

Verfassen wissenschaftlicher Texte mit
L^AT_EX

Übersicht

Umfrage

Einführung

Grundlagen

Text erstellen

Error

Aufzählungen

Struktur

Formelsatz

Mathe-Umgebungen

Zahlen und Einheiten

Chemische Formeln

Fortgeschrittener Formelsatz

Gleitumgebungen

Tabellen

Fußnoten

Literaturverzeichnis

Übersicht

Fortgeschritten

Ein bisschen Typografie

`\texorpdfstring`

Links

Makros

Makefiles

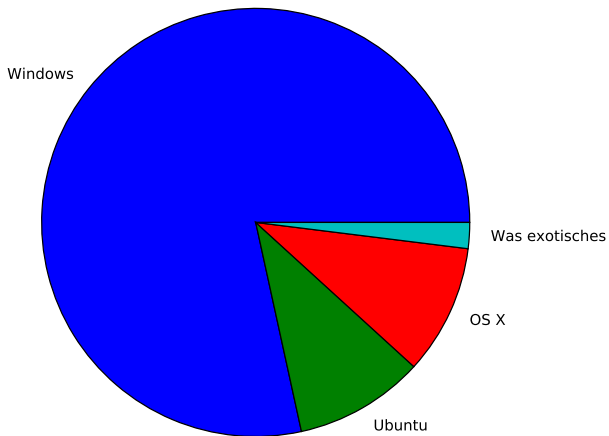
Mathe-Fonts

Breites

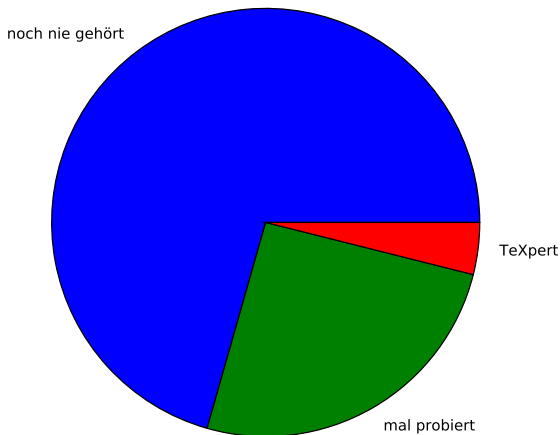
Ausblick

Umfrage

Betriebssystem



Erfahrung mit LaTeX



Einführung

Was ist L^AT_EX?

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

Warum L^AT_EX?

- 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

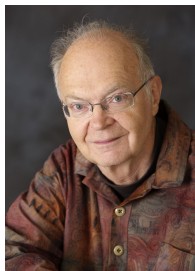
Warum L^AT_EX?

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

Geschichte

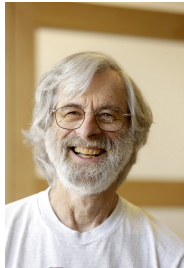
T_EX:

- Geschrieben von Donald E. Knuth 1978, um sein Buch „The Art of Computer Programming“ zu setzen
- Auf Aussprache achten!
- Version (2014): 3.14159265 → π
- Viele Erweiterungen: ϵ -T_EX, pdfT_EX, X_YT_EX, LuaT_EX



L^AT_EX:

- Geschrieben von Leslie Lamport 1984
- Version (1994): L^AT_EX 2 _{ϵ}
- L^AT_EX3 seit Anfang der Neunziger in Arbeit...



Dieser Kurs

- In L^AT_EX 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

Begriffe

T_EX-Engine Implementierung von T_EX, wird als Programm ausgeführt
T_EX-Format Paket, welches standardmäßig geladen wird, z.B. L^AT_EX

Eine Kombination davon ist oft ein neues Programm.

Beispiel: `dvilualatex` = LuaT_EX + L^AT_EX + DVI-Output (statt PDF)

Grundlagen

Das Dokument

Diese drei Zeilen braucht jedes L^AT_EX-Dokument:

Code

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

\documentclass

Dokumentenklasse=Vorlage wählen,
mit Optionen anpassen.

Präambel

Globale Optionen und zusätzliche
Pakete.

document-Umgebung

Inhalt des Dokuments.

