

## **Bachelorarbeit**

# **Vorlage für eine Abschlussarbeit mit $\LaTeX$**

zur Erlangung des akademischen Grades

Bachelor of Science (B. Sc.)

vorgelegt dem

Fachbereich Mathematik, Naturwissenschaften und Informatik  
der Technischen Hochschule Mittelhessen

von

**Burkhardt Renz**

im Juli 2020

Referent: Prof. Dr. Donald E. Knuth

Korreferent: Prof. Dr. Leslie Lamport



## **Eidesstattliche Erklärung**

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Gießen, 24. Juli 2020



## Zusammenfassung

Dieser Text beschreibt sich in einem gewissen Sinne selbst, nämlich wie die  $\text{\LaTeX}$ -Dateien aussehen, aus denen dieses Dokument erzeugt wird.

Es geht also *nicht* darum, wie man eine Abschlussarbeit gliedert, wie man in ihr argumentiert, wie man Konzepte illustriert usw.usf., sondern *nur* darum, wie man das Manuskript der Arbeit in  $\text{\LaTeX}$  setzt. Deshalb ist anzuraten, dieses Dokument parallel mit seinen Quellen zu lesen, die in der Datei `vorlage.zip` enthalten sind.

Die  $\text{\LaTeX}$ -Datei basiert auf KOMA-Script von Markus Kohm. KOMA-Script verwendet europäische typografische Konventionen. In aller Regel werden in der Vorlage Standardeinstellungen von KOMA-Skript übernommen. Darüber hinaus wird versucht eine möglichst einfache Vorlage zu erstellen, die leicht an eigene Bedürfnisse angepasst werden kann — ohne dass man tiefer gehende  $\text{\LaTeX}$ -Kenntnisse braucht.

Die Verwendung stelle ich mir so vor: Für die eigene Abschlussarbeit kopiert man `vorlage.tex` und passt die Datei entsprechend an. Für den eigentlichen Inhalt der eigenen Arbeit kann man die anderen  $\text{\LaTeX}$ -Dateien als Beispiele nehmen.



# Inhaltsverzeichnis

<b>1 Was ist <math>\LaTeX</math> und wie ist eine <math>\LaTeX</math>-Datei aufgebaut?</b>	<b>1</b>
1.1 Etwas Geschichte – oder ein paar Geschichten . . . . .	1
1.2 Der Aufbau einer $\LaTeX$ -Datei . . . . .	2
1.2.1 Dokumentklasse . . . . .	2
1.2.2 Präambel . . . . .	2
1.2.3 Die Umgebung <code>document</code> . . . . .	3
<b>2 Elemente für die Gliederung</b>	<b>5</b>
2.1 Teile . . . . .	5
2.2 Kapitel und ihre Untergliederungen . . . . .	6
2.2.1 Unterabschnitt . . . . .	6
2.2.1.1 Unterunterabschnitt . . . . .	6
2.3 Untergliederungen im Text . . . . .	6
2.3.1 Aufzählungen . . . . .	6
2.3.2 Stichwortlisten . . . . .	7
2.4 Zum Literaturverzeichnis . . . . .	8
<b>3 Elemente im Text</b>	<b>9</b>
3.1 Typografische Elemente . . . . .	9
3.2 Referenzen und Links . . . . .	9
3.3 Abbildungen . . . . .	9
3.4 Tabellen . . . . .	10
3.5 Listings . . . . .	11
3.6 Mathematische Formeln . . . . .	11
<b>Installation von <math>\LaTeX</math></b>	<b>13</b>
1 Windows . . . . .	13
2 Mac OSX . . . . .	14
3 Linux . . . . .	14
<b>Literaturverzeichnis</b>	<b>15</b>





# Abbildungsverzeichnis

3.1 Donald Knuth . . . . . 10



# Tabellenverzeichnis

3.1 Die Geschichte von T<sub>E</sub>X und L<sup>A</sup>T<sub>E</sub>X . . . . . 10



# Listings

3.1 Einbinden einer Quelldatei . . . . . 11



# 1 Was ist $\LaTeX$ und wie ist eine $\LaTeX$ -Datei aufgebaut?

Der Text dieses Kapitels steht in der Datei `aufbau.tex`. Die Erläuterungen ab Abschnitt 1.2 beziehen sich auf die Datei `vorlage.tex`.

