

**mitp**

**3., überarbeitete Auflage**

Joachim Schlosser

inklusive CD-ROM



# Wissenschaftliche Arbeiten schreiben mit

# LATEX

Leitfaden für Einsteiger

Schnell zur fertig gesetzten Arbeit –  
ohne Vorkenntnisse

Lösungsorientierte und verständliche  
Beschreibungen

Auf CD: LaTeX, Editor, Vorlagen und Beispiele

# Erste Schritte in $\text{\LaTeX}$

## 3.1 Das Arbeiten mit $\text{\LaTeX}$

Wie eingangs bereits beschrieben ist die Arbeitsweise mit  $\text{\LaTeX}$  eine andere als mit Textverarbeitungsprogrammen. Letztere zwingen den Benutzer, Layout und Aussehen der einzelnen Teile gleich interaktiv bei der Eingabe festzulegen. Anders bei  $\text{\LaTeX}$ : Hier verfassen Sie den reinen Text. Bezüglich des Layouts bekundet man zunächst seine Absichten mit Hilfe von »Befehlen«. Das können einzelne Befehle sein oder ganze sogenannte Pakete. Damit können Sie zum Beispiel das Layout von Briefen, Artikeln oder sonstiger Dokumente festlegen. Die Befehle können vollkommen unabhängig vom Erstellen des Textes benutzt werden. Erst nach dem Erfassen des Textes liest der *Übersetzer* diese Befehle und setzt sie in Layout um. Der Autor muss sich also nur um Inhalt und Struktur des Dokuments kümmern, das Aussehen wird von  $\text{\LaTeX}$  erzeugt. (Selbstverständlich haben Sie die Möglichkeit, das Aussehen ebenfalls über Befehle zu beeinflussen.) Sie öffnen also nicht etwa ein Programm namens  $\text{\LaTeX}$  und schreiben dort Ihren Text (obwohl Sie auch das tun könnten), sondern arbeiten mit mehreren Programmen. Zum Schreiben verwenden Sie dabei einen separaten Editor, den Sie frei wählen können, je nachdem, mit welchem Sie am besten zurecht kommen. Im *Editor* verfassen Sie Ihr Dokument und geben die Befehle ein. Ein anderes Programm, der  $\text{\LaTeX}$ -Prozessor, übersetzt die Eingabe dann in das gewünschte Ausgabeformat, beispielsweise PDF.

## 3.2 Der Editor

Der *Editor* ist das Programm, in dem Sie Ihren Text schreiben und die Befehle einfügen. Prinzipiell ist dazu ein simpler Texteditor ausreichend wie beispielsweise *Notepad* unter Windows oder *Kedit* unter GNU/Linux.

Um sich das Schreiben jedoch so angenehm wie möglich zu gestalten, wählen Sie besser einen Editor mit  $\text{\LaTeX}$ -Unterstützung. Solche Editoren helfen bei der Auswahl der Befehle, beim Erstellen des Dokuments und beim

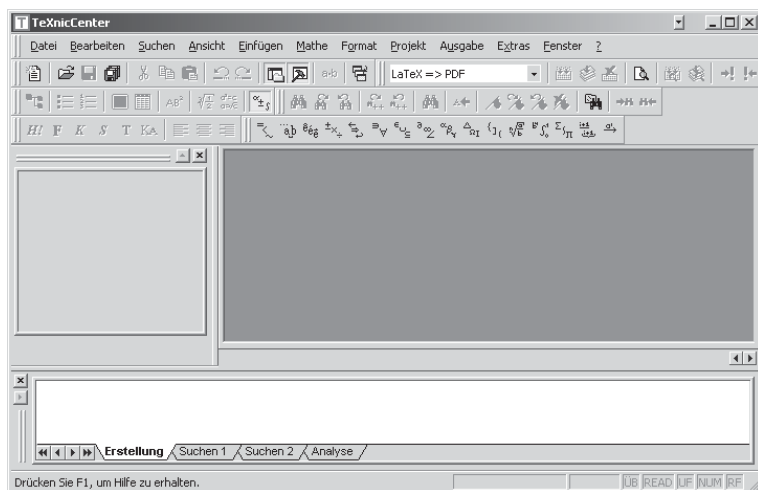
Zusammenspiel der einzelnen Dateien. Beispiele dafür sind Emacs<sup>1</sup>, Kile, WinEdt oder das in diesem Buch für Einsteiger empfohlene TeXnicCenter.

