



# TeX - LaTeX

Wissenschaftliches Arbeiten in CSE

Makros

Umgebungen

Tabellen

Bilder

Literaturverzeichnis

## Definieren von Makros

- Definition eines neuen Makros mittels
  - `\newcommand{\name}[anz]{definition}`
- Obligatorisch sind
  - Name des Makros `name`
  - Befehlsfolge des Makros `definition`
- Optional ist Anzahl `anz` der obligatorischen Parameter des Makros
  - Fehlt `anz` so ist `\name` parameterlos
  - maximal 9 Parameter, intern `#1, ... #9`
- Beispiele:
  - `\newcommand{\R}{\mathbb{R}}`
    - Aufruf: `\R`
    - Ausgabe:  $\mathbb{R}$
  - `\newcommand{\norm}[1]{\left| \#1\right|}`
    - Aufruf: `\norm{f}`
    - Ausgabe:  $\|f\|$
  - `\newcommand{\set}[2]{\big\{ \#1\,,\big| \,, \#2\right\}}`
    - Aufruf: `\set{x\in\R}{f(x)=0}`
    - Ausgabe:  $\{x \in \mathbb{R} \mid f(x) = 0\}$
- $\text{\LaTeX}$  passt auf, ob Makro-Name bereits vergeben ist
  - **! LaTeX Error: Command XXX already defined.**
  - Altes überschreiben mittels `\renewcommand`
    - Parameter/Verwendung wie bei `\newcommand`

## Warum Makros?

- Lesbarkeit des Codes, insbesondere Mathematischer Formeln
  - `\set{x\in\mathbb{R}}{f(x)=0}`  
versus
  - `\bigl\{x\in\mathbb{R}\bigr\}, \bigl| \ , f(x)=0\bigr\}`
- Code wird kürzer & übersichtlicher
- Vermeidung von Tippfehlern
- Wiederverwendbarkeit in weiteren Dokumenten
- einfache Anpassung von math. Notation
  - z.B.  $\{x \in \mathbb{R} \mid f(x) = 0\}$  vs.  $\{x \in \mathbb{R} : f(x) = 0\}$
- Umstellung der Notation im gesamten Dokument durch Änderung einer Zeile

## Was sollte man bei der Definition von Makros beachten?

- Sprechende Namen für Makros verwenden
  - `\set` , `\norm`, `\scalarproduct`
- Kurze Namen nur für reine Zeichen z.B.
  - `\bbN`, `\bbZ`, `\bbR` etc. für `mathbb`-Symbole  $\mathbb{N}$ ,  $\mathbb{Z}$ ,  $\mathbb{R}$
  - `\cA`, `\cB`, `\cC` etc. für `mathcal`-Symbole  $\mathcal{A}$ ,  $\mathcal{B}$ ,  $\mathcal{C}$
  - `\x`, `\y`, `\z` etc. für Vektoren  $\vec{x}$ ,  $\vec{y}$ ,  $\vec{z}$
- Keine Makros zur puren Abkürzung von Tipparbeit
  - Solchen Code kann man später nicht mehr lesen!

## Makros mit optionalen Parametern

- Man kann Makros definieren deren erster Parameter optional ist
  - `\newcommand{\name}[anz] [default1]{definition}`
  - name, anz, definition wie bisher
  - Parameter #1 ist optional
    - Übergabe in eckigen Klammern [parameter1]
    - Wert default1, falls nicht gegeben
  - Parameter #2, ...#anz sind obligatorisch
    - Übergabe in Klammern {parameter}
  
- Zum Beispiel:
  - `\newcommand{\norm}[2] [] {\left \| #2\right \|_{{#1}}}`
    - Aufruf: `\norm[L^2(\Omega)]{f}`      Ausgabe:  $\|f\|_{L^2(\Omega)}$
    - Aufruf: `\norm{f}`      Ausgabe:  $\|f\|$
  - `\newcommand{\set}[3] [\Big] {\#1 \setminus \#2, \#1\|, \#3\#1\}`
    - Aufruf: `\set{x\in\mathbb{R}}{f(x)=0}`      Ausgabe:  $\{x \in \mathbb{R} | f(x) = 0\}$
    - Aufruf: `\set[\Big]{x\in\mathbb{R}}{f(x)=0}\norm{f}`      Ausgabe:  $\left\{x \in \mathbb{R} | f(x) = 0\right\}$

