

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

LaTeX Crashkurs  
Mentoring SoSe 2016

Anja Wolffgramm, Diane Hanke  
Freie Universität Berlin

29/044/2016

## Was ist $\text{\LaTeX}$ ?

## Aufbau eines Dokuments

- Dokumententypen

- Gliederung

- Inhaltsverzeichnis

## Befehle und Umgebungen

- Befehle

- Umgebungen

## Textstrukturierung

- Absätze

- Auflistung, Aufzählung

- Nummerierung

- Tabulars

- Minipages

## Bilder

- Bilder einbinden

- Bildumgebung

- Bildverzeichnis

## Mathematikmodus

## Quellcode einbinden

- Liste der Programmcodes

## Eigene Kommandos und

## Umgebungen

- Eigene Kommandos erstellen

- Eigene Umgebungen erstellen

- ▶  $\LaTeX$  ist eine Sammlung von **Makros**, die die Nutzung von  $\TeX$  ermöglicht
- ▶  $\TeX$  wiederum ist ein **Drucksatzsystem**, mit dem sich Texte formatieren lassen
- ▶ Text und Formatierungselemente werden **direkt in den Quelltext** geschrieben
- ▶ **Vorteile:** Einfachheit in der Darstellung komplexer Strukturen, mathematischer Formeln, Grafiken und ähnlichem

## Was ist $\text{\LaTeX}$ ?: Installation

### ► **Windows:**

1. **MikTeX Installer** installieren: <http://miktex.org/download>
2. einzelne **Pakete** installieren: "*MiKTeX*» "*Maintenance (Admin)*» "*Package Manager (Admin)*"
3.  $\text{\LaTeX}$  **Editor** einrichten, z.B. Texmaker

### ► **Linux:**

1. `$ sudo apt-get install texlive-full`
2. `$ sudo apt-get install texlive-xetex`
3. `$ sudo apt-get install texlive texlive-doc-de texlive-latex-extra texlive-lang-german`
4. `$ sudo apt-get install latex-xcolor pgf tex-common texlive texlive-base texlive-base-bin texlive-common texlive-fonts-extra texlive-fonts-recommended texlive-lang-german texlive-latex-base texlive-latex-extra texlive-latex-recommended`
5. `$ sudo apt-get install texlive-pictures dot2tex sketch libqtengine1 texlive-humanities texlive-pstricks`
6. `$ wget`  
<http://mirror.ctan.org/graphics/pgf/contrib/tikz-qtrees.zip>
  - **im Ordner entpacken**
7. `$ sudo mv tikz-qtrees/ /usr/share/texmf/tex/latex/`
8. `$ sudo texhash`

Was ist  $\text{\LaTeX}$ ?

## Aufbau eines Dokuments

Dokumententypen

Gliederung

Inhaltsverzeichnis

## Befehle und Umgebungen

Befehle

Umgebungen

## Textstrukturierung

Absätze

Auflistung, Aufzählung

Nummerierung

Tabulars

Minipages

## Bilder

Bilder einbinden

Bildumgebung

Bildverzeichnis

## Mathematikmodus

## Quellcode einbinden

Liste der Programmcodes

## Eigene Kommandos und

## Umgebungen

Eigene Kommandos erstellen

Eigene Umgebungen erstellen

- ▶  $\LaTeX$  bringt standardmäßig verschiedene Grunddokumententypen mit, die **verschiedene Eigenschaften** haben
- ▶ in jedem Dokument muss die **Dokumentenklasse** angegeben werden  $\Rightarrow$  erster Befehl eines Dokuments
- ▶ Befehl zur Angabe der Klasse beinhaltet die Dokumentenklasse (in geschweiften Klammern) sowie die Angabe möglicher **Optionen** (in eckigen Klammern)

```
1 \documentclass[Optionen]{Dokumentenklasse}
```

- ▶ **Übersicht:** [http://www.kkittel.de/wiki/doku.php?id=grundlegende\\_einstellungen:dokumentenklassen](http://www.kkittel.de/wiki/doku.php?id=grundlegende_einstellungen:dokumentenklassen)

- ▶ **Article:** geeignet für kurze technische Artikel (*scrartcl*)
- ▶ **Report:** geeignet für längere technische Artikel (*scrreprt*)
- ▶ **Book:** Drucklayout standardmäßig zweiseitig; z.B.: automatische Kopfzeile mit Seitenzahl (*scrbook*)
- ▶ **Letter:** geeignet zum Schreiben von Briefen (*scrletter2*)

Was ist  $\text{\LaTeX}$ ?

