

Diese Skript ist noch unvollständig und kann Fehler enthalten. Falls Sie Fehler entdecken, können Sie mir gerne eine Mail an <mailto:latex.heindl@tum.de> schreiben.

1 Herkunft/Was ist \LaTeX

\TeX



\TeX ist ein Textsatzprogramm mit Makrosprache von Donald E. Knuth. Mittlerweile wurde \TeX "weiterentwickelt" und es gibt verschiedene Aufsätze auf \TeX . Hierbei ist \LaTeX das bekannteste.

\LaTeX



\LaTeX ist eine Sammlung von \TeX Makros, die die Benutzung von \TeX vereinfacht. Mittlerweile verwenden die meisten Nutzer hauptsächlich \LaTeX und kaum \TeX . Dabei wird \LaTeX nicht mehr nur von Leslie Lamport entwickelt, sondern es gibt eine große Community.

Weitere \TeX Weiterentwicklungen:

- LuaTeX
- XeTeX
- XeLaTeX
- ConTeXt

2 Vorteile

- Gute Skalierung bei größeren Dokumenten
- Auslagerung von Dokumenten in mehrere Dateien
- gut Versionskontrollbar (z.B. git)
- Sehr viele Pakete als Erweiterungen
- Einheitliche Styles gut durchsetzbar
- Keine Abhängigkeit vom eigentlichen Editor (Dateien sind im Klartext)

- v.A. für den wissenschaftlichen Bereich: gute Verarbeitung von Mathematischen Formeln
- Dokumente i.d.R. immer im pdf-Format → sehr portabel

Dabei ist es wichtig zu wissen, dass \LaTeX kein WYSIWYG (You See Is What You Get) Editor ist, sondern eher einer Markup Sprache ähnelt. Dadurch hat man den Vor-/Nachteil, dass man jegliche Formatierung direkt sieht.

3 Editor

- “IDE“ (ist out of the box für den Einstieg am einfachsten)
 - TexMaker
 - TexStudio
- Texteditor
 - (neo)vim
 - Sublime
 - Gummi
 - Jeder andere Texteditor

4 Installation

Ubuntu:

- `sudo apt-get install texlive texlive-lang-german texlive-latex-extra`
- `sudo apt-get install texlive-base` (minimale Installation)
- `sudo apt-get install texlive-full` (auch alle Sprachpakete mit dabei)

Siehe hierzu auch die jeweiligen Wikis (Ubuntuusers/ArchWiki).

5 Kompilieren

In der Regel legt man seinen \LaTeX -Code in Dateien mit der Endung *.tex* ab. Kompiliert werden können diese dann mit dem folgenden Befehl oder über den entsprechenden Button in einem Editor (mit \LaTeX Unterstützung).

`$ pdflatex .tex Datei` → gleicher Dateiname.pdf (+ ein paar Arbeitsdateien wie *.aux* oder *.toc*)

6 Hands-on

6.1 Erste Dokumente

6.1.1 Aufteilung des Dokuments

Eine `.tex` Datei besteht ganz grob aus 2 Teilen, der Präambel und dem eigentlichen Inhalt.

Präambel In der Präambel muss zunächst die Art des zu erstellenden Dokuments angegeben werden (Artikel, Bericht, Buch) `\documentclass{article}`. Außerdem werden hier eventuell zusätzliche Pakete geladen, falls gewünscht.

Inhalt Darauf folgt ein `\begin{document}` und der eigentliche Inhalt und am Ende steht noch ein `\end{document}`

6.2 Text

- Text ganz normal schreiben (Zeichen wie `$`, `_` oder `^` müssen mit einem `\` escaped werden (im Mathemodus evtl auch anders))
- Eine Leerzeile steht für einen neuen Absatz, `\\` für einen Zeilenumbruch und ein Leerzeichen für ein Leerzeichen
- Mehrere Leerzeilen und mehrere Leerzeichen werden genauso behandelt als wären es nur einzelne
- Befehle/Makros werden mit einem `\` eingeleitet
- Kommentare durch ein `%`

