

Erste Schritte in \LaTeX

Jan Rautner, Benedikt Wolf, Albin Zehe

Donnerstag, 26. November 2015

Inhaltsverzeichnis

1	Einführung	3
1.1	Motivation	3
1.2	Installation	5
1.2.1	Windows	5
1.2.2	Mac OS	5
1.2.3	Linux	6
2	Benutzung	7
2.1	Ein neues Dokument erstellen	7
2.1.1	Dokumentgerüst	7
2.1.2	Packages	7
2.2	Formatierung	8
2.2.1	Zeilenumbrüche	9
2.2.2	Textstile	9
2.2.3	Schriftgrößen	10
2.3	Strukturierung	13
2.4	Aufzählungen	14
2.4.1	Unnummerierte Listen	14
2.4.2	Nummerierte Listen	15
2.4.3	Description	15
2.5	Tabellen	16
2.6	Verschiedenes	19
2.6.1	Codestil	19
2.6.2	Nützliche Befehle	20
2.6.3	Einige Beispiele	20
3	Erweiterungen	22

1 Einführung

1.1 Motivation

Wer daran denkt, mit einem Computer Texte zu verfassen, denkt vermutlich zuerst an Microsoft Word (oder entsprechende Textverarbeitungsprogramme (WYSIWYG=What you see is what you get) anderer Hersteller). Das ist auch verständlich, schließlich ermöglichen es diese Programme, ohne große Einarbeitungszeit einen einfachen Text zu verfassen. Aber optimal sind diese Programme nicht. Wie oft ärgert man sich, dass der Text „unschön“ aussieht oder dass eingebundene Bilder plötzlich den ganzen Textfluss durcheinander bringen. Außerdem hat man häufig das Problem, dass die eigene Formatierung an einem anderen Rechner (andere Version, anderes Betriebssystem, . . .) nicht mehr funktioniert. Und spätestens, wenn man mehr als fünf Formeln mit griechischen Buchstaben oder mathematischen Symbolen braucht, können diese Programme den Anwender an den Rand des Wahnsinns bringen.

Doch das muss nicht sein. Die Antwort heißt \LaTeX .

Mit \LaTeX wird das Einbinden von Formeln mit etwas Übung sehr leicht möglich.

Will man etwa (nur) zwei Brüche addieren, z.B. $\frac{3}{5} + \frac{8}{9} = \frac{67}{45}$, wird man in einem WYSIWYG-Programm vermutlich einen Formeleditor benötigen. In \LaTeX muss man nur wissen, wie der Befehl für einen Bruch heißt und kann damit sofort schreiben

```
 $\frac{3}{5} + \frac{8}{9} = \frac{67}{45}$ 
```

Selbst komplexere Formeln wie $i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = \hat{\mathcal{H}} |\psi(t)\rangle$ werden damit gut umsetzbar.

\LaTeX setzt den Text automatisch in eine gut lesbare Form, die im universitären Bereich schon längere Zeit Standard ist und meist auch vorausgesetzt wird.

Zu diesem Stil gehört auch, dass \LaTeX automatisch Kapitel, Seiten, Abschnitte und vieles mehr nummeriert und mit nur einem Befehl ein stets korrekt geführtes Inhaltsverzeichnis erstellt. Auch Fußnoten werden automatisch verwaltet. Grafiken können problemlos eingebunden werden und erscheinen dann passend beschriftet an einer geeigneten Stelle (vgl. 1.1).

Andererseits ist \LaTeX auch durch die Einbindung von Packages (beinahe) beliebig anpassbar und erweiterbar. So können mit dem z.B. Package TikZ Grafiken direkt im Dokument erstellt werden (vgl. 1.2).

Daneben gibt es viele andere nützliche Packages, mit denen z. B. Verweise „klickbar“¹ (hyperref) gemacht werden oder die mathematischen Textsatz erlauben (amsmath, amssymb, amssymb), aber auch eher skurrile Packages wie tengwarscript, mit dem man Sindarin schreiben kann.

¹ angeklickt?

