

Verfassen wissenschaftlicher Texte mit L^AT_EX

PeP et al. Toolbox Workshop



PEP ET AL. E.V.
PHYSIKSTUDIERENDE UND
EHMALIGE PHYSIKSTUDIERENDE
DER TU DORTMUND

2015

Ergebnisse der Umfrage

Einführung

Grundlagen

Text erstellen

Error

Aufzählungen

Struktur

Formelsatz

Mathe-Umgebungen

Zahlen und Einheiten

Chemische Formeln

Fortgeschrittener Formelsatz

Gleitumgebungen

Tabellen

Fußnoten

Literaturverzeichnis

Fortgeschritten

Ein bisschen Typografie

`\texorpdfstring`

Links

Debug

Makros

Breites

Mathe: Expert

Makefiles

`latexmk`

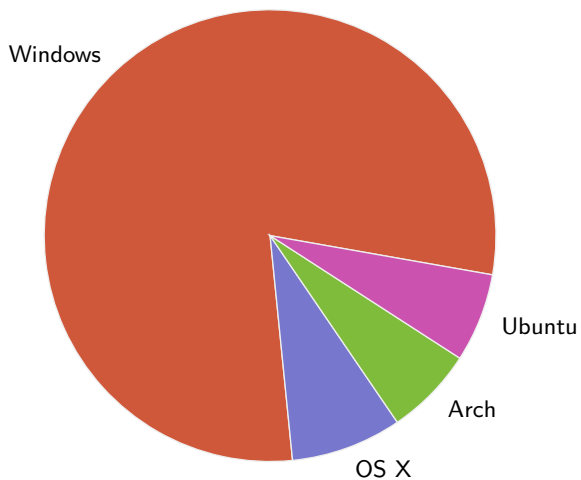
$\text{T}_{\text{E}}\text{X}$ in `matplotlib`

Präsentationen mit $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$: `beamer`

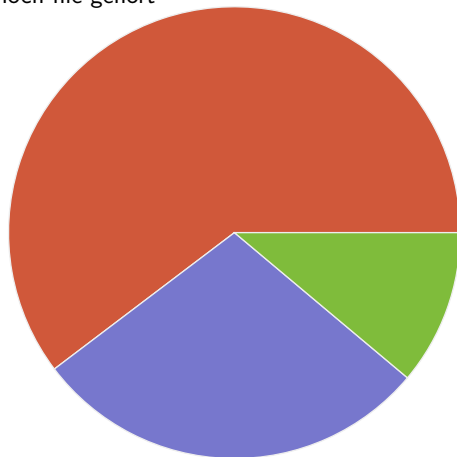
Zeichnen mit `Tikz`

Ausblick

Ergebnisse der Umfrage



noch nie gehört



TeXpert

mal probiert

Einführung

- *Programmiersprache* zum Setzen von Text
- Markup \Rightarrow kein **What-You-See-Is-What-You-Get**
- L^AT_EX-Code \rightarrow Compiler \rightarrow Ausgabedokument (meist PDF)
- Open-Source, große Erweiterungsmöglichkeit (Pakete)
- Standard-Werkzeug in der Wissenschaft

- Hervorragender Text- und Formelsatz
- Automatisierte Erstellung von Inhalts- und Literaturverzeichnis
- T_EX-Dateien sind reine Text-Dateien
 - ⇒ Gut für Versionskontrolle geeignet
- Sehr gute Vorlagen für wissenschaftliches Arbeiten

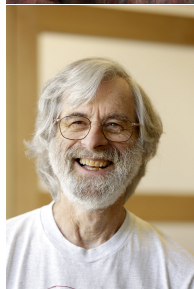
- Ausgezeichnete Dokumentation
- Erweiterbar durch zahlreiche und mächtige Pakete
- Auf allen geläufigen Betriebssystemen verfügbar
- Ausgabe direkt als PDF mit Hyperlinks

$\text{T}_{\text{E}}\text{X}$:

- Geschrieben von Donald E. Knuth 1978, um sein Buch „The Art of Computer Programming“ zu setzen
- Auf Aussprache achten!
- Version (2014): 3.14159265 $\rightarrow \pi$
- Viele Erweiterungen: $\epsilon\text{-T}_{\text{E}}\text{X}$, $\text{pdfT}_{\text{E}}\text{X}$, $\text{X}_{\text{E}}\text{T}_{\text{E}}\text{X}$, $\text{LuaT}_{\text{E}}\text{X}$

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$:

- Geschrieben von Leslie Lamport 1984
- Version (1994): $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$
- $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 3$ seit Anfang der Neunziger in Arbeit...



- In \LaTeX gibt es immer viele Möglichkeiten, ein Ziel zu erreichen
- Wir zeigen einen modernen Ansatz
- Wir erklären, warum wir diesen Ansatz gewählt haben
- Weitere Ansätze werden an manchen Stellen kurz erwähnt

$\text{T}_{\text{E}}\text{X}$ -Engine Implementierung von $\text{T}_{\text{E}}\text{X}$, wird als Programm ausgeführt

$\text{T}_{\text{E}}\text{X}$ -Format Paket, welches standardmäßig geladen wird, z.B. $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

Eine Kombination davon ist oft ein neues Programm.

Beispiel: $\text{dvi} \text{ lua} \text{ latex} = \text{Lua} \text{ T}_{\text{E}}\text{X} + \text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} + \text{DVI-Output (statt PDF)}$

Grundlagen

Diese drei Zeilen braucht jedes \LaTeX -Dokument:

Code

```
\documentclass[optionen]{klasse}
% .
% Präambel
% .
% .
\begin{document}
% Inhalt des Dokuments
\end{document}
```

\backslash documentclass

Vorlage wählen, mit Optionen anpassen.

Präambel

Globale Optionen und zusätzliche Pakete.

document-Umgebung

Inhalt des Dokuments.

Code

```
\documentclass{scrartcl}  
\begin{document}  
  Hallo Welt!  
\end{document}
```

Ergebnis

Hallo Welt!

L^AT_EX-Befehle beginnen stets mit einem \ (Backslash).

Obligatorische Argumente stehen in { }, optionale Argumente stehen in [].

Syntax

```
\befehl[optional]{obligatorisch}  
\befehl*[optional]{obligatorisch}
```

* ruft häufig eine Alternativform des Befehls auf.

Code

```
\documentclass[paper=a4]{scrartcl}  
  
\tableofcontents  
  
\frac{1}{2}  
  
% Kommentar
```

Erklärung

Dokumentenklasse scrartcl,
Papierformat DIN A4
Keine Argumente
Zwei oder mehr Pflichtargumente
%-Zeichen für Kommentare

- Einstellungen für Bereich des Dokuments
- Extrem vielseitig
- Können ggfs. auch Optionen übergeben bekommen
- Oft auch Alternativform mit *

Syntax

```
\begin{Umgebung}[optional]{obligatorisch}  
% .  
\end{Umgebung}
```

Beispiel

```
\begin{flushright}  
% .  
\end{flushright}
```

- Können weitere Umgebungen enthalten
- Diese müssen aber in der Umgebung wieder geschlossen werden

Geht:

```
\begin{document}  
  \begin{flushright}  
    % .  
  \end{flushright}  
\end{document}
```

Geht nicht:

```
\begin{itemize}  
  \begin{enumerate}  
    % .  
  \end{itemize}  
\end{enumerate}
```

Die hier aufgezählten Pakete sollten immer geladen werden, da sie wesentliche Funktionen bieten und wichtige Einstellungen vornehmen.

Paket	Funktion
<code>\usepackage[aux]{rerunfilecheck}</code>	Warnung, falls nochmal kompiliert werden muss.
<code>\usepackage{polyglossia}</code> <code>\setmainlanguage{german}</code> <code>\usepackage{fontspec}</code>	Deutsche Spracheinstellungen. Für Fonteinstellungen
% mehr Pakete hier	
<code>\usepackage[unicode]{hyperref}</code> <code>\usepackage{bookmark}</code>	Für Hyperlinks (z.B. Inhaltsverzeichnis → Kapitel). Erweiterte Bookmarks im PDF.

Die Reihenfolge ist manchmal wichtig, z.B. damit Pakete die Spracheinstellung kennen.

- scrartcl, scrreprt und scrbook
- Sehr gute Vorlagen
- Schnell global mit Klassenoptionen anpassbar

Fürs Praktikum empfohlenene Klasse

```
\documentclass[...]{scrartcl}
```

Standardeinstellung sind die Latin-Modern-Fonts.

Latin Modern

```
\usepackage{fontspec}
```

Alternativ: Tex Gyre

```
\usepackage{fontspec}  
\setmainfont{Tex Gyre Termes}  
\setsansfont{Tex Gyre Heros}  
\setmonofont{Tex Gyre Cursor}
```

- Jede System-Schriftart kann genutzt
- Das ist i.A. nicht sinnvoll: **Hallo Welt in Comic Sans!**
- Schriften müssen zueinander passen
- Schriften müssen alle benötigten Sonderzeichen enthalten
- Bei Änderung auch Mathefont anpassen → später

```
\documentclass{scrartcl}

\usepackage[aux]{rerunfilecheck}
\usepackage{polyglossia}
\setmainlanguage{german}


\usepackage{fontspec}
% mehr Pakete hier


\usepackage[unicode]{hyperref}
\usepackage{bookmark}
% Einstellungen hier, z.B. Fonts


\begin{document}
  % Text hier
\end{document}
```

Das Ausgabedokument erstellen

Es gibt verschiedene \LaTeX -Compiler, die verschiedene Ausgabeformate erzeugen können. Der modernste Compiler, der PDF-Dateien erstellt, ist **lua \LaTeX** .

\LaTeX -Dokument kompilieren

Terminal öffnen:

```
lua $\text{\LaTeX}$  MeinDokument.tex
```

Vorsicht!

- Es muss fast immer mindestens zweimal kompiliert werden.
- Es werden diverse Hilfs- und Logdateien erzeugt.
- Die Input-Dokumente müssen UTF-8 codiert sein.

L^AT_EX und (fast) alle Pakete sind hervorragend dokumentiert. Die Dokumentation wird automatisch mitinstalliert.

Dokumentation zu einem Paket

```
texdoc paket
```

Dabei ist *paket* ein Suchstring.

Nach Dokumentation suchen

```
texdoc -l name
```

Es ist wichtig zu lernen, Dokumentationen zu lesen. Probiert es an den oben genannten Paketen aus.

Alternativ kann man das Paket bei Google suchen, dann findet man auch die Dokumentation auf CTAN.

Text erstellen

Beispiel

```
% Präambel
```

```
\begin{document}
```

```
Hallo, Welt!
```

Dies ist ein dummer Beispieltext.

Er soll zeigen, dass `\LaTeX` sich nicht um
Zeilenumbrüche im Code oder zuviele
Leerzeichen kümmert.

Ein Absatz wird mit einer leeren Code-Zeile
markiert.

```
\end{document}
```

- Höchstens ein Satz pro Code-Zeile
- Absätze werden durch eine Leerzeile markiert
- Im Fließtext sollten keine Umbrüche mit `\\` erzwungen werden

Sonderzeichen

Viele Sonderzeichen sind \LaTeX -Steuerzeichen. Damit diese im Text genutzt werden können, muss meist ein `\` vorangestellt oder ein Befehl genutzt werden.

Code

```
\# \$ \% \& \_ \{ \}  
\textbackslash \textasciicircum \textasciitilde
```

Ergebnis

\$ % & _ { }
\\ ^ ~

Änderungen der Schrifteigenschaften sind mit diesen Befehlen möglich:

Code

```
\textit{kursiv} \emph{kursiv}  
\textbf{fett}  
\textbf{\textit{fett-kursiv}}  
\textrm{Serifen-Schrift}  
\texttt{Mono-Schrift}  
\textsf{Sans-Serif-Schrift}  
\textsc{Kapitälchen}
```

Ergebnis

kursiv kursiv
fett
fett-kursiv
Serifen-Schrift
Mono-Schrift
Sans-Serif-Schrift
KAPITÄLCHEN

Diese Befehle sollten sehr selten benutzt werden, semantischer Markup ist besser.

Gelten immer für den aktuellen Block, z. B. in einer Umgebung oder zwischen { }

Code

```
{\tiny tiny}  
{\small small}  
{\normalsize normal}  
{\large large}  
{\huge huge}
```

Ergebnis

tiny small normal large huge

Alle Größen

```
\tiny, \scriptsize, \footnotesize, \small, \normalsize, \large,  
\Large, \LARGE, \huge, \Huge
```

Auch diese Befehle sollten nur über semantischen Markup benutzt werden.