## 1.1 Etwas Geschichte – oder ein paar Geschichten

Die ersten Bände von *The Art of Computer Programming* (TAOCP) von Donald Knuth wurden im Bleisatz gesetzt. In den 1970er Jahren aber starb der Bleisatz aus und wurde durch den Fotosatz ersetzt. Knuth war mit den damaligen Fotosatz-Systemen sehr unzufrieden, weil sie mathematische Formeln nicht gut darstellen konnten. Das brachte ihn auf die Idee selbst ein Satzsystem zu entwickeln, das für Texte der Mathematik und Informatik typografisch hochwertige Ergebnisse erreichen sollte. Zunächst dachte er, ein solches Programm könne in einem Jahr oder so erstellt werden. Es dauerte dann doch etwa länger – siehe Tabelle 3.1.<sup>1</sup>

Knuths Idee [2] bestand darin, dass Text, Formeln und Layout gewissermaßen programmiert werden. Das Satzsystem konsumiert ein solches „Programm“ und macht daraus ein Dokument — heutzutage in der Regel ein PDF-Dokument. Das System sollte natürlich erweiterbar und anpassbar sein, weshalb die Sprache, die Knuth entwickelt hat, im Grunde eine Makrosprache ist und es deshalb auch erlaubt eigene Befehle zu schreiben. Knuth hat einen Satz an Makros entwickelt, den man `PLAIN TEX` nennt – diese Makros sind sehr nahe am Kern von `TEX`, *low level* sozusagen.

Leslie Lamport<sup>2</sup> hat eine Menge von Makros entwickelt, die auf `TEX` aufbauen und das Setzen von Büchern, Berichten und Artikeln erheblich vereinfachen. Diese Makros nannte er `\LaTeX` [4]. Leslie Lamport erzählt über die Entstehung von `\LaTeX` (<http://research.microsoft.com/en-us/um/people/lamport/pubs/pubs.html#latex>):

In the early 80s, I was planning to write the Great American Concurrency Book. I was a `TEX` user, so I would need a set of macros. I thought that, with a little extra effort, I could make my macros usable by others. Don Knuth had begun issuing early releases of the current version of `TEX`, and I figured I could write

---

<sup>1</sup>Die Entwicklung von `TEX` ist auch ein interessanter Fall von Software-Engineering, siehe Donald E. Knuth *The Errors of T<sub>E</sub>X* in: Tom de Marco und Timothy Lister (Hrsg.) *Software State-of-the-Art: Selected Papers* New York NY: Dorste Publishing House, 1990.

<sup>2</sup>Leslie Lamport hat wichtige Beiträge zur Theorie verteilter Systeme geleistet.

what would become its standard macro package. That was the beginning of  $\LaTeX$ .

...

Meanwhile, I still haven't written the Great American Concurrency Book.

$\LaTeX$  trennt im Grunde die Befehle, die den Inhalt und die Struktur eines Texts betreffen von den Befehlen, die für die typografische Gestaltung sorgen.  $\LaTeX$  stellt eine Reihe von sogenannten Dokumentklassen bereit, die das Layout von Artikeln, Berichten, Büchern und Briefen bereits enthalten, so dass man sich beim Schreiben nicht mehr darum kümmern muss.

Die Dokumentklassen von  $\LaTeX$  erzeugen eine Typografie, die an amerikanischen Konventionen orientiert ist. KOMA-Script ist eine Sammlung von Dokumentklassen, die sich an europäischer Typografie orientieren. Frank Neukam entwickelte Anfang der 1990er Jahre Script, das Markus Kohm zu KOMA-Script weiterentwickelte [3]. Wir verwenden für die Vorlage die Dokumentklasse von KOMA-Script für das Setzen von Büchern.