## Weitere Text-Umgebungen

- Wir kennen bereits `center`, `flushright`, `flushleft`-Umgebung
- für Zitate: `quote`-Umgebung
- als ob Schreibmaschine `verbatim`-Umgebung
- Aufzählungen: `itemize`-Umgebung
  - jeder Punkt wird mit `\item` eingeleitet
  - optional `\item[zeichen]` für anderes Symbol
- Nummerierte Aufzählung: `enumerate`-Umgebung
  - jeder Punkt wird mit `\item` eingeleitet
  - Art der Aufzählung über Zähler manipulierbar
    - `enumi`
    - `enumii`, `enumiii`, `enumiv` bei geschachtelten `enumerate`-Umgebungen
  - `\usepackage{enumerate}` hat mehr Funktionalität
    - Erweiterung der `enumerate`-Umgebungen um optionale Layout-Parameter

## Warum Umgebungen?

- Viele Objekte in mathematischen Texten sollten dasselbe Layout haben
  - z.B. Sätze, Definitionen, Beweise
- Umgebungen trennen Inhalt und Layout
  - Code wird lesbarer
  - Layout wird leichter veränderbar

## Definition einer Umgebung

- `\newenvironment{name}[anz]{defbegin}{defend}`
- name, anz wie bei `\newcommand`
- defbegin = Was löst `\begin{name}` aus?
- defend = Was löst `\end{name}` aus?
- `\renewenvironment` analog zu `\renewcommand`
- **Beispiel:**
  - `\newenvironment{proof}{\textbf{Beweis.}}{\hfill\textbf{q.e.d.}}`
  - **Eingabe:** `\begin{proof} ... \end{proof}`    **Ausgabe:** **Beweis. ...**    **q.e.d.**

## Wichtige mathematische Pakete

- `amsmath` = Umgebungen, Befehle
  - im Folgenden `\usepackage{amsmath}` notwendig!
- `amsth` = Theorem-Umgebungen
- `amsfonts`, `amssymb` = Schriftarten + Symbole

## Praktische Umgebungen

- `matrix`-Umgebung für Vektoren und Matrizen
  - bequemer als `array`-Umgebung, weil man die Anzahl der Spalten nicht angeben muss
  - ansonsten gleiche Syntax
    - zeilenweise Angabe
    - `&` für neue Spalte
    - `\\` für neue Zeile
- `pmatrix`-Umgebung
  - `=\left(\begin{matrix}...\end{matrix}\right)`
- `case`-Umgebung
  - `=\left(\begin{array}[1l]...\end{array}\right)`
- **Beispiel 15:** `array`, `pmatrix` und `binom`

## Die align-Umgebung

- Flexibler als `equation`- und `eqnarray`-Umgebung
- Mit (`align`) und ohne (`align*`) Formelnummer
- Erlaubt mehrzeilige Formeln, Zeilenumbruch `\\`
- Ordnet tabellarisch an
  - & neue Spalte
  - Spalten abwechselnd links/rechts ausgerichtet
  - Spaltenpaar links/rechts bildet jeweils Gruppe ohne Anstand
- `\tag{text}` ersetzt Formelnummer durch Text `\notag` unterdrückt Ausgabe der Formelnummer
  - falls nur manche Zeilen einer mehrzeiligen Formel Nummer haben sollen
- In Verbindung mit `split`-Umgebung kann man Formelnummern mehrzeiliger Formeln vertikal zentrieren
- **Beispiel 16:** `align`-Umgebung