Code

```
\input{header.tex}
\begin{document}
  \input{Teil1.tex}
  \input{Teil2.tex}
  % .
\end{document}
```

- Verschachtelung möglich
- Zur Aufteilung größerer Dokumente (z.B. diese Präsentation)
- Für häufig wiederverwendeten Code (Header, Erläuterungen zu Fehlerrechnung, ...)
- Für per Skript erzeugte Tabelleninhalte

Die richtigen Anführungszeichen, wo die Satzzeichen hingehören und vieles mehr hängt von der Sprache ab. So macht man es richtig:

Benötigte Pakete

```
\usepackage[autostyle]{csquotes} % nach polyglossia  
\setotherlanguages{english, french} % andere Sprachen laden
```

Code

```
foo \enquote{bar} baz  
\enquote{foo \enquote{bar} baz}  
\textenglish{\enquote{foo}}  
\textfrench{\enquote{foo}}  
\textcquote{root}{foo}
```

Ergebnis

```
foo „bar“ baz  
„foo ,bar‘ baz“  
“foo”  
« foo »  
„foo“ [1]
```


Error

- Alles kaputt. Was nun?
- Fehlermeldungen anfangs (und teils auch später) etwas kryptisch.

Code

```
Ich begrüße euch mit einem \enqote{Hallo Welt}
```

```
(/usr/local/texlive/2015/texmf-dist/tex/latex/latexconfig/epstopdf-sys  
.cfg))
```

```
! Undefined control sequence.
```

```
l.8 Ich begrüße euch mit einem \enqote  
                                {Hallo Welt}!
```

```
462 words of node memory still in use:
```

- Alles kaputt. Was nun?
- Fehlermeldungen anfangs (und teils auch später) etwas kryptisch.

Code

```
Ich begrüße euch mit einem \enqote{Hallo Welt}
```

```
(/usr/local/texlive/2015/texmf-dist/tex/latex/latexconfig/epstopdf-sys
.cfg))
```

```
! Undefined control sequence.
```

```
l.8 Ich begrüße euch mit einem \enqote
                                {Hallo Welt}!
```

```
462 words of node memory still in use:
```

⇒ Vertippt (es fehlt ein u in \enquote)

- Angegebene Zeile und vorherige Zeilen kontrollieren
- Teile des Codes auskommentieren um Ort des Fehlers einzugrenzen
- Google → tex.stackexchange.com

Aufzählungen

- \LaTeX bietet drei Umgebungen für Aufzählungen
- Standardeinstellungen gut, Änderungen mit Paket `enumitem`
- Verschachteln für Unterpunkte
- Unnummerierte Listen: `itemize`

Code

```
\begin{itemize}
  \item Punkt 1
  \item Punkt 2
    \begin{itemize}
      \item Unterpunkt 1
      \item Unterpunkt 2
    \end{itemize}
  \item[→] Punkt 3
\end{itemize}
```

Ergebnis

- Punkt 1
 - Punkt 2
 - Unterpunkt 1
 - Unterpunkt 2
- Punkt 3

Für nummerierte Listen wird `enumerate` genutzt.

Code

```
\begin{enumerate}  
  \item Punkt 1  
  \item Punkt 2  
    \begin{enumerate}  
      \item Unterpunkt 1  
      \item Unterpunkt 2  
    \end{enumerate}  
  \item Punkt 3  
\end{enumerate}
```

Ergebnis

1. Punkt 1
2. Punkt 2
 - a) Unterpunkt 1
 - b) Unterpunkt 2
3. Punkt 3

Zur Beschreibung von Stichwörtern wird `description` benutzt, dabei wird das Stichwort `\item` als optionales Argument übergeben.

Code

```
\begin{description}  
  \item[\LaTeX] gut  
  \item[Word] böse  
\end{description}
```

Ergebnis

LaTeX gut
Word böse

Struktur

L^AT_EX erstellt automatisch eine Titelseite aus den Metadaten.

Mit der Klassenoption `titlepage=firstiscover` wird diese als eigene Seite gesetzt.

Neue Klassenoption

```
\documentclass[... , titlepage=firstiscover, ...]{scrartcl}
```

Empfehlung fürs Praktikum:

```
\title{101 Titel des Versuchs}  
% Mehrere Autoren mit \and:  
\author{Max Mustermann \and Maria Musterfrau}  
\date{Durchführung: 26.09.2014, Abgabe: 29.09.2014}
```

Titelseite generieren

```
\maketitle
```

L^AT_EX bietet Befehle zum erstellen von Gliederungsebenen. Diese werden automatisch nummeriert und in entsprechend größerer und fatter Schrift gesetzt.

Gliederungsebenen für scrartcl

```
\section{Überschrift}  
\subsection{Überschrift}  
\subsubsection{Überschrift}  
\paragraph{Überschrift}    % wird nicht nummeriert  
\subparagraph{Überschrift} % wird nicht nummeriert
```

Höhere Gliederungsebenen für scrreprt und scrbook

```
\part{Überschrift}  
\chapter{Überschrift}  
\section{Überschrift}
```

Aus den Gliederungselementen kann automatisch das Inhaltsverzeichnis erzeugt werden.

Inhaltsverzeichnis generieren

```
\tableofcontents
```

```
\newpage
```

Formelsatz

```
\usepackage{amsmath}    % unverzichtbare Mathe-Befehle  
\usepackage{amssymb}    % viele Mathe-Symbole  
\usepackage{mathtools}  % Erweiterungen für amsmath
```

```
\usepackage{amsmath}    % unverzichtbare Mathe-Befehle  
\usepackage{amssymb}    % viele Mathe-Symbole  
\usepackage{mathtools}  % Erweiterungen für amsmath
```

```
\usepackage{fontspec} % nach amssymb
```

```
\usepackage[
```

```
]{unicode-math}          % "Does exactly what it says on the tin."
```

```
\usepackage{amsmath} % unverzichtbare Mathe-Befehle
```

```
\usepackage{amssymb} % viele Mathe-Symbole
```

```
\usepackage{mathtools} % Erweiterungen für amsmath
```

```
\usepackage{fontspec} % nach amssymb
```

```
\usepackage[  
  math-style=ISO,      % \  
  bold-style=ISO,      % |  
  sans-style=italic,   % | ISO-Standard folgen  
  nabla=upright,       % |  
  partial=upright,     % /  
{unicode-math} % "Does exactly what it says on the tin."
```



```
\usepackage{amsmath}    % unverzichtbare Mathe-Befehle  
\usepackage{amssymb}    % viele Mathe-Symbole  
\usepackage{mathtools} % Erweiterungen für amsmath
```

```
\usepackage{fontspec} % nach amssymb
```

```
\usepackage[  
  math-style=ISO,      % \  
  bold-style=ISO,      % |  
  sans-style=italic,   % | ISO-Standard folgen  
  nabla=upright,       % |  
  partial=upright,     % /  
{unicode-math}        % "Does exactly what it says on the tin."
```

```
\setmathfont{Latin Modern Math}  
% \setmathfont{Tex Gyre Pagella Math} % alternativ
```

$\$ \dots \$$ -Umgebung

Aktiviert den Mathematikmodus im Fließtext.

$\text{T}_{\text{E}}\text{X}$ sorgt für gute Abstände

$x = 5, y = 3$

Ergebnis

$x = 5, y = 3$

Satzzeichen u. Bindestriche gehören nicht in $\$ \dots \$$

Dies ist eine Variable: x .

Liste von Variablen x, y, z .

y -Achse, x - y -Ebene

Ergebnis

Dies ist eine Variable: x .

Liste von Variablen x, y, z .

y -Achse, x - y -Ebene

Vorsicht bei der Höhe von Formeln im Text

Text ohne eine Bedeutung.

Mit einer Formel:

$\frac{1}{1 - \frac{1}{1 - x}}$

Text ohne eine Bedeutung.

Ergebnis

Text ohne eine Bedeutung.

Mit einer Formel: $\frac{1}{1 - \frac{1}{1 - x}}$

Text ohne eine Bedeutung.

Code

```
\epsilon \theta \kappa \pi \rho \sigma \phi  
\varepsilon \vartheta \kappa \varpi \varrho \varsigma \varphi  
  \varsigma \varphi  
\Alpha \Beta \Gamma  
\hbar \imath \jmath \ell  
\partial \nabla \square \Delta  
\infty \diameter
```

Ergebnis

$\epsilon \theta \kappa \pi \rho \sigma \phi$
 $\varepsilon \vartheta \kappa \varpi \varrho \varsigma \varphi$
 $A B \Gamma$
 $\hbar \imath \jmath \ell$
 $\partial \nabla \square \Delta$
 $\infty \varnothing$

Operatoren und Relationen

Code

```
+ - / \cdot \times  
\pm \mp  
< > \leq \geq  
= \simeq \equiv \cong  
\approx \propto \sim  
\coloneq \eqcolon  
\to \iff \implies  
\mapsto \leadsto  
\forall \exists \in \subset
```

Ergebnis

$+ - / \cdot \times$
 $\pm \mp$
 $< > \leq \geq$
 $= \simeq \equiv \cong$
 $\approx \propto \sim$
 $:= =:$
 $\rightarrow \iff \implies$
 $\mapsto \leadsto$
 $\forall \exists \in \subset$

Negierte Variante mit n bzw. not

```
\neq \nsimeq \nexists \nni \notin
```

Ergebnis

$\neq \nsimeq \nexists \nni \notin$

Häufig möchte man etwas über eine Relation schreiben:

```
\stackrel{!}{=} \stackrel{\text{def}}{=}
```

Ergebnis

$\stackrel{!}{=} \stackrel{\text{def}}{=}$

Indizes / Exponenten

Code

`x^2` `x_2` `x^2`

Ergebnis

x^2 x_2 x^2

Lange o. doppelte Indizes/Exponenten

`x^10` `x^{10}`
`x^{2^2}` `x^{2^2}`
`x_{\sqrt{3}[2]}` `x_{\sqrt{3}[2]}`

Ergebnis

x^{10} x^{10}
error x^{2^2}
error $x_{\sqrt[3]{2}}$

Text in Indizes

falsch: `x_{min}`, richtig: `x_{\text{min}}`

Ergebnis

falsch: x_{min} , richtig: x_{\min}

Striche / linksseitig

`x'` `x^{'}` `x''` `x'^2`
`{ }^2` `x`

Ergebnis

x' x' x'' x'^2
 2x

Nur wenige Befehle können ohne { } im Index stehen.

Code

```
\bar{x}  
\hat{x}  
\tilde{x}  
\vec{x}  
\mathring{x}  
\dot{x} \ddot{x} \dddot{x} \ddddot{x}  
\underline{xy} \overline{xy}
```

Ergebnis

\bar{x}
 \hat{x}
 \tilde{x}
 \vec{x}
 \mathring{x}
 \dot{x} \ddot{x} \dddot{x} \ddddot{x}
 $\underline{x y}$ \overline{xy}

Auf Position des Akzents achten:

```
\hat{x}_\text{min}  
\hat{x}_\text{min}
```

Ergebnis

\hat{x}_{\min}
 \hat{x}_{\min}

Code

```
x \sin y
x \sin(y)
\cos \tan \exp \ln \log_{10}(x)

\lim_{x \to \infty} x^2
```

Ergebnis

$x \sin y$
 $x \sin(y)$
 $\cos \tan \exp \ln \log_{10}(x)$

$$\lim_{x \rightarrow \infty} x^2$$

Man kann auch eigene Funktionen definieren:

```
% direkt in der Matheumgebung:
\operatorname{xyz}_i(a)
\operatorname*{xyz}_i(a)

% in Präambel definieren
\DeclareMathOperator{\xyz}{xyz}
\DeclareMathOperator*{\Xyz}{Xyz}

% dann überall im Dokument nutzbar:
\xyz_i(a)
\Xyz_i(a)
```

Ergebnis

$\operatorname{xyz}_i(a)$
 $\operatorname*{xyz}_i(a)$

$\operatorname{xyz}_i(a)$
 $\operatorname{Xyz}_i(a)$

Code

```
\sum_{i=0}^{\infty} x_i
```

```
\prod_{x \neq 0}
```

```
\int_0^1 \iiint \oint
```

```
\int_0^1 f(x) \, , \, \symup{d}x
```

```
\int x \quad \int_0 x \quad \int^{} x \quad \int_0^{} x  
% LuaTeX Bug: immer obere Grenze angeben
```

Ergebnis

$$\sum_{i=0}^{\infty} x_i$$

$$\prod_{x \neq 0}$$

$$\int_0^1 \iiint \oint$$

$$\int_0^1 f(x) \, dx$$

$$\int x \quad \int_0 x \quad \int^{} x \quad \int_0^{} x$$

Auslassungspunkte

Auslassungspunkte sind sehr ... wichtig.

Code

```
a_1,      \dotsc , a_n  
a_1 +     \dotsb + a_n  
a_1       \dotsm   a_n  
\int^{} \dotsi \int^{} 
```

Ergebnis

a_1, \dots, a_n
 $a_1 + \dots + a_n$
 $a_1 \cdots a_n$
 $\int \cdots \int$

Für andere Fälle gibt es Befehle mit festen Positionen:

Code

```
x \ldots x  
x \cdots x  
  \vdots  
  \ddots  
  \adots
```

Ergebnis

$x \dots x$
 $x \cdots x$
 \vdots
 \ddots
 \adots

Im Text kann man einfach `\dots` benutzen.