Ebenfalls in der Verantwortung des Editors liegt – sofern gewünscht – die Rechtschreibprüfung.

Selbst wenn man schon eine Weile mit  $\text{\LaTeX}$  arbeitet, sollte man sich nicht scheuen, den Editor zu wechseln, wenn man einen besseren findet. »Besser« ist dabei natürlich von den einzelnen Bedürfnissen abhängig. Der eine arbeitet lieber mit der Maus und entsprechenden Knöpfen, der andere hat lieber Tastaturkürzel und eine integrierte Voransicht. Das Schönste daran: Für jeden gibt es einen passenden Editor!

In diesem Buch gehen wir ein Stück weit auf den Editor TeXnicCenter ein, der für Windows verfügbar ist und für den Einsteiger viele Hilfen bietet. Für GNU/Linux sei an dieser Stelle Kile empfohlen, der ungefähr in derselben Liga spielt.

Um Ihr erstes  $\text{\LaTeX}$ -Dokument erstellen zu können, öffnen Sie also zunächst einmal Ihren Editor. Sofern Sie das TeXnicCenter verwenden, sehen Sie jetzt in etwa Abb. 3.1 vor sich.



**Abb. 3.1:** Bildschirmfoto des TeXnicCenter-Hauptfensters

<sup>1</sup>Die Installation von Emacs unter Windows habe ich in meinem Online-Tutorial beschrieben (Schlosser, 2008).

## 3.3 Ein Dokument erstellen und übersetzen

Erzeugen Sie mittels der entsprechenden Schaltfläche ein neues Dokument. Geben Sie jetzt den folgenden Quelltext ein. Um Tippfehler zu vermeiden, können Sie ihn auch direkt von der Begleit-CD ins TeXnicCenter kopieren.

```

1 \documentclass{scrartcl}
2 \begin{document}
3 \section{Ein Abschnitt}
4 Das ist mein erstes Dokument.
5 \end{document}

```

Auf den ersten Blick sieht das Dokument wahrscheinlich ein bisschen kryptisch aus. Welchem Zweck die Befehle im Einzelnen dienen, erfahren Sie später in Kapitel 4 auf Seite 37. Speichern Sie das Dokument jetzt erst einmal als `beispiel0.tex` ab.

Nachdem Sie Ihr Dokument nun verfasst und abgespeichert haben, verwenden Sie einen Übersetzer, um daraus eine Ausgabedatei zu erzeugen. Mehrere Ausgabeformate sind möglich: So kann entweder DVI erzeugt werden – ein L<sup>A</sup>T<sub>E</sub>X-eigenes Format –, welches dann nach PostScript weiterverarbeitet werden kann, oder aber PDF (Adobes »Portable Document Format«).

Welches Format Sie wählen sollten, hängt zunächst einmal davon ab, was Sie mit dem fertigen Dokument tun möchten. Sie sollten allerdings auch beachten, dass hier zwei verschiedene Übersetzer zum Einsatz kommen (nämlich L<sup>A</sup>T<sub>E</sub>X und pdfL<sup>A</sup>T<sub>E</sub>X), die teilweise unterschiedlich arbeiten und vor allem unterschiedliche Grafikformate verarbeiten, worauf später in Abschnitt 8.1 auf Seite 131 genauer eingegangen wird.

Enthält die Datei einen fehlerhaften Befehl, den der Übersetzer nicht verarbeiten kann, gibt dieser eine Fehlermeldung aus und unterbricht den Übersetzungsvorgang (auch L<sup>A</sup>T<sub>E</sub>X-Lauf genannt). Genauereres hierzu finden Sie in Abschnitt 15.2 auf Seite 271.

Oft ist es notwendig, zwei oder mehr Übersetzungsvorgänge durchzuführen, um Dinge, die L<sup>A</sup>T<sub>E</sub>X nicht gleich beim ersten Mal komplett erledigen kann (z. B. das Indizieren von Überschriften), vollständig abzuarbeiten.

Beim aktuellen Beispiel müssen Sie das Dokument aber nur einmal übersetzen. Im Folgenden wird zunächst erklärt, wie Sie dazu im TeXnicCenter vorgehen müssen. Für den Fall, dass Sie einen Editor verwenden, der diese Funktion nicht vorsieht, ist aber auch die alternative Vorgehensweise über die Kommandozeile beschrieben.

