

# Verfassen wissenschaftlicher Texte mit $\text{\LaTeX}$

PeP et al. Toolbox Workshop



**PeP et al. e.V.**

Physikstudierende und  
ehemalige Physikstudierende  
der TU Dortmund

2021

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

Makros

Mathe: Expert

`\texorpdfstring`

Links

Makefiles

Breites

`tlmgr`

`latexmk`

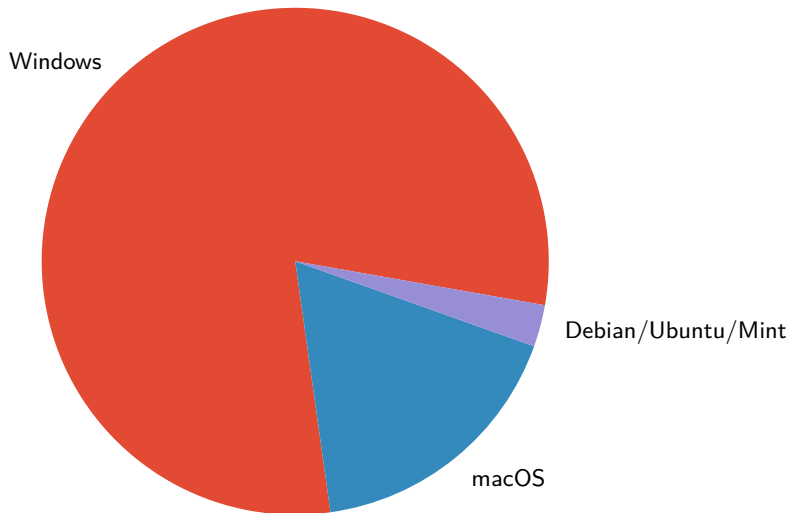
`TEX` in `matplotlib` in `TEX`

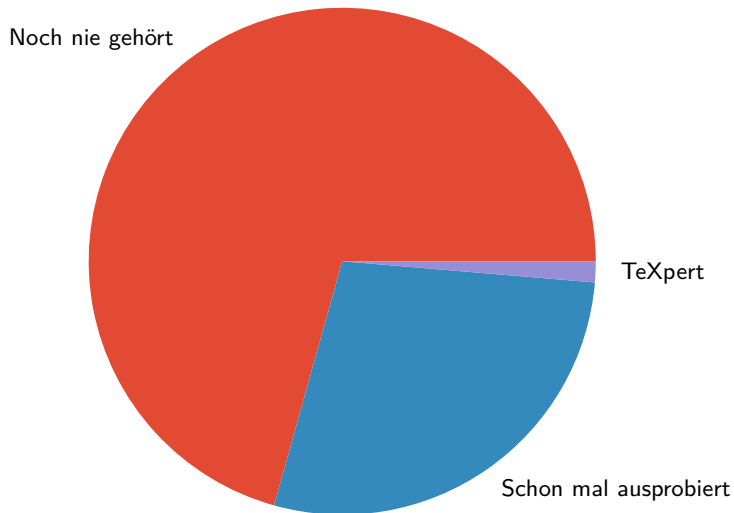
Präsentationen mit `ΛTEX`: `beamer`

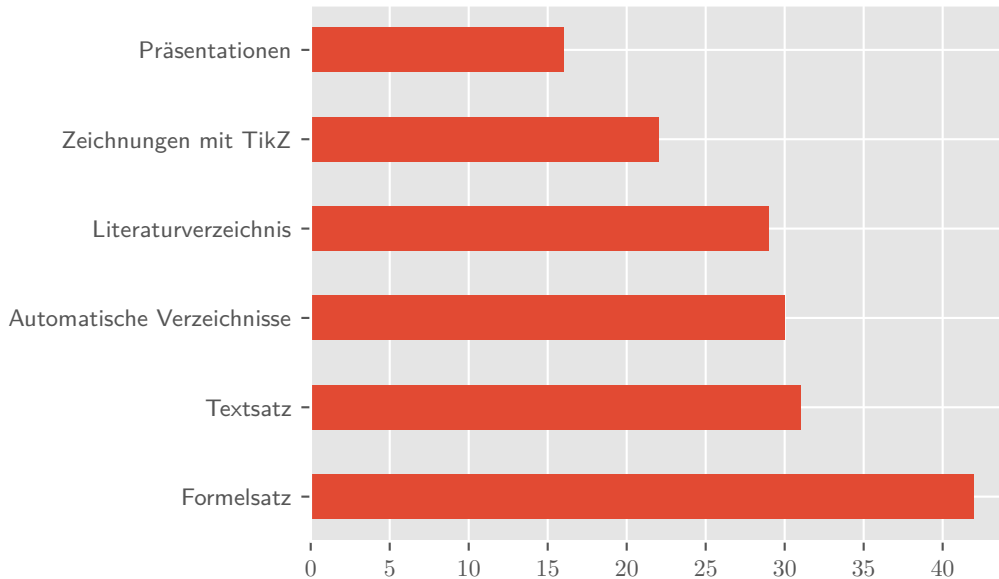
Zeichnen mit Tikz

Ausblick

# Ergebnisse der Umfrage







# Einführung



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

- Hervorragender Text- und Formelsatz
- Automatisierte Erstellung von Inhalts- und Literaturverzeichnis
- $\text{\TeX}$ -Dateien sind reine Text-Dateien
  - ⇒ Gut für Versionskontrolle geeignet
- Sehr gute Vorlagen für wissenschaftliches Arbeiten

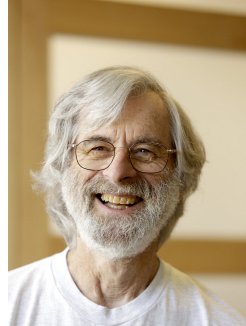
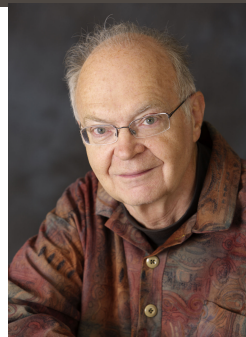
- Ausgezeichnete Dokumentation
- Erweiterbar durch zahlreiche und mächtige Pakete
- Auf allen geläufigen Betriebssystemen verfügbar
- Ausgabe direkt als PDF mit Hyperlinks

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

- Geschrieben von Donald E. Knuth 1978, um sein Buch „The Art of Computer Programming“ zu setzen.
- Auf Aussprache achten!
- Version (2021): 3.141592653 →  $\pi$
- Viele Erweiterungen:  $\epsilon\text{-T}_{\text{E}}\text{X}$ ,  $\text{pdfT}_{\text{E}}\text{X}$ ,  $\text{X}_{\text{F}}\text{T}_{\text{E}}\text{X}$ ,  $\text{LuaT}_{\text{E}}\text{X}$

$\text{\LaTeX}$ :

- Geschrieben von Leslie Lamport 1984
- Version (1994):  $\text{\LaTeX} 2_{\epsilon}$
- $\text{\LaTeX} 3$  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

**T<sub>E</sub>X-Engine** Implementierung von T<sub>E</sub>X, wird als Programm ausgeführt

**T<sub>E</sub>X-Format** Paket, welches standardmäßig geladen wird, z.B. L<sup>A</sup>T<sub>E</sub>X

Eine Kombination davon ist oft ein neues Programm.

Beispiel: `dvilualatex` = LuaT<sub>E</sub>X + L<sup>A</sup>T<sub>E</sub>X + DVI-Output (statt PDF)

# Grundlagen

Diese drei Zeilen braucht jedes  $\LaTeX$ -Dokument:

Code

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

`\documentclass`

Vorlage wählen, mit Optionen anpassen.

Präambel

Globale Optionen und zusätzliche Pakete.

document-Umgebung

Inhalt des Dokuments.



## Code

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

## Ergebnis

Hallo Welt!