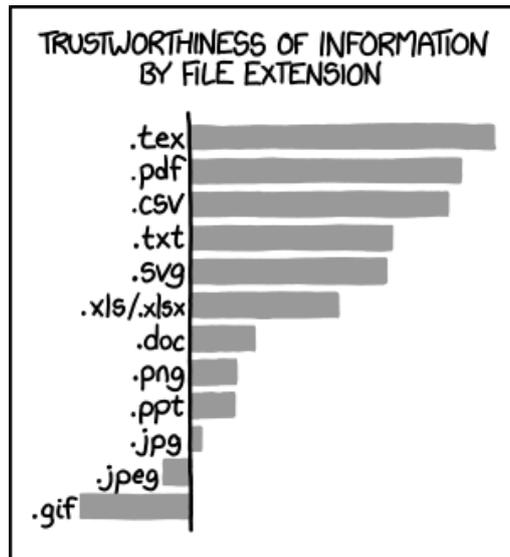


Abbildung 1.1: Ein xkcd von <http://www.xkcd.com>

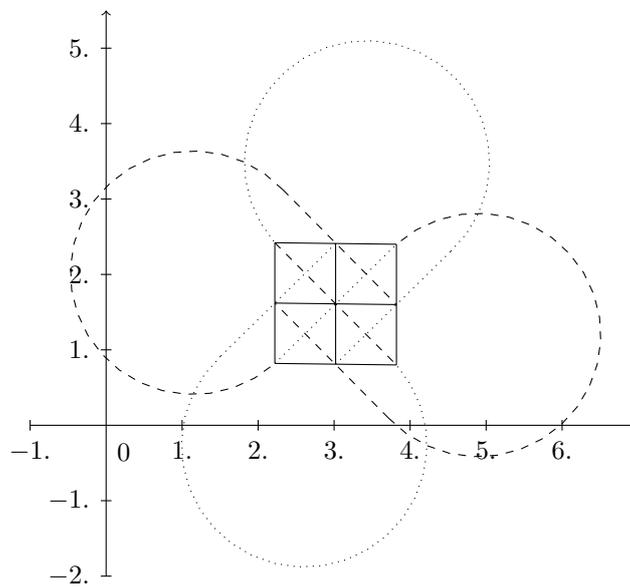


Abbildung 1.2: Eine (semi)geTikZte Grafik

Was \LaTeX besonders auszeichnet, ist, dass es Open Source ist, also nichts kostet und zudem von einer recht großen Nutzergemeinde verbessert und erweitert wird. Außerdem ist es (im Wesentlichen) plattformunabhängig, d.h. wie das erstellte Dokument aussieht, hängt nicht vom Betriebssystem ab. Dadurch wird es auch möglich, perfekt passende Formatvorlagen zu erstellen, auszutauschen und weiterzuverwenden.²

Der einzige Nachteil an \LaTeX ist, dass es beim Einstieg eine gewisse Einarbeitungszeit braucht, weil es anders als WYSIWYG-Programme nicht direkt das Ergebnis anzeigt, sondern erst nach der Übersetzung ein Ergebnis liefert, das dann aber exakt dem entspricht, was festgelegt wurde (WYSIWYM=What you see is what you mean). Diese Einführung dient dem Zweck, diese Einarbeitung zu erleichtern.

1.2 Installation

Die Installation besteht allgemein aus zwei Teilen: Der \LaTeX -Distribution und einem Editor. Theoretisch kann auch ein simpler Texteditor verwendet werden und anschließend über die Kommandozeile übersetzt werden, sodass Teil 2 eigentlich optional ist. Ein Editor ist aber deutlich komfortabler.

Wir empfehlen als Editor den \TeX maker, der für alle gängigen Betriebssysteme verfügbar ist.

1.2.1 Windows

Unter Windows lädt man zunächst auf <http://miktex.org/download> MikTeX herunter. Im heruntergeladenen Installer folgt man den Anweisungen und wählt die gewünschten Optionen aus:

- Preferred Paper: *A4*
- Install missing packages on-the-fly: Sollen fehlende Packages, die in einem Dokument verwendet werden, automatisch aus dem Internet heruntergeladen werden? Es empfiehlt sich, hier *Yes* oder *Ask me first* auszuwählen.

Anschließend lädt man den \TeX maker von <http://www.xm1math.net/texmaker/> herunter und installiert auch diesen.

1.2.2 Mac OS

Unter Mac OS lädt man zunächst auf <https://tug.org/mactex/> MacTeX herunter. Im heruntergeladenen Installer folgt man einfach den Anweisungen.