Code

```
x \alpha \symup{x \alpha}
\symbf{x\alpha}
\symbfsf{x \alpha}
\sybbb{R N 1 0 x}
\symcal{I A O} \symbfcal{I A O}
\symfrac{A B c} \sybbffrak{A B c}
```

Ergebnis

$x \alpha \ x \alpha$
 $\mathbf{x \alpha}$
 $\mathbf{x \alpha}$
 $\mathbb{R} \mathbb{N} 1 0 x$
 $\mathcal{I} \mathcal{A} \mathcal{O} \ \mathbf{\mathcal{I} \mathcal{A} \mathcal{O}}$
 $\mathfrak{A} \mathfrak{B} c \ \mathbf{\mathfrak{A} \mathfrak{B} c}$

Für mehrbuchstabige Bezeichnungen gibt es andere Befehle:

Code

```
Re \mathit{Re}
diff \quad \mathit{diff}
\mathbf{NP} \subseteq \mathbf{PSPACE}
```

Ergebnis

$Re \ Re$
 $diff \ diff$
 $\mathbf{NP} \subseteq \mathbf{PSPACE}$

Manchmal muss man manuell eingreifen, um das Spacing zu perfektionieren.

Code

```
% Kein Space  
\,  
\:  
\;  
\quad  
\quad\quad
```

Ergebnis

$\Rightarrow \times \Leftarrow$
 $\Rightarrow \Leftarrow$
 $\Rightarrow \Leftarrow$
 $\Rightarrow \Leftarrow$
 $\Rightarrow \quad \Leftarrow$
 $\Rightarrow \quad \quad \Leftarrow$

Negativer Space um zu viel Platz zu korrigieren:

Code

```
% kein Space  
\! % negativer \,
```

Ergebnis

$\Rightarrow \times \Leftarrow$
 $\Rightarrow \times \Leftarrow$

Code

```
^2            ^{\!\!\! 2}
```

Ergebnis

$\left(\frac{2^2}{2}\right)^2$ $\left(\frac{2^2}{2}\right)^2$

Code

```
(x) [x] \{x\} \langle x\rangle  
\lvert x\rvert \lVert x\rVert
```

Ergebnis

(x) $[x]$ $\{x\}$ $\langle x \rangle$
 $|x|$ $\|x\|$

Häufig braucht man größere Klammern

```
\bigl(x\bigr) \Bigl(x\Bigr) \biggl(x\biggr)  
 \Biggl(x\Biggr)
```

```
\bigl<x\bigr> \bigl|x\bigr| \bigl\|\bigr|x\bigr\|
```

Ergebnis

(x) (x) (x) (x)
 $\langle x \rangle$ $|x|$ $\|x\|$

Klammern: Automatische Größe

- Größe des Ausdrucks zwischen `\left` und `\right` bestimmt Größe der Klammern
- Ein `\left` muss in der gleichen Zeile wieder mit `\right` geschlossen werden
- `\left.` oder `\right.` falls nur eine Klammer gewünscht wird

Code

```
\left(\frac{1}{2} \right) \left(\frac{1}{2}\right.
```

```
\left\{x \,,\middle|\,, x<\frac{1}{2} \right\}
```

Ergebnis

$$\left(\frac{1}{2}\right) \quad \left(\frac{1}{2}\right.$$
$$\left\{x \mid x < \frac{1}{2}\right\}$$

Hat kein optimales Spacing:

```
\sin(x)
```

```
\sin\left(x\right)
```

```
\sin\!\left(x\right)
```

Ergebnis

$$\sin(x)$$
$$\sin(x)$$
$$\sin(x)$$

Praktischer Link:

<http://detexify.kirelabs.org/classify.html>

(Symbol malen und \LaTeX -Code angezeigt bekommen)

- Variablen/Größen werden kursiv gesetzt
- Mathematikmodus: alles erstmal Variable
- Alles, was keine Variable ist: aufrecht

→ Konstanten: e , i , π

`$\text{\textit{\textit{e}}}$` , `$\text{\textit{\textit{i}}}$` , `$\text{\textit{\textit{\pi}}}$`

→ Infinitesimales: dx

`$\text{\textit{d}}x$`

→ Indizes wie „min“ oder „max“

`x_{min}`

- dx wird durch kleines Leerzeichen (`\,`) vom Integranden abgetrennt
- `\,` auch zwischen verschiedenen dx_i

$$\int_0^1 \int_0^\pi \int_0^{2\pi} r^2 \sin(\vartheta) \, d\phi \, d\vartheta \, dr = \frac{4}{3}\pi$$

```
\int_0^1 \int_0^{\symup{\pi}} \int_0^{2 \symup{\pi}}  
r^2 \sin(\vartheta)  
\,, \symup{d}\varphi \,, \symup{d}\vartheta \,, \symup{d}r  
= \frac{4}{3} \symup{\pi}
```


Formelsatz

Mathe-Umgebungen

- amsmath stellt Mathe-Umgebungen für alles was man so braucht zur Verfügung
- Alle Gleichungen werden automatisch nummeriert
- * nach dem Umgebungsnamen sorgt für unnummerierte Gleichung
- Unnummerierte Gleichungen sollten selten sein

Code

```
Es gilt
\begin{equation}
  \nabla \cdot \vec{E}
    = \frac{\rho}{\varepsilon_0} .
  \label{eqn:maxwell1}
\end{equation}
Schon Gauß hatte das Durchflutungsgesetz
\eqref{eqn:maxwell1} aufgestellt.
```

Ergebnis

Es gilt

$$\nabla \cdot \vec{E} = \frac{\rho}{\varepsilon_0}. \quad (1)$$

Schon Gauß hatte das Durchflutungsgesetz (1) aufgestellt.

- Satzzeichen gehören in die equation-Umgebung!
- Gleichung ist grammatikalisch ein Substantiv
- Gleichungen müssen immer Teil eines vollständigen Satzes sein

- Für mehrere Gleichungen
- `\\` erzeugt neue Zeile
 - Kein `\\` nach der letzten Zeile!
- Jede Zeile bekommt eine Gleichungsnummer

Code

```
\begin{gather}  
  (a + b)^2 = a^2 + 2ab + b^2 \\  
  (a - b)^2 = a^2 - 2ab + b^2 \\  
  (a+b) \cdot (a-b) = a^2 - b^2  
\end{gather}
```

Ergebnis

$$(a + b)^2 = a^2 + 2ab + b^2 \quad (2)$$

$$(a - b)^2 = a^2 - 2ab + b^2 \quad (3)$$

$$(a + b) \cdot (a - b) = a^2 - b^2 \quad (4)$$

- Abhängig vom Fall ist die gather-Umgebung grammatikalisch ein Substantiv oder eine Aufzählung

- Für mehrere Gleichungen, die aneinander ausgerichtet werden
- `&` steuert Ausrichtung
- `\\` erzeugt neue Zeile
- Jede Zeile bekommt eine Gleichungsnummer

Code

```
\begin{align}
a &= 1 & b &= 2 \\
a \cdot b &= 5 & \frac{a}{b} &= 0.5
\end{align}
```

Ergebnis

$$\begin{array}{lll} a = 1 & b = 2 & (5) \\ a \cdot b = 2 & \frac{a}{b} = 0.5 & (6) \end{array}$$

- Um überlange Gleichungen auf zwei Zeilen aufzuteilen.
- Kommt in den anderen Umgebungen zum Einsatz
- & steuert Ausrichtung
- \\ erzeugt neue Zeile
- Gemeinsame Gleichungsnummer

Code

```
\begin{equation}
  \begin{split}
    (a+b)^3 = {} & a^3 + 3a^2b \\
    & + 3ab^2 + b^3
  \end{split}
\end{equation}
```

Ergebnis

$$(a + b)^3 = a^3 + 3a^2b + 3ab^2 + b^3 \quad (7)$$

Zahlen und Einheiten

- Einheiten werden aufrecht gesetzt
- Zwischen Zahl und Einheit steht ein kleines Leerzeichen `$5\,\mathrm{kg}$`
- siunitx stellt Befehle zur Verfügung, die das korrekte Setzen von Zahlen und Einheiten stark vereinfachen
- Funktioniert in Fließtext und Matheumgebung
- ⇒ Dieses Paket sollte **immer** und für **jede** Zahl mit oder ohne Einheit verwendet werden.

Benötigte Pakete

```
\usepackage[  
  locale=DE,  
  separate-uncertainty=true, % Immer Fehler mit ±  
  per-mode=symbol-or-fraction, % m/s im Text, sonst \frac  
  % alternativ:  
  % per-mode=reciprocal,      % m s-1  
  % output-decimal-marker=., % . statt , für Dezimalzahlen  
{siunitx}
```


Zahlen mit automatischen 3er-Gruppen

```
\num{1.23456}
\num{987654321}
```

Ergebnis

1,234 56
987 654 321

Einfaches Eingeben von 10er Potenzen

```
\num{6.022e23}
```

Ergebnis

6,022 · 10²³

Angabe von Fehlern

```
\num{1.54 +- 0.1}
\num{1.54(10)}
\num{1.54 \pm 0.1}
\num[separate-uncertainty=false]{1.54 +- 0.1}
\num{3.5(1)e6}
```

Ergebnis

1,54 ± 0,10
1,54 ± 0,10
1,54 ± 0,10
1,54(10)
(3,5 ± 0,1) · 10⁶

Einheiten

```
\si{\meter\per\second}
\si[per-mode=fraction]{\meter\per\second}
\si{\meter\per\second\squared}
\si[per-mode=reciprocal]{\gram\per\cubic\centi\meter}
\si{\kelvin\tothe{4}}
```

Ergebnis

m/s
 $\frac{\text{m}}{\text{s}}$
 m/s^2
 g cm^{-3}
 K^4

per-mode=symbol-or-fraction

```
\begin{equation}
\si{\kilo\gram\meter\per\second\squared}
\end{equation}
 $\si{\kilo\gram\meter\per\second\squared}$ 
```

Ergebnis

$\frac{\text{kg m}}{\text{s}^2}$ (8)
 kg m/s^2

Meter mal Sekunde oder Millisekunde?

```
\si{\milli\second}
\si{\meter\second}
\si[inter-unit-product=\cdot]{\meter\second}
```

Ergebnis

ms
 m s
 $\text{m} \cdot \text{s}$

\SI = Kombination aus \num und \si

```
\SI{5}{\percent}  
\SI{10}{\celsius}  
\SI{2.5(1)e6}{\kilo\gram\meter  
  \per\second\squared}
```

1. Argument Kann alles, was \num kann

2. Argument Kann alles, was \si kann

Winkel

```
\ang{5;55;59}
```

Ergebnis

5 %

10 °C

$(2,5 \pm 0,1) \cdot 10^6 \text{ kg m}^2/\text{s}^2$

Ergebnis

5°55′59″

Chemische Formeln

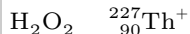
Benötigte Pakete

```
\usepackage[  
  version=4,  
  math-greek=default,  
  text-greek=default,  
{mhchem}
```

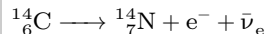
Code

```
$\ce{H2O2}$  
$\ce{^{227}_{90}Th+}$  
$\ce{c_{\ce{H2O}} = \SI{4184}{\joule\per  
  \kilo\gram\per\kelvin}}$  
$\ce{^{14}_6C -> ^{14}_7N + e-  
  + \bar{\symup{\nu}}_e}$  
$\ce{CO2 + C <=> 2CO}$
```

Ergebnis



$$c_{\text{H}_2\text{O}} = 4184 \text{ J}/(\text{kg K})$$



Fortgeschrittener Formelsatz

Matrizen (und damit Vektoren) werden fett geschrieben.

```
\sympbf{M}^{\top}      \sympbf{M}^{\star}
\sympbf{M}^{\dagger}   \sympbf{M}^{-1}   M_{12}
```

Ergebnis

$$\mathbf{M}^{\top} \quad \mathbf{M}^{\star} \\ \mathbf{M}^{\dagger} \quad \mathbf{M}^{-1} \quad M_{12}$$

Verschiedene Klammern

```
\begin{pmatrix} x & y \\ z & abc \end{pmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} \begin{Bmatrix} -1 \\ 2 \end{Bmatrix}
\end{pmatrix} \end{bmatrix} \end{Bmatrix}
```

Ergebnis

$$\begin{pmatrix} x & y \\ z & abc \end{pmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} \begin{Bmatrix} -1 \\ 2 \end{Bmatrix}$$

Unterschiedliche Ausrichtung mit *

```
\begin{pmatrix*}[l] 1 & -1 \\ -1 & 2 \end{pmatrix*} \begin{pmatrix*}[c] 1 & -1 \\ -1 & 2 \end{pmatrix*}
\end{pmatrix*} \end{pmatrix*}
\begin{pmatrix*}[r] 1 & -1 \\ -1 & 2 \end{pmatrix*}
\end{pmatrix*}
```

Ergebnis

$$\begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix} \quad \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix} \\ \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix}$$

Mehr Möglichkeiten in der Doku.