## Aufbau eines Dokuments

Dokumententypen

**Gliederung**

Inhaltsverzeichnis

Befehle und Umgebungen

Befehle

Umgebungen

Textstrukturierung

Absätze

Auflistung, Aufzählung

Nummerierung

Tabulars

Minipages

Bilder

Bilder einbinden

Bildumgebung

Bildverzeichnis

Mathematikmodus

Quellcode einbinden

Liste der Programmcodes

Eigene Kommandos und

Umgebungen

Eigene Kommandos erstellen

Eigene Umgebungen erstellen



Eine Überschrift wird mit folgendem Befehl erstellt:

```
1 \section{my Headline}
```

Eine Überschrift wird mit folgendem Befehl erstellt:

```
1 \section{my Headline}
```

Unterüberschrift:

```
1 \subsection{my Subhead}
```

Eine Überschrift wird mit folgendem Befehl erstellt:

```
1 \section{my Headline}
```

Unterüberschrift:

```
1 \subsection{my Subhead}
```

Überschrift dritten Grades:

```
1 \subsubsection{my Subhead}
```

Was ist  $\text{\LaTeX}$ ?

## Aufbau eines Dokuments

Dokumententypen

Gliederung

Inhaltsverzeichnis

Befehle und Umgebungen

Befehle

Umgebungen

Textstrukturierung

Absätze

Auflistung, Aufzählung

Nummerierung

Tabulars

Minipages

Bilder

Bilder einbinden

Bildumgebung

Bildverzeichnis

Mathematikmodus

Quellcode einbinden

Liste der Programmcodes

Eigene Kommandos und

Umgebungen

Eigene Kommandos erstellen

Eigene Umgebungen erstellen

Das Inhaltsverzeichnis wird automatisch mit dem Befehl

```
1 \tableofcontents
```

erstellt. Dabei werden alle Überschriften übernommen.

Das Inhaltsverzeichnis wird automatisch mit dem Befehl

```
1 \tableofcontents
```

erstellt. Dabei werden alle Überschriften übernommen.

Möchte man nicht, dass eine Überschrift im Verzeichnis auftaucht, benutzt man das Sternchen:

```
1 \section*{diese Überschrift taucht nicht auf}
```

Was ist  $\text{\LaTeX}$ ?

Aufbau eines Dokuments

Dokumententypen

Gliederung

Inhaltsverzeichnis

**Befehle und Umgebungen**

**Befehle**

Umgebungen

Textstrukturierung

Absätze

Auflistung, Aufzählung

Nummerierung

Tabulars

Minipages

Bilder

Bilder einbinden

Bildumgebung

Bildverzeichnis

Mathematikmodus

Quellcode einbinden

Liste der Programmcodes

Eigene Kommandos und

Umgebungen

Eigene Kommandos erstellen

Eigene Umgebungen erstellen

Ein Befehl wird mit einem Backslash eingeleitet:

```
1 \huge nachfolgender Text
```



Ein Befehl wird mit einem Backslash eingeleitet:

```
1 \huge nachfolgender Text
```

Und so sieht es aus: **nachfolgender Text**

Ein Befehl wird mit einem Backslash eingeleitet:

```
1 \huge nachfolgender Text
```

Und so sieht es aus: **nachfolgender Text**

Manche Befehle erhalten Argumente in geschweiften Klammern:

```
1 \textcolor{Pink}{mein pinker Unicorn-Text}
```

Ein Befehl wird mit einem Backslash eingeleitet:

```
1 \huge nachfolgender Text
```

Und so sieht es aus: **nachfolgender Text**

Manche Befehle erhalten Argumente in geschweiften Klammern:

```
1 \textcolor{Pink}{mein pinker Unicorn-Text}
```

Und so sieht es aus: **mein pinker Unicorn-Text**

Was ist  $\text{\LaTeX}$ ?