# Syntax: Befehle

$\LaTeX$ -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

# Syntax: Umgebungen

- 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}
```

# Standardpakete

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

## Paket

```
\usepackage[aux]{rerunfilecheck}
```

```
\usepackage{fontspec}
```

```
\usepackage[ngerman]{babel}
```

```
% mehr Pakete hier
```

```
\usepackage[unicode]{hyperref}
```

```
\usepackage{bookmark}
```

## Funktion

Warnung, falls nochmal kompiliert werden muss.

Für Fonteinstellungen

Deutsche Spracheinstellungen.

Mehreren Sprache: **english**, **ngerman**

Hauptsprache als letztes

Für Hyperlinks (z.B. Inhaltsverzeichnis → Kapitel).

Bessere Bookmarks im PDF, korrekte Seitenzahlen im Viewer, wenn mehrere Numerierungen

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 empfohlene Klasse

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

Standardeinstellung sind die Latin-Modern-Fonts.

Latin Modern

```
\usepackage{fontspec}
```

Alternativ zum Beispiel: Libertinus

```
\usepackage{fontspec}  
\setmainfont{Libertinus Serif}  
\setsansfont{Libertinus Sans}  
\setmonofont{Libertinus Mono}
```

- Jede System-Schriftart kann genutzt werden
- 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
- Welche Schriftarten zueinander passen ist eine Wissenschaft für sich.

VORSICHT BEIM KOPIEREN AUS PDFs! Besser selbst tippen

```
\documentclass{scrartcl}

\usepackage[aux]{rerunfilecheck}

\usepackage{fontspec}

\usepackage[ngerman]{babel}

% 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  $\text{\LaTeX}$ -Kompiler, die verschiedene Ausgabeformate erzeugen können. Der modernste Kompiler, der PDF-Dateien erstellt, ist **lua $\text{\LaTeX}$** .

## $\text{\LaTeX}$ -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.

$\LaTeX$  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  $\text{\TeX}$ -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                                       | Ergebnis                  |
|--|---------------------------|
| <code>\textit{kursiv} \emph{kursiv}</code> | <i>kursiv kursiv</i>      |
| <code>\textbf{fett}</code>                 | <b>fett</b>               |
| <code>\textbf{\textit{fett-kursiv}}</code> | <b><i>fett-kursiv</i></b> |
| <code>\textrm{Serifen-Schrift}</code>      | Serifen-Schrift           |
| <code>\texttt{Mono-Schrift}</code>         | Mono-Schrift              |
| <code>\textsf{Sans-Serif-Schrift}</code>   | Sans-Serif-Schrift        |
| <code>\textsc{Kapitälchen}</code>          | 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

```
% babel mit anderen Sprachen laden
\usepackage[english, french, ngerman]{babel}
\usepackage[autostyle]{csquotes} % babel
```

## Code

```
foo \enquote{bar} baz
\enquote{foo \enquote{bar} baz}
\foreignlanguage{english}{\enquote{foo}}
\foreignlanguage{french}{\enquote{foo}}
% siehe Kapitel über Bibliographie
\textcquote{numpy}{foo}
```

## Ergebnis

```
foo „bar“ baz
„foo ‚bar‘ baz“
“foo”
« foo »

„foo“ [5]
```

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](https://tex.stackexchange.com)

# Aufzählungen

- $\text{\LaTeX}$  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

# Titelseite und Metadaten

L<sup>A</sup>T<sub>E</sub>X erstellt automatisch eine Titelei 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
```

LaTeX 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{fontspec}
```

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

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

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

```
\usepackage{fontspec}
```

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

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

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

```
\usepackage[
```

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

```
% Laden von OTF-Mathefonts
```

```
% Ermöglich Unicode Eingabe von Zeichen:  $\alpha$  statt \alpha
```



```
\usepackage{fontspec}
```

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

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

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

```
\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."
```

```
% Laden von OTF-Mathefonts
```

```
% Ermöglich Unicode Eingabe von Zeichen:  $\alpha$  statt \alpha
```

```
\usepackage{fontspec}

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

\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."
% Laden von OTF-Mathfonts
% Ermöglich Unicode Eingabe von Zeichen: α statt \alpha

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

## `$. . . $`-Umgebung

Aktiviert den Mathematikmodus im Fließtext.

`TEX` sorgt für gute Abstände

```
$x = 5$, $y=3$
```

Satzzeichen u. Bindestriche gehören nicht in `$. . . $`

```
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

$x = 5, y = 3$

Ergebnis

Dies ist eine Variable:  $x$ .  
Liste von Variablen  $x, y, z$ .  
 $y$ -Achse,  $x$ - $y$ -Ebene

Ergebnis

Text ohne eine Bedeutung.  
Mit einer Formel:  $\frac{1}{1 - \frac{1}{1 - x}}$   
Text ohne eine Bedeutung.

## Code

```
\epsilon \theta \kappaappa \pi \rhoho \sigma \phi  
\varepsilon \vartheta \varkappaappa \varpi \varrho \varsigma \varphi  
  \varphi  
\Alpha \Beta \Gamma  
\hbar \imath \jmath \ell  
\partial \nabla \square \increment  
\infty \diameter
```

## Ergebnis

$\epsilon \theta \kappa \pi \rho \sigma \phi$   
 $\varepsilon \vartheta \varkappa \varpi \varrho \varsigma \varphi$

$A B \Gamma$   
 $\hbar \iota \jmath \ell$   
 $\partial \nabla \square \Delta$   
 $\infty \emptyset$

## Code

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

## Ergebnis

+ − / · ×  
± ∓  
< > ≤ ≥  
= ≈ ≡ ≐  
≈ ∝ ~  
:= =:  
→ ↔ ⇒  
↦ ↷  
∀ ∃ ∈ ⊂ ∪ ∩

## Negierte Variante mit n bzw. not

```
\neq \nsime \nexists \nni \notin
```

## Ergebnis

≠ ≠̸ ≠̇ ≠̈

## Häufig möchte man etwas über eine Relation schreiben:

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

## Ergebnis

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

# Indizes / Exponenten

Code

```
x^2 x_2 x²
```





Ergebnis

$x^2$   $x_2$   $x^2$

Falsch

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

Ergebnis

$x^{10}$   
 Error   
 Error 

Richtig

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

Ergebnis

$x^{10}$   
 $x^{2^2}$   
 $x_{\sqrt{3}}$

Text in Indizes

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

Ergebnis

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

Striche & Linksseitiges

```
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}

```

Auf Position des Akzents achten:

```

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

```

## Ergebnis

$$\bar{x}$$

$$\hat{x}$$

$$\tilde{x}$$

$$\vec{x}$$

$$\mathring{x}$$

$$\dot{x} \ddot{x} \dddot{x} \ddddot{x}$$

$$\underline{xy} \overline{xy}$$

## Ergebnis

$$x_{\text{min}}^{\hat{}}$$

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

## 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

$xyz_i(a)$   
 $xyz_i(a)$

$$xyz_i(a)$$
  
$$Xyz_i(a)$$





# 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}
\sympfsf{x \alpha}
\sybbb{R N 1 0 x}
\symcal{I A O} \sympfcal{I A O}
\symfrac{A B c} \sympffrak{A B c}
```

## Ergebnis

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

Für mehrbuchstabile Bezeichnungen gibt es andere Befehle:

## Code

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

## Ergebnis

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

# Spaces

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

Code

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

Ergebnis

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

Negativer Space um zu viel Platz zu korrigieren:

Code

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

Ergebnis

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

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
```

## Häufig braucht man größere Klammern

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

## Ergebnis

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

## 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

$(\frac{1}{2})$      $(\frac{1}{2})$

$\{x \mid x < \frac{1}{2}\}$

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:  $e$ ,  $i$ ,  $\pi$   
 `$\backslash\symup{e}$` ,  `$\backslash\symup{i}$` ,  `$\backslash\symup{\pi}$`
  - Infinitesimales:  $dx$   
 `$\backslash\symup{d}x$`
  - Indizes wie „min“ oder „max“  
 `$x_{\text{min}}$`
- Vektoren und Matrizen: fett  
 `$\backslash\symbf{r}$`  =  $(0, 1, -1)^{\top}$   `$r = (0, 1, -1)^{\top}$`



- $dx$  wird durch kleines Leerzeichen ( $\backslash,$ ) vom Integranden abgetrennt
- $\backslash,$  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^\pi \int_0^{2\pi} r^2 \sin(\vartheta) \, d\phi \, d\vartheta \, dr = \frac{4}{3}\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
- Achtung: Leere Zeilen führen in allen Mathe-Umgebungen zu einem Fehler

## Code

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

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

## Ergebnis

Es gilt

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

Schon Gauß hatte das Durchflutungsgesetz (1) aufgestellt.