## 1.2 Der Aufbau einer $\LaTeX$ -Datei

Eine  $\LaTeX$ -Datei beginnt mit der Angabe der Dokumentklasse mitsamt ihren Optionen. Darauf folgt die sogenannte Präambel, in der benötigte Pakete angegeben, Einstellungen festgelegt und auch eigene Makros definiert werden. Darauf folgt in der Umgebung `document` der eigentliche Text.

### 1.2.1 Dokumentklasse

Wir verwenden für die Abschlussarbeit die Dokumentklasse `scrbook` von KOMA-Script. Diese Dokumentklasse ist vorgesehen für den Satz von Büchern und sorgt dafür, dass Kapitel immer auf einer rechten (also vorderen) Seite beginnen.

Die Dokumentklasse hat in der Regel geeignete Voreinstellungen für eine Abschlussarbeit. Deshalb ändern wir nur einige wenige der Optionen von `scrbook`. In [5] werden weitere Optionen erläutert.

### 1.2.2 Präambel

In der Präambel binden wir benötigte Pakete ein. Pakete sind vorgefertigte Sammlungen von Makros für  $\LaTeX$ . Es gibt für nahezu jede denkbare Aufgabe solche Pakete, die man im  $\TeX$  Catalogue <http://texcatalogue.ctan.org> finden kann.

Unsere Vorlage bindet die wichtigsten und für eine Arbeit im Feld der Informatik in der Regel benötigten Pakete ein. Auch hierzu findet man weitere Informationen in [5].



Darüber hinaus stehen Einstellungen in der Präambel, bei uns z.B. zur Nummerierungstiefe.

Eigene Makros kann man auch in der Präambel unterbringen.

### 1.2.3 Die Umgebung `document`

Nach der Präambel kommt die Umgebung `document`. In L<sup>A</sup>T<sub>E</sub>X nennt man Blöcke, die durch `\begin{umgebung}` und `\end{umgebung}` begrenzt sind *Umgebung*. In der Regel werden zu Beginn der Umgebung bestimmte Einstellungen ein- und am Ende der Umgebung wieder ausgeschaltet.

Der eigentliche Text der Arbeit steht also in der Klammer

```
\begin{document}

% hierher kommt der eigentliche Text

\end{document}
```

Es ist ratsam, den eigentlichen Text in verschiedene Dateien zu verteilen, etwa pro Kapitel eine Datei. Diese Dateien können dann in die führende Datei eingebunden werden, bei uns durch folgende Befehle:

```
\input{aufbau}      % Kapitel 1
\input{gliederung}  % Kapitel 2
\input{elemente}    % Kapitel 3
```



## 2 Elemente für die Gliederung

Der Text dieses Kapitels steht in `gliederung.tex`. Er bezieht sich auf `vorlage.tex` und `gliederung.tex`.

### 2.1 Teile

Eine Abschlussarbeit besteht aus drei großen Teilen:

- Dem Vorderteil (`frontmatter`) mit
  - der Titelseite,
  - der eidesstattlichen Erklärung,
  - der Zusammenfassung,
  - dem Inhaltsverzeichnis und
  - den Verzeichnissen von Abbildungen, Tabellen und Listings,
- dem Hauptteil (`mainmatter`) mit den Kapiteln der Arbeit und
- dem Anhang (`backmatter`) mit Anhängen und dem Literaturverzeichnis

Manchmal erwarten Dozentinnen oder Dozenten, dass eine Abschlussarbeit Verzeichnisse der Abbildungen, Tabellen und Listings oder auch ein Glossar der verwendeten Begriffe enthält. Deshalb enthält unsere Vorlage diese Verzeichnisse. Ein Glossar kann man mit der Umgebung `description` oder `labeling` erzeugen, die in Abschnitt [2.3.2](#) beschrieben werden.

Ich persönlich finde, dass man auf diese Verzeichnisse verzichten kann. Und ein Glossar ist meines Erachtens nur nötig, wenn man viele Fachbegriffe in der Arbeit verwendet, die einem in der Informatik Kundigen nicht geläufig sind.