Syntax: Befehle

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

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}
```

Syntax: Umgebungen

- Können weitere Umgebungen enthalten
- Diese müssen aber in 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{fixltx2e}
\usepackage[aux]{rerunfilecheck}

\usepackage{polyglossia}
\setmainlanguage{german}
\usepackage{fontspec}
% mehr Pakete hier

\usepackage[unicode]{hyperref}

\usepackage{bookmark}
```

Funktion

L^AT_EX 2_ε korrigieren.
Warnung, falls nochmal kompiliert werden muss.

Deutsche Spracheinstellungen.

Für Fonteinstellungen

Für Hyperlinks (z.B. Inhaltsverzeichnis → Kapitel).
Erweiterte Bookmarks im PDF.

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

KOMA-Script-Klassen

Doku: KOMA-Skript

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

Fürs Praktikum empfohlene Klasse

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

Fonteinstellungen

Standardeinstellung sind die Latin-Modern-Fonts.

Latin Modern

```
\usepackage{fontspec}
```

Alternativ: Tex Gyre

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

- Jede System-Schriftart kann genutzt
- Das ist i.A. nicht sinnvoll: *Hallo Welt in Comic Sans MS!*
- Schriften müssen zueinander passen
- Schriften müssen alle benötigten Sonderzeichen enthalten
- Bei Änderung auch Mathefont anpassen → später in 8

Gerüst

```
\documentclass{scrartcl}

\usepackage{fixltx2e}
\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 L^AT_EX-Kompiler, die verschiedene Ausgabeformate erzeugen können. Der modernste Compiler, der PDF-Dateien erstellt, ist `lualatex`.

L^AT_EX-Dokument kompilieren

Terminal öffnen:

```
lualatex 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.

texdoc

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

Text schreiben

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}
```

Konventionen für Text

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

Sonderzeichen

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

Code

```
\% \& \_ \textbackslash \$ \{ \}
```

Ergebnis

```
% & _ \ $ { }
```

Textauszeichnung

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.

Schriftgrößen

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.

Inhalt auslagern

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 L^AT_EX-Code (header, Erläuterungen zu Fehlerrechnung, ...)
- Für per Skript erzeugte Tabelleninhalte

Anführungszeichen

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

Errors

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

Code

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

```
(/usr/local/texlive/2014/texmf-dist/tex/generic/oberdiek/gettitlestring.sty)
(./test.out) (./test.out)
! Undefined control sequence.
l.16 Ich begrüße euch mit einem \enquote
                                {Hallo Welt}.
? █
```

Errors

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

Code

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

```
(/usr/local/texlive/2014/texmf-dist/tex/generic/oberdiek/gettitlestring.sty))  
./test.out) ./test.out)  
! Undefined control sequence.  
l.16 Ich begrüße euch mit einem \enquote  
                                {Hallo Welt}.  
? █
```

⇒ Vergessen csquotes zu laden.

Lösungsstrategien

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

Aufzählungen

Aufzählungen: Itemize

- L^AT_EX 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

Aufzählungen: Enumerate

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

Aufzählungen: Description

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

L^AT_EX gut
Word böse

Struktur

Titelseite und Metadaten

L^AT_EX 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
```

Gliederung

L^AT_EX bietet Befehle zum erstellen von Gliederungsebenen. Diese werden automatisch nummeriert und in entsprechend größerer und fetter 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}
```

Inhaltsverzeichnis

Aus den Gliederungselementen kann automatisch das Inhaltsverzeichnis erzeugt werden.

Inhaltsverzeichnis generieren

```
\tableofcontents  
\newpage
```

Formelsatz

Benötigte Pakete

Doku: [amsmath](#)

Doku: [mathtools](#)

Doku: [unicode-math](#)

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

Benötigte Pakete

Doku: `amsmath`

Doku: `mathtools`

Doku: `unicode-math`

```
\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."
```

Benötigte Pakete

Doku: `amsmath`

Doku: `mathtools`

Doku: `unicode-math`

```

\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."

```

Benötigte Pakete

Doku: `amsmath`

Doku: `mathtools`

Doku: `unicode-math`

```

\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

```


\$...\$-Umgebung

Aktiviert den Mathematikmodus im Fließtext.

Code