# Die gather-Umgebung

- 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

$$a = 1 \qquad b = 2 \qquad (5)$$
$$a \cdot b = 2 \qquad \frac{a}{b} = 0.5 \qquad (6)$$

# Die split-Umgebung

- 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



# Zahlen und Einheiten

- Regeln zur Benutzung der SI-Einheiten:  
<https://www.bipm.org/utils/common/pdf/si-brochure/SI-Brochure-9-EN.pdf>
- Einheiten werden aufrecht gesetzt
- Zwischen Zahl und Einheit steht ein kleines Leerzeichen
- Ab 5 Stellen wird ein kleines Leerzeichen als 1000er Trennzeichen genutzt:

Zahl mit Einheit

```
$5\, \mathrm{kg}$
```

Ergebnis

5 kg

Zahl mit mehr als vier Stellen

```
$10\,000$
```

Ergebnis

10 000

Zehnerpotenz mit Unsicherheit in Klammern

```
$(5{,}34 \pm 0{,}54) \cdot 10^{-3}\, \mathrm{GeV}$
```

Ergebnis

$(5,34 \pm 0,54) \cdot 10^{-3} \text{ GeV}$

# Zahlen und Einheiten

- Regeln zur Benutzung der SI-Einheiten:  
<https://www.bipm.org/utis/common/pdf/si-brochure/SI-Brochure-9-EN.pdf>
- Einheiten werden aufrecht gesetzt
- Zwischen Zahl und Einheit steht ein kleines Leerzeichen
- Ab 5 Stellen wird ein kleines Leerzeichen als 1000er Trennzeichen genutzt:

Zahl mit Einheit

```
$5\, \mathrm{kg}$
```

Ergebnis

5 kg

Zahl mit mehr als vier Stellen

```
$10\,000$
```

Ergebnis

10 000

Zehnerpotenz mit Unsicherheit in Klammern

```
$(5{,}34 \pm 0{,}54) \cdot 10^{-3}\, \mathrm{GeV}$
```

Ergebnis

$(5,34 \pm 0,54) \cdot 10^{-3} \text{ GeV}$

Das muss einfacher gehen

- 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 Unsicherheit 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 · 10<sup>23</sup>

## Angabe von Unsicherheiten

```
\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) · 10<sup>6</sup>

## Einheiten

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

## Ergebnis

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

## per-mode=symbol-or-fraction

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

## Ergebnis

$$\frac{\text{kg m}}{\text{s}^2} \quad (8)$$
  
kg m/s<sup>2</sup>

## Meter mal Sekunde oder Millisekunde?

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

## Ergebnis

ms  
m s  
m · s

# siunitx: Physikalische Größe, eine Zahl mit Einheit: `\qty`

`\qty` = Kombination aus `\num` und `\unit`

```
\qty{5}{\percent}  
\qty{10}{\degreeCelsius}  
\qty{2.5(1)e6}{\kilo\gram\square\meter\per\second  
  \squared}
```

- 1. Argument** Kann alles, was `\num` kann
- 2. Argument** Kann alles, was `\unit` kann

Ergebnis

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

Winkel

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

Ergebnis

5°  
5'  
5"  
5°55'  
5°55'59"

# Chemische Formeln

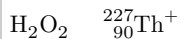
## Benötigte Pakete

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

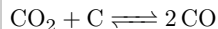
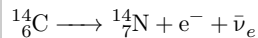
## Code

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

## Ergebnis



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





# Fortgeschrittener Formelsatz

# Displaystil vs. Textstil

$\LaTeX$  besitzt zwei Stile zum darstellen von Formeln

**textstyle** Standardmäßig in  $\$. . . \$$ . Kleiner, weniger hoch.

**displaystyle** Formeln werden in der vollen Höhe dargestellt, standardmäßig in allen Mathematik-Umgebungen.

Integral

```
\int_0^1 x^2$
```

```
\displaystyle\int_0^1 x^2$
```

Ergebnis

$$\int_0^1 x^2$$

$$\int_0^1 x^2$$

Bruch

```
\frac{1}{1 + \frac{1}{x^2}}$
```

```
\displaystyle\frac{1}{1 + \frac{1}{x^2}}$
```

Ergebnis

$$\frac{1}{1 + \frac{1}{x^2}}$$

$$\frac{1}{1 + \frac{1}{x^2}}$$

Matrizen (und damit Vektoren) werden fett geschrieben.

```
\sympf{M}^{\top}      \sympf{M}^*
\sympf{M}^{\dagger}  \sympf{M}^{-1}  M_{12}
```

Ergebnis

$$M^{\top} \quad M^*$$

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

Verschiedene Klammern

```
\begin{pmatrix} x & y \\ z & abc \end{pmatrix}
\begin{bmatrix} -1 \\ 2 \end{bmatrix}
\begin{Bmatrix} -1 \\ 2 \end{Bmatrix}
\end{pmatrix} \quad \end{bmatrix} \quad \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*}
\begin{pmatrix*}[r] 1 & -1 \\ -1 & 2 \end{pmatrix*}
\end{pmatrix*} \quad \end{pmatrix*} \quad \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
```

```
\cramped
```

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

Ergebnis

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

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

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

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

Ergebnis

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

# Fallunterscheidungen

- Umgebungen mit \* aktivieren den Textmodus nach dem &
- Umgebungen mit d am Anfang nutzen den Displaystil

## 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

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

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

`\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

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

# Ausrichtung trotz fehlender Symbole mit `\phantom`

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}} + x \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  $\text{\LaTeX}$  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}
```

## Code

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

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

## Ergebnis



**PeP et al. e.V.**

Physikstudierende und  
ehemalige Physikstudierende  
der TU Dortmund

Abbildung 1: Das PeP-Logo.

Benötigte Pakete

```
\usepackage{subcaption}
```



(a) PeP-Logo.



(b) Das TU-Logo.

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

## Subfigures: Code

In  $\text{\TeX}$  wirkt ein Zeilenende wie ein Leerzeichen, dies ist oft unerwünscht und kann durch ein % am Ende der Zeile behoben werden.

### Code

```
\begin{figure}%  
  \begin{subfigure}{0.48\textwidth}%  
    \centering%  
    \includegraphics[height=0.75cm]{logos/pep.pdf}%  
    \caption{PeP-Logo.}%  
    \label{fig:pep2}%  
  \end{subfigure}%  
  \hfill% Fills available space in the center -> space between figures  
  \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.  
In \autoref{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.

In Abbildung 2a sehen Sie das PeP-Logo.

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

`\autoref` erfordert eine Sprachoption für hyperref: `\usepackage[german, ...]{hyperref}`



# 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
    $$$ & $l_{\text{start}}$ & $l_1$ & $l_{\text{kor}}$ \\
    ,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.

| $f$ | $l_{\text{start}}$ | $l_1$ | $l_{\text{kor},1}$ | $B_1$  |
|-----|--------------------|-------|--------------------|--------|
| 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 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.

| $f$ | $l_{\text{start}}$ | $l_1$ | $l_{\text{kor},1}$ | $B_1$  |
|-----|--------------------|-------|--------------------|--------|
| 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}
    \toprule
    & \multicolumn{2}{c}{Technetium} & \multicolumn{2}{c}{Molybdän} & \\
    \cmidrule(lr){2-3}\cmidrule(lr){4-5}
    &  $\lambda$  &  $\mathbin{/}$  &  $\text{unit}\{\text{nano}\text{meter}\}$  & \\
    &  $\phi_1$  &  $\phi_2$  &  $\phi_1$  &  $\phi_2$  & \\
    \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.

| $\lambda / \text{nm}$ | Technetium |          | Molybdän |          |
|-----------------------|------------|----------|----------|----------|
|                       | $\phi_1$   | $\phi_2$ | $\phi_1$ | $\phi_2$ |
| 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}{ $x \ \mathbin{/} \ \text{\unit{\ohm}}$ } \\  
 \midrule  
 663.0 & 12.1 \\  
 670.0 & 10.9 \\  
 678.0 & 9.1 \\  
 684.0 & 8.2 \\  
 \bottomrule  
\end{tabular}
```

## Ergebnis

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

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

# Fußnoten

## Code

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

## Ergebnis

In diesem Versuch werden PMTs<sup>1</sup> eingesetzt.