Anschließend wird auch hier der \TeX maker von <http://www.xm1math.net/texmaker/> geladen und installiert.

²Weitere Gründe für \LaTeX <http://latex.tugraz.at/dokumentation/mythen>

1.2.3 Linux

Unter Linux ist die Installation noch ein wenig einfacher: Dort müssen nur die folgenden zwei Befehle im Terminal eingetippt werden.

```
sudo apt-get install texlive texlive-doc-de \  
    texlive-latex-extra texlive-lang-german  
sudo apt-get install texmaker
```

2 Benutzung

2.1 Ein neues Dokument erstellen

L^AT_EX-Dokumente haben üblicherweise die Endung `.tex`. Ein neues Dokument erstellt man wie sonst auch über das passende Icon im T_EXmaker.

Das Dokument muss zwingend ein bestimmtes Format einhalten, um übersetzt werden zu können.

2.1.1 Dokumentgerüst

Die erste Zeile definiert die Klasse des Dokuments. In den meisten Fällen wird hier `scrartcl`, `scrreprt` oder `scrbook` verwendet.

`scrartcl` ist für kurze Dokumente normalerweise die Einstellung der Wahl. `scrbook` und `scrreprt` sind für längere Dokumente gedacht und auf doppelseitige Darstellung ausgelegt. Damit werden beispielsweise die Seiten für den Ausdruck im doppelseitigen Format gestaltet, sodass die Abstände zum Rand auf den Außenseiten (links bei geraden Seitenzahlen und rechts bei ungerade Seitenzahlen) größer sind als auf den Innenseiten. Ein weiterer Unterschied liegt in der Aufteilung in Kapitel oder Abschnitte, vgl. 2.3.

Das Grundgerüst eines L^AT_EX-Dokuments sieht wie folgt aus:

```
\documentclass[12pt,a4paper]{scrartcl}
...
Einbindung von Packages
...
Eventuell Definitionen und weitere Konfigurationen

\begin{document}
Dokumententext
\end{document}
```

Der Teil vor `\begin{document}` wird als Präambel bezeichnet.

2.1.2 Packages

Der vielleicht wichtigste Bestandteil von L^AT_EX sind Packages. Diese stellen Erweiterungen der Grundfunktionalität dar, die von zusätzlichen Zeichen oder mathematischen Umgebungen über die Einbindung und direkte Erstellung von Grafiken bis hin zur Integration ganzer Programmiersprachen in ein Dokument reichen.

Um ein Paket zu verwenden, schreibt man in den Kopf der Datei den Befehl

```
\usepackage[optionale Argumente]{Paketname}
```

Optionale Argumente sind dabei Einstellungen für das Paket.

Um beispielsweise das Paket `hyperref` mit der Option `implicit` einzubinden, verwendet man folgenden Befehl:

```
\usepackage[implicit=true]{hyperref}
```

Hier werden einige der wichtigsten Packages aufgeführt, die in fast allen Dokumenten eingebunden werden sollten.

inputenc, babel `inputenc` legt die Zeichencodierung (Unicode, ASCII) fest, `babel` stellt verschiedene Sprachen zur Verfügung (wichtig u. A. für Darstellung von Datumsangaben und die Sprache des Inhaltsverzeichnisses). Für unsere Zwecke werden die Pakete mit den Optionen `utf8` bzw. `ngerman` verwendet. Muss immer eingebunden werden.

amsmath, amsfonts, amssymb Viele Umgebungen zur Darstellung von Gleichungen und längeren Rechnungen, sowie mathematische Symbolen. Sollte in jedem Dokument eingebunden werden, dass irgendwelche mathematischen Inhalte behandelt.

microtype Verbessert einige typographische Aspekte des erzeugten Dokuments, darunter Zeichenabstände etc. Die genaue Funktionsweise zu kennen ist nicht unbedingt wichtig, im Endeffekt erhöht das Package die Lesbarkeit des Dokuments. Sollte immer eingebunden werden.

hyperref Ein sehr umfangreiches Paket, das auch viele Funktionen zur Konfiguration von pdfs bietet. Die meisten davon werden selten bis nie benötigt, interessanter ist die vom Paket bereitgestellte Verlinkung von Inhalten im Dokument. Mit dem Parameter `implicit=true` wird angegeben, dass die Abschnittstitel im Inhaltsverzeichnis, die Verweise auf Fußnoten, andere Abschnitte oder Literaturverweise automatisch auf die passende Seite verlinken. Sollte immer eingebunden werden.