```
Dies ist eine Variable:  $x$ .
Liste von Variablen  $x$ ,  $y$ ,  $z$ .
Kleine Formel:  $a^2 + b^2 = c^2$ .
Vorsicht Höhe:  $x^{\{2^{\{2^{\{2^2\}}\}}\}}$ 
 $x_{\{2_{\{2_{\{2_2\}}\}}\}}$ 
Mehr Text. Mehr Text. Mehr
 $x$ -Achse,  $x$ - $y$ -Ebene
```

Ergebnis

Dies ist eine Variable: x .
 Liste von Variablen x , y , z .
 kleine Formel: $a^2 + b^2 = c^2$.
 Vorsicht Höhe: $x^{2^{2^2}}$ $x_{2_{2_2}}$
 Mehr Text. Mehr Text. Mehr
 x -Achse, x - y -Ebene

- Leerzeichen werden im Mathe-Modus ignoriert.
- T_EX hat Algorithmen für das richtige Spacing.
- Satzzeichen gehören nicht in die \dots -Umgebung!

Griechisch und mehr

Code

```
\epsilon \theta \kappaappa \pi \rho \sigma \phi
\varepsilon \vartheta \varkappa \varpi \varrho
\varsigma \varphi
\Alpha \Beta \Gamma
\hbar \imath \jmath \ell \wp
\aleph \beth \gimel
\partial \eth \nabla \square \triangle \infty
\diameter \ldots \cdots
```

Ergebnis

$\epsilon \theta \kappa \pi \rho \sigma \phi$
 $\varepsilon \vartheta \varkappa \varpi \varrho$
 $\varsigma \varphi$
 $A B \Gamma$
 $\hbar \iota \jmath \ell \wp$
 $\aleph \beth \lambda$
 $\partial \eth \nabla \square \triangle \infty$
 $\oslash \dots \cdots$

Operatoren und Relationen

Code

```
+ - / \pm \mp \cdot \times
= \simeq \equiv \cong \approx \propto \sim
\coloneq
\to \iff \implies
\mapsto \leadsto
```

Ergebnis

+ - / ± ∓ · ×
= ≈ ≡ ≅ ≈ ∝ ~
:=
→ ↔ ⇒
↦ ↷

Die meisten Relationen lassen sich durch ein n negieren:

Code

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

Ergebnis

≠ ≠̄ ∄ ∄̄

Häufig möchte man etwas über eine Relation schreiben:

Code

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

Ergebnis

! def

Indizes

Code

```
x^2 x_2 x^2
x^10 x^{10}
x' x' x'' x''^2
{}^2 x
x_{min} x_{\text{min}}

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

Ergebnis

x^2 x_2 x^2
 x^{10} x^{10}
 x' x' x'' x''^2
 2x
 x_{min} x_{min}
Fehler
 x^{2^2} x^{2^2}
Fehler
 $x_{\sqrt{3}}$

- Man muss häufig den Index in { } schreiben
- Beim mehrfachen Hochstellen jeweils { } nötig
- Nur wenige Befehle können ohne { } im Index stehen

Akzente

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}
 $\dot{x} \ddot{x} \dddot{x}$
 $x y \overline{xy}$

Aufpassen, wo der Akzent hinkommt:

Code

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

Ergebnis

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

Funktionen

Code

```
x \sin y
x \sin(y)
\cos \tan \exp \ln
\lim_{x \to \infty} x^2
\lim_{\mathclap{x \to \infty}} x^2
```

Ergebnis

$x \sin y$
 $x \sin(y)$
 $\cos \tan \exp \ln$
 $\lim_{x \rightarrow \infty} x^2 \lim_{x \rightarrow \infty} x^2$

Man kann auch eigene Funktionen definieren:

Code

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

% in Präambel
\DeclareMathOperator{\xyz}{xyz} \xyz_i(a)
\DeclareMathOperator*{\Xyz}{Xyz} \Xyz_i(a)
```

Ergebnis

$xyz_i(a)$
 $xyz_i(a)$
 $xyz_i(a)$
 $X_{yz_i}(a)$

Große Operatoren

Code