---

<sup>1</sup>Photo-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.}
```

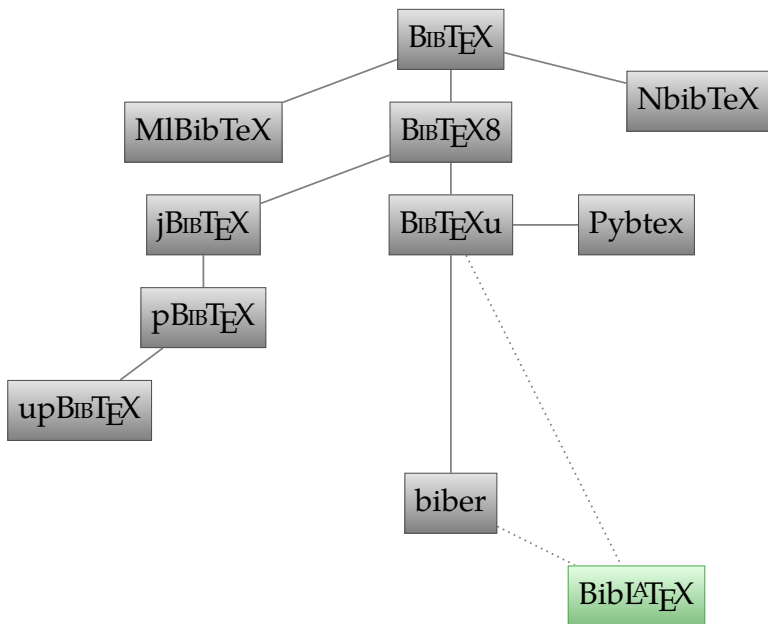
- `\footnotemark` an der Stelle wo die Fußnote sein soll
- In einer `\caption` muss dem `\footnotemark` ein `\protect` vorangestellt werden.
  - Aufpassen, wenn man eine `\listoffigures` hat
- `\footnotetext{...}` außerhalb der Umgebung für den Text der Fußnote
- Fußnoten in Abbildungen sollten vermieden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

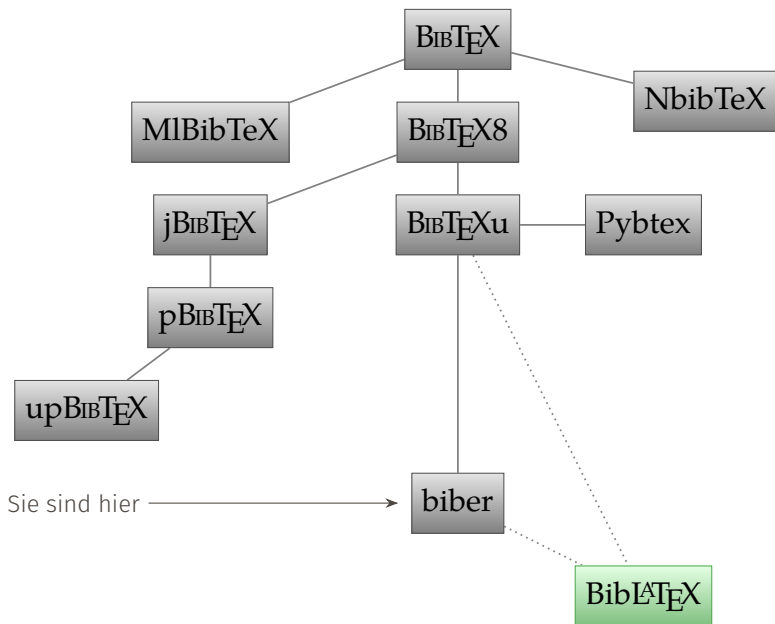
# Literaturverzeichnis

- Wichtiger Teil vieler Dokumente, für wissenschaftliche Texte zwingend
- Bib<sub>La</sub>T<sub>E</sub>X 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}
```







# Warum biber?

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

# .bib-Dateien

- Datenbank eurer Literatur
- Textdateien
- Bib $\text{\TeX}$  definiert viele Eintragstypen und dazugehörige Felder
- Typen haben Pflichtfelder und weitere optionale Felder
- Viele Felder fordern ein bestimmtes Format, z. B. Autorenlisten, Daten
  - Namensliste** Nachname1, Vorname1 and Nachname2, Vorname2 and others
  - Datum** YYYY-MM-DD (ISO8601)
  - Seitenzahlen** X--Y

## Allgemeine Syntax von BibTeX-Einträgen

```
@TYPE{entryname,  
  field1 = {value2},  
  field2 = {value2},  
}
```

Viele Dienste stellen Zitationen im **.bib**-Format zur Verfügung, zum Beispiel:

<https://scholar.google.com>

*Vorsicht: müssen oft noch angepasst werden.*

## Pflichtfelder

title,      year oder date

## Beispiel

```
@manual{fp01,  
  title = {Versuchsanleitung zu Versuch Nr. 01 Lebensdauer der Myonen},  
  year = 2004,  
  organization = {Fakultät Physik, TU Dortmund},  
}
```

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

## Pflichtfelder

author, title, journal, year oder date

## Beispiel

```
@article{photoeffekt,  
  title = {  
    Über einen die Erzeugung und Verwandlung des  
    Lichtes betreffenden Gesichtspunkt  
  },  
  author = {Einstein, Albert},  
  year = 1905,  
  journal = {Annalen der Physik},  
  volume = 322,  
  number = 6,  
  pages = {132--148},  
  doi = {10.1002/andp.19053220607},  
}
```

Albert Einstein. „Über einen die Erzeugung und Verwandlung des Lichtes betreffenden Gesichtspunkt“. In: *Annalen der Physik* 322.6 (1905), S. 132–148. DOI: [10.1002/andp.19053220607](https://doi.org/10.1002/andp.19053220607)

## Pflichtfelder

author, title, year oder date

## Beispiel

```
@book{gelb1,  
  title      = {Das gelbe Rechenbuch 1},  
  subtitle   = {Lineare Algebra, Differentialrechnung},  
  author     = {Furlan, Peter},  
  year       = {1995},  
  isbn       = {3931645002},  
  publisher  = {Verlag Martina Furlan},  
}
```

Peter Furlan. *Das gelbe Rechenbuch 1. Lineare Algebra, Differentialrechnung*. Verlag Martina Furlan, 1995.  
ISBN: 3931645002

## Pflichtfelder

|         |        |              |                 |                       |
|---------|--------|--------------|-----------------|-----------------------|
| author, | title, | institution, | year oder date, | type (nur für thesis) |
|---------|--------|--------------|-----------------|-----------------------|

## Beispiel

```
% or mastersthesis, or thesis with field "type"
@phdthesis{phd_mnoethe,
  title      = {Monitoring the High Energy Universe},
  subtitle   = {
    Open, reproducible, machine learning based analysis
    for the First G-APD Cherenkov Telescope
  },
  author     = {Nöthe, Maximilian},
  year      = {2020},
  institution = {TU Dortmund},
  doi       = {10.17877/DE290R-21143},
}
```

Maximilian Nöthe. „Monitoring the High Energy Universe. Open, reproducible, machine learning based analysis for the First G-APD Cherenkov Telescope“. Diss. TU Dortmund, 2020. doi: [10.17877/DE290R-21143](https://doi.org/10.17877/DE290R-21143)

## Pflichtfelder

author, title, booktitle, year oder date

## Beispiel

```
@inproceedings{hillas,  
  title      = {Cerenkov light images of EAS produced by primary gamma},  
  author     = {Hillas, A. Michael},  
  booktitle  = {  
    Proceedings from the 19th International Cosmic Ray Conference  
  },  
  volume    = {3},  
  pages     = {445--448},  
  year      = {1985},  
}
```

A. Michael Hillas. „Cerenkov light images of EAS produced by primary gamma“. In: *Proceedings from the 19th International Cosmic Ray Conference*. Bd. 3. 1985, S. 445–448

## Pflichtfelder

`title`, `url` oder `doi` oder `eprint`

## Beispiel

```
@online{toolbox-dualboot,  
  url      = {https://toolbox.pep-dortmund.org/install/dualboot.html},  
  urldate  = {2020-10-07},  
  title    = {Installation eines Dual-Boot Systems},  
  author   = {PeP et al.},  
  year     = {2020},  
}
```

