneOne TWOtwo
- Two Three
- Three FourfourFour

```
1 \storestyleof {itemize}
2 \begin{ listliketab }
3 \begin{ tabular }{ lll }
4     \textbullet & OneOneOne & TWOtwo\\
5     \textbullet & Two & Three\\
6     \textbullet & Three & FourfourFour \\
7 \end{ tabular }
8 \end{ listliketab }
```

- a) **Red** or green or blue
- b) **Short** or stout or tall
- c) **Happy** or sad or confused

```
1 \storestyleof {enumerate}
2 \begin{ listliketab }
3 \newcounter{tabenum}\setcounter{tabenum}{0}
4 \newcommand{\nextnum}{\addtocounter{tabenum}{1}\alph{
    tabenum}}
5 \begin{ tabular }{ l>{\bf}l@{~or~}>{\bf}l@{~or~}>{\bf}l }
6     \nextnum & Red & green & blue\\
7     \nextnum & Short & stout & tall \\
8     \nextnum & Happy & sad & confused\\
9 \end{ tabular }
10 \end{ listliketab }
```

10.4 tabularx-Package

`\usepackage{tabularx}`
`\begin{tabularx}{Breite}{Spaltendefinition}`
 Körper
`\end{tabularx}`

Diese Umgebung ist nah verwandt mit der `tabular`-Umgebung, somit kann man sich auch an deren Syntax orientieren. Die Zellen werden durch `&` getrennt und mit `\\` beendet.

Hier kann man allerdings die Tabellenbreite explizit vorgeben. Die mit `X` markierten Spalten sind Spalten variabler Breite. Zwischen diesen Spalten wird der verbliebene Platz aufgeteilt, der sich aus der Gesamtbreite und der schon verbrauchten Breite für die „normalen“ Spalten ergibt.

- *Breite* Hier wird die gewünschte Tabellenbreite angegeben. Häufig verwendet werden hier `\textwidth` oder `\linewidth`.
- *X* neue Spaltendefinition → Spalte mit variabler Breite.
- `\hspace` sind 2 oder mehr Spalten mit `X` markiert würde der verbliebene Platz gleichmäßig auf diese aufgeteilt werden. Mit `\hspace` lässt sich ein Verhältnis zwischen diesen variablen Spalten angeben.
- Will man mehrere `tabularx` Umgebungen schachteln, muss man diese in geschweifte Klammern einschließen.

10.5 longtable-Package

```
\usepackage{longtable}
\begin{longtable}{ Format }
  Körper
\end{longtable}
```

Wie die `tabular` Umgebung, erlaubt jedoch auch mehrseitige Tabellen.

Um die Tabelle automatisch auf die Seiten umzubrechen, muss LaTeX die Tabelle mit seinem Seitenumbruchsalgorithmus verarbeiten können. Dazu ist es jedoch notwendig, das Dokument mehrfach zu kompilieren.

Um diese Anzahl der Kompilierungsvorgänge zu verringern, kann die erste Zeile der Tabelle mit `\kill` anstatt mit `\end{longtable}` abgeschlossen werden.

10.6 ltxtable-Package

```
\usepackage{ltxtable}
\LTxtable{Breite}{Dateiname}
```

Tabelle als `longtable`-Umgebung in eine eigene Datei schreiben und dabei die X-Spezifikation von `tabularx` benutzen. Danach Tabelle mit obiger Anweisung in das Dokument einfügen.

Eine andere Möglichkeit wäre mit dem `filecontents`-Package zu arbeiten. Hier kann die Tabellendefinition im gleichen Dokument erfolgen. Dabei wird die Tabelle dann in die entsprechende Datei geschrieben (überschrieben) und an der gewünschten Stelle eingefügt.

```
\documentclass[a4paper,12pt]{scrreprt}
```

```
\usepackage{filecontents}
\begin{filecontents*}{LONGTAB.tex}
\begin{longtable}{lX}
EINS &
Auf der Mauer
\\
DREI &
sitzt 'ne kleine Wanze
\end{longtable}
\end{filecontents*}
```

```
\usepackage[german]{babel}
\usepackage[latin1]{inputenc}
```

```
\usepackage{ltxtable}
```

```
\begin{document}
\LTxtable{\textwidth}{LONGTAB}
\end{document}
```

11 Farben

11.1 Schrift und Hintergrund

- `\color{red}`: Der folgende Text ist rot bis zum nächsten Farbwechsel.
- `\textcolor{green}`{Der eingeklammerte Text ist grün}.
- `\pagecolor{blue}`: Setzen der Seitenhintergrundfarbe.

11.2 Farbboxen

- `\colorbox{red}`{Rot hinterlegte Box}.
- `\fcolorbox{blue}{green}`{Grüne Box mit blauem Rand}.
- Ändern der Randstärke mit `\setlength{\fboxrule}{5pt}`: 5pt Rand
- Ändern des Randabstandes mit `\setlength{\fboxsep}{0pt}`: 5pt Rand

11.3 Das framed-Paket

- `\usepackage{framed}`
- Zeilen- und Seitenumbrüche innerhalb des Rahmens möglich!
- rahmt per Voreinstellung über die gesamte Seitenbreite ein!
- die definieren Umgebungen sind:
 - framed
 - shaded
 - snugshade
 - leftbar


 TEST

`\imp{TEST}`

 TEST

`\defi{TEST}`

 TEST

`\bsp{\centerline{TEST}}`

 TEST

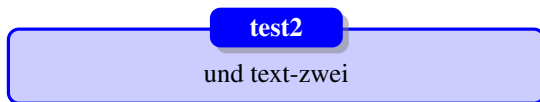
`\note{TEST}`

 TEST

`\Note{TEST}`



```
\colorbox{0.6}{orange}{wichtig}{und text}
```



```
\colorbox{0.95}{blue}{\textcolor{white}{test 2}}{\centerline{und text-zwei}}
```

12 Indexregister erstellen

- `\usepackage{makeidx}`
- `\makeindex`
- beides direkt untereinander und noch vor dem Begin des Dokuments!
- Einfügen von `\printindex` vor dem Dokument-Ende (bzw. an der Stelle, wo das Indexregister erscheinen soll)
- An den Stellen/Wörtern, die in den Index eingetragen werden sollen, folgendes anfügen:
`\index{Indexeintrag_bzw_Wort}`
- oder `\index{Eintrag!Untereintrag}`
- `\index{Virtuell@Eintrag}`
Virtuelle Einträge sind notwendig, um Sonderzeichen oder mathematische Symbole in den Index einzuordnen
`\index{wünschenswert@wünschenswert}`
`\index{R@R}`