```
\sum_{i=0}^{\infty} x_i
\prod \bigotimes
\int_0^1 \iiint \oint
\sum\nolimits_0^1 \int\limits_0^1
\setsideset{a^b}{c^d}\sum_{i=0}^n
\sum_{i=1+2+3+4+5+6} x_i
\sum_{\mathclap{i=1+2+3+4+5+6}} x_i
```

Ergebnis

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

$$\prod \bigotimes$$

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

$$\sum_0^1 \int_0^1$$

$$a \sum_{i=0}^n c^d$$

$$\sum_{i=1+2+3+4+5+6} x_i$$

$$\sum_{\mathclap{i=1+2+3+4+5+6}} x_i$$

Fonts

Doku: unicode-math

Code

```
x \alpha \mathup{x \alpha}
\mathbf{x\alpha}
\mathbfsf{x \alpha}
\mathbb{R N 1 0 x}
\mathcal{I A O} \mathbf{cal}{I A O}
\mathfrak{A B c} \mathbf{ffrak}{A B}
```

Ergebnis

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

Spaces

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

Code

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

Ergebnis

```
|
||
|||
|||
||
| |
|  |
|   |
```

Negativer Space um zu viel Platz zu korrigieren:

Code

```
\! % negativer \,
```

Ergebnis

```
||
```

Code

```
~2 ~{\!\! 2}
```

Ergebnis

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

Klammern

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|
```

Ergebnis

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

$$\langle x \rangle |x|$$

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

$$\sin(x)$$

$$\sin(x)$$

$$\sin(x)$$

Symbol-Sammlung

Doku: symbols-a4

Doku: unimath-symbols

Praktischer Link:

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

(Symbol malen und L^AT_EX-Code angezeigt bekommen)

Konventionen: Variablen, Zahlen, Einheiten, Indizes

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

```
\mathup{e}$, \mathup{i}$, \mathup{\pi}$
```
 - Infinitesimales: dx


```
\mathup{d}x$
```
 - Indizes wie „min“ oder „max“


```
x_{\text{min}}
```

Konventionen: Variablen, Zahlen, Einheiten, Indizes

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

Formelsatz

Mathe-Umgebungen

Mathe-Umgebungen

Doku: `amsmath`

- `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

Die equation-Umgebung

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 sollten immer Teil eines vollständigen Satzes sein

Die gather-Umgebung

- Für mehrere Gleichungen
- `\` erzeugt neue Zeile
 - Kein `\` nach letzter 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

Die align-Umgebung

- 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

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

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

Das siunitx-Paket

Doku: siunitx

- Einheiten werden aufrecht gesetzt
 - Zwischen Zahl und Einheit steht ein kleines Leerzeichen \,
 - siunitx stellt Befehle zur Verfügung, die das korrekte Setzen von Zahlen und Einheiten stark vereinfachen
- ⇒ 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 Bruch  
{siunitx}
```

Das siunitx-Paket

Code

```
\num{1.23456} und \num{987654321}
\num{1.2e2}
\si{\newton} = \si{\kilo\gram\metre\per
  \second\squared}
\SI{1.2}{\metre\per\second}
\SI{4.3+-1.2}{\micro\second}
\SI{4.3(12)e-6}{\second}
\si[per-mode=reciprocal]{\kilo\gram
  \meter\per\second\squared}
\si[per-mode=fraction]{\kilo\gram\meter
  \per\second\squared}
\num[output-decimal-marker=.]{1.2}
```

Ergebnis

1,234 56 und 987 654 321
 $1,2 \cdot 10^2$
 N = kg m/s²
 1,2 m/s
 $(4,3 \pm 1,2) \mu\text{s}$
 $(4,3 \pm 1,2) \cdot 10^{-6} \text{ s}$
 kg m s⁻²
 $\frac{\text{kg m}}{\text{s}^2}$
 1.2

Statt +- kann `\pm` verwendet werden.

Chemische Formeln

Chemische Formeln

Benötigte Pakete