Unnötigen Leerraum loswerden

mathtools stellt zwei wichtige Befehle, um Leerraum zu eliminieren:

`\mathclap`

`\lim_{x\to\infty} f(x)`

`\lim_{\mathclap{x\to\infty}} f(x)`

`\sum_{i \in \{1, 2, 3, 4, 5\}} i`

`\sum_{\mathclap{i \in \{1, 2, 3, 4, 5\}}} i`

Ergebnis

$$\lim_{x \rightarrow \infty} f(x)$$

$$\lim_{x \rightarrow \infty} f(x)$$

$$\sum_{i \in \{1, 2, 3, 4, 5\}} i$$

$$\sum_{i \in \{1, 2, 3, 4, 5\}} i$$

`\cramped`

`2^{2^2} \cramped{2^{2^2}}`

Ergebnis

$$2^{2^2} 2^{2^2}$$

Befehle mit * aktivieren den Textmodus nach dem &.

Code

```
f(x) =  
\begin{cases}  
  x , & x \ge 0 \\  
  \int_0^1 x , & \text{sonst}  
\end{cases}  
  
\begin{drcases*}  
  x , & $x < 0$ \\  
  \int_0^1 x , & \text{sonst}  
\end{drcases*}  
\ne - \lvert x \rvert
```

Ergebnis

$$f(x) = \begin{cases} x, & x \geq 0 \\ \int_0^1 x, & \text{sonst} \end{cases}$$

$$\left. \begin{array}{ll} x, & x < 0 \\ \int_0^1 x, & \text{sonst} \end{array} \right\} \neq -|x|$$

`\intertext` erhält die Ausrichtung der `align`-Umgebung.

Code

```
Es gilt
\begin{align*}
f &= xyz, \\
\intertext{wobei dies ein langer
Erklärungstext ist, und dass}
g' &= \symup{e}^x \\
\shortintertext{von}
g &= \symup{e}^x
\end{align*}
gelöst wird.
```

Ergebnis

Es gilt

$$f = xyz,$$

wobei dies ein langer
Erklärungstext ist, und dass

$$g' = e^x$$

von

$$g = e^x$$

gelöst wird.

Code

```
f(x)= \underbrace{g(x)}
_{x + x^2 + x^3}
+ \overbrace{h(x)}
^{\mathclap{\text{mehr Erklärung}}}
+ 2
```

Ergebnis

$$f(x) = \underbrace{g(x)}_{x+x^2+x^3} + \overbrace{h(x)}^{\text{mehr Erklärung}} + 2$$

Code

```
\begin{align*}
f_{123} &= 2x + 3y + z \\
g &= \phantom{2}x + 3y + z \\
h &= x + 3y + z
\end{align*}
```

Ergebnis

$$\begin{aligned}
 f_{123} &= 2x + 3y + z \\
 g &= x + 3y + z \\
 h &= x + 3y + z
 \end{aligned}$$

Code

```
\begin{align*}
f &= \frac{1}{2}x \\
g &= \hphantom{\frac{1}{2}}x \\
\end{align*}
\begin{align*}
f &= \frac{1}{2}x \\
g &= \phantom{\frac{1}{2}}x
\end{align*}
```

Ergebnis

$$\begin{aligned}
 f &= \frac{1}{2}x & f &= \frac{1}{2}x \\
 g &= x & g &= x
 \end{aligned}$$

`\hphantom` wirkt nur horizontal und hat keine Höhe.

Code

```

\begin{align*}
f = {} & \& \left( \frac{1}{2} \right) \& \left. \vphantom{\frac{1}{2}} \right. + x \\
& \& \left. \right)^{2} \\
\end{align*}

```

Ergebnis

$$f = \left(\frac{1}{2} + x \right)^2$$

`\vphantom` wirkt nur vertikal und hat keine Breite.

Gleitumgebungen

- Zum setzen von Elementen, die nicht Fließtext sind
- Hauptsächlich Grafiken und Tabellen
- Position wird von $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ automatisch bestimmt
- Nicht auf früherer Seite als umgebender Text
- Bekommen meist `\caption` und `\label`

Benötigte Pakete

```
% Floats innerhalb einer Section halten  
\usepackage[section, below]{placeins}  
\usepackage[...]{caption} % Captions schöner machen
```

`\FloatBarrier` kann benutzt werden, um alle vorigen Floats zu setzen.

Benötigte Pakete

```
\usepackage{graphicx}  
\usepackage{grffile}
```

Code

```
\begin{figure}  
  \centering  
  \includegraphics[width=\textwidth]{  
    logos/pep.pdf}  
  \caption{Das Pep-Logo.}  
  \label{fig:peplogo}  
\end{figure}
```

Ergebnis



PEP ET AL. E.V.
PHYSIKSTUDIERENDE UND
EHMALIGE PHYSIKSTUDIERENDE
DER TU DORTMUND

Abbildung 1: Das PeP-Logo.

- Auch möglich: `height=...`, `scale=...`
- `\caption` endet immer mit einem Punkt.

Benötigte Pakete

```
\usepackage{subcaption}
```



(a) PeP-Logo.



(b) Das TU-Logo.

Abbildung 2: Zwei Logos, Abbildung b: das TU-Logo.

Code

```
\begin{figure}
  \centering
  \begin{subfigure}{0.48\textwidth}
    \centering
    \includegraphics[height=0.75cm]{logos/pep.pdf}
    \caption{PeP-Logo.}
    \label{fig:pep2}
  \end{subfigure}
  \begin{subfigure}{0.48\textwidth}
    \centering
    \includegraphics[height=0.75cm]{logos/tu.pdf}
    \caption{Das TU-Logo.}
    \label{fig:TU}
  \end{subfigure}
  \caption{Zwei Logos, Abbildung \subref{fig:TU}: Das TU-Logo.}
  \label{fig:logos}
\end{figure}
```

Code

```
\section{Messung mit Apparatur 2}  
\label{sec:apparatur2}  
% .  
\section{Auswertung}  
Wie in \ref{sec:apparatur2} beschrieben, ...
```

- Auch für Gleichungen, Grafiken, Tabellen
- Für Übersichtlichkeit sollten Labels den Typ der Referenz nennen:

Sections sec:

Gleichungen eqn:

Abbildungen fig:

Tabellen tab:

- Bei Gleichungen: `\eqref` statt `\ref` → setzt Klammern: (1)
- `\label` immer nach dem, worauf verwiesen wird

Code

In Abbildung `\ref{fig:logos}`
sehen Sie zwei Logos.

In Abbildung `\ref{fig:pep2}`
sehen Sie das PeP-Logo.

In Abbildung `\subref{fig:pep2}`
sehen Sie das PeP-Logo.

Ergebnis

In Abbildung 2 sehen Sie zwei Logos.

In Abbildung 2a sehen Sie das PeP-Logo.

In Abbildung a sehen Sie das PeP-Logo.

`\subref` nur in `\caption{...}` zu Subfigures sinnvoll.

Positionen der Gleitumgebungen

- \LaTeX hat 4 Regionen, in die es Float-Umgebungen platziert
 - h** here, zwischen Text
 - t** top, oben auf einer Seite
 - b** bottom, unten auf einer Seite
 - p** page, eigene Seite nur für Floats
- Standardmäßig nur t, b, p genutzt
- **Nicht** empfohlen: Änderung mit optionalem Argument an Umgebung
- Änderung des Standards mit dem Paket float

Benötigte Pakete

```
\usepackage{scrhack} % nach \documentclass

\usepackage{float}
\floatplacement{figure}{htbp}
\floatplacement{table}{htbp}
```

Tabellen

Benötigte Pakete

```
\usepackage{booktabs}
```

Neue Klassenoption

```
\documentclass[... , captions=tableheading, ...]{scrartcl}
```

Code

```
\begin{table}
  \centering
  \caption{Eine Tabelle mit Messdaten.}
  \label{tab:some_data}
  \begin{tabular}{c c c c c}
    \toprule
    $f$ & $l$ & \text{start} & $ & $l_1$ & $l_-$ \\
    \text{kor},1 & $ & $B_1$ & \\\
    \midrule
    100 & 1.14 & 3.51 & 0.00 & 4.30 & \\\
    300 & 1.27 & 2.42 & 0.13 & 41.14 & \\\
    500 & 1.21 & 1.70 & 0.25 & 168.73 & \\\
    \bottomrule
  \end{tabular}
\end{table}
```

- Äußere table-Umgebung behandelt Tabelle wie ein float
- Innere tabular-Umgebung für eigentlichen Tabelleninhalt
- l, c oder r geben Ausrichtung der einzelnen Spalten an
- \caption, \label oberhalb von tabular

Tabelle 1: Eine Tabelle mit Messdaten.

f	l_{start}	l_1	$l_{\text{kor},1}$	B_1
100	1.14	3.51	0.00	4.30
300	1.27	2.42	0.13	41.14
500	1.21	1.70	0.25	168.73

- Keine vertikalen Linien!
- Keine horizontalen Linien zwischen Daten!

Code

```

\begin{table}
  \centering
  \caption{Eine schöne Tabelle mit Messdaten.}
  \label{tab:some_data}
  \sisetup{table-format=1.2}
  \begin{tabular}{S[table-format=3.0] S S S[table-format=3.2]}
    \toprule
    {\$f\$} & {\$l_\text{start}\$} & {\$l_1\$} & {\$l_\text{kor},1\$} & {\$B_1\$} \\
    \midrule
    100 & 1.14 & 3.51 & 0.00 & 4.30 \\
    200 & 1.30 & 2.99 & 0.06 & 25.98 \\
    300 & 1.27 & 2.42 & 0.13 & 41.14 \\
    400 & 1.28 & 1.47 & 0.20 & 53.76 \\
    500 & 1.21 & 1.70 & 0.25 & 168.73 \\
    \bottomrule
  \end{tabular}
\end{table}

```

Tabelle 2: Eine schöne Tabelle mit Messdaten.

f	l_{start}	l_1	$l_{\text{kor},1}$	B_1
100	1,14	3,51	0,00	4,30
200	1,30	2,99	0,06	25,98
300	1,27	2,42	0,13	41,14
400	1,28	1,47	0,20	53,76
500	1,21	1,70	0,25	168,73

- s-Spalte eröffnet mehr Ausrichtungsmöglichkeiten mit `\sisetup` und `[...]`
- s-Spalte für Einheiten
- Standard: Ausrichtung an Dezimalkomma
- Spaltennamen durch `{ }` schützen

Gruppieren von mehreren Spalten

Kommandostruktur

```
\multicolumn{#Spalten}{Ausrichtung}{Inhalt}
```

Beispiel

```
\begin{table}
  \centering
  \caption{Messdaten für dubiose Elemente.}
  \sisetup{table-format=2.1}
  \begin{tabular}[S[table-format=3.1] S S S S]{S}
    \toprule
    & \multicolumn{2}{c}{Technetium} & \multicolumn{2}{c}{Molybdän} & \\
    & {\$ \lambda} & \:/\: & \si{nano\meter} & \\
    & {\$ \phi_1} & & {\$ \phi_2} & & {\$ \phi_1} & & {\$ \phi_2} & \\
    \midrule
    663.0 & 12.1 & 14.4 & 13.1 & 16.9 & \\
    670.0 & 10.9 & 12.9 & 11.8 & 15.7 & \\
    678.0 & 9.1 & 11.4 & 10.3 & 14.6 & \\
    684.0 & 8.2 & 10.2 & 9.5 & 13.5 & \\
    \bottomrule
  \end{tabular}
\end{table}
```

Tabelle 3: Messdaten für dubiose Elemente.

λ / nm	Technetium		Molybdän	
	ϕ_1	ϕ_2	ϕ_1	ϕ_2
663,0	12,1	14,4	13,1	16,9
670,0	10,9	12,9	11,8	15,7
678,0	9,1	11,4	10,3	14,6
684,0	8,2	10,2	9,5	13,5

→ Einheiten werden im Tabellenkopf herausdividiert.

Code

```
\begin{tabular}{  
  S[table-format=3.1]  
  @{$$}\pm{$$}  
  S[table-format=2.1]  
}  
 \toprule  
 \multicolumn{2}{c}{$x \text{ } \text{:/\text{:}} \text{ } \text{\si{\ohm}}$} \\\br/> \midrule  
 663.0 & 12.1 \\\br/> 670.0 & 10.9 \\\br/> 678.0 & 9.1 \\\br/> 684.0 & 8.2 \\\br/> \bottomrule  
\end{tabular}
```

Ergebnis

x / Ω	
663,0	$\pm 12,1$
670,0	$\pm 10,9$
678,0	$\pm 9,1$
684,0	$\pm 8,2$

@{...} ersetzt den Spaltenabstand durch ...

Fußnoten

Code

```
In diesem Versuch werden  
PMTs\footnote{Photo-Multiplier-  
Tubes}  
eingesetzt.
```

Ergebnis

In diesem Versuch werden PMTs¹
eingesetzt.