Aufbau eines Dokuments

Dokumententypen

Gliederung

Inhaltsverzeichnis

**Befehle und Umgebungen**

Befehle

**Umgebungen**

Textstrukturierung

Absätze

Auflistung, Aufzählung

Nummerierung

Tabulars

Minipages

Bilder

Bilder einbinden

Bildumgebung

Bildverzeichnis

Mathematikmodus

Quellcode einbinden

Liste der Programmcodes

Eigene Kommandos und

Umgebungen

Eigene Kommandos erstellen

Eigene Umgebungen erstellen

Eine Umgebung fasst einen Bereich ein. Sie hat immer die folgende Form:

```
1 \begin{meineUmgebung}  
2 Teil dazwischen  
3 \end{meineUmgebung}
```

Eine Umgebung fasst einen Bereich ein. Sie hat immer die folgende Form:

```
1 \begin{meineUmgebung}
2 Teil dazwischen
3 \end{meineUmgebung}
```

Als Beispiel dient da die Dokument-Umgebung:

```
1 \begin{document}
2 mein cooles LaTeX-Dokument
3 \end{document}
```

# Inhalt

Was ist  $\text{\LaTeX}$ ?

Aufbau eines Dokuments

Dokumententypen

Gliederung

Inhaltsverzeichnis

Befehle und Umgebungen

Befehle

Umgebungen

**Textstrukturierung**

**Absätze**

Auflistung, Aufzählung

Nummerierung

Tabulars

Minipages

Bilder

Bilder einbinden

Bildumgebung

Bildverzeichnis

Mathematikmodus

Quellcode einbinden

Liste der Programmcodes

Eigene Kommandos und

Umgebungen

Eigene Kommandos erstellen

Eigene Umgebungen erstellen

## Absätze und Zeilenumbrüche

- ▶ sind inhaltliche Einheiten
- ▶ erste Zeile eingerückt und Abstand zwischen zwei Absätzen



## Absätze und Zeilenumbrüche

- ▶ sind inhaltliche Einheiten
- ▶ erste Zeile eingerückt und Abstand zwischen zwei Absätzen

Entweder durch eine einfache Leerzeile oder mit

```
1 \par
```

## Absätze und Zeilenumbrüche

- ▶ sind inhaltliche Einheiten
- ▶ erste Zeile eingerückt und Abstand zwischen zwei Absätzen

Entweder durch eine einfache Leerzeile oder mit

```
1 \par
```

Alternativen:

```
1 \newline
```

## Absätze und Zeilenumbrüche

- ▶ sind inhaltliche Einheiten
- ▶ erste Zeile eingerückt und Abstand zwischen zwei Absätzen

Entweder durch eine einfache Leerzeile oder mit

```
1 \par
```

Alternativen:

```
1 \newline
```

```
1 \\
```

Was ist  $\text{\LaTeX}$ ?

Aufbau eines Dokuments

Dokumententypen

Gliederung

Inhaltsverzeichnis

Befehle und Umgebungen

Befehle

Umgebungen

Textstrukturierung

Absätze

Auflistung, Aufzählung

Nummerierung

Tabulars

Minipages

Bilder

Bilder einbinden

Bildumgebung

Bildverzeichnis

Mathematikmodus

Quellcode einbinden

Liste der Programmcodes

Eigene Kommandos und

Umgebungen

Eigene Kommandos erstellen

Eigene Umgebungen erstellen

# Auflistung, Aufzählung, Nummerierung

- ▶ unnummerierte Auflistung mit **itemize**
  - ▶ kann bis zu vier Ebenen tief geschachtelt werden

```
1 \begin{itemize}
2 \item
3 ...
4 \end{itemize}
```