Übung

Erstelle ein (sehr) einfaches Dokument!

Das Dokument sollte mindestens eine automatisch generierte Titelseite mit eurem Namen, einem beliebigen Titel und dem heutigen Datum enthalten¹.

2.2 Formatierung

Wie bereits erwähnt erfolgt die Formatierung von Text in \LaTeX nicht, wie z.B. in Word, über Schaltflächen, sondern über Befehle. Zwar gibt es in den meisten Editoren Menüs

¹Tipp: Auf das heutige Datum kann man mit dem Befehl `\today` zugreifen

und Knöpfe, die wichtige Funktionen einfügen; es empfiehlt sich aber, diese nur zu benutzen, wenn man den Befehl für eine Formatierung gerade nicht auswendig weiß, oder um die Befehle zu lernen. Normalerweise ist man per Tastatur wesentlich schneller als mit der Maus.

Einige Hinweise zu Formatierungen in \LaTeX :

2.2.1 Zeilenumbrüche

Ein Zeilenumbruch im Editor hat zunächst keinen Einfluss auf das erzeugte Dokument. Um einen Zeilenumbruch in der pdf zu bekommen, verwendet man entweder den Befehl `\` oder lässt eine Zeile frei.

Beispiel:

```
Zeile 1
Zeile 2
erzeugt
```

```
Zeile 1 Zeile 2
```

Um

```
Zeile 1
Zeile 2
```

zu bekommen, muss man

```
Zeile 1 \
Zeile 2
```

oder

```
Zeile 1
Zeile 2
```

oder

```
Zeile 1 \ Zeile 2
```

schreiben.

Es ist also egal, ob im Quelltext überhaupt ein Zeilenumbruch auftaucht, solange der `\`-Befehl gegeben wird.

Anzumerken ist noch, dass eine Leerzeile nicht nur einen Zeilenumbruch, sondern auch einen neuen Absatz erzeugt. Ein neuer Absatz beginnt in \LaTeX standardmäßig eingerückt. Wenn man das nicht möchte, kann man mit `\setlength{\parindent}{0cm}` in der Präambel die Einrückung auf 0 setzen.

2.2.2 Textstile

- Für Hervorhebungen im Text sollte der `\emph{Text}`-Befehl verwendet werden. Es ist zwar auch möglich, Wörter zu unterstreichen (`\underline{Text}`), fett zu setzen (`\textbf{Text}`) und vieles mehr (siehe `Texmaker -> LaTeX -> Schriftart`). Üblicherweise wird das aber nur in Ausnahmesituationen getan.

Der Effekt von `\emph{Text}` ist folgender:

```
Dies ist ein \emph{Beispieltext}, der zeigt, wie der emph-Befehl
verwendet wird.
```

Dies ist ein *Beispieltext*, der zeigt, was der `emph`-Befehl tut.

- Wenn Teile des Textes nicht übersetzt werden sollen, kann der `\verb`-Befehl verwendet werden.

Beispiel:

Textteile im `\verb|\verb|`-Befehl werden nicht uebersetzt und in Schreibmaschinenschrift ausgegeben.

Textteile im `\verb`-Befehl werden nicht uebersetzt und in Schreibmaschinenschrift ausgegeben.

Damit können beispielsweise andere Befehle in der pdf auftauchen. Die senkrechten Striche („|“) sind dabei die Begrenzer des Abschnitts, der nicht übersetzt werden soll. Statt | können viele andere Zeichen verwendet werden, sie dürfen nur nicht im eingegrenzten Bereich vorkommen.

Die Schreibung von ü als ue oben geht auf eine Einschränkung des `minted`-Pakets zurück, mit dem hier die Codestellen ausgegeben werden. Mehr dazu später.

- Fußnoten werden mit `\footnote{Fußnotentext}` erstellt².

2.2.3 Schriftgrößen

Natürlich kann man in \LaTeX auch verschiedene Schriftgrößen verwenden. Anders als bei WYSIWYG-Programmen werden diese normalerweise nicht in absoluten Zahlen angegeben, sondern richten sich nach der Größe des normalen Textes, die ganz am Anfang als Option von `\documentclass[]{}` festgelegt wird.