```
\usepackage[version=3]{mhchem}
```

Code

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

Ergebnis

H_2O_2
 $^{227}_{90}\text{Th}^+$
 $c_{\text{H}_2\text{O}} = 4184 \text{ J}/(\text{kg K})$
 $^{14}_6\text{C} \longrightarrow ^{14}_7\text{N} + e^- + \bar{\nu}_e$
 $\text{CO}_2 + \text{C} \rightleftharpoons 2 \text{CO}$

Fortgeschrittener Formelsatz

Matrizen

Doku: `amsmath`

Doku: `mathtools`

Matrizen (und damit Vektoren) werden fett geschrieben.

Code

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

Ergebnis

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

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

Code

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

Ergebnis

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

Mehr Möglichkeiten in der Doku.

Fallunterscheidungen

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

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

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

`\intertext`

Code

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

\underbrace

Code

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

Ergebnis

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

`\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_{123} &= \frac{1}{2}x + y \\
g &= \hphantom{\frac{1}{2}}x + y \\
h &= \phantom{\frac{1}{2}}x + y
\end{align*}
```

Ergebnis

$$\begin{aligned}
 f_{123} &= \frac{1}{2}x + y \\
 &\quad \vdots \\
 g &= x + y \\
 f_{123} &= \frac{1}{2}x + y \\
 &\quad \vdots \\
 h &= x + y
 \end{aligned}$$

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

`\vphantom`

Code

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

Ergebnis

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

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

Gleitumgebungen

Gleitumgebungen

Doku: placeins

Doku: caption

- Zum setzen von Elementen, die nicht Fließtext sind
- Hauptsächlich Grafiken und Tabellen
- Position wird von L^AT_EX 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.

Bilder einbinden

Doku: `graphicx`

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
EHEMALIGE PHYSIKSTUDIERENDE
DER TU DORTMUND

Abbildung 1: Das PeP-Logo.

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

Subfigures

Doku: subcaption

Benötigte Pakete

```
\usepackage{subcaption}
```



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

(a) PeP-Logo.



technische universität
dortmund

(b) Das TU-Logo.

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

Subfigures: Code

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}
```

Referenzen

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

`\ref` vs. `\subref`

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

- L^AT_EX 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{float}

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


Tabellen

Tabellen

Doku: booktabs

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
    $f$ & $I_1$ & \text{start}$ & $I_1$ & $I_1$ \\
    \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

Ergebnis

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!

Schönere Tabellen mit siunitx

Doku: siunitx

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}
```

Ergebnis

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}
```

Resultat

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

Fehler in Tabellen

Code

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

Fußnoten

Code

```
In diesem Versuch benutzen wir
PMTs\footnote{Photo-Multiplier-
Tubes}.
```

Ergebnis

In diesem Versuch benutzen wir
PMTs^a.

^aPhoto-Multiplier-Tubes

- Anpassung von Fußnoten mit dem Paket `footmisc`

Fußnoten in Floats

Vorsicht bei Float-Umgebungen!

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

- `\footnotemark` an der Stelle wo die Fußnote sein soll
- Bei `\caption` muss der Inhalt der Caption nocheinmal ohne `\footnotemark` in `[]` gegeben werden
- `\footnotetext{...}` außerhalb der Umgebung für den Text der Fußnote

Literaturverzeichnis

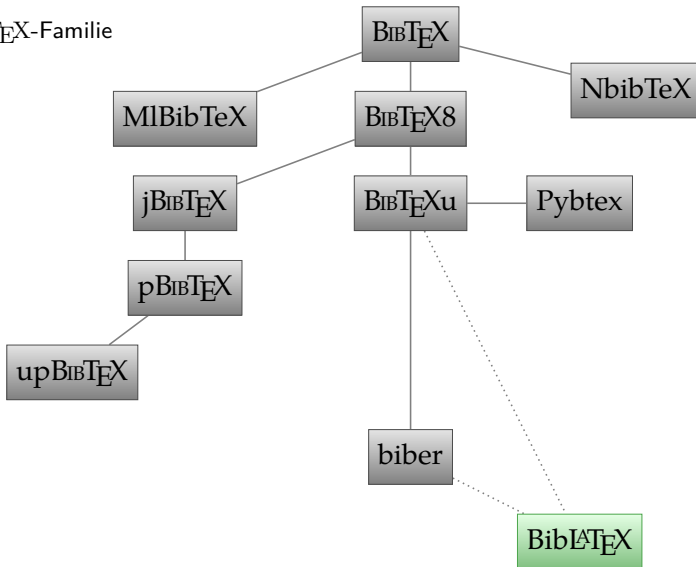
Literaturverzeichnis

- Wichtiger Teil vieler Dokumente, für wissenschaftliche Texte zwingend
- Bib_LA_TE_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

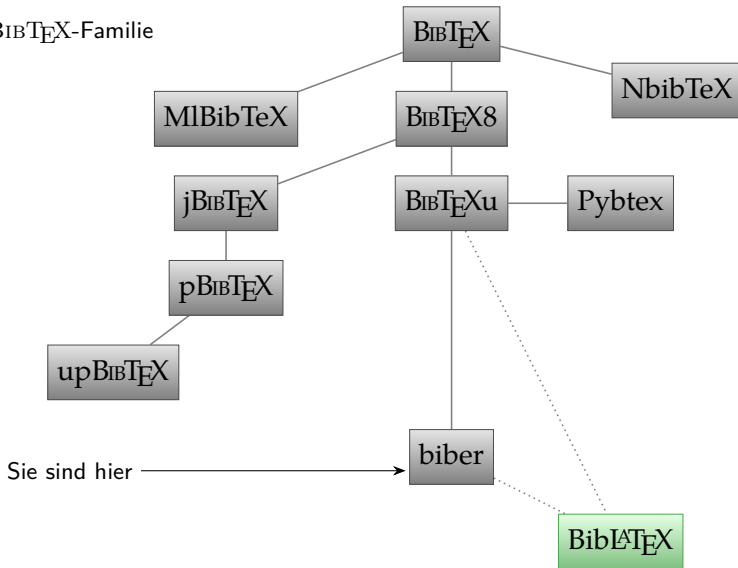
Neue Klassenoption

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

BIB_TE_X-Familie



BIB_TE_X-Familie



Warum biber?

- 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

.bib-Dateien (I)

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

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

.bib-Dateien (II)