PeP et al. *Installation eines Dual-Boot Systems*. 2020. URL:  
<https://toolbox.pep-dortmund.org/install/dualboot.html> (besucht am 07.10.2020)



## Pflichtfelder

title, year oder date

## Beispiel

```
@software{python,  
  title   = {Python},  
  url     = {https://www.python.org},  
  version = {3.9.0},  
  date    = {2020-10-05},  
}
```

*Python*. Version 3.9.0. 5. Okt. 2020. URL: <https://www.python.org>

# Zu Software gehörige Veröffentlichungen

Viele Programmbibliotheken haben auch wissenschaftliche Veröffentlichungen, die zusätzlich zitiert werden sollten:

Zum Beispiel: <https://www.scipy.org/citing.html>

## Beispiel

```
@article{numpy,  
  author = {  
    Harris, Charles R. and Millman, K. Jarrod and van der Walt, Stéfan J and Gommers, Ralf  
    and Virtanen, Pauli and Cournapeau, David and Wieser, Eric and Taylor, Julian  
    and Berg, Sebastian and Smith, Nathaniel J. and Kern, Robert and Picus, Matti  
    and Hoyer, Stephan and van Kerkwijk, Marten H. and Brett, Matthew and Haldane, Allan  
    and Fernández del Río, Jaime and Wiebe, Mark and Peterson, Pearu  
    and Gérard-Marchant, Pierre and Sheppard, Kevin and Reddy, Tyler and Weckesser, Warren  
    and Abbasi, Hameer and Gohlke, Christoph and Oliphant, Travis E.  
  },  
  title = {Array programming with {NumPy}},  
  journal = {Nature},  
  year = {2020},  
  volume = {585},  
  pages = {357--362},  
  doi = {10.1038/s41586-020-2649-2}  
}
```

Charles R. Harris u. a. „Array programming with NumPy“. In: *Nature* 585 (2020), S. 357–362. DOI: 10.1038/s41586-020-2649-2

# Closed Access und (Pr)e-prints

Viele wissenschaftliche Artikel sind leider nicht öffentlich einsehbar.

Für manche gibt es sogenannte E-Prints oder Pre-Prints, zum Beispiel auf dem arXiv.

## Beispiel

```
@article{higgs,  
  title = {  
    Observation of a new particle in the search for the  
    Standard Model Higgs boson with the ATLAS detector at the LHC  
  },  
  author = {Aad, G. and others},  
  journal = {Physics Letters B},  
  volume = {716},  
  number = {1},  
  doi = {10.1016/j.physletb.2012.08.020},  
  publisher = {Elsevier BV},  
  date = {2012-09},  
  eprint = {1207.7214},  
  eprinttype = {arxiv},  
  eprintclass = {hep-ex},  
}
```

G. Aad u. a. „Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC“. In: *Physics Letters B* 716.1 (Sep. 2012). DOI: [10.1016/j.physletb.2012.08.020](https://doi.org/10.1016/j.physletb.2012.08.020). arXiv: [1207.7214](https://arxiv.org/abs/1207.7214) [[hep-ex](#)]

## Benötigte Pakete

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

## Zitieren

```
\cite{numpy}  
\cite[20]{numpy}  
\cite[1--3]{numpy}  
\cite{gelb1, fp01}
```

## Ergebnis

```
[5]  
[5, S. 20]  
[5, S. 1–3]  
[4, 9]
```

## Verzeichnis ausgeben

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



???

Die Idee ist:

1. Bib<sub>La</sub>T<sub>E</sub>X 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<sub>La</sub>T<sub>E</sub>X liest `.bbl`, gibt Verzeichnis aus

Also:

Aufrufe mit Literaturverzeichnis

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

- [1] G. Aad u. a. „Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC“. In: *Physics Letters B* 716.1 (Sep. 2012). DOI: [10.1016/j.physletb.2012.08.020](https://doi.org/10.1016/j.physletb.2012.08.020). arXiv: [1207.7214](https://arxiv.org/abs/1207.7214) [[hep-ex](#)].
- [2] PeP et al. *Installation eines Dual-Boot Systems*. 2020. URL: <https://toolbox.pep-dortmund.org/install/dualboot.html> (besucht am 07.10.2020).
- [3] Albert Einstein. „Über einen die Erzeugung und Verwandlung des Lichtes betreffenden Gesichtspunkt“. In: *Annalen der Physik* 322.6 (1905), S. 132–148. DOI: [10.1002/andp.19053220607](https://doi.org/10.1002/andp.19053220607).
- [4] Peter Furlan. *Das gelbe Rechenbuch 1. Lineare Algebra, Differentialrechnung*. Verlag Martina Furlan, 1995. ISBN: 3931645002.
- [5] Charles R. Harris u. a. „Array programming with NumPy“. In: *Nature* 585 (2020), S. 357–362. DOI: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2).
- [6] A. Michael Hillas. „Cerenkov light images of EAS produced by primary gamma“. In: *Proceedings from the 19th International Cosmic Ray Conference*. Bd. 3. 1985, S. 445–448.
- [7] Maximilian Nöthe. „Monitoring the High Energy Universe. Open, reproducible, machine learning based analysis for the First G-APD Cherenkov Telescope“. Diss. TU Dortmund, 2020. DOI: [10.17877/DE290R-21143](https://doi.org/10.17877/DE290R-21143).
- [8] *Python*. Version 3.9.0. 5. Okt. 2020. URL: <https://www.python.org>.
- [9] *Versuchsanleitung zu Versuch Nr. 01 Lebensdauer der Myonen*. Fakultät Physik, TU Dortmund. 2004.



- 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 Grauteil

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

$\frac{1}{2}$   
 $\frac{\pi}{2}$

# Geschützte Leerzeichen

Es gibt Leerzeichen, an denen nicht umgebrochen werden soll.

- Zwischen Titel und Name
- Bei Referenzen
- Bei Datumsangaben
- Zweiteilige Ortsnamen
- Zweiteilige Abkürzungen (kleines Leerzeichen)
- Zwischen Zahl und Einheit (→ `siunitx`)

Dafür gibt es die Tilde `~` (normaler Abstand) und `\,` (kleiner Abstand).

## Code

```
Prof.~Dr.~Dr.~Rhode  
Abbildung~\ref{fig:peplogo}  
2.~Oktober~2014  
St.~Helena  
z.\,B.\  
\qty{3}{\newton\meter}
```

## Ergebnis

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

Es gibt vier verschiedene Striche:

Code

```
- $-$ -- ---
```

Ergebnis

```
- - - -
```

- **Bindestrich** → Bindestrich  
→ zwischen Doppelnamen der selben Person  
Levi-Civita-Symbol

- **Halbgeviertstrich (en-dash)** → Gedankenstrich (wird mit Leerzeichen abgetrennt)  
Text -- oh, Gedankenstriche -- Text  
→ zwischen Namen von versch. Personen  
Maxwell--Boltzmann-Verteilung

– **Geviertstrich (em-dash)** → nicht im Deutschen genutzt, Gedankenstrich im Englischen  
text---oh, em-dashes---text



# Trennung bei Strichen

## Benötigte Pakete

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

Falls ein Wort Striche enthält, trennt  $\TeX$  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  $\LaTeX$  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

## Makros

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

## Code

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

\ExplSyntaxOn

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

\ExplSyntaxOff
```

## Erklärung

experimental  $\LaTeX$ 3

bequeme Syntax für Definition von Befehlen

Befehl  $\backslash\text{I}$  definieren, keine Argumente  
Ergebnis von  $\backslash\text{I}$

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{\\symup{d} #1}
}
```

Code

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

Ergebnis

$$\int dx d^2 \mathbf{y} x^2 |\mathbf{y}| \quad (9)$$

Das Prinzip gilt auch für  $Dx$ ,  $\delta x$ ,  $\Delta x$ .  
Dabei sind  $D$ ,  $\delta$ ,  $\Delta$  gerade, weil sie keine Variablen sind.