**TeXnicCenter** In TeXnicCenter gibt es dazu das Auswahlfeld »Ausgabeprofil«. Wählen Sie das Ausgabeformat PDF, und drücken dann den Button für AKTIVES DOKUMENT COMPILIEREN.

**Kommandozeile** Ganz allgemein, das heißt ohne an einen bestimmten Editor gebunden zu sein, können Sie auch Folgendes tun: Öffnen Sie eine Kommandozeile unter Windows mit **START | PROGRAMME | ZUBEHÖR | EINGABEAUFFORDERUNG** und wechseln Sie in das Verzeichnis, in dem Sie Ihre Datei gespeichert haben, beispielsweise mit

```
cd c:\Textdateien
```

Unter Linux haben Sie meist auf dem Desktop ein Shell-Icon und wechseln das Verzeichnis mit

```
cd ~/textdateien
```

Geben Sie nun folgenden Befehl ein:

```
pdflatex beispiel0.tex
```

Wollen Sie PostScript über die Kommandozeile erzeugen, so benötigen Sie zwei Schritte: Zunächst geben Sie

```
latex beispiel0.tex
```

ein, um das Zwischenformat DVI zu erzeugen, bevor Sie mit

```
dvips beispiel0.dvi
```

die PostScriptdatei erhalten.

Welchen Weg auch immer Sie beschritten haben, in Ihrem Verzeichnis sollte es jetzt eine Datei namens `beispiel0.pdf` oder `beispiel0.ps` geben. Diese können Sie nun anzeigen lassen.

## 3.4 Der Betrachter

Die erzeugte Ausgabedatei `beispiel1.ps` bzw. `beispiel1.pdf` können wir mit `GSview` bzw. `Adobe Reader` betrachten, je nachdem, welches Ausgabeformat gewählt wurde.

Auch für DVI-Dateien gibt es Betrachter. DVI-Dateien haben den Vorteil, dass sie relativ klein sind und schneller erzeugt werden als die endgültigen PostScript- oder PDF-Dateien. Aber Vorsicht: der DVI-Betrachter zeigt nur eine *Voransicht*. Will heißen: Zeilenumbrüche usw. sind korrekt, grafische Elemente wie Linien oder Grafiken können jedoch durchaus einmal anders angezeigt werden als in der späteren Ausgabe. Deswegen sollte eine Endkontrolle immer auch mit dem Betrachter für das endgültige Format durchgeführt werden, also mit dem PDF- oder PostScriptbetrachter.

## 3.5 Das Dokument

Ein weiteres Beispiel eines Dokuments ist nachfolgend zu sehen (mit Zeilennummern zur besseren Lesbarkeit, diese sind natürlich nicht im Dokument enthalten!):

```

1 \documentclass[a4paper]{scrartcl}
2 \usepackage[T1]{fontenc}
3 \usepackage[latin1]{inputenc}
4 \usepackage[ngerman]{babel}
5 \begin{document}
6 \author{Max Muster}
7 \title{Mein erstes Dokument}
8 \maketitle
9 \tableofcontents
10
11 \section{Ein Abschnitt}
12 Dieser Abschnitt hat ein wenig Text, und zwar gerade so viel, dass
13 es auch zu einem \emph{Zeilenumbruch} kommt.
14
15 Ein zweiter Absatz wird durch mindestens eine Leerzeile vom ersten
16 getrennt.
17
18 \section{Noch ein Abschnitt}
19 "‘Wunderbar,“' sagte er dazu, "‘das sehe ich zum 1.~Mal!“'
20
21 \subsection{Ein Unterabschnitt}
22 Damit kann man tolle Sachen machen:
23
24 \begin{enumerate}
25 \item schreiben
26 \item strukturieren
27 \item sich freuen
28 \end{enumerate}
29 \end{document}

```

In diesem zweiten Beispiel sind bereits die meisten wesentlichen Grundelemente eines Dokuments zu sehen, die hier kurz erklärt, später aber noch detaillierter abgehandelt werden.

**Befehle** sind alle Anweisungen, die L<sup>A</sup>T<sub>E</sub>X verarbeiten soll. Ein Befehl beginnt immer mit einem Backslash-Zeichen (»\«). Ein Beispiel ist der

`\section`-Befehl in Zeile 11, der dort einen neuen Abschnitt entstehen lässt. Befehle können *obligatorische* und *optionale* Argumente haben. *Obligatorische* Argumente werden in geschweiften Klammern (`{ }`) angegeben und sind für den jeweiligen Befehl unbedingt notwendig, *optionale* Argumente werden in eckigen Klammern (`[ ]`) angegeben und können wahlweise auch weggelassen werden.