In den üblichen Dokumentklassen gibt es die zehn Schriftgrößen `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge` und `Huge`. Zur Verwendung der Schriftgrößen gibt es drei Möglichkeiten:

- Setzt man einen `\` vor den Namen der Schriftgröße, so wird der gesamte folgende Text in diese Größe gesetzt. (Nur bei sehr langen Passagen zu empfehlen). Ein Beispiel:

Der Text ist noch normal.
`\tiny` Jetzt wir der Text klein und auch alles danach.
 Selbst in der zweiten Zeile bleibt der Text klein.

Der Text ist noch normal. Jetzt wir der Text klein und auch alles danach. Selbst in der zweiten Zeile bleibt der Text klein.

²Die Nummerierung und Platzierung der Fußnote erfolgt dann automatisch. Mit eingebundenem `hyperref` wird auch automatisch ein Link von der Fußnotenmarkierung zur Erklärung gesetzt.

- Setzt man geschweifte Klammern um die Passage mit dem Befehl, wird nur diese in die entsprechende Größe gesetzt.

```
Dies ist ein {\tiny kleiner} Text.
```

Ergibt also

```
Dies ist ein kleiner Text.
```

- Schließlich kann man jede Schriftgröße auch in einer Umgebung verwenden, d.h. der Text zwischen `\begin{Schriftgroesse}` und `\end{Schriftgroesse}` bekommt diese Schriftgröße. Das ist bei längeren Passagen deutlich übersichtlicher als die geschweiften Klammern.

```
Das ist die Einleitung in normaler Schriftgroesse.
```

```
\begin{tiny}
    Jetzt mache ich Randbemerkungen.
\end{tiny}

\begin{LARGE}
    Jetzt schreibe ich furchtbar wichtige Dinge.
\end{LARGE}
```

```
Und jetzt geht der normale Text weiter.
```

Ergibt

```
Das ist die Einleitung in normaler Schriftgroesse.
```

```
Jetzt mache ich Randbemerkungen, die eigentlich nichts mit dem Thema zu tun haben.
```

```
Jetzt schreibe ich furchtbar wichtige Dinge.
```

```
Und jetzt geht der normale Text weiter.
```

Übung

Baut das Dokument auf der folgenden Seite nach!

Euren Namen dürft ihr natürlich eintragen, sonst soll das Dokument exakt nachgebildet werden.

Ein erstes Dokument

Albin Zehe, Jan Rautner, Benedikt Wolf

15. Oktober 2015

Mr. und Mrs. Dursley im Ligusterweg Nummer 4 waren stolz darauf, ganz und gar normal zu sein, *sehr* stolz sogar.

Niemand wäre auf die Idee gekommen, sie könnten sich in eine merkwürdige und geheimnisvolle Geschichte verstricken, denn mit solchem Unsinn wollten sie nichts zu tun haben.

Mr. Dursley war Direktor einer Firma namens **Grunnings**, die Bohrmaschinen herstellte.¹

Die Fußnote wurde mit dem `\footnote`-Befehl erstellt.

¹vgl. Rowling, J.,K., Harry Potter und der Stein der Weisen, Seite 5

2.3 Strukturierung

Auch Überschriften werden mit Befehlen erzeugt, die Nummerierung übernimmt L^AT_EX. Ein Inhaltsverzeichnis kann mit dem Befehl `\tableofcontents` erstellt werden (*Hyperref* setzt wieder Links auf die einzelnen Abschnitte).

Welche Arten von Abschnitten es gibt, hängt von der gewählten Dokumentklasse ab. Scrartel bietet

- `\section{Section-Titel}`
- `\subsection{Subsection-Titel}`
- `\subsubsection{Subsubsection-Titel}`
- `\paragraph{Paragraph-Titel}`
- `\subparagraph{Subparagraph-Titel}`

Standardmäßig tauchen 3 Gliederungsebenen im Inhaltsverzeichnis auf. Sowohl die Tiefe des Inhaltsverzeichnis als auch die Nummerierung können beliebig angepasst werden, das ist aber Stoff für eigene Nachforschungen.