Code

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

Ergebnis

$dx Dx \delta x \Delta x$

## Beispiel: \v

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

Code

```
\v{ä}
\begin{equation}
  \int^{} \dif{x} \dif{{}^2 \v{y}} x^2 |\v{y}|
\end{equation}
```

Ergebnis

ä

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

## Beispiel: Umgebung

```
\NewDocumentEnvironment {CenterStrip} {0{\textwidth} m}
{
  \begin{minipage}[c][#2\baselineskip][c]{#1}           % Code für \begin
}{
  \end{minipage}                                       % Code für \end
  \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

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, bfcalf}, StylisticSet=1]
```

`\Re` und `\Im` sehen nicht aus, wie erwartet:

Code

```
\Re z    \Im z
```

Ergebnis

$\Re z$     $\Im z$

```
\AtBeginDocument{ % wird bei \begin{document} ausgeführt  
  % werden sonst wieder von unicode-math überschrieben  
  \RenewDocumentCommand \Re {} {\operatorname{Re}}  
  \RenewDocumentCommand \Im {} {\operatorname{Im}}  
}
```

Besser:

Code

```
\Re z    \Im z
```

Ergebnis

$\operatorname{Re} z$     $\operatorname{Im} z$

# 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

- Mit dem `mathtools`-Befehl `\DeclarePairedDelimiter` können Befehle erzeugt werden, die Symbole um Ausdrücke setzen
- Automatische \*-Variante, die mitwächst
- Automatisch richtiges 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

$$|x| \quad \left|\frac{1}{x}\right|$$
$$\|\mathbf{y}\|$$
$$\sin\left|\frac{1}{2}\right|$$
$$\sin\left|\frac{1}{2}\right|$$

# \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

```
⟨Ψ|
|Ψ⟩
⟨Ψ1|Ψ2⟩
```

# \delimitershortfall

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

`\texorpdfstring`

## Motivation

- `hyperref` bezieht den Text für seine Bookmarks, z.B. aus `\section{ }`
- Bookmarks mögen meist keine  $\text{\TeX}$ -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}
```

## Beispiel

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

# Fortgeschritten

## Links

# Links und Mail-Adressen

- Es ist sehr empfehlenswert, auf der Titelseite eure Mailadressen anzugeben!
- `\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

## Makefiles

# build-Ordner

Lua $\TeX$  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
biber build/file.bcf
```

- `TEXINPUTS`, `BIBINPUTS`: Suchpfade für  $\TeX$ - und `.bib`-Dateien
- Elemente getrennt mit `:`, der erste Treffer wird genommen (wie `PATH`)
- Hilfreich um z. B. den Header nur einmal für alle Protokolle abzuspeichern. (Siehe `latex-template`)
- `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
- Endet ein Element mit `//`, werden auch alle Unterordner durchsucht

# nonstopmode

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
```

# Fortgeschritten

Breites



## \OverfullCenter

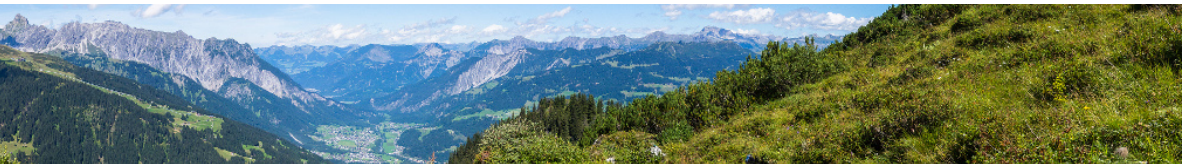
```
\includegraphics[width=\textwidth+15pt]{figures/Panorama.jpg}
```



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

Ist hier natürlich etwas merkwürdig, da Beamer-Slides bereits im Landscape-Modus sind ...

⟨insert wide table here⟩

tlmgr

- TeX-Live kommt mit einem eigenen Verwaltungstool
- Neue Pakete installieren, updaten, suchen, ...

## Ein neues Paket installieren

Wenn man die Dokumentation auch haben möchte, `--with-doc` nutzen:

```
tlmgr install [--with-doc] <name>
```

## Welches Paket muss ich installieren?

```
tlmgr search --global --file booktabs.sty
```

Findet heraus, welches Paket eine bestimmte Datei zur Verfügung stellt.  
Hilfreich bei Fehlermeldungen wie `booktabs.sty not found`.

## Updates installieren

```
tlmgr update --self --all --forcibly-removed
```

Besonders wichtig, wenn man nicht das ganze TeX-Live installiert hat, um Platz zu sparen.

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 $\text{\LaTeX}$  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: \
  max_print_line=1048576 \
  latexmk \
    --lualatex \
    --output-directory=build \
    --interaction=nonstopmode \
    --halt-on-error \
  file.tex

FORCE:

.PHONY: FORCE all clean
```

- 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

$\TeX$  in `matplotlib` in  $\TeX$

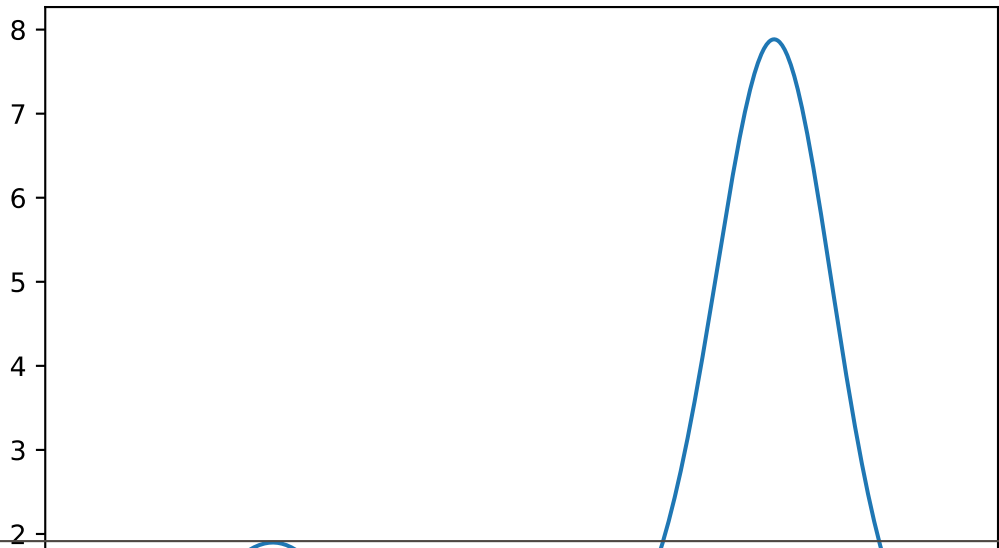
```
import matplotlib.pyplot as plt
import numpy as np
```

```
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')
```

# Ergebnis (1)



```
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, h_pad=1.08, w_pad=1.08)
plt.savefig('build/figures/mattex2.pdf')
```