## Mathematische Sätze

- Umgebung für math. Sätze etc. können leicht(!) erstellt werden, d.h. `\newenvironment` hier unnötig!
- `\newtheorem{name}[counter]{text}[supercounter]`
  - Obligatorisch:
    - **name**: Name der neuen Umgebung
    - **text**: Überschrift der neuen Umgebung, z.B. Satz, Lemma
  - Optional:
    - **counter**: falls kein neuer Zähler angelegt werden soll, sondern vorhandener mitbenutzt wird
    - **supercounter**: spezifiziert übergeordneten Zähler, z.B. **section**: wenn Section erhöht, wird **counter** auf 0 gesetzt
    - gleiche Funktion wie `\numberwithin`
- **Beispiele**:
  - `\newtheorem{satz}{Satz}[section]`
    - Satz-Umgebung
    - Zähler zählt in jeder section neu
  - `\newtheorem{lemma}{Satz}[Lemma]`
    - Satz & Lemma werden gemeinsam nummeriert
  - `\newtheorem{bemerkung}{Bemerkung}[section]`
    - Bemerkungen werden unabhängig nummeriert
    - Zähler zählt in jeder Section neu
- **Beispiel 17**: `newtheorem.tex`

## Die tabbing-Umgebung

- Zur Spaltenweisen Ausrichtung von Text
- \= Markierung setzen
- \kill Zeile nicht ausgeben
  - für Definitionszeile
- \> Textposition auf nächste Markierung setzen

## Quelldatei ( $\LaTeX$ )

```
% tabbing.tex
\documentclass[a4paper,12pt]{report}
\usepackage{fullpage}
\usepackage[ngerman]{babel}
\usepackage[applemac]{inputenc}

\begin{document}
\noindent Jetzt kommt eine \texttt{tabbing}-Umgebung:
\begin{tabbing}
% Definition der Tabulator-Stops
\hspace*{25mm} \= \hspace*{3cm} \= \hspace*{5cm} \= \kill
% Der ausgerichtete Text
Spalte 1 \> Spalte 2 \> Spalte 3 \> Spalte 4\\
Text A   \> Text B   \> Text C           \\
         \> Weiter   \> so               \> !!
\end{tabbing}
\end{document}
```

## Ausgabe-Datei (PDF)

Spalte 1	Spalte 2	Spalte 3	Spalte 4
Text A	Text B	Text C	
	Weiter	so	!!

## Die tabular-Umgebung

- Benutzung wie array-Umgebung
  - Anzahl Spalten angeben & Ausrichtung
    - mittig {c}, links {l}, rechts {r}
    - Blocksatz mit fester Spaltenbreite p{Breite}
  - vertikale Trennlinie mit Pipe | angeben
    - oder eigene Trennlinie mit
  - Zeilenumbruch mit \\
  - horizontale Trennlinie mit \hline
- kann Trennlinien auch in array-Umgebung nutzen
- Verwende \cline{von-bis}, falls horizontale Linie nur Spalte von bis bis
- Verwende \multicolumn{anz}{style}{text} für Eintrag text über mehrere Spalten
  - anz = Anzahl der betroffenen Spalten
  - style = analog zu tabular-Style, z.B. {l|c|}

## Die tabular-Umgebung

### Quelldatei (L<sup>A</sup>T<sub>E</sub>X)

```
% tabular.tex
\documentclass[a4paper,12pt]{report}
\usepackage{fullpage}
\usepackage[ngerman]{babel}
\usepackage[applemac]{inputenc}

\begin{document}
\begin{tabular}{|l| |c| |c| |r|}
\hline
links & mittig & rechts & rechts\\
\hline\hline
1 & 2 & 3 & 4\\
5 & 6 & 7 & 8\\
\hline
\end{tabular}
\end{document}
```

### Ausgabe-Datei (PDF)

links	mittig	rechts	rechts
1	2	3	4
5	6	7	8

- [Beispiel 18: tabular.tex](#)