# Auflistung, Aufzählung, Nummerierung

- ▶ unnummerierte Auflistung mit **itemize**
  - ▶ kann bis zu vier Ebenen tief geschachtelt werden

```
1 \begin{itemize}
2 \item
3 ...
4 \end{itemize}
```

- ▶ nummerierte Auflistung mit **enumerate**
  - ▶ kann ebenfalls bis zu vier Ebenen tief geschachtelt werden

```
1 \begin{enumerate}
2 \item
3 ...
4 \end{enumerate}
```

## Auflistung, Aufzählung, Nummerierung

- ▶ unnummerierte Auflistung mit **itemize**
  - ▶ kann bis zu vier Ebenen tief geschachtelt werden

```
1 \begin{itemize}
2 \item
3 ...
4 \end{itemize}
```

- ▶ nummerierte Auflistung mit **enumerate**
  - ▶ kann ebenfalls bis zu vier Ebenen tief geschachtelt werden

```
1 \begin{enumerate}
2 \item
3 ...
4 \end{enumerate}
```

**Protipp:** Die Aufzählungszeichen lassen sich mit eigenen Kommandos verändern, aber dazu später mehr!

# Inhalt

Was ist  $\text{\LaTeX}$ ?

Aufbau eines Dokuments

Dokumententypen

Gliederung

Inhaltsverzeichnis

Befehle und Umgebungen

Befehle

Umgebungen

**Textstrukturierung**

Absätze

Auflistung, Aufzählung

Nummerierung

**Tabulars**

Minipages

Bilder

Bilder einbinden

Bildumgebung

Bildverzeichnis

Mathematikmodus

Quellcode einbinden

Liste der Programmcodes

Eigene Kommandos und

Umgebungen

Eigene Kommandos erstellen

Eigene Umgebungen erstellen



## Einfachste Tabelle mit:

```
1 \begin{tabular}{FORMAT}  
2 Tabelleninhalt  
3 \end{tabular}
```

- ▶ in das Feld {FORMAT} wird für jede Spalte die Formatierung angegeben
  - ▶ c - für zentrierten Text
  - ▶ l - für linksbündigen Text
  - ▶ r - für rechtsbündigen Text
  - ▶ p{SIZEcm} - Spalte soll SIZE cm breit sein
  - ▶ pipe - zeichnet einen senkrechten Strich zwischen die beiden Spalten

# Tabulars

Einfachste Tabelle mit:

```
1 \begin{tabular}{FORMAT}  
2 Tabelleninhalt  
3 \end{tabular}
```

- ▶ in das Feld {FORMAT} wird für jede Spalte die Formatierung angegeben
  - ▶ c - für zentrierten Text
  - ▶ l - für linksbündigen Text
  - ▶ r - für rechtsbündigen Text
  - ▶ p{SIZEcm} - Spalte soll SIZE cm breit sein
  - ▶ pipe - zeichnet einen senkrechten Strich zwischen die beiden Spalten

Beispiel:

```
1 \begin{tabular}{lcr}  
2 Spalte 1 & Spalte 2 & Spalte 3 \\  
3 1 & 2 & 3 \\  
4 \end{tabular}
```

# Tabellenverzeichnis

- ▶ mit `\listoftables` kann ein Tabellenverzeichnis erstellt werden
- ▶ die Tabellen müssen sich dafür in der `Table` Umgebung befinden

```
1 \begin{table}
2 \caption{\LaTeX \ Kursübersicht}
3 \begin{tabular}{|c|c|c|}
4 \hline
5 Zeitpunkt & Kursleiter & Titel \\
6 \hline
7 ...
8 \end{tabular}
9 \end{table}
```

- ▶ Im Tabellenverzeichnis steht dann `\LaTeX Kursübersicht S.??` .

Was ist  $\text{\LaTeX}$ ?