Mehr dazu finden Sie in Kapitel 4 auf Seite 37.

**Umgebungen** beginnen immer mit `\begin{umgebung}`<sup>2</sup> und enden mit `\end{umgebung}`. Umgebungen erlauben es, längere Textteile mit bestimmten Eigenschaften zu versehen. Ein Beispiel dafür ist die `enumerate`-Umgebung in den Zeilen 24 bis 28, die eine Aufzählung beschreibt.

Mehr dazu finden Sie ebenfalls in Kapitel 4 auf Seite 37.

**Präambel und Textteil** bilden die zwei Teile eines L<sup>A</sup>T<sub>E</sub>X-Dokuments. Als Präambel werden alle Zeilen bis zum `\begin{document}` bezeichnet. Sie legt alle wesentlichen Eigenschaften des Dokuments fest, also das prinzipielle Aussehen, die Dokumentart etc. Danach folgt – in der `document`-Umgebung – der Textteil, in dem das eigentliche Dokument steht.

**Präambel** In der Präambel wird das Aussehen des Dokuments und seine Art festgelegt. Dazu wählen Sie eine *Dokumentklasse* aus, laden *Zusatzpakete* und verwendet eventuell einige weitere Befehle.

**Klasse** Üblicherweise in der ersten Zeile der Datei wird die Klasse und damit die Dokumentart festgelegt. Der Befehl `\documentclass[a4paper]{scrartcl}` legt die Klasse `scrartcl` als Dokumentart fest (obligatorisches Argument) und gibt außerdem die Option `a4paper` mit (optionales Argument). Das bedeutet, unser Beispieldokument soll das Layout eines Artikels haben und DIN A4-Format aufweisen.

Neben der Klasse der Artikels gibt es Klassen für Bücher, für wissenschaftliche Arbeiten, für Briefe, für Präsentationen und für fast alles, was man sich sonst an Dokumenten vorstellen kann.

Mehr dazu finden Sie in Kapitel 6 auf Seite 85.

---

<sup>2</sup>Oft wird in der Informatik in Syntaxbeschreibungen bei variablen Anteilen mit spitzen Klammern gearbeitet, so dass man `umgebung>\begin<` schriebe. Da ich jedoch festgestellt habe, dass dies viele Leser verwirrt, lasse ich die Klammern weg.

**Pakete** stellen Zusatzfunktionalitäten zur Verfügung. Sie werden mit dem `\usepackage`-Befehl eingebunden. Im Beispiel:

Das `fontenc`-Paket, das mit der Option `T1` aufgerufen wird, aktiviert die europäischen Zeichensätze, was z. B. der Silbentrennung zugute kommt. Das `inputenc`-Paket mit der Option `latin1` erlaubt die direkte Eingabe von Umlauten und anderen Sonderzeichen. Andernfalls müssten wir anstelle von `ä` etwa `\"a` schreiben. Schließlich stellen wir mit `babel` auf die deutsche Sprache `german` um. Damit stehen deutsche Trennmuster zur Verfügung und alle etwaigen Bezeichnungen im ausgegebenen Dokument sind auf Deutsch. Für die neue Rechtschreibung verwenden Sie `ngerman` statt `german`.

**Titelei** ist alles, was zum Titel gehört. Also zum Beispiel der Titel des Dokuments (`\title`) und der Autor (`\author`). Mit dem Befehl `\maketitle` werden diese Informationen entsprechend aufbereitet und gesetzt. Mehr dazu in Abschnitt 6.4 auf Seite 101.

**Struktur** Wir können das Dokument mit Befehlen wie `\section` oder `\subsection` strukturieren. Die Nummerierung erfolgt automatisch. Ein Inhaltsverzeichnis erzeugen wir daraus mit `\tableofcontents`. Wie das im Einzelnen funktioniert, steht in Abschnitt 4.2.2 auf Seite 43.

**Textauszeichnung** heißt es, wenn Sie einen Textteil besonders behandeln möchten. Wollen wir etwa ein Wort hervorheben (engl. »emphasize«), so können wir dies mit dem Befehl `\emph{}` erreichen wie in Zeile 13.