## Die table-Umgebung

- I.d.R. soll Tabelle nicht Teil von Text sein, sondern herausgehoben mit Unterschrift und Nummer
  - verwende `table`-Umgebung
- `\caption` gibt der Tabelle eine Unterschrift
- `table`-Umgebung erzeugt eine so genanntes **float object**
  - wird von  $\LaTeX$  automatisch platziert
  - wird intern in Liste eingetragen und sobald als möglich gesetzt
    - First-In-First-Out Prinzip
    - `\clearpage` arbeitet Float-Liste ab, danach Seitenumbruch (`\newpage` = neue Seite)
- Präferenz für Platzierung kann optional als Liste angegeben werden
  - z.B. `\begin{table}[!thpb]`
    - `!` = egal, ob es vernünftig scheint
    - `t` = top
    - `h` = here
    - `t` = page = Extraseite nur mit floats
    - `b` = bottom
  - wird in der angegebenen Reihenfolge von  $\LaTeX$  in Erwägung gezogen
- erzeugt Tabellen-Verzeichnis
  - Einträge werden aus `\caption{...}` übernommen
    - erstes `latex name.tex` erzeugt `name.lot`
    - zweites `latex name.tex` bindet Verzeichnis ein
  - Falls Unterschrift zu lang ist, Kurztitel festlegen
    - `\caption[kurztitel]{unterschrift}\verb`

## Bilder einbinden

- Voraussetzung: `\usepackage[dvips]{graphicx}`
- Einbinden: `\includegraphics[options]{filename}`
  - Optionale Parameter sind:
    - `width = unit`: Breite festlegen (& ggf. skalieren)
    - `height = unit`: Höhe festlegen (& ggf. skalieren)
    - `scale = num`: Bild skalieren
    - `angle = unit`: Bild drehen (math. pos. Grad)

## Quelldatei (L<sup>A</sup>T<sub>E</sub>X)

```
%includegraphics.tex
\documentclass[a4paper,12pt]{report}
\usepackage{fullpage}
\usepackage[ngerman]{babel}
\usepackage[applemac]{inputenc}
\usepackage[dvips]{graphicx}

\begin{document}
\includegraphics[width=0.75\textwidth]{Grafiken/UniLogo.pdf}

\includegraphics[width=0.75\textwidth,angle=25]{Grafiken/UniLogo.pdf}

\includegraphics[width=3cm,angle=-25]{Grafiken/TeXLogo.pdf}
\end{document}
```

## Ausgabe-Datei (PDF)



ulm university universität

uulm



ulm university universität

uulm

TeX-Latex

## Die figure-Umgebung

- Verwendung von `figure` analog zu `table`
- `\listoffigures` erzeugt Abbildungsverzeichnis
  - erzeugt Datei `name.laf`

## Quelldatei (L<sup>A</sup>T<sub>E</sub>X)

```
%figure.tex
\documentclass[a4paper,12pt]{report}
\usepackage{fullpage}
\usepackage[ngerman]{babel}
\usepackage[applemac]{inputenc}
\usepackage[dvips]{graphicx}

\begin{document}
\listoffigures
\clearpage
\begin{figure}[t]
\begin{center}
\includegraphics[width=0.75\textwidth]{Grafiken/UniLogo.pdf}
\caption{Logo der Universität Ulm}
\label{fig::bsp1}
\end{center}
\end{figure}
\begin{figure}[t]
\begin{center}
\includegraphics[width=0.4\textwidth]{Grafiken/UniLogo.pdf}
\includegraphics[width=0.4\textwidth]{Grafiken/TeXLogo.pdf}
\caption{Links: Logo der Universität Ulm,
rechts: \TeX-\LaTeX-Logo}
\label{fig::bsp2}
\end{center}
\end{figure}
\end{document}
```

## Ausgabe-Datei (PDF)



ulm university universität

uulm

Abbildung: Logo der Universität Ulm



ulm university universität

uulm

TeX-LaTeX

Abbildung: Links: Logo der Universität Ulm, rechts: T<sub>E</sub>X-L<sub>A</sub>T<sub>E</sub>X-Logo