¹Photo-Multiplier-Tubes

→ Anpassung von Fußnoten mit dem Paket footmisc

Vorsicht bei Float-Umgebungen!

```
\begin{figure}  
  \includegraphics[height=0.5cm]{pep.pdf}  
  \caption{Bla\protect\footnotemark}  
\end{figure}  
\footnotetext{Unsinnige Caption.}
```

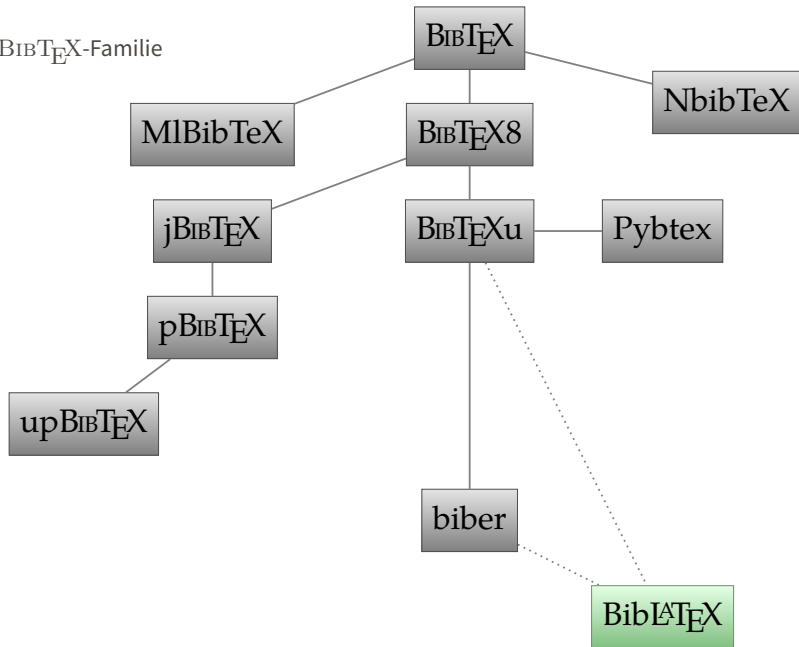
- `\footnotemark` an der Stelle wo die Fußnote sein soll
- In einer `\caption` muss dem `\footnotemark` ein `\protect` vorangestellt werden.
 - Aufpassen, wenn man eine `\listoffigures` hat
- `\footnotetext{...}` außerhalb der Umgebung für den Text der Fußnote
- Fußnoten in Abbildungen sollten vermieden werden.

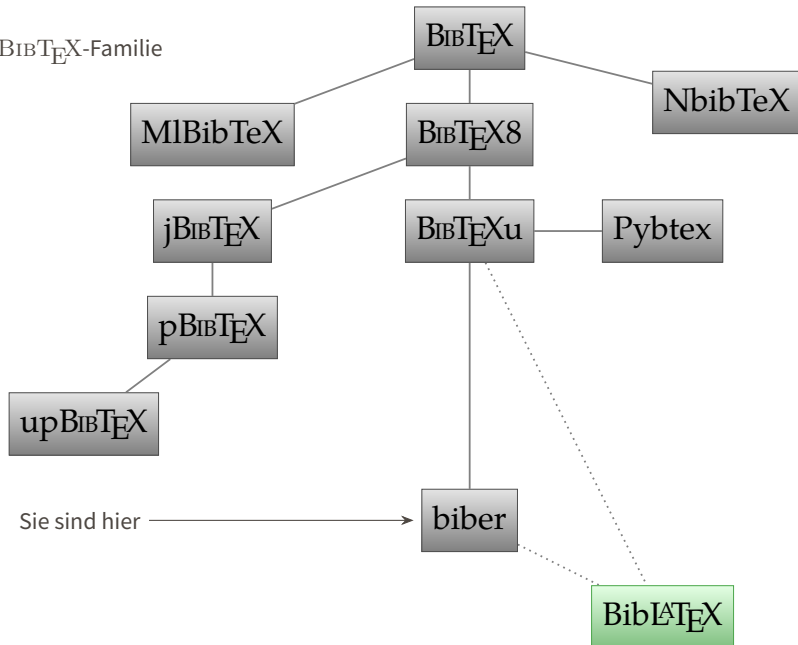
Literaturverzeichnis

- Wichtiger Teil vieler Dokumente, für wissenschaftliche Texte zwingend
- Bib \LaTeX und biber bieten eine sehr angenehme Arbeitsweise
- Auch für sehr große Referenzdatenbanken geeignet
- Es gibt viele unterschiedliche Stile
- Standardstil fürs Praktikum geeignet
- Referenzen in .bib-Dateien

Neue Klassenoption

```
\documentclass[..., bibliography=totoc, ...]{scrartcl}
```





- Unterstützt Unicode-Input
- Wird weiterentwickelt, zusammen mit Bib \LaTeX
- Sortiert richtig, nach regeln der jeweiligen Sprache
- Kann noch viele weitere Formate außer `.bib` lesen
- Unterstützt alle Funktionen von Bib \LaTeX

```
@manual{anleitung01,  
  author = "TU Dortmund", % alternativ {...} statt "... " möglich  
  title = "Versuchsanleitung zu Versuch Nr. 01 Lebensdauer der Myonen  
  ",  
  year = 2004,  
}
```

TU Dortmund. *Versuchsanleitung zu Versuch Nr. 01 Lebensdauer der Myonen*. 2004

```
@article{numpy,  
  author = "Oliphant, Travis E.",  
  title = "Python for Scientific Computing",  
  publisher = "IEEE",  
  year = "2007",  
  journal = "Computing in Science & Engineering",  
  volume = "9",  
  number = "3",  
  pages = "10--20",  
  url = "http://link.aip.org/link/?CSX/9/10/1",  
  version = "1.8.1",  
}
```

Travis E. Oliphant. „Python for Scientific Computing“. Version 1.8.1. In: *Computing in Science & Engineering* 9.3 (2007), S. 10–20. URL: <http://link.aip.org/link/?CSX/9/10/1>

```
@inproceedings{root,  
  author = "Brun, Rene and Rademakers, Fons",  
  booktitle = "AIHENP'96 Workshop, Lausanne",  
  url = "http://root.cern.ch/",  
  journal = "Nucl. Inst. \& Meth. in Phys. Res. A",  
  pages = "81--86",  
  title = "ROOT -- An Object Oriented Data Analysis Framework",  
  volume = 389,  
  year = 1996,  
  version = "5.34.18",  
}
```

Rene Brun und Fons Rademakers. „ROOT – An Object Oriented Data Analysis Framework“. In: *AIHENP'96 Workshop, Lausanne*. Bd. 389. 1996, S. 81–86. URL: <http://root.cern.ch/>


```
@online{splot,  
  author = "Pivk, Muriel and Le Diberder, Francois R.",  
  title = "sPlot: a statistical tool to unfold data distributions",  
  date = "2005-09-02",  
  eprinttype = "arXiv",  
  eprint = "physics/0402083v3",  
}
```

Muriel Pivk und Francois R. Le Diberder. *sPlot: a statistical tool to unfold data distributions*.
2. Sep. 2005. arXiv: physics/0402083v3

```
@online{wingate,  
  author = "Liu, Zhaofeng and Meinel, Stefan and Hart, Alistair and  
    Horgan, Ron R. and Müller, Eike H. and Wingate, Matthew",  
  title = "A lattice calculation of  $\text{\textcolor{blue}{\$}\symup{B} \rightarrow \text{\textcolor{red}{\$}\symup{K}^{(*)}}\text{\textcolor{blue}{\$}}$   
    form factors",  
  date = "2011-01-14",  
  eprinttype = "arXiv",  
  eprint = "1101.2726v1",  
  eprintclass = "hep-ph",  
}
```

Zhaofeng Liu u. a. *A lattice calculation of $B \rightarrow K^{(*)}$ form factors*. 14. Jan. 2011. arXiv:
1101.2726v1 [[hep-ph](#)]

Benötigte Pakete

```
\usepackage{biblatex} % nach polyglossia  
\addbibresource{lit.bib}
```

Zitieren

```
\cite{numpy}  
\cite[20]{numpy}  
\cite[1--3]{numpy}  
\cite{splot, root}
```

Ergebnis

```
[4]  
[4, S. 20]  
[4, S. 1–3]  
[5, 1]
```

Verzeichnis ausgeben

```
\nocite{wingate} % ins Verzeichnis, obwohl nicht explizit zitiert  
\nocite{*} % alles aus .bib ins Verzeichnis  
\printbibliography
```


???

Die Idee ist:

1. Bib_{La}T_EX erstellt eine Liste der .bib-Dateien und der benötigten Referenzen
→ .bcf-Datei
2. biber liest Anweisungen, liest .bib, sucht und sortiert Referenzen
→ .bbl-Datei
3. Bib_{La}T_EX liest .bbl, gibt Verzeichnis aus

Also:

Aufrufe mit Literaturverzeichnis

```
lualatex file.tex  
biber file.bcf  
lualatex file.tex
```

- [1] Rene Brun und Fons Rademakers. „ROOT – An Object Oriented Data Analysis Framework“. In: *AIHENP'96 Workshop, Lausanne*. Bd. 389. 1996, S. 81–86. URL: <http://root.cern.ch/>.
- [2] TU Dortmund. *Versuchsanleitung zu Versuch Nr. 01 Lebensdauer der Myonen*. 2004.
- [3] Zhaofeng Liu u. a. *A lattice calculation of $B \rightarrow K^{(*)}$ form factors*. 14. Jan. 2011. arXiv: 1101.2726v1 [[hep-ph](#)].
- [4] Travis E. Oliphant. „Python for Scientific Computing“. Version 1.8.1. In: *Computing in Science & Engineering* 9.3 (2007), S. 10–20. URL: <http://link.aip.org/link/?CSX/9/10/1>.
- [5] Muriel Pivk und Francois R. Le Diberder. *sPlot: a statistical tool to unfold data distributions*. 2. Sep. 2005. arXiv: [physics/0402083v3](#).

- Standardstil ist „numeric“
- Häufig genutzte Alternative: „alphabetic“
- Kombination aus Autorennamen und Jahr: z.B. [Oli07]
- Viele weitere Stile → Doku
- Setzen mit `style=...` als Option für `biblatex`

Code

```
\usepackage[style=alphabetic]{biblatex}
```


Fortgeschritten

Fortgeschritten

Ein bisschen Typografie

- Zur Erinnerung: Leerzeile im Code erzeugt neuen Absatz
- Zwei Möglichkeiten: Einzug der ersten Zeile oder vertikaler Abstand
- Standard ist Einzug
- halbzeiliger vertikaler Abstand mit:

Klassenoption

```
\documentclass[parskip=half, ...]{scrartcl}
```

- Ihr werdet den Effekt kaum sehen
- Das ist Absicht!
- Kleine Korrekturen, die das Schriftbild verbessern
- z. B. - etwas in den Rand hinein für homogenen Grauanteil

Benötigte Pakete

```
\usepackage{microtype}
```

Benötigte Pakete

```
\usepackage{xfrac}
```

- Problem: `\frac{1}{2}` zu hoch
- unschöne Alternative: $1/2$
- schön: `\sfrac{1}{2}`

Code

```
\sfrac{1}{2}  
\sfrac{$\symup{\pi}$}{2}
```

Ergebnis

 $1/2$ $\pi/2$

- Es gibt Leerzeichen an denen nicht umgebrochen werden soll
- Zwischen Titel und Name
- Bei Referenzen
- Zweiteilige Abkürzungen (aber ein kleines!)
- Bei Datumsangaben
- Zweiteilige Ortsnamen
- Zwischen Zahl und Einheit (→ `siunitx`)

Code

```
Prof.~Dr.~Dr.~Rhode  
Abbildung~\ref{fig:peplogo}  
z.~,B.  
2.~Oktober~2014  
St.~Helena
```

Ergebnis

Prof. Dr. Dr. Rhode
Abbildung 1
z. B.
2. Oktober 2014
St. Helena

Es gibt vier verschiedene Striche:

Code
- \$-\$ -- ---

Ergebnis
- -- --

- Bindestrich** → Bindestrich
 - zwischen Doppelnamen der selben Person
Levi-Civita-Symbol
- Halbgeviertstrich (en-dash)** → Gedankenstrich:
 - Text -- oh, Gedankenstriche -- Text
 - zwischen Namen von versch. Personen
Maxwell--Boltzmann-Verteilung
 - ist auch der Bis-Strich
1 bis 10 ist 1--10
- Geviertstrich (em-dash)** → nicht im Deutschen, englischer Gedankenstrich
text---oh, em-dashes---text

Trennung bei Strichen

Benötigte Pakete

```
\usepackage[shortcuts]{extdash} % nach hyperref, bookmark
```

Falls ein Wort Striche enthält, trennt \LaTeX ausschließlich an diesen.
So ermöglicht man mehr Trennung:

Trennbare Striche

```
\-/ \-- \---
```

Normal-Verteilung
Normal\-/Verteilung

Ergebnis

- - -
Normal-
Verteilung
Normal-
Vertei-
lung

So verhindert man die Trennung an den Strichen:

```
\=/ \== \===  
$x$ \=/Achse
```


- Manchmal kann $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ein Wort nicht richtig trennen
- Manche Fachwörter sollten nicht nach deutschen Regeln getrennt werden

Trennung für Wort vorgeben

```
% Präambel
\hyphenation{Dia-mag-ne-tis-mus hy-phen-ate hy-phen-a-tion}
% statt Di-a-mag-ne-tis-mus

hy\ -phen\ -ate % im Text
```

Fortgeschritten

`\texorpdfstring`

Motivation

- hyperref bezieht den Text für seine Bookmarks, z.B. aus `\section{ }`
- Bookmarks mögen meist keine $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Befehle
- hyperref fängt vieles ab, aber nicht alles
- `\section{$\alpha + 1$}`
ergibt Fehlermeldung:
Package hyperref Warning:
Token not allowed in a PDFDocEncoded string:
! Improper alphabetic constant.