Tipp: Bei Zeilenumbrüchen kann man zusätzlich manuell angeben, wieviel Platz gelassen werden soll `\\[längenangabe]`

6.3 Hervorhebungen

fett `\{textbf}`

kursiv `\{textit}`

slanted `\textsl{text}`

unterstrichen `\underline{text}` (Achtung bei Zeilenumbrüchen)

Es ist jedoch guter Stil Hervorhebungen durch den Befehl `\emph{Text}` zu machen (standardmäßig kursiv), da dieser Befehl falls gewünscht leicht undefiniert werden kann (dadurch weiterhin einheitlich). Zudem wirkt eine fette Hervorhebung z.B. häufiger als zu aufdringlich.

Schriftgröße

Die Schriftgröße kann mit einem der folgenden Befehle geändert werden:

<code>\tiny</code>	sample text	<code>\large</code>	sample text
<code>\scriptsize</code>	sample text	<code>\Large</code>	sample text
<code>\footnotesize</code>	sample text	<code>\LARGE</code>	sample text
<code>\small</code>	sample text	<code>\huge</code>	sample text
<code>\normalsize</code>	sample text	<code>\Huge</code>	sample text

Dabei bleibt die Schriftgröße erhalten, bis die aktuelle Umgebung beendet wird oder die Schriftgröße erneut geändert wird (evtl auch durch einen anderen Befehl).

Falls die Schriftgröße global geändert werden soll, so ist das als optionaler Parameter bei der Dokumentklasse anzugeben (hier nochmal ein großer Vorteil bei KOMA-Skript)

Schriftart

Roman Font Family `\textrm{text}`

SMALL CAPS `\textsc{text}`

Sans serif font family `\textsf{text}`

teletype font family `\texttt{text}`

6.4 Zusatz zu Befehlen

Ein Befehl wird durch ein `\` eingeleitet, er kann feste Parameter haben, diese werden in `{...}` angegeben und optionale Parameter, diese werden in `[...]` angegeben.

Streng genommen ist `\begin{env-Name}... \end{env-Name}` eine Umgebung (Environment). Ein Beispiel hiervon ist die den Inhalt des Dokuments umgebende `document` Umgebung.

6.4.1 Dokumentklassen

Da \LaTeX ursprünglich auf den amerikanischen Raum ausgelegt war, sind viele Dinge für den europäischen Raum vielleicht etwas ungewöhnlich (z.B. das letter-Papierformat). Das kann umgangen werden, indem man die *KOMA-Skript* Klassen verwendet (die noch zusätzliche Features bereitstellen), also z.B. statt `article` `scrartcl`. Alternativ kann man aber auch der Klasse erstmal das gewünschte Papierformat als optionalen Parameter übergeben, also z.B. `\documentclass[a4paper]{article}`.

6.4.2 Pakete

Da der Funktionsumfang von \LaTeX out of the Box erstmal recht abgespeckt ist, werden in der Regel Pakete nachgeladen, die dann bestimmte Funktionalitäten bereitstellen.

Geladen werden diese durch den `\usepackage[parameter]{packagename}` Befehl (wenn möglich können auch Parameter übergeben werden).

6.4.3 Sonderzeichen

Die deutschen Sonderzeichen müssen normal durch Kombinationen, wie z.B. `\ss` für ein β geschrieben werden. Abhilfe geschaffen werden kann, indem man die \LaTeX angibt, dass die Eingabedatei in *utf8* codiert ist (`\usepackage[utf8]{inputenc}`). Ebenso sollte das encoding der Schrift auf *T1* gestellt werden (`\usepackage[T1]{fontenc}`).

6.4.4 Einteilung des Dokuments und Inhaltsverzeichnis

Ein Dokument kann in mehrere Segmente eingeteilt werden, dazu wird der `\section{Section Name}` Befehl verwendet. Eine section kann wiederum `\subsection{Subsection Name}` enthalten, die wiederum `\subsubsection{Subsubsection Name}` enthalten kann.