```
@article{numpy,  
  author = "Travis E. Oliphant",  
  title = "Python for Scientific Computing",  
  publisher = "IEEE",  
  year = "2007",  
  journal = "Computing in Science & Engineering",  
  volume = "9",  
  number = "3",  
  pages = -"1020",  
  url = "http://link.aip.org/link/?CSX/9/10/1",  
  addendum = "Version 1.8.1"  
}
```

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

.bib-Dateien (III)

```
@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 = -"8186",  
  title = "ROOT - An Object Oriented Data Analysis Framework",  
  volume = 389,  
  year = 1996,  
  addendum = "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/>. Version 5.34.18

.bib-Dateien (IV)

```
@online{splot,  
  author = "Muriel Pivk and Francois R. Le Diberder",  
  title = "sPlot: a statistical tool to unfold data distributions",  
  date = "2005-09-02",  
  archivePrefix = "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](https://arxiv.org/abs/physics/0402083v3)

.bib-Dateien (V)

```
@online{wingate,  
  author = "Zhaofeng Liu and Stefan Meinel and Alistair Hart and  
           Ron R. Horgan and Eike H. Müller and Matthew Wingate",  
  title = "A lattice calculation of  $\text{HepProcess}\{\text{PB} \to \text{PK}^{\{*\}}\}$   
          form factors",  
  date = "2011-01-14",  
  archivePrefix = "arXiv",  
  eprint = "1101.2726v1",  
  primaryClass = "hep-ph"  
}
```

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

BibL^AT_EX

Doku: biblatex

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
```

Literaturverzeichnis

Literaturverzeichnis

???

biber

Doku: biber

Die Idee ist:

- 1 Bib^LA_TE_X erstellt eine Liste der .bib-Dateien und der benötigten Referenzen
→ .bcf-Datei
- 2 biber liest Anweisungen, liest .bib, sucht und sortiert Referenzen
→ .bb1-Datei
- 3 Bib^LA_TE_X liest .bb1, gibt Verzeichnis aus

Also:

Aufrufe mit Literaturverzeichnis

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

Literaturverzeichnis

- [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/>. Version 5.34.18.
- [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](https://arxiv.org/abs/1101.2726v1) [hep-ph].
- [4] Travis E. Oliphant. „Python for Scientific Computing“. In: *Computing in Science & Engineering* 9.3 (2007), S. 10–20. URL: <http://link.aip.org/link/?CSX/9/10/1>. Version 1.8.1.
- [5] Muriel Pivk und Francois R. Le Diberder. *sPlot: a statistical tool to unfold data distributions*. 2. Sep. 2005. arXiv: [physics/0402083v3](https://arxiv.org/abs/physics/0402083v3).