Lösung

```
\texorpdfstring{LaTeX-Code}{Unicode-Äquivalent}
```

Code

```
\section{\texorpdfstring{$\alpha + 1$}{ $\alpha + 1$ }}
```

Fortgeschritten

Links

- Es ist sehr empfehlenswert, auf der Titelseite eure Mailadressen anzugeben!
- hyperref stellt den `\href{link}{text}` Befehl
- `\url{url} = \href{url}{url}`

Code

```
\href{www.google.de}{Google}  
\href{mailto:max@mustermann.de}{  
  max@mustermann.de}
```

Ergebnis

Google
max@mustermann.de

Autoren mit Mailadressen:

```
\author{  
  Max Mustermann\\  
  \texorpdfstring{\href{mailto:max@mustermann.de}{max@mustermann.de}  
    \and}{,}  
  Felix Mustermann\\  
  \texorpdfstring{\href{mailto:felix@mustermann.de}{felix@mustermann.  
    de}}{}}  
}
```

Fortgeschritten

Debug

```
fooooooooooooooooooooooo
```

Overfull `\hbox` (14.97614pt too wide) in paragraph at lines 10--10

Wo genau ist die problematische Stelle?

```
\setlength\overfullrule{5pt}
```

```
fooooooooooooooooooooooo■
```

Zeilen, die über den Rand ragen, werden markiert.

Entweder als Option für die Dokumentklasse

```
\documentclass[... , draft, ...]{...}
```

oder auch nur für ein Bild

```
\includegraphics[draft, height=2cm]{logos/pep.pdf}
```



logos/pep.pdf

Vorteile:

- Ränder des Bilds sind sichtbar
- Bild muss nicht existieren (Größe stimmt dann aber nicht)

Manchmal möchte man den Textbereich auf der Seite grafisch sehen.

Das geht mit

```
\usepackage{showframe}
```

Manchmal möchte man sicher gehen, dass Sachen ausgerichtet sind oder die richtige Größe haben.

Dabei hilft.

`usepackage lua-visual-debug`

Hier noch eine Gleichung.

$$\int_0^{\infty} e^{-x^2} dx = \frac{\sqrt{\pi}}{2} \quad (9)$$

Fortgeschritten

Makros

Nach 20 Mal `\symup{e}` oder `\symup{i}` schreiben hat man keine Lust mehr.

Code

```
% in Präambel
\usepackage{expl3}
\usepackage{xparse}

\ExplSyntaxOn

\NewDocumentCommand \I {}
{
  \symup{i}
}

\ExplSyntaxOff
```

Erklärung

experimental L^AT_EX3

bequeme Syntax für Definition von Befehlen

Befehl `\I` definieren, keine Argumente

Ergebnis von `\I`

Syntax wieder ausschalten, wichtig!

`\ExplSyntaxOn`

- Leerzeichen werden völlig ignoriert
- ~ gibt ein Leerzeichen

`\NewDocumentCommand \Befehl {Argumente} { Code }`

- `\Befehl` sollte nicht vorher existieren
- Argumente: ab 1 nummeriert
 - m** (mandatory) Pflichtargument (in `{}`)
 - O{foo}** optional mit Standardwert foo (in `[]`)
- Weitere Argumenttypen in der Doku
- Argument im Code mit #1 usw. verwenden
- ## gibt ein echtes #

Beispiel: \dif

```
\NewDocumentCommand \dif {m}  
{  
  \mathinner{\\sympup{d} #1}  
}
```

Code

```
\begin{equation}  
  \int^{\!}\! \dif{x} \dif{^2 \sympbf{y}} x^2  
  | \sympbf{y} |  
\end{equation}
```

Ergebnis

$$\int dx d^2 \mathbf{y} x^2 |\mathbf{y}| \quad (10)$$

Das Prinzip gilt auch für Dx , δx , Δx .

Dabei sind D , δ , Δ gerade, weil sie keine Variablen sind.

Code

```
\dif{x} \Dif{x} \del{x} \Del{x}
```

Ergebnis

$dx Dx \delta x \Delta x$

Beispiel: $\backslash v$

```
\let\vaccent=\v          % alten Befehl kopieren
\RenewDocumentCommand \v {} % Befehl überschreiben
{
  \TextOrMath{
    \vaccent              % Textmodus
  }{
    \symbf                 % Mathemodus
  }
}
```

Code

```
\v{a}
\begin{equation}
  \int^{\v{a}} \frac{dx}{x^2} \frac{d^2 y}{dy^2} x^2 |\v{y}|
\end{equation}
```

Ergebnis

ä

$$\int^{\v{a}} dx d^2 \mathbf{y} x^2 |\mathbf{y}| \quad (11)$$

Beispiel: Umgebung

```
\NewDocumentEnvironment {CenterStrip} {0{\textwidth} m}
{
    % Code für \begin
    \begin{minipage}[c][#2\baselineskip][c]{#1}
}{
    % Code für \end
    \end{minipage}
    \ignorespacesafterend % Einrückung von Text nach Umgebung vermeiden
    % #1 und #2 können auch hier benutzt werden
}
```

Code

```
\begin{CenterStrip}{3}
    vertikal zentriert!
\end{CenterStrip}
\\[2\baselineskip]
\hfill
\begin{CenterStrip}
    [0.6\textwidth]{4}
    vertikal zentriert!
\end{CenterStrip}
```

Ergebnis

vertikal zentriert!

vertikal zentriert!

Alte Befehle, die man häufig trifft:

```
\newcommand*\Befehl [Anzahl Argumente] {Code}
```

```
\newcommand*\Befehl [Anzahl Argumente] [Default] {Code}
```

```
\newenvironment* {Umgebung} [Anzahl Argumente] {\begin-Code} {\end-Code}
```

- Nur ein optionales Argument möglich, muss erstes Argument sein
- \end-Code kann Argumente nicht benutzen

Fortgeschritten

Breites

\OverfullCenter

```
\includegraphics[width=\textwidth+15pt]{figures/Panorama.jpg}
```

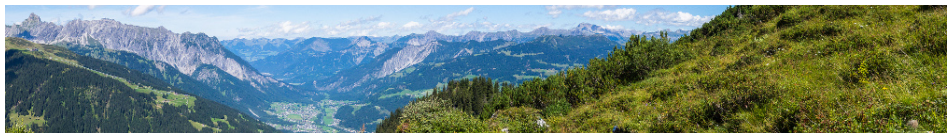
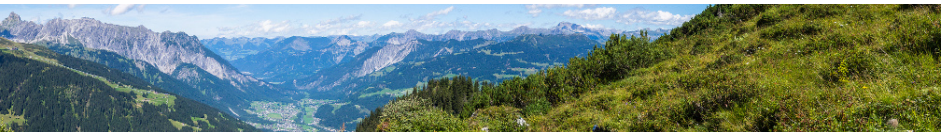


Bild oder Tabelle ist zu Breit, passt aber auf die Seite.
Wie kriegt man es in die Mitte?

```
\OverfullCenter{\includegraphics[width=\textwidth+15pt]{figures/  
Panorama.jpg}}
```



Code

```
\NewDocumentCommand \OverfullCenter {+m} {  
  \noindent\makebox[\linewidth]{#1} }
```

Falls das Bild oder die Tabelle wirklich breiter als die Seite ist, ist vielleicht eine gedrehte Seite die Lösung.

Benötigte Pakete

```
\usepackage{pdfscape}
```

Code

```
\begin{landscape}  
  \begin{table}  
    % .  
  \end{table}  
\end{landscape}
```

- Inhalt der landscape-Umgebung wird horizontal gesetzt (separate Seite)
- Seite wird im PDF-Reader horizontal angezeigt → schöner zu lesen

⟨insert wide table here⟩

Fortgeschritten

Mathe: Expert

Manchmal braucht man einen Script-Font oder einen zweiten kalligraphischen.

Code

```
\symcal{IA} \symbfcal{IA} % Latin Modern  
\symcal{IA} \symbfcal{IA} % XITS Math, StylisticSet=1  
\symscr{IA} \symbfscr{IA} % XITS Math
```

Ergebnis

I A I A
I A I A
I A I A

Mathe-Fonts einstellen

```
\setmathfont{XITS Math}[range={scr, bfscr}]  
\setmathfont{XITS Math}[range={cal, bfcal}, StylisticSet=1]
```

$\backslash\text{Re}$, $\backslash\text{Im}$

$\backslash\text{Re}$ und $\backslash\text{Im}$ tun nicht das, was man will:

Code

```
 $\backslash\text{Re } z$       $\backslash\text{Im } z$ 
```

Ergebnis

$\Re z$ $\Im z$

```
 $\text{\AtBeginDocument}$ { % wird bei  $\text{\begin}$ {document} ausgeführt  
   $\text{\let}\text{\symIm}=\text{\Im}$  % werden sonst wieder von unicode-math überschrieben  
   $\text{\RenewDocumentCommand}$   $\backslash\text{Re}$  {}  
  {  
     $\text{\operatorname}$ {Re}  
  }  
   $\text{\let}\text{\symIm}=\text{\Im}$   
   $\text{\RenewDocumentCommand}$   $\backslash\text{Im}$  {}  
  {  
     $\text{\operatorname}$ {Im}  
  }  
}
```

Besser:

Code

```
 $\backslash\text{Re } z$       $\backslash\text{Im } z$ 
```

Ergebnis

$\text{Re } z$ $\text{Im } z$

Richtiges Spacing für `\left`, `\right`

Benötigte Pakete

```
\usepackage{mleftright}
```

Code

```
\sin \left( x \right) y  
\sin \mleft( x \mright) y
```

Ergebnis

$\sin(x)y$
 $\sin(x)y$

Man kann natürlich eigene kurze Makros für `\mleft` und `\mright` definieren.
Beispiel: `\l` und `\r` (Textbedeutungen beachten!).

```
\let\ltext=\l  
\RenewDocumentCommand \l {}  
{  
  \TextOrMath{ \ltext }{ \mleft }  
}  
\let\raccent=\r  
\RenewDocumentCommand \r {}  
{  
  \TextOrMath{ \raccent }{ \mright }  
}
```

`\DeclarePairedDelimiter`

- Man kann mit dem `mathtools`-Befehl `\DeclarePairedDelimiter` Befehle erzeugen, die Symbole um Ausdrücke setzen.
- Automatische \star -Variante, die mitwächst.
- Mit richtigem Spacing!

Code

```
% in Präambel
\DeclarePairedDelimiter{\abs}{\lvert}{\rvert}
\DeclarePairedDelimiter{\norm}{\lVert}{\rVert}

% in Mathe:
\abs{x} \abs*{\frac{1}{x}}
\norm{\symbf{y}}

\sin\abs*{\frac{1}{2}}
\sin\left|\frac{1}{2}\right|
```

Ergebnis

$$\begin{array}{cc} |x| & \left|\frac{1}{x}\right| \\ \|\mathbf{y}\| & \end{array}$$

$$\sin\left|\frac{1}{2}\right|$$

$$\sin\left|\frac{1}{2}\right|$$

`\bra`, `\ket`, `\braket`

Schonmal für Physik IV und Quantenmechanik vormerken.

In der Präambel

```
\DeclarePairedDelimiter{\bra}{\langle}{\rvert}
\DeclarePairedDelimiter{\ket}{\lvert}{\rangle}
% <name> <#arguments> <left> <right> <body>
\DeclarePairedDelimiterX{\braket}[2]{\langle}{\rangle}{
  #1 \delimsize| #2
}
```

→ `\delimsize` gibt Größe der äußeren Klammern in `<body>`

Code

```
\bra{\Psi}
\ket{\Psi}
\braket*{\Psi_1}{\Psi_2}
```

Ergebnis

$$\langle \Psi |$$
$$| \Psi \rangle$$
$$\langle \Psi_1 | \Psi_2 \rangle$$

Klammern wachsen nicht immer:

Code

```
\left( \left( \left( \left(
  x
\right) \right) \right) \right)

% in Präambel
\setlength{\delimitershortfall}{-1sp}

\left( \left( \left( \left(
  x
\right) \right) \right) \right)
```

Ergebnis

$$(((x)))$$

$$((((x))))$$

Fortgeschritten

Makefiles

LuaT_EX und biber bieten Optionen an, um einen build-Ordner zu benutzen.

Aufrufe

```
lualatex --output-directory=build file.tex  
biber build/file.bcf
```

Um Dateien aus dem build-Ordner zu finden (Plots, Tabellen)

Aufrufe

```
TEXINPUTS=build: lualatex --output-directory=build file.tex  
BIBINPUTS=build: biber build/file.bcf
```

- TEXINPUTS, BIBINPUTS: Suchpfade für T_EX- und .bib-Dateien
- Elemente getrennt mit :, der erste Treffer wird genommen (wie PATH)
- TEXINPUTS auch für `\includegraphics`
- : am Ende der Liste: Standardsuchpfade anhängen (wichtig!)
- . (der aktuelle Ordner) ist am Anfang der Standardliste, braucht man also nicht selbst angeben

In Makefiles will man keine Interaktion.