Aufbau eines Dokuments

Dokumententypen

Gliederung

Inhaltsverzeichnis

Befehle und Umgebungen

Befehle

Umgebungen

**Textstrukturierung**

Absätze

Auflistung, Aufzählung

Nummerierung

Tabulars

**Minipages**

Bilder

Bilder einbinden

Bildumgebung

Bildverzeichnis

Mathematikmodus

Quellcode einbinden

Liste der Programmcodes

Eigene Kommandos und

Umgebungen

Eigene Kommandos erstellen

Eigene Umgebungen erstellen

```
1 \begin{minipage}[ÄUSSERE POSITION][HÖHE][INNERE POSITION]{BREITE}
2 Beispieltext
3 \end{minipage}
```

- ▶ **ÄUSSERE POSITION** richtet die minipage relativ zur aktuellen Grundlinie aus
  - ▶ c = center
  - ▶ t = top
  - ▶ b = bottom
- ▶ **HÖHE**, ist eine gültige Längenangabe, durch die die Gesamthöhe der minipage bestimmt wird
- ▶ **INNERE POSITION**, richtet den Inhalt der minipage innerhalb der angegebenen HÖHE aus

Was ist  $\text{\LaTeX}$ ?

Aufbau eines Dokuments

Dokumententypen

Gliederung

Inhaltsverzeichnis

Befehle und Umgebungen

Befehle

Umgebungen

Textstrukturierung

Absätze

Auflistung, Aufzählung

Nummerierung

Tabulars

Minipages

**Bilder**

**Bilder einbinden**

Bildumgebung

Bildverzeichnis

Mathematikmodus

Quellcode einbinden

Liste der Programmcodes

Eigene Kommandos und

Umgebungen

Eigene Kommandos erstellen

Eigene Umgebungen erstellen

Ein Bild bindet man mit folgendem Code ein:

```
1 \includegraphics[OPTION]{path/name}
```

Ein Bild bindet man mit folgendem Code ein:

```
1 \includegraphics[OPTION]{path/name}
```

Options:

- ▶ `scale=1` (Werte  $> 0$ )
- ▶ `width=0.5\linewidth`
- ▶ `keepaspectratio=true`



Ein Bild bindet man mit folgendem Code ein:

```
1 \includegraphics[OPTION]{path/name}
```

Options:

- ▶ `scale=1` (Werte  $> 0$ )
- ▶ `width=0.5\linewidth`
- ▶ `keepaspectratio=true`

Beispiel:

```
1 \includegraphics[width=\linewidth]%  
2 {img/latex_is_beautiful}
```

# Bilder

Ein Bild bindet man mit folgendem Code ein:

```
1 \includegraphics[OPTION]{path/name}
```

Options:

- ▶ `scale=1` (Werte > 0)
- ▶ `width=0.5\linewidth`
- ▶ `keepaspectratio=true`

Beispiel:

```
1 \includegraphics[width=\linewidth]%  
2 {img/latex_is_beautiful}
```



Was ist  $\text{\LaTeX}$ ?

Aufbau eines Dokuments

Dokumententypen

Gliederung

Inhaltsverzeichnis

Befehle und Umgebungen

Befehle

Umgebungen

Textstrukturierung

Absätze

Auflistung, Aufzählung

Nummerierung

Tabulars

Minipages

**Bilder**

Bilder einbinden

**Bildumgebung**

Bildverzeichnis

Mathematikmodus

Quellcode einbinden

Liste der Programmcodes

Eigene Kommandos und

Umgebungen

Eigene Kommandos erstellen

Eigene Umgebungen erstellen



Abbildung: Bildbeschreibung

Was ist  $\text{\LaTeX}$ ?

Aufbau eines Dokuments

Dokumententypen

Gliederung

Inhaltsverzeichnis

Befehle und Umgebungen

Befehle

Umgebungen

Textstrukturierung

Absätze

Auflistung, Aufzählung

Nummerierung

Tabulars

Minipages

**Bilder**

Bilder einbinden

Bildumgebung

**Bildverzeichnis**

Mathematikmodus

Quellcode einbinden

Liste der Programmcodes

Eigene Kommandos und

Umgebungen

Eigene Kommandos erstellen

Eigene Umgebungen erstellen

Ein Bildverzeichnis erstellt  $\LaTeX$  automatisch mittels:

```
1 \listoffigures
```

- ▶ ist eine Umgebung, die dafür optimiert ist, mathematische Formeln und Symbole effizient zu erstellen
- ▶ zahlreiche Befehle, Symbole und Umgebungen funktionieren nur in diesem Modus
- ▶ andere funktionieren gerade in diesem Modus nicht und man muss für diese entsprechend andere Modi verwenden.