## 2.2 Kapitel und ihre Untergliederungen

Die Dokumentklasse `scrbook` hat als Möglichkeiten der Gliederung zunächst den Teil (`part`). Er wird in diesem Dokument nicht verwendet und meistens ist eine Bachelor- oder Masterarbeit nicht so umfangreich, als dass man sie in Teile unterteilen müsste.

Die nächste Ebene ist das Kapitel (`chapter`), wie wir auf der vorigen Seite den Anfang eines solchen sehen. Kapitel beginnen immer auf einer rechten Seite.

Dann kommt der Abschnitt (`section`) — in einem solchen befinden wir uns im Moment.

### 2.2.1 Unterabschnitt

Dies ist eine `subsection`.

#### 2.2.1.1 Unterunterabschnitt

Jetzt sind wir noch eine Ebene tiefer, in der `subsubsection`. Eine Gliederung sollte ausgewogen sein, auch in Bezug auf die Gliederungstiefe. Deshalb sollte man eher nicht bis zum Unterunterabschnitt gehen.

**Absatz mit Überschrift** Dies ist ein Absatz mit Überschrift `paragraph`. Die Überschrift wird im Dokument hervorgehoben, hat aber keine eigene Zeile. Typografen nennen das auch einen „Spieß“.

**Unterabsatz** Dies ist ein Unterabsatz `subparagraph`, auch ein „Spieß“.

## 2.3 Untergliederungen im Text

### 2.3.1 Aufzählungen

Man kann nummerierte Aufzählungen mit der Umgebung `enumerate` erzeugen. Dies eignet sich für Aufzählungen, die eine Reihenfolge haben und ist oft einer Aneinanderreihung im Text vorzuziehen, weil übersichtlicher.

Eine Bachelorarbeit besteht aus

1. einem ersten Kapitel
  - a) einem ersten Abschnitt darin,
  - b) einem zweiten Abschnitt darin,
2. einem zweiten Kapitel

3. usw. usf.

Aufzählungen, die keine inhaltliche Reihenfolge haben, kann man mit der Umgebung `itemize` darstellen:

- eine wichtige Aussage
- noch eine wichtige Aussage
  - mit einer Ausprägung
  - und noch einer Ausprägung
- ...

### 2.3.2 Stichwortlisten

Ein Beispiel für die Umgebung `description` habe ich aus der Dokumentation von KOMA-Script übernommen:

**empty** ist der Seitenstil, bei dem Kopf- und Fußzeile vollständig leer bleiben.

**plain** ist der Seitenstil, bei dem keinerlei Kolummentitel verwendet wird.

**headings** ist der Seitenstil für automatische Kolummentitel.

**myheadings** ist der Seitenstil für manuelle Kolummentitel.

Es gibt auch noch die Umgebung `labeling`, bei der man durch ein Muster angeben kann, wie breit die Einrückung ist. Im Beispiel mit den Seitenstilen könnte man das so machen:

**empty** ist der Seitenstil, bei dem Kopf- und Fußzeile vollständig leer bleiben.

**plain** ist der Seitenstil, bei dem keinerlei Kolummentitel verwendet wird.

**headings** ist der Seitenstil für automatische Kolummentitel.

**myheadings** ist der Seitenstil für manuelle Kolummentitel.

## 2.4 Zum Literaturverzeichnis

In der Vorlage findet man das Literaturverzeichnis in der Datei `litverz.tex`.

Das Literaturverzeichnis kann man in  $\text{\LaTeX}$  auf zwei Arten erstellen:

Die einfache Variante besteht darin, dass man die Umgebung `bibliography` verwendet. So haben wir das in diesem Dokument gemacht. Die Angaben zur Literatur stehen in `litverz`. Jeder Eintrag hat einen *key*, den wir im Text im Befehl `\cite` referenzieren können. Die Referenz wird dann als Nummer im Text angegeben und die entsprechende Nummer erscheint auch im Literaturverzeichnis.