Anführungszeichen gibt es in vielen Variationen, die deutschen Anführungszeichen können mit `"` (Gänsefüßchen Akzent rechts) und `'` (Gänsefüßchen Apostroph) erzeugt werden, die französischen mit `">` und `"<`.

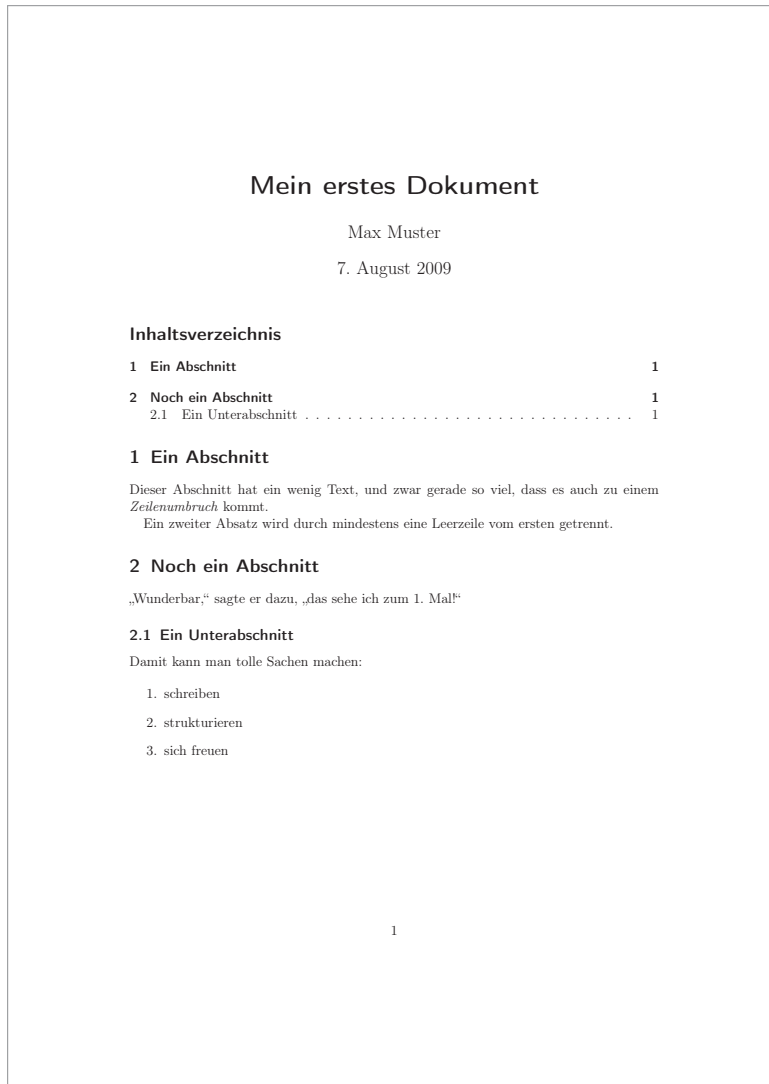
Mehr dazu finden Sie in Abschnitt 4.2.4 auf Seite 47.

**Leerraum.** Zwei Absätze werden durch beliebig viele Leerzeilen getrennt.  $\text{\LaTeX}$  erkennt dann von selbst, dass hier ein neuer Absatz beginnt. Ebenso können zwischen Wörtern beliebig viele Leerzeichen stehen, der Wortabstand wird von  $\text{\LaTeX}$  selbst berechnet.

Möchten Sie einen bestimmten Wortzwischenraum, so können Sie dies angeben. Das Zeichen `~` in Zeile 19 erzeugt einen sogenannten *geschützten Leerraum*, d. h. die beiden dadurch verbundenen Worte stehen immer in einer Zeile. Weitere Möglichkeiten in diesem Umfeld werden später in Abschnitt 4.3 auf Seite 50 erläutert.

**Listen** sind Umgebungen und heißen etwa `enumerate` für nummerierte Listen und `itemize` für Aufzählungen. Die einzelnen Listenelemente werden mit dem Befehl `\item` eingeleitet. Alles zu Listen gibt es in Abschnitt 4.4 auf Seite 56.

Diese grundlegenden Befehle sind in der einen oder anderen Form in wohl jedem Dokument vorhanden. In den folgenden Kapiteln werden diese sukzessive eingeführt. Das fertige Dokument können Sie in Abb. 3.2 sehen.



**Abb. 3.2:** Ausgabe für Beispieldokument 1