- ▶ Im Fließtext wird der betreffende Ausdruck mit Dollarzeichen  $\$$  umgeben. z.B.:  $\$E = mc^2\$$

Mitten im Text steht die Formel  $E = mc^2$  von Einstein.



- ▶ Im Fließtext wird der betreffende Ausdruck mit Dollarzeichen  $\$$  umgeben. z.B.:  $\$E = mc^2\$$

Mitten im Text steht die Formel  $E = mc^2$  von Einstein.

- ▶ Um einzeilige Formeln hervorzuheben, baut man eine Umgebung auf, innerhalb derer der Mathematikmodus eingeschaltet ist:

```
1 \[E=mc^2\]
```

Mit der **align-Umgebung** schaltet man den Mathematikmodus ein, hebt Formeln hervor und nummeriert sie:

```
1 \begin{align}
2 \sum_{i=0}^{n+1} i &= \sum_{i=0}^n i + (n+1) \\
3 &= \frac{n(n+1)}{2} + (n+1)
4 \end{align}
```

Mit der **align-Umgebung** schaltet man den Mathematikmodus ein, hebt Formeln hervor und nummeriert sie:

```

1 \begin{align}
2 \sum_{i=0}^{n+1} i &=& \sum_{i=0}^n i + (n+1) \\
3 && =& \frac{n(n+1)}{2} + (n+1)
4 \end{align}

```

Und so sieht es aus:

$$\sum_{i=0}^{n+1} i = \sum_{i=0}^n i + (n+1) \tag{1}$$

$$= \frac{n(n+1)}{2} + (n+1) \tag{2}$$

Will man keine Nummerierung, nutzt man **align\***:

```

1 \begin{align*}
2 \sum_{i=0}^{n+1} i &=& \sum_{i=0}^n i + (n+1) \\
3 && =& \frac{n(n+1)}{2} + (n+1)
4 \end{align*}

```

- ▶ Beispiele für Befehle, die nur im Mathematikmodus funktionieren
  - ▶  $\bar{x}$
- ▶ Textmodusbefehle, die im Mathematikmodus ersetzt werden müssen:
  - ▶ `tabular`  $\Rightarrow$  `array`
  - ▶ `textbf`  $\Rightarrow$  `mathbf`
  - ▶ `v`  $\Rightarrow$  `breve`

Wenn Text nicht vom  $\text{\LaTeX}$ -Compiler interpretiert werden soll, kann man ihn in ein **verb**-Befehl setzen:

```
1 | \verb| nicht interpretierter Text |
```

Dabei muss dem Befehl mitgeteilt werden, wann dies endet. Dies geschieht durch ein Zeichen, das den Bereich einleitet und abschließt.

Wenn Text nicht vom  $\text{\LaTeX}$ -Compiler interpretiert werden soll, kann man ihn in ein **verb**-Befehl setzen:

```
1 \verb| nicht interpretierter Text |
```

Dabei muss dem Befehl mitgeteilt werden, wann dies endet. Dies geschieht durch ein Zeichen, das den Bereich einleitet und abschließt.

Für mehrere Zeilen gibt es die **verbatim**-Umgebung:

```
1 \begin{verbatim}  
2 diese Zeilen  
3 sollen nicht interpretiert  
4 werden  
5 \end{verbatim}
```

Um Programmcode in das Dokument einzubinden, möchte man jedoch Syntax-Highlighting und Zeilennummerierungen haben.

## Quellcode inline einbinden

Um Programmcode in das Dokument einzubinden, möchte man jedoch Syntax-Highlighting und Zeilennummerierungen haben.