Oberhalb dieser Hierarchie gibt es noch `\part{Teil-Titel}`. Damit kann ein Dokument weiter in Teile untergliedert werden, wobei die Nummerierung der Sections in einem neuen Teil nicht zurückgesetzt wird, sondern einfach weiterläuft.

Parts sind eher in größeren Projekten relevant.

In den Klassen `scrreprt` und `scrbook` gibt es zusätzlich noch Kapitel, die mit `\chapter{Kapitel-Titel}` erzeugt werden. Kapitel stehen in der Hierarchie zwischen Section und Part.

Hinweis

Wenn die Gliederung verändert wurde, muss zweimal übersetzt werden, damit alle Informationen im Inhaltsverzeichnis aktualisiert werden.

Übung

Erstellt ein Dokument mit Kopf und mindestens 3 Sections, wobei einige davon noch Subsections und Paragraphen enthalten sollten.³

Das Dokument soll auch ein automatisch generiertes Inhaltsverzeichnis enthalten!

Wer schon fertig ist, kann noch mit `scrbook` und `scrreprt` experimentieren. Was passiert, wenn man eine Subsection beginnt, ohne vorher eine Section zu haben? Lässt sich das mithilfe von `renewcommand{\thesubsection}` beheben?

³Tipp: Wer keinen Text schreiben will, kann sich das Paket `lipsum` ansehen.

2.4 Aufzählungen

LaTeX bietet im Wesentlichen zwei Arten von Aufzählungen:

1. Nummerierte Listen
 - Nicht nummerierte Aufzählungen
3. Mischformen⁴

2.4.1 Unnummerierte Listen

Eine Liste ohne automatische Nummerierung wird mit

```
\begin{itemize}
  \item Punkt 1
  \item Punkt 2
  \item Punkt 3
\end{itemize}
```

erzeugt. Das liefert folgende Ausgabe:

- Punkt 1
- Punkt 2
- Punkt 3

Es ist möglich, den • bei einzelnen oder auch allen Items durch andere Zeichen zu ersetzen. `\begin{itemize}[label=gewünschtes Label]... \end{itemize}` ändert das Symbol für alle Items, `\item[gewünschtes Label]` nur für das aktuelle.

```
\begin{itemize}[label=x]
  \item Bla
  \item[y] Bla
  \item Bla
\end{itemize}
```

liefert also

- x Bla
- y Bla
- x Bla

⁴Es gibt 11 Arten von Menschen...

2.4.2 Nummerierte Listen

Nummerierte Listen erhält man so:

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

1. Punkt 1
2. Punkt 2
3. Punkt 3

Auch hier ist es natürlich wie oben möglich, einzelnen Items beliebige andere Labels zu geben. Auch die Art der Nummerierung kann angepasst werden:

```
\begin{enumerate}[label=\alph*]
  \item Punkt 1
  \item Punkt 2
  \item Punkt 3
\end{enumerate}
```

- a) Punkt 1
- b) Punkt 2
- c) Punkt 3

Mögliche Nummerierungstypen sind dabei unter anderem `\alph*` und `\roman*`, wobei `\alph*` die Buchstaben des Alphabets als Labels verwendet und `\roman*` römische Ziffern. Das Label kann sich aus diesen Typen in Verbindung mit beliebigen anderen Zeichen zusammensetzen (vgl. Beispiel: `[label=\alph*]`). Wenn die Label zu lang werden, kann es zu Formatierungsproblemen kommen, die zu beheben etwas Handarbeit erfordert (das Package `enumitem` kann sehr hilfreich sein), sodass man sich hier nicht allzu sehr austoben sollte.

2.4.3 Description

Eine weitere Art von Aufzählung ist die `description`, die im Wesentlichen funktioniert wie ein `itemize`, nur dass die Labels mehr hervorgehoben werden:

```
\begin{description}
  \item[Label 1] Punkt 1
  \item[Label 2] Punkt 2
\end{description}
```

Label 1 Punkt 1

Label 2 Punkt 2

Übung

Bilde die folgende Seite nach!

Hinweis

Hinweis: Der Befehl für die Nummerierung mit arabischen Zahlen ist `\arabic*`.

Aufgabe 1: Bilde den Inhalt der aktuellen Seite genau nach. Auch die Aufgabenzählung ist eine nummerierte Liste! Jede neue Aufgabe (außer dieser) ist auch eine neue Liste.