Die etwas aufwändigere Variante besteht darin, `bibtex` zu verwenden. Dies lohnt sich insbesondere dann, wenn man in verschiedenen Manuskripten immer wieder dieselben Literaturverweise verwendet.

Mit `bibtex` speichert man die Literaturangaben in einer Art Datenbank, einer BibTeX-Datei. Die im Manuskript referenzierten Arbeiten werden dann von `bibtex` automatisch in das Literaturverzeichnis übernommen. Die Formatierung des Verweises im Text und der Einträge im Literaturverzeichnis wird dabei durch eine eigene Datei, dem BibTeX-Stil, gesteuert. Will man die Verweise durch Nummern, wählt man als Stil `plain`, will man die Verweise durch die Abkürzung des Autorennamens mit Angabe des Jahres, wählt man z.B. den Stil `alpha`. Es gibt eine Vielzahl solcher BibTeX-Stile, siehe <https://www.ctan.org/tex-archive/biblio/bibtex/contrib>. Man kann sogar eigene BibTeX-Stile definieren mit `tex makebst`.

Mehr über BibTeX findet man z.B. bei [https://de.wikibooks.org/wiki/LaTeX-Kompodium:\\_Zitieren\\_mit\\_BibTeX](https://de.wikibooks.org/wiki/LaTeX-Kompodium:_Zitieren_mit_BibTeX).

## 3 Elemente im Text

Der Text dieses Kapitels steht in `elemente.tex` und bezieht auch auf diese Datei.

### 3.1 Typografische Elemente

Im laufenden Text kann man alles Mögliche machen. Gute Typografie geht mit diesen Möglichkeiten sparsam um. Was in Texten der Informatik oft vorkommt ist die *Hervorhebung* mit dem Befehl `\emph`. Man verwendet kursive Schrift auch für fremdsprachige Worte im Text, etwa *digital typography*. Für Schlüsselworte, Befehle der Kommandozeile u.ä. nimmt man gerne einen passenden Zeichensatz, etwa `make vorlage.pdf`.

Das Paket `csquotes` sorgt dafür dass Anführungszeichen typografisch korrekt verwendet werden. Im Deutschen werden die „Gänsefüßchen“ nämlich anders als die angelsächsischen „quotation marks“ gesetzt.

Mehr über das Setzen von Text findet man in [1].

### 3.2 Referenzen und Links

Verweise auf die Literatur macht man mit dem Befehl `\cite`. Braucht man Querverweise auf Abschnitte im Text oder Abbildungen etc., so versieht man das Verweisziel mit einem `\label` und verweist dann mit `\ref` oder `\pageref`. Mehr dazu in [5, S.50].

Links auf Quellen im Internet werden durch den Befehl `\url` angegeben und dadurch in PDF „anklickbar“, etwa wie <https://tug.org/mactex/src/WelcomeToMacTeX.pdf>.

### 3.3 Abbildungen

Abbildungen bindet man mit `\includegraphics` in einer Umgebung `figure` ein. Man kann dann im Text auf die Abbildung 3.1 verweisen. Man muss beachten, dass die Abbildung in einer sogenannten „fließenden“ Umgebung eingebunden wird, d.h. beim Setzen des Dokuments bestimmt  $\text{T}_{\text{E}}\text{X}$ , an welcher Stelle genau im Dokument die Abbildung erscheint.

Wie man an diesem Beispiel sieht, kann man beliebige Abbildungen etwa `jpg` wie in diesem Beispiel, aber auch `pdf` einbinden. Man kann also Abbildungen mit einem Grafikprogramm erstellen, als `pdf` speichern und dann einbinden.



Abbildung 3.1: Donald Knuth

Es gibt aber auch TikZ (= TikZ ist kein Zeichenprogramm), das der deutsche Informatiker Till Tantau entwickelt hat. Damit ist es möglich, Abbildungen zu „programmieren“. Die Projektseite von TikZ ist <https://sourceforge.net/projects/pgf/>. Interessant sind auch die Beispiele auf <http://www.texample.net/tikz/examples/>.