Stile

- 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

Absatzauszeichnung

- 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}
```

microtype

- 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}
```

Schönere Brüche im Text

Benötigte Pakete

```
\usepackage{xfrac}
```

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

Code

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

Ergebnis

1/2
 $\pi/2$

Geschützte Leerzeichen

- 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
```

Striche

Benötigte Einstellung

```
\defaultfontfeatures{Ligatures=TeX} % nach fontspec
```

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 Streckenstrich
1 bis 10 ist 1--10
- Geviertstrich (em-dash)
 - nicht im Deutschen, englischer Gedankenstrich
text---oh, em-dashes---text

Trennung bei Strichen

Doku: `exdash`

Benötigte Pakete

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

Falls ein Wort Striche enthält, trennt L^AT_EX ausschließlich an diesen.
So ermöglicht man mehr Trennung:

Trennbare Striche

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

Maxwell--Boltzmann-Verteilung
Maxwell\--Boltzmann\-/Verteilung

So verhindert man die Trennung an den Strichen:

```
\=/ \== \===
```

`x`\=/Achse

Ergebnis

```
- - -
```

Maxwell-
Boltzmann-
Verteilung
Max-
well-
Boltz-
mann-
Vertei-
lung

Silbentrennung

- Manchmal kann L^AT_EX 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`

`\texorpdfstring`

Motivation

- hyperref bezieht den Text für seine Bookmarks, z.B. aus `\section{ }`
- Bookmarks mögen meist keine L^AT_EX-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{LateXcode}{Unicode-Äquivalent}
```

Code

```
\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

Makros

Eigene L^AT_EX-Kommandos

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

Kommandostruktur

```
\newcommand*\Kommandoname [#Argumente]{Code mit #1, #2, ...}  
\newcommand*\Command [#Argumente] [Default]{Code}  
\newenvironment*{Umgebungsname} [#Argumente]{\begin-Code}{\end-Code}
```

- ## gibt ein wirkliches #
- Ohne *, um mehrere Absätze als Argument übergeben zu können
- `\end`-Code kann Argumente nicht nutzen
- Man sollte neue Befehle nur in der Präambel definieren

Beispiele

Code

```
\newcommand*\I{\mathup{i}}
\newcommand*\t[1]{\text{#1}}
\newcommand*\dd[2]{\frac{\mathup{d}#1}{\mathup{d}#2}}

% Überschreiben bereits belegter Kommandos
\renewcommand*\v[1]{\mathbf{#1}}

\newenvironment*{eqn}{\begin{equation}}{\end{equation}
\ignorespacesafterend}
```

Verwendung

```
\I
\t{foo}
\dd{x}{y}
\v{u}
\begin{eqn}
% .
\end{eqn}
```

Fortgeschritten

Makefiles

build-Ordner

Lua_TE_X 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
```

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
```

Log schöner machen

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

Fortgeschritten

Mathe-Fonts

Alternative Mathe-Fonts

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

Code

```
\mathcal{IA} \mathbf{fcal}{IA} % Latin Modern  
\mathcal{IA} \mathbf{fcal}{IA} % XITS Math, StylisticSet=1  
\mathscr{IA} \mathbf{fscr}{IA} % XITS Math
```

Ergebnis

$\mathcal{I} \mathcal{A} \mathcal{I} \mathcal{A}$
 $\mathcal{I} \mathcal{A} \mathcal{I} \mathcal{A}$
 $\mathcal{I} \mathcal{A} \mathcal{I} \mathcal{A}$

Mathe-Fonts einstellen

```
\setmathfont [range={\mathcal, \mathbfcal}, StylisticSet=1] {XITS Math}  
\setmathfont [range={\mathscr, \mathbfscr}] {XITS Math}
```


Fehlende Zeichen

Manchmal sind Sonderzeichen nicht im Font enthalten:

Code