Keine Interaktion

```
lualatex --interaction=nonstopmode file.tex
```

Beim ersten Fehler abbrechen

```
lualatex --interaction=nonstopmode --halt-on-error file.tex
```

Neben nonstopmode gibt es auch batchmode, was die Ausgabe nur in der .log-Datei speichert, aber nicht ausgibt.

Log schöner machen

```
max_print_line=1048576 lualatex file.tex
```

latexmk

- Problem: Mehrfaches Kompilieren von Dokumenten ist aufwändig und fehleranfällig
- latexmk ist ein Kommandozeilenwerkzeug, das automatisch tex (und andere Programme wie biber) oft genug aufruft
- Bei TeXLive mitgeliefert
- Auswahl von Lua^LA^T_EX durch Parameter `--lualatex`.
- Versteht auch viele tex-Argumente wie `--interaction` und `--halt-on-error`.

Aufruf auf der Kommandozeile

```
latexmk --lualatex --output-directory=build --interaction=nonstopmode
--halt-on-error file.tex
```

- Noch mehr Kontrolle durch Konfigurationsdatei `latexmkrc`
- Siehe dazu Dokumentation

Im Makefile

```
build/file.pdf: FORCE plots... tabellen...
    TEXINPUTS=build: \
    BIBINPUTS=build: \
    max_print_line=1048576 \
    latexmk \
    --lualatex \
    --output-directory=build \
    --interaction=nonstopmode \
    --halt-on-error \
    file.tex

FORCE:
```

- latexmk bestimmt Abhängigkeiten selbst
- Sollte also immer ausgeführt werden
 - FORCE

```
latexmk -pvc --interaction=nonstopmode ... document.tex
```

- latexmk merkt, wenn ihr eure Dateien ändert
- Kompiliert automatisch neu
- Öffnet den Standard-PDF-Betrachter
- Einfach im Hintergrund laufen lassen

$\text{T}_{\text{E}}\text{X}$ in matplotlib

```
import matplotlib.pyplot as plt
import numpy as np
```

```
x = np.linspace(0, 10, 1000)
y = x ** np.sin(x)
```

```
plt.plot(x, y)
plt.xlabel(r'$\alpha$ / $\Omega$')
```

```
plt.savefig('build/figures/mattex1.pdf')
```

8

7

6

5

4

```
import matplotlib.pyplot as plt
import numpy as np
```

```
x = np.linspace(0, 10, 1000)
y = x ** np.sin(x)
plt.figure(figsize=(4.76, 2.94))
plt.plot(x, y)
plt.xlabel(r'$\alpha / \Omega$')
plt.tight_layout(pad=0)
plt.savefig('build/figures/mattex2.pdf',
            bbox_inches='tight', pad_inches=0)
```

```
plt.figure(figsize=(4.76, 2.94))
```

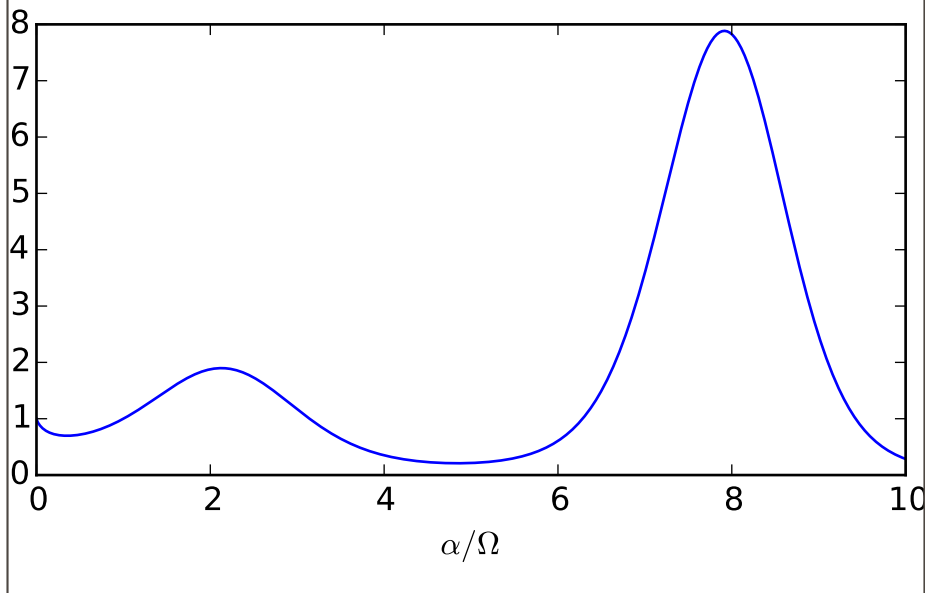
- Größe der Zeichenfläche setzen (in Zoll)
- Breite des textes kann mit `\the\textwidth` ins Dokument geschrieben werden
- $1\text{ in} = 72,27\text{ pt}$
- Goldener Schnitt für Höhe
- Für `scrartcl` mit Standardeinstellungen: 5.78, 3.57

```
plt.tight_layout(pad=0)
```

```
plt.savefig(..., bbox_inches='tight', pad_inches=0)
```

- Weiße Leerräume am Rand eliminieren
- Inhalt des Bilds ist genauso breit wie der Text

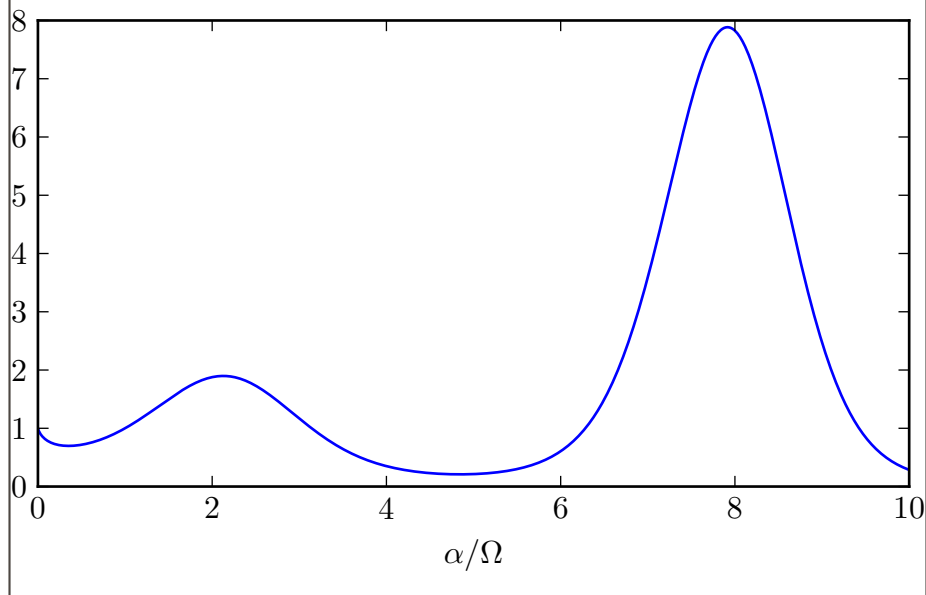
Ergebnis (2)



```
import matplotlib as mpl
mpl.use('pgf')
import matplotlib.pyplot as plt
import numpy as np
mpl.rcParams.update({
    'font.family': 'serif',
    'text.usetex': True,
    'pgf.rcfonts': False,
    'pgf.texsystem': 'luatex',
    'pgf.preamble': r'\usepackage{unicode-math}\usepackage{siunitx}',
})

x = np.linspace(0, 10, 1000)
y = x ** np.sin(x)
plt.figure(figsize=(4.76, 2.94))
plt.plot(x, y)
plt.xlabel(r'$\alpha$ / \si{\ohm}$')
plt.tight_layout(pad=0)
plt.savefig('build/figures/mattex3.pdf',
            bbox_inches='tight', pad_inches=0)
```

Ergebnis (3)



```
import matplotlib as mpl
mpl.use('pgf')
import matplotlib.pyplot as plt
import numpy as np
mpl.rcParams.update({
    'font.family': 'serif',
    'text.usetex': True,
    'pgf.rcfonts': False,
    'pgf.texsystem': 'luatex',
    'pgf.preamble': r'\input{header-matplotlib.tex}',
})

x = np.linspace(0, 10, 1000)
y = x ** np.sin(x)
plt.figure(figsize=(4.76, 2.94))
plt.plot(x, y)
plt.xlabel(r'$\alpha$ / $\si{\ohm}$')
plt.tight_layout(pad=0)
plt.savefig('build/figures/mattex4.pdf',
            bbox_inches='tight', pad_inches=0)
```

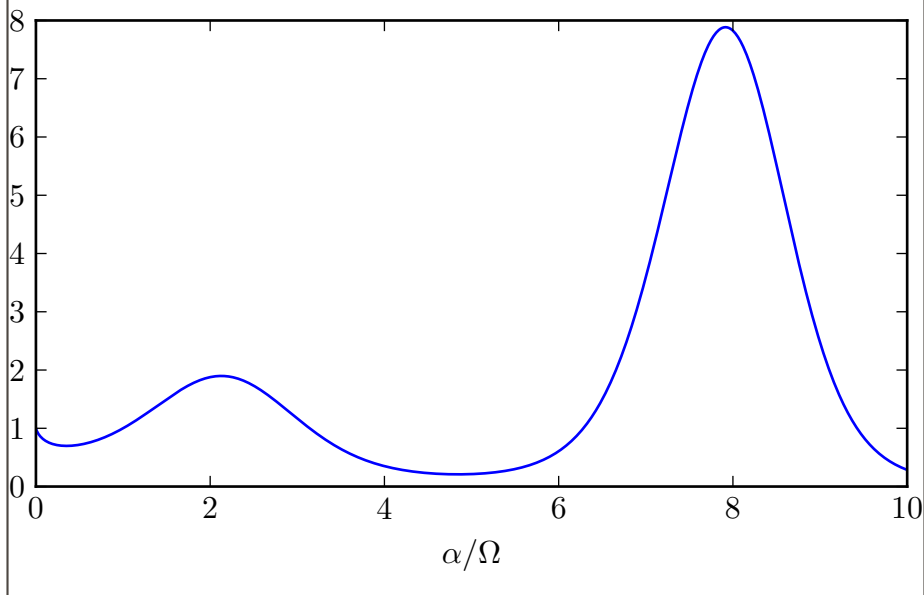
```

\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{mathtools}
\usepackage{fontspec}
\usepackage[
  math-style=ISO,
  bold-style=ISO,
  sans-style=italic,
  nabla=upright,
  partial=upright,
]{unicode-math}
\setmathfont{Latin Modern Math}
\usepackage[
  per-mode=reciprocal,
]{siunitx}

```

- \TeX wird von matplotlib in /tmp ausgeführt
 - Datei kann nicht gefunden werden
- Lösung: TEXINPUTS setzen!
- `TEXINPUTS=$(pwd): python script/mattex4.py`
- Makefile: `TEXINPUTS=$(shell pwd): python script/mattex4.py`

Ergebnis (4)



```
import matplotlib.pyplot as plt
import numpy as np
```

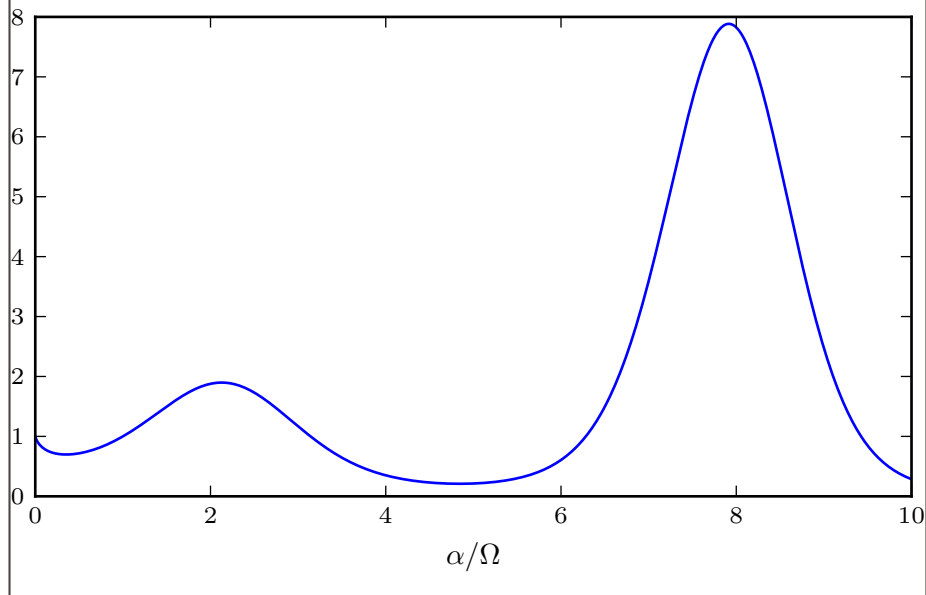
```
x = np.linspace(0, 10, 1000)
y = x ** np.sin(x)
```

```
plt.plot(x, y)
plt.xlabel(r'$\alpha / \si{\ohm}$')
plt.tight_layout(pad=0) # pad=0 in matplotlibrc leider nicht möglich
plt.savefig('build/figures/mattex5.pdf')
```

```
backend : pgf # mpl.use('...')
figure.figsize : 4.76, 2.94 # 5.78, 3.57 für scrartcl
font.family : serif
font.size : 11 # standard Textgröße in scrartcl
legend.fontsize : medium
pgf.rcfonts : False
pgf.texsystem : lualatex
pgf.preamble : \input{header-matplotlib.tex}
savefig.bbox : tight # savefig(bbox_inches=...)
savefig.pad_inches : 0 # savefig(pad_inches=...)
text.usetex : True
xtick.labelsize : 9
ytick.labelsize : 9
```

