

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

Fortgeschritten

Links

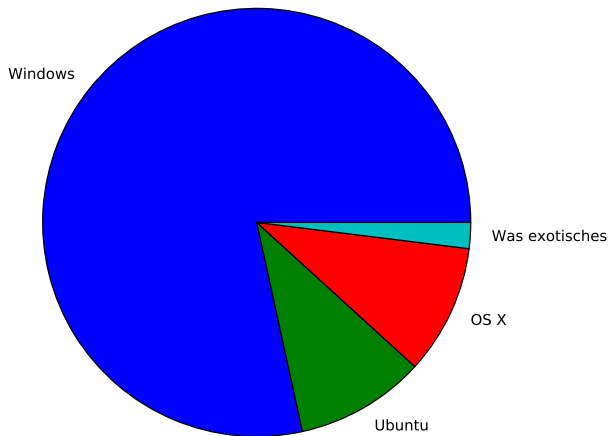
Makros

Tricks

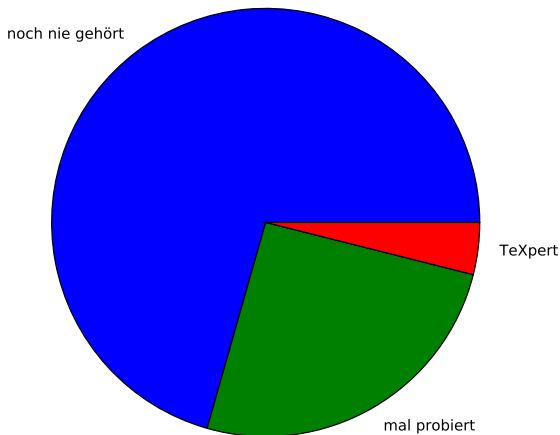
Ausblick

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 Compiler \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

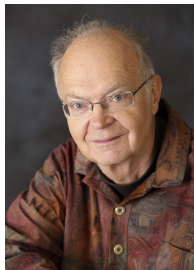
Warum L^AT_EX?

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

Geschichte

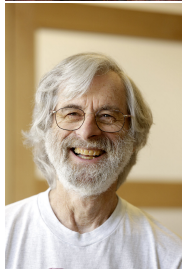
T_EX:

- Geschrieben von Donald E. Knuth 1978, um sein Buch „The Art of Computer Programming“ zu setzen
- Auf Aussprache achten!
- Version (2014): 3.14159265 $\rightarrow \pi$
- 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}

\frac{1}{2}

% Kommentar
```

Erklärung

Dokumentenklasse scrartcl,
Papierformat DIN A4.
Zwei oder mehr Pflichtargumente,
z.B. der Bruch.
%-Zeichen für Kommentare.

Syntax: Umgebungen

- Einstellungen für Bereich des Dokuments
- extrem vielseitig
- Können ggfs. auch Optionen übergeben bekommen

Syntax

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

Beispiel

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


Syntax: Umgebungen

Umgebungen können weitere Umgebungen enthalten, diese müssen aber in dieser 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	Funktion
<code>\usepackage{fixltx2e}</code> <code>\usepackage[aux]{rerunfilecheck}</code>	L ^A T _E X 2 _ε korrigieren. Warnung, falls nochmal kompiliert werden muss.
<code>\usepackage{polyglossia}</code> <code>\setmainlanguage{german}</code> <code>\usepackage{fontspec}</code> % mehr Pakete hier	Deutsche Spracheinstellungen. Für Fonteinstellungen
<code>\usepackage[unicode]{hyperref}</code> <code>\usepackage{bookmark}</code>	Für Hyperlinks (z.B. Inhaltsverzeichnis → Kapitel). Erweiterte Bookmarks im PDF.

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

KOMA-Script-Klassen

Doku: KOMA-Skript

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

Fürs Praktikum empfohlenene 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.

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 drei Umgebungen für Aufzählungen
- Unnummerierte Listen: `itemize`
- Verschachteln

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

Anpassung der Listen mit dem Paket `enumitem`.

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 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}
```