Aufgabe 2: x Das globale Symbol in dieser Aufzählung soll ein x sein.
• Hier soll ein • verwendet werden.
x Dieses Item hat wieder das globale Symbol.

Aufgabe 3: i. Hier ist eine Aufzählung mit römischen Zahlen.
ii. Nach der römischen Zahl steht ein Punkt.

Zusatzaufgaben:

- Bilde die Seite exakt nach! (Querformat, Schriftgröße 19)
- Welche vordefinierten Möglichkeiten zur automatischen Nummerierung gibt es? (Außer arabic und roman).
- Fortgeschritten: Definiere eine eigene Aufzählungsnummerierung (google/Stackexchange)

2.5 Tabellen

Tabellen in \LaTeX sind ein Thema, mit dem man vermutlich alleine ein Buch füllen könnte. Hier werden nur die grundlegenden Befehle erklärt, die für eine Seminararbeit aber ausreichend sein sollten.

Die einfachste Möglichkeit, eine Tabelle zu erzeugen, führt im Texmaker über den Menüpunkt *Assistent* \rightarrow *Tabellen-Assistent*. Dieser öffnet einen grafischen Tabelleneditor,

in dem man relativ komfortabel seine gewünschte Tabelle füllen und dann Texmaker den Code dafür generieren lassen kann.

Zum besseren Verständnis wird trotzdem der zugehörige Befehl beschrieben:

```
\begin{tabular}{|c|c|}
\hline
Das & ist \\ \hline
eine & Beispieltabelle \\ \hline
\end{tabular}
```

Das	ist
eine	Beispieltabelle

Das Argument `{|c|c|}` beschreibt das Format der Tabelle: Es gibt zwei Spalten, deren Inhalt zentriert wird (dafür das `c`), und die von einer senkrechten Linie getrennt werden (`|`). Auch an den Außenrändern der Spalten ist je eine senkrechte Linie. Wenn man das Argument `{cc}` übergibt, erhält man die gleiche Tabelle, aber ohne Trennlinien zwischen den Spalten und am Rand, `{|cc|}` ergibt Linien am Rand, aber keine zwischen den Spalten. Statt `c` können auch weitere Parameter übergeben werden, die Bedeutung ist in der folgenden Tabelle angegeben:

<code>c</code>	<code>l</code>	<code>r</code>	<code>p{3cm}</code>
Zentrierte Spalte	Linksbündige Spalte	Rechtsbündige Spalte	Spalte mit fester Breite von 3 cm

Im Tabellenkörper sind die wichtigsten Befehle `\hline` (erzeugt eine horizontale Linie, dabei ist es auch möglich, diese nicht über die gesamte Tabellenbreite, sondern beispielsweise mit `\cline{2-3}` nur über die Spalten 2-3 anzuzeigen), `&` (wechselt in die nächste Spalte) und `\\` (erzeugt wie üblich einen Zeilenumbruch).

Übung

Bilde die folgende Tabelle nach!

Der Inhalt kann beliebig sein, das Format muss genau nachgebaut werden.

Kopfzeile	Spaltenname1	Spaltenname2
11	12	13
21	22	23

Übung

Füge einen Rahmen um die Tabelle ein:

Kopfzeile	Spaltenname1	Spaltenname2
11	12	13
21	22	23

Ein weiterer nützlicher Befehl ist `\multicolumn{2}{r|}{Inhalt}`. Damit ist es möglich,

Texte über mehrere Zellen auszubreiten. In der ersten Klammer wird angegeben, über wie viele Zellen sich der Inhalt erstrecken soll, die zweite Klammer gibt ein Format für die große Zelle an (hier rechtsbündig mit einem senkrechten Rand am Ende), in der 3. Klammer wird dann der Inhalt gegeben. Im nächsten Beispiel werden einige der beschriebenen Befehle veranschaulicht:

```
\begin{tabular}{l|cc}
      & & a & & b & \\
      1 & & 1a & & 1b & \\
      2 & & \multicolumn{2}{c}{2ab} & & \\
      3 & & 3a & & 3b & \\
\end{tabular}
```

	a	b
1	1a	1b
2	2ab	
3	3a	3b

Übung

Bilde die folgende Tabelle nach!