Diese Unterteilung des Dokuments kann in einem Inhaltsverzeichnis festgehalten werden, das durch den Befehl `\tableofcontents` (typischerweise am Anfang oder am Ende) erstellt wird.

Hierbei ist zu beachten, dass das Dokument eventuell mehrfach kompiliert werden muss, bis das Inhaltsverzeichnis korrekt gesetzt ist, da \LaTeX streng von vorne bis hinten das Dokument durchläuft (Informationen über das Inhaltsverzeichnis werden dann z.B. in der `.toc` Datei gespeichert).

Als zusätzliches Paket kann an dieser Stelle das `hyperref` Paket empfohlen werden, womit Hyperlinks innerhalb des Dokuments ermöglicht werden. Das Inhaltsverzeichnis z.B. wird nur durch das Laden dieses Pakets klickbar gemacht.

Solche Verzeichnisse gibt es auch für Quellen, Tabellen und Grafiken. Hierauf wird im Folgenden jedoch nicht eingegangen.

6.4.5 Lokalisierung und Silbentrennung

Da \LaTeX automatisch Wörter am Zeilenende umbricht, sollte für eine Deutsche Silbentrennung das `babel`-Paket mit der Option `ngerman` (n für neue Rechtschreibung) geladen werden, also `\usepackage[ngerman]{babel}`.

Um zusätzlich manuell an manchen Stellen Silben zu trennen kann in der *Präambel* ein `\hyphenation{silbe1-silbe2}` oder im Text ein `\-` an der jeweiligen Stelle eingefügt werden (hilfreich für Fremdwörter). Mit einem `~` kann ein Umbruch an dieser Stelle verhindert werden (hartes Leerzeichen).

6.4.6 Umbrüche

Mit einem `\\` kann ein Zeilenumbruch manuell erzeugt werden (mit optionalem Parameter wie viel Platz gelassen werden soll). Mit einem `\pagebreak` kann ein Seitenumbruch erzeugt werden.

6.4.7 Längen

Es gibt in \LaTeX verschiedene Möglichkeiten eine Länge anzugeben. Dabei kann zunächst einmal eine normale Einheit wie z.B. `cm/mm/in/pt` verwendet werden, jedoch kann auch `em` (Breite des Buchstabens “M“ in aktueller Schrift) oder `ex` (Höhe des Buchstabens “X“ in aktueller Schrift) als Einheit verwendet werden. Die letzten beiden sind somit relative Maße.

Zusätzlich dazu können bestimmte Längen der Umgebung genutzt werden, wie zum Beispiel `\textwidth` für die Breite des Textes. Diese Längenangabe kann ebenfalls mit einem Vorfaktor versehen werden. (Man kann diese Längen auch verändern und so manuell das Erscheinungsbild der Seite verändern, aber um soetwas sollten sich Pakete oder die Dokumentenklasse kümmern).

6.4.8 Alignment / Ausrichtung

\LaTeX setzt den Text standardmäßig im Blocksatz. Dies mag vielleicht am Anfang etwas ungewöhnlich wirken, ist aber tatsächlich Standard und sieht normal tatsächlich auch besser aus (vor allem bei viel Text).

Zum **zentrieren** kann der Befehl `\centering` (Gültigkeit wie bei Schriftgrößen) oder die `center`-Umgebung verwendet werden.

Für einen **linksbündigen** Teil kann die Umgebung `flushleft`, für einen **rechtsbündigen** `flushright` verwendet werden.

Bei einzelnen Zeilen können auch die Befehle `\centerline{}`, `\raggedleft` (linker Flatterrand → rechtsbündig) und `\raddedright` (rechter Flatterrand → linksbündig) verwendet werden.

Dabei ist zu beachten, dass nur im Blocksatz eine Silbentrennung durchgeführt wird.

6.4.9 parskip / Verhalten beim Absatz

Auffällig ist, dass nach einem Absatz standardmäßig nicht mehr Platz zwischen den Zeilen gelassen wird, sondern die nächste Zeile etwas eingerückt wird. Dies ist wieder ein Überbleibsel aus dem amerikanischen Raum.

