Kapitel 1

Warum LaTeX?

1.1 Inhalt und Layout: Zwei separate Aufgaben

Wenn Sie dieses Buch in Händen halten, haben Sie Ihre Entscheidung wahrscheinlich bereits getroffen: Sie wollen eine wissenschaftliche Arbeit schreiben, und zwar nicht mithilfe gängiger Textverarbeitungsprogramme, sondern mit LaTeX. Warum ist das eine gute Entscheidung? Einige Fakten:

LaTeX (ausgesprochen »Lah-Tech«) ist anders als Textverarbeitungsprogramme. Ganz anders. In einem Textverarbeitungsprogramm wie Word™ schreiben Sie Ihren Text und bestimmen gleichzeitig das Aussehen Ihres Dokuments. Da Sie jede Änderung des Dokuments sofort sehen können, spricht man hier gemeinhin von *What you see is what you get* (WYSIWYG).

In LaTeX sehen Sie hingegen die Ausgabe während des Schreibens nicht immer vor sich. Die Gestaltung der Textabschnitte steht also zunächst einmal nicht im Vordergrund, sondern es wird der reine Text geschrieben und zusätzlich ein sogenanntes *Markup*. Das bedeutet, man fügt spezielle Layoutbefehle ein. Der Gestalter – LaTeX – nimmt dieses *Markup* und interpretiert es, stellt das Layout zusammen und legt fest, wie die einzelnen Textteile auszusehen haben. Der Setzer schließlich – TeX – erstellt die druckfertige Ausgabe aus den Anweisungen des Gestalters.

Das hat natürlich auch ein paar wenige Nachteile. Zum einen ist der größte zweifelsohne der, dass man die Layoutbefehle kennen muss – wobei man sich von sogenannten LaTeX-Editoren unterstützen lassen kann. Zum anderen kann das Dokument gerade bei umfangreichen Texten unter Umständen während der Eingabe unübersichtlicher werden, da bei vielen Editoren die Tabellen, Abbildungen und Formeln nicht sofort angezeigt werden. Warum geht das nicht in einem Aufwasch wie bei Word? Warum ist es die Mühe wert?

An ein Dokument, insbesondere an eine wissenschaftliche Arbeit, wird nicht nur ein inhaltlicher Anspruch gestellt. Es geht auch darum, dass das Lesen ermüdungsfrei und angenehm gestaltet ist. Das Layout des Dokuments soll also nicht besonders auffällig oder künstlerisch extravagant sein, sondern in erster Linie diesen Zweck erfüllen. Nicht umsonst war und ist

die Gestaltung eines Schriftstücks und dessen Satz eine Handwerkskunst, die seit Jahrhunderten gelehrt wird.

Seit es Textverarbeitungsprogramme gibt, ist diese Trennung der Aufgaben nicht mehr gewährleistet. LaTeX stellt die ursprüngliche Trennung der Aufgaben wieder her. Während des Schreibens kümmern Sie sich zunächst nicht darum, wie etwas später *aussehen* soll, sondern konzentrieren sich allein auf Inhalt und *Bedeutung* des Textabschnitts. Sie werden sehen, wie dies an Qualität gewinnt, sobald der Kopf frei dafür ist, eben *weil* man nicht immer das endgültige Aussehen vor sich hat. Seien Sie ehrlich: Wie oft unterbrachen Sie bis jetzt die inhaltliche Arbeit an einem Text, weil Sie zum Beispiel am Layout einer Tabelle feilten? LaTeX trennt den Inhalt vom Layout, stellt aber die Option zur Verfügung, jederzeit eine Voransicht zu erzeugen, die Sie am Bildschirm begutachten oder drucken können.

1.2 Portabilität und Sicherheit

Textverarbeitungsprogramme speichern ihre Dokumente in sogenannten *Binärdateien*, also Dateien, die man nur mit dem Programm lesen kann, mit dem sie auch gespeichert wurden. Ist die Datei defekt, weil beispielsweise durch einen Absturz eines Programms oder des Betriebssystems Teile durcheinandergekommen sind oder fehlen, ist das Dokument in der Regel nicht zu retten.

LaTeX wird in simplen Textdateien gespeichert, aus denen später die Ausgabe erzeugt wird. Diese Textdateien können mit jedem beliebigen Editor bearbeitet werden. Selbst wenn ein partieller Datenverlust entsteht oder Teile durcheinanderkommen, ist der Rest auf jeden Fall weiterhin lesbar. Die Gefahr eines totalen Datenverlusts geht gegen null. Dieses Speichern in einfachen Textdateien sorgt auch dafür, dass LaTeX-Dokumente unter jedem Betriebssystem bearbeitet werden können, selbst wenn kein LaTeX-System installiert oder verfügbar sein sollte. Aus dem gleichen Grund sind mit LaTeX gesetzte Dokumente auch immer *reproduzierbar*. Selbst Jahre später können alte Dokumente immer noch gelesen und verarbeitet werden.

Schlussendlich hat dies auch Auswirkungen auf den Umgang mit sehr großen Dokumenten: Selbst das Bearbeiten von umfangreichen Dokumenten mit mehreren Tausend Seiten ist kein Hindernis unter LaTeX. Diese Zuverlässigkeit ist ebenfalls ein Ergebnis der Trennung von Inhalt und Layout.