## Wissenschaftliches Arbeiten

- In offiziellen mathematischen Dokumenten muss der Autor Quellen angeben
  - im Literaturverzeichnis am Ende
    - vollständige Liste aller verwendeter Hilfen
  - im Fliesstext genaue Angaben
    - woher Ergebnisse, Ideen oder Beweise übernommen wurden
    - ob Teile wörtlich übernommen wurden
- Eigenleistung des Autors muss klar werden
  - z.B. einheitliche Darstellung eines Stoffs aus mehreren Quellen
    - genaue Angabe: Was stammt woher?
  - z.B. zusammenfassende Darstellung eines Stoffs
  - z.B. eigene Beweisidee, aber bekanntes Resultat
  - eigenes Resultat & eigener Beweis
- Im Extremfall: Vorwurf des Plagiats
  - juristisches Nachspiel (z.B. Exmatrikulation!)
  - Aberkennung akademischer Titel

## Literatursuche

- <http://www.uni-ulm.de/einrichtungen/kiz/bibliothek/literatursuche.html>
  - Bibliothekskatalog (Bücher und Zeitschriften der Uni Ulm)
- <http://rzblx1.uni-regensburg.de/ezeit/>
  - elektronische Zeitschriftenbibliothek mit Links zu Online-Journals (inkl. Ampel-Darstellung)
- <http://www.zentralblatt-math.org/zmath/de>
  - bibliographische Datenbank math. Veröffentlichungen
  - eingeschränkter Zugang innerhalb der Uni-Ulm
- <http://www.ams.org/mathscient>
  - bibliographische Datenbank math. Veröffentlichungen
  - Abkürzungsverzeichnis für Zeitschriften
  - freier Zugang innerhalb der Uni Ulm

## Literaturverzeichnis anlegen

- thebibliography-Umgebung:
  - startet mit `\begin{thebibliography}{string}`
    - `string` gibt nur maximale Länge von Markern an
  - Einträge mittels `\bibitem[marker]{label}`
    - `label`: definiert Label zum zitieren
    - `marker`: optional, gibt Kennung für Eintrag
    - falls `marker` fehlt, wird Nummer zugewiesen
- Zitieren im Text mittels
  - `\cite[string]{referenz}`
    - `referenz = label` vom `\bibitem`
    - `string`: optional, wird zusätzlich ausgegeben, z.B. expliziter Verweis auf einen Satz
  - Listen `\cite{ref1, ref2, ...}` sind erlaubt

## Grundsätzliches

- Einträge im Literaturverzeichnis einheitlich!
  - alle Vornamen abkürzen oder ausschreiben
  - gleiches Layout für alle Einträge
  - insb. einheitliche Groß-Kleinschreibung
  - am Ende jedes Eintrags ein Punkt oder nicht
- gewisse Sortierung
  - alphabetisch nach Erstautor (Autorenreihenfolge nicht ändern!)
  - chronologisch nach Veröffentlichungsjahr
  - chronologisch nach Reihenfolge des Zitierens
- korrekte Abkürzungen bei Zeitschriften (<http://www.ams.org/mathscient>)

## Beispiel zum Literaturverzeichnis

```
% literatur.tex
\documentclass[a4paper ,12pt]{report}
\usepackage{fullpage}
\usepackage[utf8]{inputenc}
\usepackage[ngerman]{babel}
\usepackage{soul}

\begin{document}

\addcontentsline{toc}{chapter}{Literaturverzeichnis}
\begin{thebibliography}{99}
\bibitem{Alberty:1999} \caps{J. Alberty, C. Carstensen, S. A. Funken}:
\emph{Remarks around 50 lines of Matlab: short finite element implementation},
Numerical Algorithms 20, 1999.

\bibitem{Alt:1980} \caps{H.W. Alt}: \emph{Lineare Funktionalanalysis},
Springer, Berlin, 1980.

\bibitem{Alt:2003}\caps{H.W. Alt}:
\emph{Partielle Differentialgleichungen, Vorlesungsskript WS 2002/03 - SoSe 2003},
Institut für Angewandte Mathematik, Universität Bonn.

\bibitem{Axelsson:1984}\caps{O. Axelsson, V. A. Barker}:
\emph{Finite Element Solution of Boundary Value Problems}, Academic Press, Orlando, 1984.

\bibitem{Baumgaertel:1985}\caps{H. Baumgaertel}:
\emph{Analytic Perturbation Theory for Matrices and Operators}, Birkhäuser, Boston, 1985.
\end{thebibliography}

\end{document}
```