Das Verhalten kann durch die KOMA-option `parskip=half` abgestellt werden. (Optional könnten zwei Längen undefiniert werden, jedoch soll hierauf an dieser Stelle nicht eingegangen werden)

6.4.10 Aufzählungen

Es gibt verschiedene Arten der Aufzählung, dabei werden bei jeder die einzelnen Elemente der Aufzählung durch ein `\item` angegeben. Um die Aufzählung wird eine `itemize` (nicht nummerierte Aufzählung), `enumerate` (nummerierte Aufzählung) oder eine `description` Umgebung gespannt.

Dabei ist die `description` Umgebung etwas besonders, hier wird dem `\item` Befehl noch ein optionaler Parameter übergeben, der dann Fett vor den Eintrag geschrieben wird (perfekt für kurze Begriffserklärungen).

Man kann die Aufzählungszeichen von `itemize` und `description` auch verändern, darauf wird hier jedoch nicht eingegangen.

6.4.11 Fußnoten

Fußnoten können mit dem Befehl `\footnote{Fußnotentext}` erzeugt werden. Der Rest geht automatisch (evtl wird ein zusätzlicher Kompilier-run benötigt). Der Rest kann auch benutzerdefiniert werden (z.B. Art der Nummerierung, Pseudo-Fußnoten).

6.5 Tabellen

Tabellen werden in \LaTeX in einer `tabular`-Umgebung gesetzt. Beispiel:

```
\begin{tabular}{spaltendef}
  zelle1 & zelle2 \\
  zelle3 & zelle4 \\
\end{tabular}
```

Dabei beschreibt die *Spaltendefinition* die Ausrichtung und die Anzahl der Spalten. `l,r,c` steht dabei für eine links-/rechtsbündige oder zentrierte Spalte, die nicht umgebrochen wird. `|` kann verwendet werden um senkrechte Striche zu ziehen. Die `&`-Zeichen stehen für den Beginn einer neuen Spalte, `\\` steht für eine neue Zeile.

Zusätzlich kann in der Spaltendefinition noch `*{num}{spaltendef}` verwendet werden, die angegebene Spaltendefinition wird dann *num*-mal wiederholt. Mit einem `p{breite}` kann auch eine Spalte im Blocksatz gesetzt werden.

Da in Tabellen der Befehl `\\` schon vergeben ist, muss der Befehl `\newline` verwendet werden um einen Zeilenumbruch zu erzeugen.

Vertikale Linien können in Tabellen mit dem `\hline` oder dem `\cline{i-j}` gezogen werden.

Da es häufig gewollt ist, dass die Tabelle eine feste Breite hat, empfehle ich den Einsatz des *tabularx* Paketes. Diese stellt eine neue Umgebung (`tabularx`) bereit, die zusätzlich zur Spaltendefinition noch die Tabellenbreite als Parameter benötigt (häufig `\textwidth`). In der Spaltendefinition muss dann mindestens eine Spalte mit *X* deklariert werden. Das steht für eine Spalte im Blocksatz, deren Länge abhängig vom verfügbaren Platz ist. Wird *X* mehrfach verwendet, so sind alle Spalten, die als *X* definiert sind gleich breit.

Bsp: `\begin{tabularx}{\textwidth}{XX}`

6.6 Grafiken

Um Bilder einzufügen wird das Paket `graphicx` benötigt. Im Dokument kann dann der Befehl `\includegraphics{pfad/zum/Bild}` verwendet werden, wobei der Pfad relativ zum Ort der *.tex*-Datei ist.

In der Regel möchte man jedoch auch noch die Größe des Bilder verändern, dazu gibt es die optionalen Parameter `height=höhe` und `width=breite` für den `includegraphics`-Befehl. Wird nur einer der Beiden verwendet, so bleibt das Seitenverhältnis gleich.

Unterstützte Bildformate sind *pdf*, *png*, *jpg* (*eps*). Vektorgrafiken können in pdfs umgewandelt werden. Mit *inkscape* kann jedoch auch die Vektorgrafik direkt eingebettet werden¹ (dabei wird vorhandener Text auch direkt durch \LaTeX gesetzt, was zu einem einheitlichen Schriftbild führt).