```
\setmathfont[range={\lblkbrbrak,\rblkbrbrak}]{XITS Math}

% make bar horizontal, use \hslash for slashed h
\let\hbar\relax
\DeclareMathSymbol{\hbar}{\mathord}{AMSb}{"7E}
\DeclareMathSymbol{\h}{\mathord}{AMSb}{"7E} % mit Unicode hbar
```

Fortgeschritten

Breites

`\OverfullCenter`

```
\includegraphics [width=\textwidth+25pt] {figures/Panorama.jpg}
```

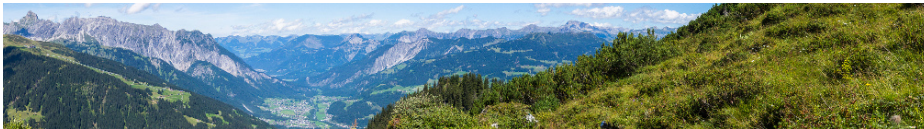


Bild oder Tabelle ist zu Breit, passt aber auf die Seite.
Wie kriegt man es in die Mitte?

```
\OverfullCenter{\includegraphics [width=\textwidth+25pt] {figures/  
Panorama.jpg}}
```



Code

```
\newcommand\OverfullCenter [1] {\noindent\makebox [\linewidth] {#1}}
```

pdflscape

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

Benötigte Pakete

```
\usepackage{pdflscape}
```

Code

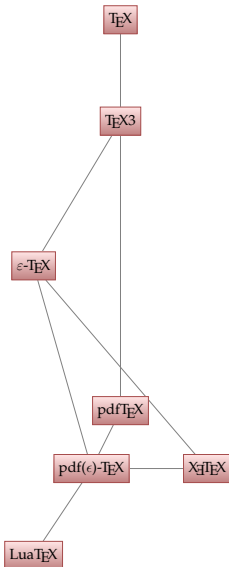
```
\begin{landscape}  
  \begin{table}  
    % .  
  \end{table}  
\end{landscape}
```

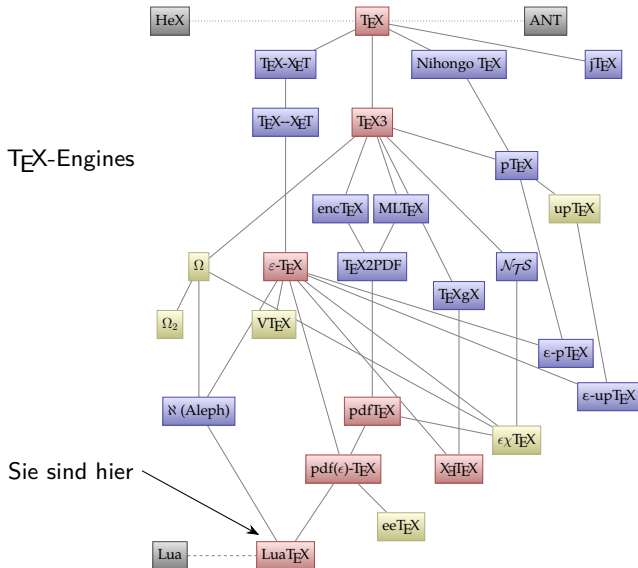
- Inhalt der landscape-Umgebung wird horizontal gesetzt (separate Seite)
- Seite wird im PDF-Reader horizontal angezeigt → schöner zu lesen

⟨insert wide table here⟩

Ausblick

T_EX-Engines





Warum eigentlich Lua_TE_X

- 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 an

T_EX-Formate

L^AT_EX 2.09

L^AT_EX 2_ε

Warum L^AT_EX3?

- L^AT_EX3 existiert (noch) nicht
- exp13 ist L^AT_EX3 unter L^AT_EX2_ε
- xpackages sind Pakete, die auf exp13 aufbauen und neue Möglichkeiten bieten
- xparse macht das schreiben neuer (auch komplizierter) Befehle sehr einfach
- viele Pakete benutzen jetzt schon exp13 und xparse

Möglichkeiten mit L^AT_EX

beamer Präsentationen mit L^AT_EX

scrlettr2 Briefe

MusiXTeX, Lilypond Notensatz

TikZ Grafiken direkt in L^AT_EX erstellen

Poster Verschiedene Pakete

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?