11	12	13
21	22	23
31	323	

Übung

Ersetze die erste Trennlinie durch eine gestrichelte Linie! (→ google/Stackexchange)

Ein nützliches Package im Zusammenhang mit Tabellen ist *tabularx*. Es führt eine gleichnamige Umgebung `tabularx` ein, der man eine Breite vorgeben kann. Mit der \LaTeX -eigenen Tabelle ist es recht kompliziert, eine Tabelle genau auf Seitenbreite zu strecken, *tabularx* ermöglicht das folgendermaßen:

```
\begin{tabularx}{\textwidth}{l|X}
      Spalte mit passender Breite & Spalte ueber den Rest der Seite \\
\end{tabularx}
```

Spalte mit passender Breite	Spalte ueber den Rest der Seite	
-----------------------------	---------------------------------	--

Das erste Argument setzt die Breite der gesamten Tabelle fest (hier die aktuelle Textbreite), das zweite Argument ist genau wie oben die Formatierung der Tabelle, mit

dem zusätzlich verfügbaren Parameter X . Eine Spalte mit Format X bekommt den Rest der verfügbaren Platzes zugeteilt. Wenn mehrere Spalten dieses Format haben, wird der Platz gleichmäßig aufgeteilt.

Übung

Baue eine Notentabelle nach dem angegebenen Schema!

Name	Vorname	Note 1	Note 2
Euler	Leonhard	12	13
Hilbert	David	11	15
Noether	Emmy	13	13

Übung

Zusatzaufgabe Erstelle einen zweispaltigen Text (zum Füllen kann das Paket `\lipsum` verwendet werden). Füge in diesen die eben erstellte Tabelle (ohne die beiden Notenspalten) passend ein. Die Tabelle soll außerdem einen Rahmen erhalten. Was passiert, wenn das Dokument wieder einspaltig gemacht wird?

2.6 Verschiedenes

2.6.1 Codestil

Da \LaTeX code recht leicht unübersichtlich wird, ist es notwendig, den eigenen Code so geordnet wie möglich zu halten. Dazu gehören vor allem Einrückungen und sinnvolle Zeilenumbrüche (diese werden nicht übersetzt und können daher zur Verbesserung des Codes genutzt werden). Ein weiterer Baustein sind außerdem wie bei jeder Programmiersprache Kommentare. Kommentare sind dazu da, zu erläutern, was Code bewirkt, vor allem, wenn dies nicht offensichtlich ist. In \LaTeX wird ein Kommentar durch `%` eingeleitet. Der Rest der Zeile nach dem `%` Symbol wird nicht mehr übersetzt und hat daher keinen Einfluss auf das Dokument. Will man ein „%“ schreiben, so verwendet man `\%`. Ein Beispiel:

```
%Dies ist die Zeile von oben:
```

```
Will man ein “\%” schreiben, so verwendet man \verb|\%|.
```

Wird zu

```
Will man ein „%“ schreiben, so verwendet man \%.
```

Hinweis

Wichtig Für \LaTeX endet eine Codezeile erst beim manuellen Umbruch (im \TeX maker auch an der Zeilennummerierung erkennbar). Kommentare werden vom \TeX maker

grau eingefärbt.

2.6.2 Nützliche Befehle

hrule Horizontale Linie über die ganze Textbreite

bigskip Ganze Leerzeile beim nächsten Zeilenumbruch

smallskip Halbe Leerzeile beim nächsten Zeilenumbruch

Etwa so

vfill Füllt den vertikalen Freiraum

hfill Füllt den horizontalen Freiraum

Mehrere **hfills/vfills** teilen den Platz gleichmäßig auf.

2.6.3 Einige Beispiele

```
oben \\
\vfill
unten \hfill rechts
```

```
oben

unten                                rechts
```

```
oben \\
\vfill
mittig \hfill mittigst \hfill rechts mittig \\
\vfill
unten
```

```
oben

mittig                                mittigst                                rechts mittig

unten
```

Mit den bisher gelernten Befehlen können wir uns an den ersten Baustein der Seminararbeit wagen:

Übung

Erstellt das Deckblatt eurer Seminararbeit! Eine Vorlage dazu befindet sich auf der Kursseite im Wiki.

3 Erweiterungen