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."
```

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

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} % standard
% \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: $x^{2^{2^{2}}}$

$x_{2_{2_{2}}}$

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 \kappa \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 \increment \infty
\diameter \ldots \cdots
```

Ergebnis

$\epsilon \theta \kappa \pi \rho \sigma \phi$
 $\varepsilon \vartheta \varkappa \varpi \varrho$
 $\varsigma \varphi$
 $\Alpha \Beta \Gamma$
 $\hbar \imath \jmath \ell \wp$
 $\aleph \beth \gimel$
 $\partial \eth \nabla \square \Delta \infty$
 $\oslash \dots \cdots$

Operatoren und Relationen

Code

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

Ergebnis

+ - / \pm \mp \cdot \times
 $=$ \simeq \equiv \cong \approx \propto \sim
 $:=$
 \rightarrow \leftrightarrow \Rightarrow
 \mapsto \leadsto

Die meisten Relationen lassen sich durch ein `n` negieren:

Code

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

Ergebnis

\neq \nsimeq \nexists \nni

Häufig möchte man etwas über eine Relation schreiben:

Code

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

Ergebnis

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

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} \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}
\hat{x}_{\text{min}} \hat{x}_{\text{min}}
\underline{xy} \overline{xy}
```

Ergebnis

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

Funktionen

Code

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

Ergebnis

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

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

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

Große Operatoren

Code

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

\prod \bigotimes

\int_0^1 \iiint \oint

\sum\nolimits_0^1 \int\limits_0^1

\sideset{_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

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

$$\prod \bigotimes$$

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

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

$$b \sum_{i=0}^n d$$

$$a \sum_c^d$$

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

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

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}
\mathfrak{A B c} \mathbfrak{A B}
```

Ergebnis

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

Spaces

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

Code

```
%  
\,  
\:  
\;  
\quad  
\qquad  
\! % negativer \,
```

Ergebnis

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

Code

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

Ergebnis

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

Klammern

Code

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

Ergebnis

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

Häufig braucht man größere Klammern.

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

Ergebnis

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

Automatische Bestimmung der Größe:

```
\left(\right) \left(\right.
\left\{ \,,\middle|\,, \right\}
```

Ergebnis

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

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 , dx

$\text{\texttt{\textcolor{red}{e}}}$, $\text{\texttt{\textcolor{red}{i}}}$, $\text{\texttt{\textcolor{red}{\pi}}}$, $\text{\texttt{\textcolor{red}{d}}}\text{\texttt{\textcolor{blue}{x}}}$

- Indizes wie „min“ oder „max“

$\text{\texttt{x}}\text{\texttt{\textcolor{red}{_}}}\text{\texttt{\textcolor{red}{text}}}\text{\texttt{\textcolor{blue}{min}}}$

- dx wird durch kleines Leerzeichen (\,) vom Integranden abgetrennt.

Konventionen: Variablen, Zahlen, Einheiten, Indizes

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

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.

Die equation-Umgebung

Abgesetzte Gleichung, automatische Nummerierung.
equation* erzeugt unnummerierte Gleichung.

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

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

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(12)}{\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
 $1,2 \cdot 10^2$
 $\text{N} = \text{kg m/s}^2$
 $1,2 \text{ m/s}$
 $(4,3 \pm 1,2) \mu\text{s}$
 $(4,3 \pm 1,2) \cdot 10^{-6} \text{ s}$
 kg m s^{-2}
 $\frac{\text{kg m}}{\text{s}^2}$
 1.2

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

$$\text{H}_2\text{O}_2$$

$$^{227}_{90}\text{Th}^+$$

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

$$^{14}_6\text{C} \longrightarrow ^{14}_7\text{N} + \text{e}^- + \bar{\nu}_e$$

$$\text{CO}_2 + \text{C} \rightleftharpoons 2 \text{CO}$$

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

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

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

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

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

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