### 3.4 Tabellen

Tabellen werden in einer „fließenden“ Umgebung namens `table` eingegeben. Der eigentliche Inhalt der Tabelle kommt in die Umgebung `tabular`.

Tabelle 3.1: Die Geschichte von  $\text{T}_{\text{E}}\text{X}$  und  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 

Jahr	Entwicklung
1977	Donald Knuth beginnt mit der Entwicklung von $\text{T}_{\text{E}}\text{X}$ .
1985	$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (mächtige Makros, die die Verwendung von $\text{T}_{\text{E}}\text{X}$ vereinfachen, in dem sie Struktur und Layout trennen) wird von Leslie Lamport in der Version 2.09 freigegeben.
1986	Feier der Fertigstellung von $\text{T}_{\text{E}}\text{X}$ im Computer Museum in Boston
1986	Leslie Lamport veröffentlicht <i>L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System</i> .
1993	$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$
2000	pdf $\text{T}_{\text{E}}\text{X}$ (entwickelt von Hàn Thé Thành)
?	Nach dem Tod von Donald Knuth bekommt $\text{T}_{\text{E}}\text{X}$ die Versionsnummer $\pi$ .

Wie man am Beispiel des  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Texts der Tabelle 3.1 sieht, kann die Formatierung von



Tabellen etwas „sperrig“ sein. Gut, dass man sich in einer der vielen Anleitungen dazu erkundigen kann, z.B. in der  $\text{\LaTeX}$  2 $\epsilon$ -Kurzbeschreibung [1, S.23].

## 3.5 Listings

`listings` ist ein Paket, das es erlaubt, Code-Beispiele in die Abschlussarbeit zu setzen. Wie das Beispiel 3.1 zeigt, kann man auch externe Dateien einbinden und im Text verbatim einsetzen. In diesem Fall kann der eingebundene Text auch in der Zeichenkodierung `utf8` sein, wohingegen dies bei direkt im Text geschriebenen Codebeispielen nicht unterstützt wird.

Listing 3.1: Einbinden einer Quelldatei

```
% Der Inhalt der Datei myclass.java wird verbatim hier eingefügt
\lstinputlisting[caption={Eine Java-Klasse}\label{lst:java}]{myclass.java}
```

Für Listings gibt es viele Optionen des Layouts, insbesondere ist es möglich die Programmiersprache des eingebundenen Code-Beispiels anzugeben, was dazu führt, dass Schlüsselworte, Bezeichner und Kommentare der Sprache im Layout hervorgehoben werden.

## 3.6 Mathematische Formeln

Mathematisches kann man in den laufenden Text einbauen, wie etwa bei folgender Definition:

Mit  $k!$ , der Fakultät einer natürlichen Zahl  $k$  bezeichnet man das Produkt  $1 \cdot 2 \cdot \dots \cdot k$ .

Oft braucht man aber auch ganze Abschnitte im Mathematik-Modus, wie in folgendem Beispiel:

Die sogenannte Collatz-Folge<sup>1</sup> zu einer natürlichen Zahl  $n$  wird folgendermaßen gebildet:

$$n_1 = n$$

$$n_{i+1} = \begin{cases} n_i/2 & \text{falls } n_i \text{ gerade} \\ 3n_i + 1 & \text{falls } n_i \text{ ungerade} \end{cases}$$

Startet man etwa mit der Zahl 7 erhält man

7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1 4 2 1 ...

<sup>1</sup>nach Lothar Collatz, deutscher Mathematiker 1910 - 1990

Wie man sieht, geht die Folge schließlich in den Zyklus 1, 4, 2 über. Die *Collatz-Vermutung* besagt, dass dies für jeden Startwert  $n$  der Fall ist, d.h. jede Collatz-Folge erreicht irgendwann den Wert 1.

Mehr über das Setzen von mathematischen Formeln steht in [1, Kapitel 4].

# Installation von L<sup>A</sup>T<sub>E</sub>X

Der Text dieses Anhangs steht in `install.tex`.