- Datei heißt matplotliblibrc ohne Endung!
- Wird im aktuellen Verzeichnis gesucht
 - nicht unbedingt gleich dem Ordner, wo das Skript liegt

Ergebnis (5)



Präsentationen mit \LaTeX : beamer

- Dokumentenklasse für Präsentationen
- frame-Umgebung erzeugt eine Folie
- Bei Nutzung mit `fontspec` und `unicode-math` muss die Option `professional fonts` gesetzt werden.
- Aussehen wird durch „themes“ gesteuert.
- Viele themes werden mit $\text{T}_\text{E}\text{X}$ -Live mitgeliefert.
- Sehen leider alle fast gleich aus.

```
\documentclass[
  professionalfonts,
]{beamer}

\usepackage{fontspec}
\usepackage[
  math-style=ISO,
  bold-style=ISO,
  nabla=upright,
  partial=upright,
  sans-style=italic,
]{unicode-math}
\setmathfont{Latin Modern Math}

\begin{document}
  \begin{frame}{title}
    Hallo Welt!
  \end{frame}
\end{document}
```

- columns-Umgebung für Bereich mit mehreren Spalten
- Option onlytextwidth damit nichts in den Rand ragt
- Mögliche option für vertikale Ausrichtung der Spalten:
 - t** top, funktioniert nicht bei Bildern
 - c** center
 - b** bottom
 - T** wie t, funktioniert aber auch bei Bildern
- column-Umgebung erzeugt Spalte, Breite ist Pflichtargument

```
\begin{columns}[onlytextwidth]
  \begin{column}[0.45\textwidth]
    Hallo
  \end{column}
  \begin{column}[0.45\textwidth]
    Welt
  \end{column}
\end{columns}
```

- (Zu?) Oft genutztes Element in beamer-Präsentationen
- Standardblöcke können nicht viel → tcolorbox

Code

```
\begin{block}{Titel}  
  Block Body  
\end{block}
```

```
\begin{exampleblock}{Titel}  
  Block Body  
\end{exampleblock}
```

```
\begin{alertblock}{Titel}  
  Block Body  
\end{alertblock}
```

Ergebnis

Titel

Block Body

Titel

Block Body

Titel

Block Body

Nervige Buttons abschalten

```
\documentclass[...]{beamer}
% ...
% packages here
% ...

\setbeamertemplate{navigation symbols}{}

\begin{document}
  \begin{frame}{title}
    Hallo Welt!
  \end{frame}
\end{document}
```

```
\documentclass[professionalfonts]{beamer}
% ...
% packages here
% ...

\usepackage{siunitx}

\AtBeginDocument{
  \sisetup{
    math-rm=\mathrm,
    math-micro=μ, % AltGr+m = MICRO SIGN, Unicode: U+00B5
  }
}

\begin{document}
  \begin{frame}{title}
    \SI{5}{\micro\ohm}
  \end{frame}
\end{document}
```


Zeichnen mit Tikz

Benötigte Pakete

```
\usepackage{tikz}
```

- Tikz ist **kein** Zeichenprogramm
- Zeichnen mit Befehlen
 - Sehr präzise
 - programmierfähig
 - automatisierbar
 - Versionkontrolle!
- Extrem umfangreiche Doku mit *zahlreichen* Beispiel (>1000 Seiten)
- Basis-Einheit ist cm

Code

```
\begin{tikzpicture}  
  \draw[thick, ->] (0, 0) -- (1, 0);  
\end{tikzpicture}
```

Ergebnis



cycle

```
\begin{tikzpicture}
  \draw[thick] (0, 0) -- (1, 0) -- (1, 1) -- cycle;
\end{tikzpicture}
```

Ergebnis



Polarkoordinaten

```
\begin{tikzpicture}
  \foreach \ang in {0, 45, 90, 135, 180, 215, 270, 315}
  {
    \draw (0, 0) -- (\ang: 10pt);
  }
\end{tikzpicture}
```

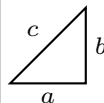
Ergebnis



nodes

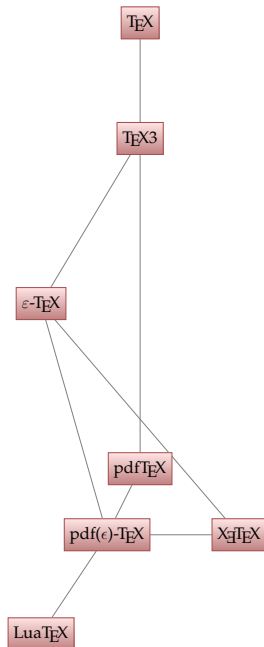
```
\begin{tikzpicture}
  \draw[thick] (0, 0)
    -- (1, 0) node[midway, below] {$a$}
    -- (1, 1) node[midway, right] {$b$}
    -- cycle node[midway, above left] {$c$};
\end{tikzpicture}
```

Ergebnis

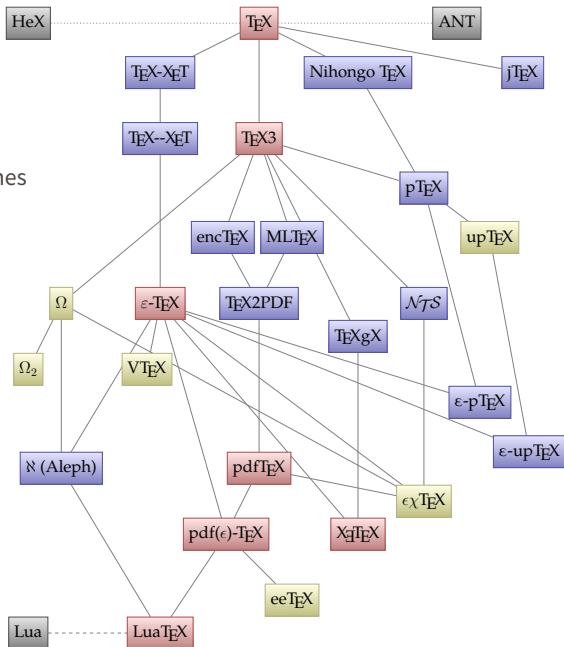


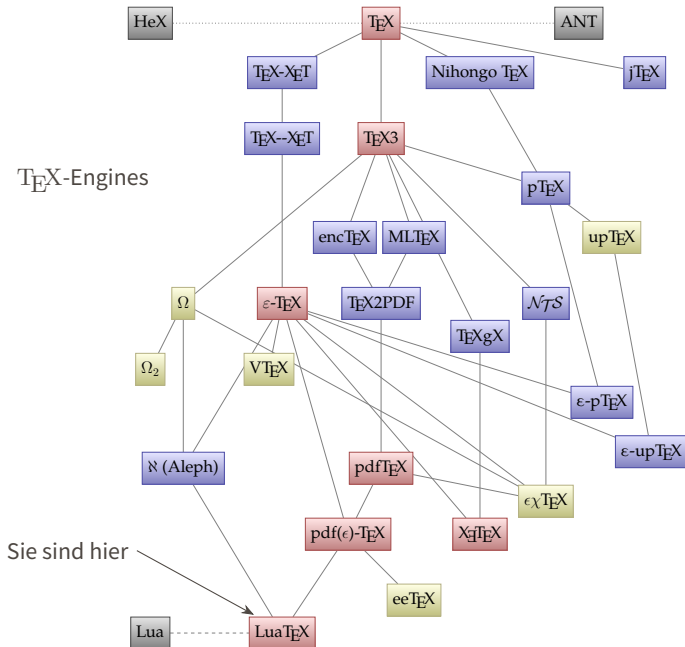
Ausblick

$\text{T}_\text{E}\text{X}$ -Engines



$\text{T}_\text{E}\text{X}$ -Engines





Unicode-Input → Bequem, äöüßêë funktioniert einfach

OTF-Fonts → Alle Fonts benutzen, die man auf dem Rechner hat

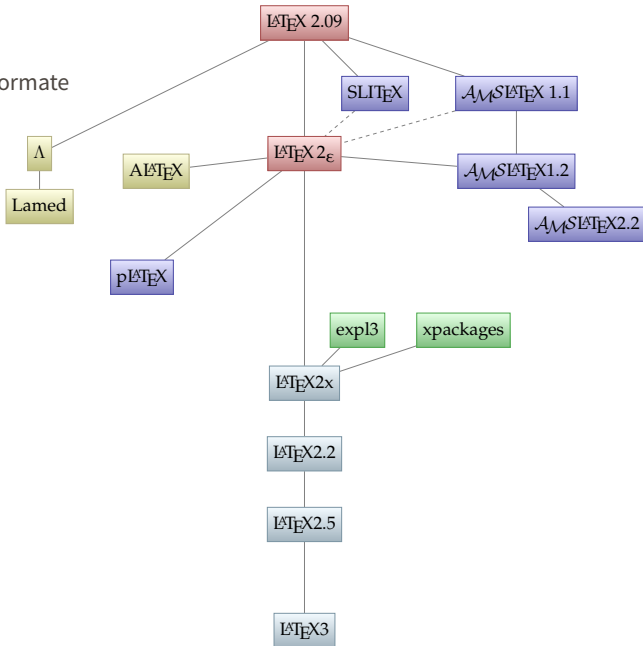
Unicode-Math → Mathe-Input über Unicode
→ Stichwort: Compose-Key (XCompose, Linux)
→ Code lesbarer, Tippen schneller
→ Mehr Font-Möglichkeiten

Lua-Programmierung → T_EX-Programmierung ist nicht besonders einfach
→ Manche Pakete bieten weitergehende Funktionen nur über Lua

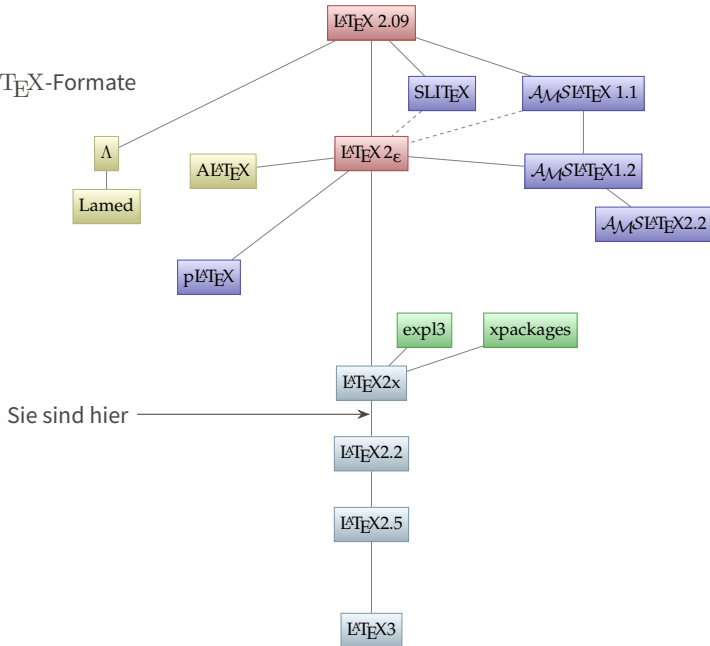
T_EX-Format



T_EX-Formate



T_EX-Formate



- L^AT_EX3 existiert (noch) nicht
- `expl3` ist L^AT_EX3 unter L^AT_EX 2_ε
- `xpackages` sind Pakete, die auf `expl3` aufbauen und neue Möglichkeiten bieten
- `xparse` macht das schreiben neuer (auch komplizierter) Befehle sehr einfach
- viele Pakete benutzen jetzt schon `expl3` und `xparse`

scrlettr2 Briefe

MusiXTeX, Lilypond Notensatz

IEEEtrantools Mächtigere Matheumgebungen

Poster beamerposter, tcolorbox

todonotes TODOs im Text, Liste am Ende, Platzhalter für Grafiken

L^AT_EX:

```
\DeclareRobustCommand{\LaTeX}{%  
  L\kern-.36em%  
  {\sbox\z@ T%  
    \vbox to\ht\z@{\hbox{%  
      \check@mathfonts  
      \fontsize\sf@size\z@  
      \math@fontsfalse\selectfont A}%  
      \vss}%  
    }%  
  \kern-.15em%  
  \TeX}
```

... alles klar?