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

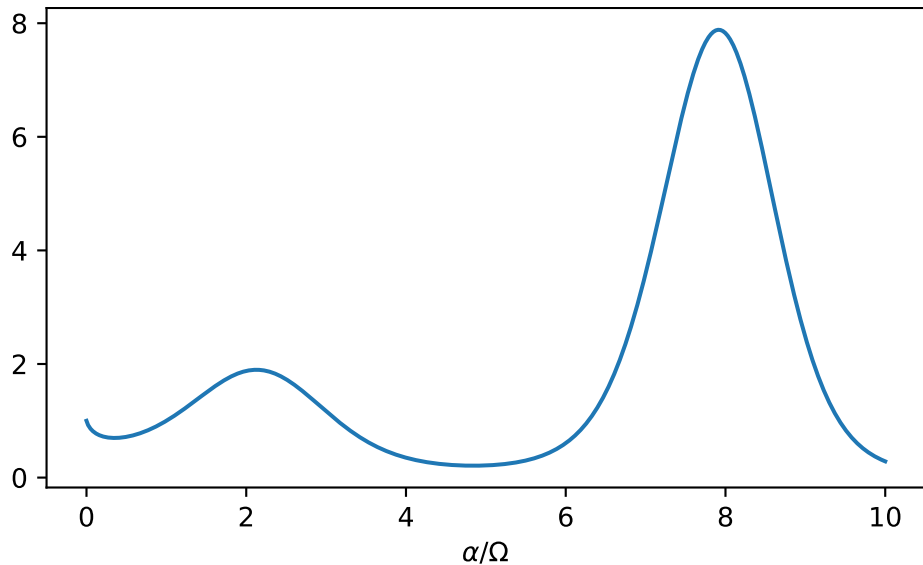
- Größe der Zeichenfläche setzen (in Zoll)
- Breite des Textes kann mit `\the\textwidth` ins Dokument geschrieben werden
- 1 in = 72,27 pt
- Goldener Schnitt für Höhe
- Für `scrartcl` mit Standardeinstellungen: 5.78, 3.57

```
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)



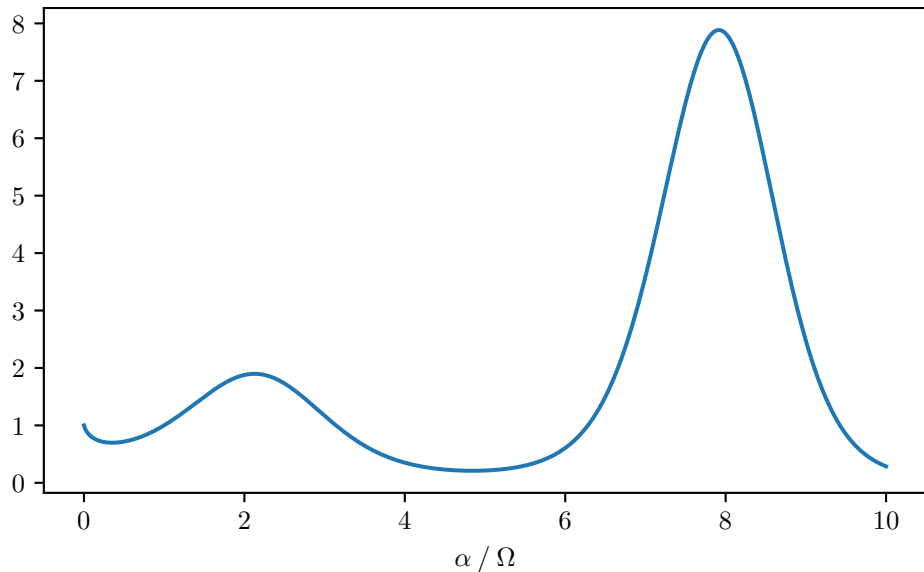
```
import matplotlib as mpl
mpl.use('pgf')
import matplotlib.pyplot as plt
import numpy as np
mpl.rcParams.update({
    'font.family': 'serif',
    'text.usetex': True,
    'pgf.rcfonts': False,
    'pgf.texsystem': 'luatex',
    'pgf.preamble': r'\usepackage{unicode-math}\usepackage{siunitx}',
})

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'$\alpha \mathbin{/} \unit{\ohm}$')

plt.tight_layout(pad=0, h_pad=1.08, w_pad=1.08)
plt.savefig('build/figures/mattex3.pdf')
```



## 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': 'lualatex',
    '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'$\alpha \mathbin{/} \text{\unit{\ohm}}$')

plt.tight_layout(pad=0, h_pad=1.08, w_pad=1.08)
plt.savefig('build/figures/mattex4.pdf')
```

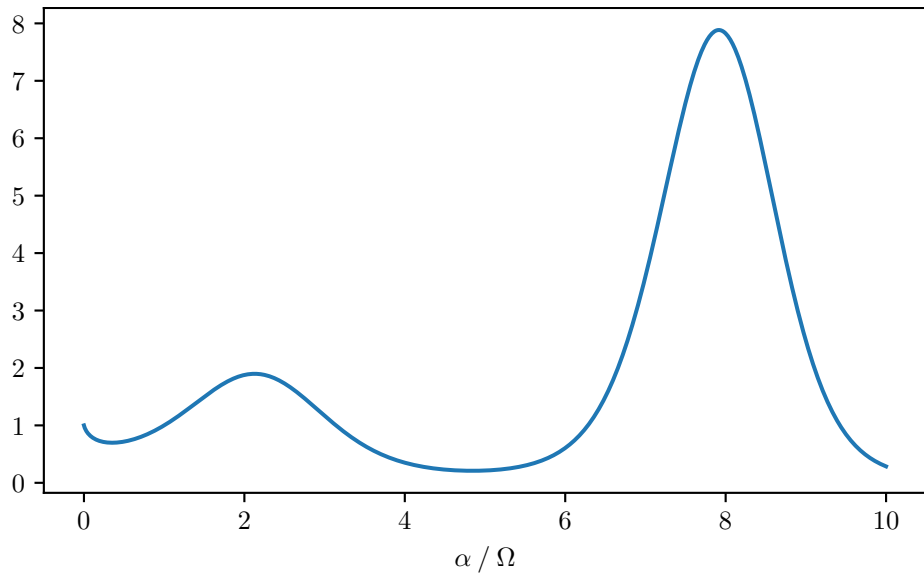
```
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{mathtools}
\usepackage{fontspec}
\usepackage[
  math-style=ISO,
  bold-style=ISO,
  sans-style=italic,
  nabla=upright,
  partial=upright,
]{unicode-math}
\setmathfont{Latin Modern Math}
\usepackage[
  per-mode=reciprocal,
]{siunitx}
```

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

```
build/document.pdf: ...  
    TEXINPUTS=build: ...
```

```
build/figures/mattex4.pdf: script/mattex4.py  
    TEXINPUTS=$(pwd): python script/mattex4.py
```

## Ergebnis (4)



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

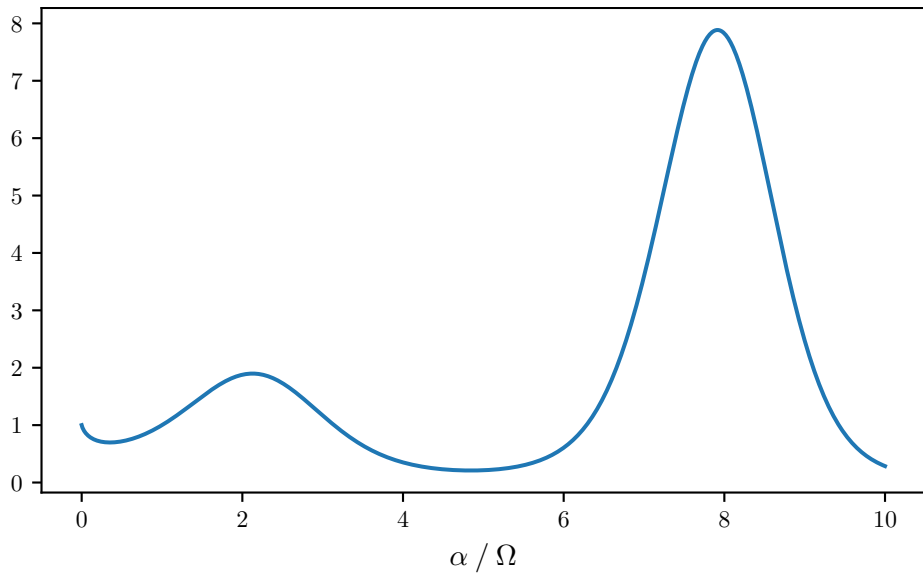
x = np.linspace(0, 10, 1000)
y = x ** np.sin(x)

plt.plot(x, y)
plt.xlabel(r'$\alpha \mathbin{/} \text{unit}\{\text{ohm}\}$')
# in matplotlibrc leider (noch) nicht möglich
plt.tight_layout(pad=0, h_pad=1.08, w_pad=1.08)
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
xtick.labelsize : 9
ytick.labelsize : 9
pgf.rcfonts : False
text.usetex : True
pgf.texsystem : lualatex
pgf.preamble : \input{header-matplotlib.tex}
```

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

## Ergebnis (5)





# Präsentationen mit $\text{\LaTeX}$ : **beamer**

- Dokumentenklasse für Präsentationen
- `frame`-Umgebung erzeugt eine Folie
- Bei Nutzung mit `fontspec` und `unicode-math` muss das `fonttheme professionalfonts` genutzt werden.
- Aussehen wird durch „themes“ gesteuert
- Viele themes werden mit T<sub>E</sub>X-Live mitgeliefert
- Sehen leider alle fast gleich aus
- Alternativen: z. B. `mtheme`

# Minimal-Beispiel