## 1 Windows

### Installationsschritte

1. Downloadseite von MiKTeX <http://www.miktex.org/download>.
2. Die Basisversion **Basic MiKTeX 2.9 Installer** herunterladen.
3. Die Installationsdatei namens `basic-miktex-2.9.xxxx.exe` starten.
4. Dem Installationswizard folgen (am einfachsten die vorgeschlagenen Werte für Verzeichnisse usw. übernehmen).
5. Einen Ordner für die eigenen Dokumente erstellen, z.B. in *Eigene Dokumente*.

### Erste Schritte mit T<sub>E</sub>Xworks

1. Im Startmenü oder den Apps das Programm *TeXworks* suchen und starten. Optional: Zur späteren Bequemlichkeit das Programm an die Taskleiste anheften.  
Es erscheint das Editierfenster auf der linken Hälfte des Bildschirms.
2. Zum ersten Ausprobieren im Menü *File* den Unterpunkt *New from Template* auswählen und in dem dann erscheinenden Dialog *Basic LaTeX documents* und *article.tex*
3. Die Datei wird im Editorfenster geöffnet. Um daraus das PDF-Dokument zu erstellen drückt man auf den grünen Button links oben.  
Nun öffnet sich ein Dialog zum Abspeichern des Dokuments. Danach:
4. Die LaTeX-Datei wird übersetzt. Im linken Fenster unten sieht man die Meldungen über den Fortschritt dieses Vorgangs. (Beim ersten Mal dauert das relativ lange, weil diverse Pakete für LaTeX aus dem Internet heruntergeladen werden.)
5. Nach einiger Zeit erscheint das Ergebnis in einem neuen Fenster auf der rechten Seite des Bildschirms.

6. Jetzt kann es losgehen: links editieren, Erstellen des Dokuments mit dem grünen Button starten und rechts das Ergebnis überprüfen.

## 2 Mac OSX

### Installation von MacT<sub>E</sub>X

1. Das Package MacTeX.pkg erhältlich bei <http://www.tug.org/mactex/> herunterladen.
2. Öffnet man das Paket mit Doppelklick, startet die Installation und sie wird schrittweise durchgeführt.

### Arbeiten mit TeXShop

Nach der Installation hat man im Launchpad eine App namens TeXShop. Dies ist ein Editor für T<sub>E</sub>X und L<sup>A</sup>T<sub>E</sub>X.

## 3 Linux

Die einfachste Variante besteht darin, eine komplette T<sub>E</sub>XLive-Installation durchzuführen mittels

```
sudo apt-get install texlive-full
```

Dabei wird allerdings vieles installiert, das man voraussichtlich niemals braucht. Andererseits muss man aber nicht wissen, was man mindestens installieren muss, damit L<sup>A</sup>T<sub>E</sub>X verwendet werden kann.

Auf Github findet man Skripte, die Installationen z.B. unter Ubuntu steuern können. Beispiel: <https://github.com/scottkosty/install-tl-ubuntu>. Ich habe aber keine Ahnung, wie gut diese Skripte tatsächlich sind.

# Literaturverzeichnis

- [1] Marco Daniel, Patrick Gundlach, Walter Schmidt, Jörg Knappen, Hubert Partl und Irene Hyna *L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>-Kurzbeschreibung*, <http://mirror.unicorncloud.org/CTAN/info/lshort/german/l2kurz.pdf>, 2015.
- [2] Donald E. Knuth *The T<sub>E</sub>Xbook*, Reading, MA: Addison-Wesley, 1986.
- [3] Markus Kohm *Die Anleitung KOMA-Script*, <http://www.komascript.de/~mkohm/scrguide.pdf> 2016.
- [4] Leslie Lamport *L<sup>A</sup>T<sub>E</sub>X: a document preparation system*, 2nd edition, Reading, MA: Addison-Wesley, 1994.
- [5] Günter Partosch *Anforderungen an wissenschaftliche Abschlussarbeiten und wie sie mit L<sup>A</sup>T<sub>E</sub>X gelöst werden können*, <https://www.staff.uni-giessen.de/partosch/unterlagen/abschlussarbeit.pdf>, 2015.