Dafür gibt es die **lstlisting**-Umgebung:

```
1 \begin {lstlisting}[style=tex, caption=Beschriftung]
2   % hier kann man wild coden
3 \end {lstlisting}
```



## Quellcode inline einbinden

Um Programmcode in das Dokument einzubinden, möchte man jedoch Syntax-Highlighting und Zeilennummerierungen haben.

Dafür gibt es die **lstlisting**-Umgebung:

```
1 \begin {lstlisting}[style=tex, caption=Beschriftung]
2   % hier kann man wild coden
3 \end {lstlisting}
```

Dafür haben wir vier styles vorbereitet: *java*, *hs*, *tex* und *py*. Hier ein Beispiel für Python:

### Listing 3: Beschriftung

```
1 def myFunction():
2   # hier kann man wild coden
3   return True
```

In der Regel programmiert man in einer Datei und möchte diese im  $\LaTeX$ -Dokument einbinden. Dies geht mittels **lstinputstring**:

## Listing 4: Istlisting

```
1 \lstinputlisting[style=tex, caption={Beschriftung}]{pfad/filename.tex}
```

In der Regel programmiert man in einer Datei und möchte diese im  $\text{\LaTeX}$ -Dokument einbinden. Dies geht mittels **lstinputstring**:

## Listing 5: Istlisting

```
1 \lstinputstring[style=tex, caption={Beschriftung}]{pfad/filename.tex}
```

Man kann auch nur ein paar Zeilen des Programmcodes einbinden:

```
1 \lstinputstring[style=tex, firstline=17, lastline=42]{pfad/filename.tex}
```

Was ist  $\text{\LaTeX}$ ?

Aufbau eines Dokuments

Dokumententypen

Gliederung

Inhaltsverzeichnis

Befehle und Umgebungen

Befehle

Umgebungen

Textstrukturierung

Absätze

Auflistung, Aufzählung

Nummerierung

Tabulars

Minipages

Bilder

Bilder einbinden

Bildumgebung

Bildverzeichnis

Mathematikmodus

**Quellcode einbinden**

Liste der Programmcodes

Eigene Kommandos und

Umgebungen

Eigene Kommandos erstellen

Eigene Umgebungen erstellen

Eine Liste der `lstlistings` wird wie folgt erstellt:

```
1 \lstlistoflistings
```

# Inhalt

Was ist  $\text{\LaTeX}$ ?

Aufbau eines Dokuments

Dokumententypen

Gliederung

Inhaltsverzeichnis

Befehle und Umgebungen

Befehle

Umgebungen

Textstrukturierung

Absätze

Auflistung, Aufzählung

Nummerierung

Tabulars

Minipages

Bilder

Bilder einbinden

Bildumgebung

Bildverzeichnis

Mathematikmodus

Quellcode einbinden

Liste der Programmcodes

**Eigene Kommandos und**

**Umgebungen**

**Eigene Kommandos erstellen**

Eigene Umgebungen erstellen

Man kann sich eigene Kommandos erstellen, um z. B. Schreibaufwand einzusparen.

```
1 \newcommand{\name}{Was es tun soll}
```

## Eigene Kommandos erstellen

Man kann sich eigene Kommandos erstellen, um z. B. Schreibaufwand einzusparen.

```
1 \newcommand{\name}{Was es tun soll}
```

Hier ein Beispiel:

```
1 \newcommand\zz{\ensuremath{\raisebox{+0.25ex}{Z}% zu-zeigen-Symbol  
2 \kern-0.4em\raisebox{-0.25ex}{Z}%  
3 \;\xspace}%  
4 }
```



## Eigene Kommandos erstellen

Man kann sich eigene Kommandos erstellen, um z. B. Schreibaufwand einzusparen.

```
1 \newcommand{\name}{Was es tun soll}
```