```
\documentclass[aspectratio=1610]{beamer}
\usefonttheme{professionalfonts}
\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}
```

# Mehrere Spalten

- `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}
```

# Zeichnen mit Tikz

## Benötigte Pakete

```
\usepackage{tikz}
```

- Tikz ist kein Zeichenprogramm
- Zeichnen mit Befehlen
  - Sehr präzise (Kleinste Einheit in  $\TeX \approx 5 \text{ nm}$ )
  - programmierfähig
  - automatisierbar
  - Versionskontrolle!
- Extrem umfangreiche Doku mit *zahlreichen* Beispiel (>1 000 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}
```

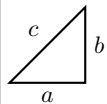
## Ergebnis



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



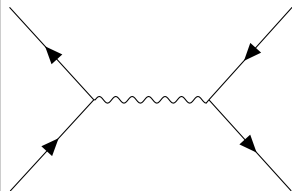
## Benötigte Pakete

```
\usepackage{tikz-feynman}
```

## Tree-Graph

```
\feynmandiagram [horizontal=a to b] {  
  i1 -- [fermion] a -- [fermion] i2,  
  a -- [photon] b,  
  f1 -- [fermion] b -- [fermion] f2,  
};
```

## Ergebnis



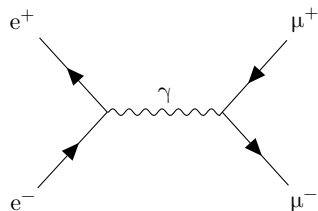
## Tree-Graph

```

\feynmandiagram [horizontal=a to b] {
  i1 [particle=\symup{e}^-]
    -- [fermion] a
    -- [fermion] i2 [particle=\symup{e}^+],
  a -- [photon, edge label=\symup{\gamma}] b,
  f1 [particle=\symup{\mu}^-]
    -- [fermion] b
    -- [fermion] f2 [particle=\symup{\mu}^+],
};

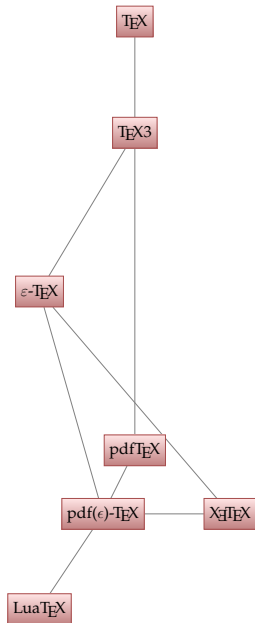
```

## Ergebnis

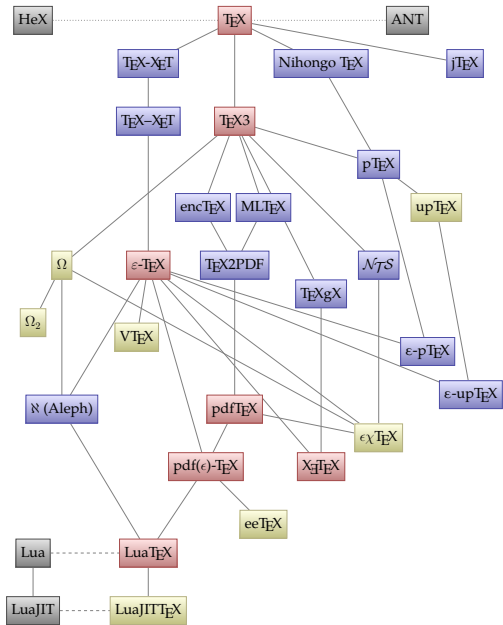


# Ausblick

T<sub>E</sub>X-Engines

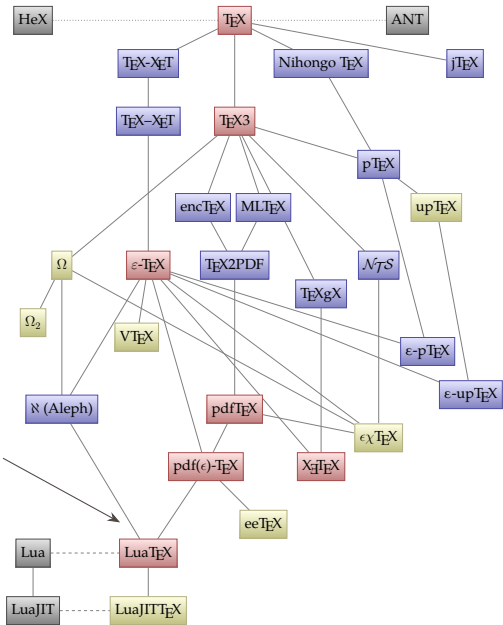


# TeX-Engines



T<sub>E</sub>X-Engines

Sie sind hier





**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

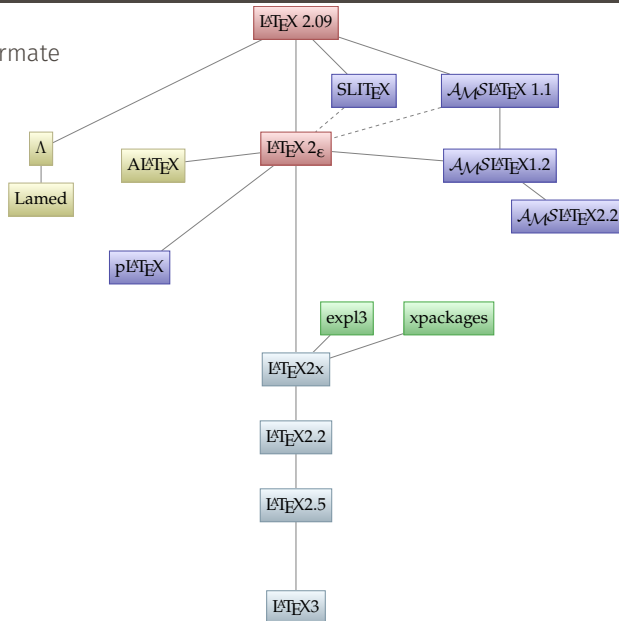
**Lua-Programmierung** → T<sub>E</sub>X-Programmierung ist nicht besonders einfach  
→ Manche Pakete bieten weitergehende Funktionen nur über Lua

T<sub>E</sub>X-Formate

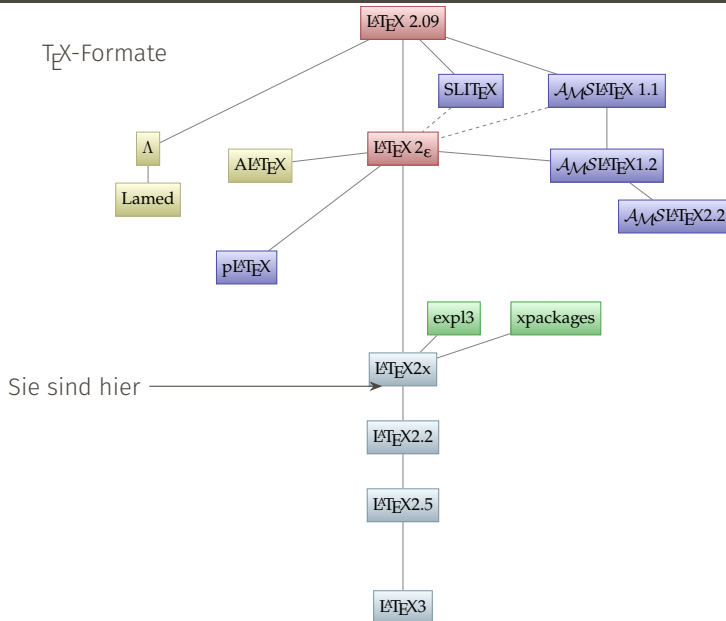
L<sup>A</sup>T<sub>E</sub>X 2.09

L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>

T<sub>E</sub>X-Formate



T<sub>E</sub>X-Formate



Sie sind hier

- $\LaTeX$ 3 existiert (noch) nicht
- `expl3` ist  $\LaTeX$ 3 unter  $\LaTeX$ 2 <sub>$\epsilon$</sub>
- `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`

**scrletter2** Briefe

**MusiXTeX, Lilypond** Notensatz

**IEEEtrantools** Mächtigere Matheumgebungen

**Poster** beamerposter, tcolorbox

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

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

```
\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?