6.7 Floating environments

Häufig sollen Bilder und Tabellen auch durchnummeriert werden und eine Unterschrift haben. Dies ist mit den sogenannten floating-Environments `table`

¹<http://tug.ctan.org/tex-archive/info/svg-inkscape/InkscapePDFLaTeX.pdf>

und `figure` möglich. Die Unterschrift wird dabei mittels `\caption{text}` angegeben (in der Ausgabe kommt dann noch die Nummerierung davor).

Durch den `\label{name}` Befehl kann dem Bild noch ein Name gegeben werden, wodurch man sich im Text durch ein `\ref{label}` wieder darauf beziehen kann (Nummer wird eingefügt).

Dadurch ist die `positioning` des Elements nicht mehr so wichtig und \LaTeX gibt es eventuell auch an einer etwas anderen Stelle aus.

6.8 Mathe

Formeln können entweder inline (`$code$`) oder abgesetzt gesetzt werden (`\[code\]`). Zusätzlich stehen noch verschiedene Umgebungen zur Verfügung, die zusätzliche Funktionalitäten mitbringen (v.A. bereitgestellt durch das *amsmath* Paket).

Darunter ist u.A. die `align(*)` Umgebung, die (un)nummerierte Formeln setzt, Tabellenartig ausgerichtet werden können. Dabei werden die Spalten klassisch durch `&s`, Zeilen durch `\\` von einander getrennt. Ausgerichtet sind die Spalten immer abwechselnd rechts,links,rechts,...!

Als Code können zunächst einmal ganz normal mathematische Formeln geschrieben werden, wie z.B. $a + b = c$. Wie in normalem \LaTeX Code auch, werden Leerzeichen automatisch gesetzt unabhängig davon, wieviele davon vorhanden sind.

Mit einem `^x` kann z.B. das Zeichen x hochgestellt werden. Um mehrere Zeichen hochzustellen, werden `{}` benötigt.

Mit einem `_x` kann z.B. das Zeichen x tiefgestellt werden. Um mehrere Zeichen tiefzustellen, werden `{}` benötigt.

Besondere Zeichen wie z.B. griechische Buchstaben können mit Befehlen gesetzt werden (z.B. `\alpha \beta \varphi`).

Dabei ist darauf hinzuweisen, dass häufig statt einem `*` auf ein `\cdot` (`\cdot`) als Multiplikationszeichen zurückgegriffen wird.

Für große Summen- (`\sum^{ }_{ }`) und Produktzeichen (`\prod^{ }_{ }`) gibt es ebenfalls spezielle Befehle. Bei diesen ist darauf zu achten, dass diese im inline Modus evtl etwas anders aussehen als im abgesetzten Modus oder in einer Mathe Umgebung.

Da man häufig auch größere Klammern benötigt, gibt es die Befehle `\left(mathe code \right)`, die Klammern setzen, die abhängig von ihrem Inhalt skaliert werden. Dabei benötigt ein `\right` auch immer ein matchendes `\left` und andersrum (kann mit `\right.` umgangen werden). Statt `(` kann auch z.B. `[` als große Klammer verwendet werden und anderes.

Mit `\text{text-code}` kann auch normaler L^AT_EX Code im Mathemodus gesetzt werden (dieser ist dann nicht kursiv).

Es gibt noch viel weiteres im Mathemodus (z.B. Integrale und Matrizen (empfehlung: NiceMatrix package)), das lässt sich aber i.d.R. leicht im Internet finden.

Literatur

- [1] Abbildung 1: Donald E. Knuth, Von Flickr user Jacob Appelbaum, uploaded to en.wikipedia by users BeSherman, Duozmo - Flickr.com (via en.wikipedia), CC BY-SA 2.5, <https://commons.wikimedia.org/w/index.php?curid=1303242>
- [2] Abbildung 1: Leslie Lamport, Copyrighted free use, <https://commons.wikimedia.org/w/index.php?curid=487587>