Hier ein Beispiel:

```
1 \newcommand\zz{\ensuremath{\raisebox{+0.25ex}{Z}}% zu-zeigen-Symbol  
2 \kern-0.4em\raisebox{-0.25ex}{Z}}%  
3 \;\xspace}%  
4 }
```

Und so sieht es aus:  $\mathbb{Z}$

Es lassen sich auch Kommandos erstellen, welche Argumente erhalten und diese benutzen:

```
1 \newcommand{\name}[Argumentzahl]{was es mit dem Argument #1 tun soll}
```

## Eigene Kommandos

Es lassen sich auch Kommandos erstellen, welche Argumente erhalten und diese benutzen:

```
1 \newcommand{\name}[Argumentzahl]{was es mit dem Argument #1 tun soll}
```

Hier ein Beispiel:

```
1 \newcommand{\Quellcode}[3]{\lstinputlisting[%  
2 style=#2, % 2. Argument: filename  
3 caption={#3}] % 3. Argument: Beschriftung  
4 {#1.#2}} % 1. Argument: path
```

## Eigene Kommandos

Es lassen sich auch Kommandos erstellen, welche Argumente erhalten und diese benutzen:

```
1 \newcommand{\name}[Argumentzahl]{was es mit dem Argument #1 tun soll}
```

Hier ein Beispiel:

```
1 \newcommand{\Quellcode}[3]{\lstinputlisting[%  
2 style=#2, % 2. Argument: filename  
3 caption={#3}] % 3. Argument: Beschriftung  
4 {#1.#2}} % 1. Argument: path
```

In der Anwendung:

```
1 \Quellcode{src/myFile}{py}{Ein Python-Programm}
```

Was ist  $\text{\LaTeX}$ ?

Aufbau eines Dokuments

Dokumententypen

Gliederung

Inhaltsverzeichnis

Befehle und Umgebungen

Befehle

Umgebungen

Textstrukturierung

Absätze

Auflistung, Aufzählung

Nummerierung

Tabulars

Minipages

Bilder

Bilder einbinden

Bildumgebung

Bildverzeichnis

Mathematikmodus

Quellcode einbinden

Liste der Programmcodes

**Eigene Kommandos und**

**Umgebungen**

Eigene Kommandos erstellen

Eigene Umgebungen erstellen

# Eigene Umgebungen erstellen

Möchte man eine bereits vorhandene Umgebung modifizieren, geht dies nicht mittels Kommando, sondern mit `\newenvironment` .

```
1 \newenvironment{name}[Argumentzahl]{Befehlsbeginn}{Befehlsende}
```

## Eigene Umgebungen erstellen

Möchte man eine bereits vorhandene Umgebung modifizieren, geht dies nicht mittels Kommando, sondern mit `\newenvironment` .

```
1 \newenvironment{name}[Argumentzahl]{Befehlsbeginn}{Befehlsende}
```

Nachfolgend ein Beispiel:

```
1 \newenvironment{Magic}[1][Pink]% hat 1 optionales Arg., Standardwert: Pink
2   {\begin{center}\begingroup\textcolor{#1}}%
3   {\endgroup\end{center}}
```

## Eigene Umgebungen erstellen

Möchte man eine bereits vorhandene Umgebung modifizieren, geht dies nicht mittels Kommando, sondern mit `\newenvironment`.

```
1 \newenvironment{name}[Argumentzahl]{Befehlsbeginn}{Befehlsende}
```

Nachfolgend ein Beispiel:

```
1 \newenvironment{Magic}[1][Pink]% hat 1 optionales Arg., Standardwert: Pink
2   {\begin{center}\begingroup\textcolor{#1}}%
3   {\endgroup\end{center}}
```

Hier ein Beispiel:

```
1 \begin{Magic}
2   mein toller Text
3 \end{Magic}
```

mein toller Text



Um euch den Einstieg etwas zu erleichtern, haben wir für euch ein **Template** erstellt.

Dieses könnt ihr für die Abgabe von Übungszetteln, zum Schreiben von Skripten oder einfach nur zum Üben benutzen.

**Möge L<sup>A</sup>T<sub>E</sub>X mit euch sein!**