1.3 Seit Langem verfügbar und weiterentwickelt

Seit nunmehr über fünfunddreißig Jahren ist das Satzprogramm TEX ein stabiles Programm, bei dem schon seit Langem keine Fehler mehr gefunden wurden. Donald E. Knuth hat es seinerzeit für sein Buch »The Art of Computer Programming« entwickelt, weil es kein vernünftiges System für den rechnergestützten Satz von Büchern und mathematischen Formeln gab. Da TEX aber sehr schwierig zu benutzen ist, wurde vor mehr als fünfundzwanzig Jahren der Aufsatz für die Gestaltung – LATEX – entwickelt, der es wesentlich einfacher machte, das Satzprogramm anzuwenden. Hierfür wiederum entstanden seitdem viele Erweiterungen für fast alle Arten von Dokumenten. Fortlaufend erfährt das Programm Verbesserungen und Veränderungen aus aller Welt. Leistungsfähige Pakete und Programme erlauben das Erzeugen von PDF inklusive Verlinkungen, Zitierstilen aller Art, speziellen Formatierungen und so weiter.

Die Weiterentwicklungen werden auch von Vereinen gefördert: den *TEX User Groups*. Die deutsche TUG heißt Dante e. V., *Deutsche Anwendervereinigung TEX*. Werden Sie Mitglied oder spenden Sie an Dante e. V., wenn Sie etwas an die Entwicklung zurückgeben wollen.

1.4 Warum dieses Buch?

Dieses Buch möchte Ihnen helfen, die Anfangshürden zu überwinden, damit Sie rasch zu vollständigen, druckfähigen Dokumenten gelangen. Wir halten uns deswegen nicht lange mit Theorien auf, sondern wagen den direkten Einstieg. Durch zahlreiche Praxisbeispiele werden Sie in der Lage sein, nachzuvollziehen, wie das Programm arbeitet. Das erklärte Ziel ist es, Sie bereits während der Lektüre dieses Buches in die Lage zu versetzen, eine wissenschaftliche Arbeit mit LATEX zu erstellen, sei es eine Fach-, Studien-, Diplom-, Bachelor- oder Masterarbeit oder eine Dissertation. Andere Dokumentarten wie Briefe werden zwar erwähnt, jedoch nicht weiter ausgeführt.

Dieses Buch will kein allumfassendes Werk zu LATEX sein: Der Fokus liegt ganz klar auf der *Nutzung*, was für die allermeisten Aufgaben ausreicht. Deshalb wird nicht erklärt, wie Sie eigene *Klassen* und *Styles* selbst schreiben, sondern lediglich behandelt, wie Sie vorhandene verwenden. Die *Erweiterung* oder *Entwicklung* ist also ausgeklammert.

Anders als viele andere Einsteigerwerke zu LATEX erläutert dieses Buch einfachere Zusatzpakete in der Regel gleich beim entsprechenden Thema. Auf diese Weise haben Sie die Informationen an *einer* Stelle zum Nachlesen.

Und damit auch garantiert nichts schiefgeht, wird natürlich auch die Installation eines LaTeX-Systems sowie das Arbeiten mit Literatur und Grafiken beschrieben. Der Umfang, in dem auf die einzelnen Themen eingegangen wird, ist auf das Erstellen einer größeren wissenschaftlichen Arbeit abgestimmt. Sollten darüber hinaus spezielle Fragen auftauchen, so verweise ich am Ende des Buches auf Hilfen, die Sie über dieses Buch hinaus nutzen können.

1.5 Website zum Buch – www.latexbuch.de

Zu diesem Buch gehört die Website `www.latexbuch.de`. Dort finden Sie neben den Codebeispielen zu allen Kapiteln auch eine aktuelle Installationsanleitung mit regelmäßig aktualisierten Quellen Programmpakete. Eventuelle Fehlerkorrekturen finden Sie ebenfalls dort.

Tipps und Tricks zu LaTeX und dem Schreiben von Texten können Sie in meinem Blog auf derselben Seite lesen.

Bitte laden Sie sich die Beispiele von `www.latexbuch.de/beispiele` herunter.

Kapitel 3

Erste Schritte in LaTeX

3.1 Das Arbeiten mit LaTeX

Wie eingangs bereits beschrieben, ist die Arbeitsweise mit LaTeX eine andere als mit Textverarbeitungsprogrammen. Letztere zwingen den Benutzer, Layout und Aussehen der einzelnen Teile gleich interaktiv bei der Eingabe festzulegen. Anders bei LaTeX: Hier verfassen Sie den reinen Text. Bezüglich des Layouts bekundet man zunächst seine Absichten mithilfe 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 LaTeX erzeugt. (Selbstverständlich haben Sie die Möglichkeit, das Aussehen ebenfalls über Befehle zu beeinflussen.) Sie öffnen also nicht etwa ein Programm namens 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 zurechtkommen. Im *Editor* verfassen Sie Ihr Dokument und geben die Befehle ein. Ein anderes Programm, der 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. Editoren mit LaTeX-Unterstützung helfen bei der Auswahl der Befehle, beim Erstellen des Dokuments und beim Zusammenspiel der einzelnen Dateien. Beispiele dafür sind Emacs[1], Kile, WinEdt oder das in diesem Buch für Einsteiger empfohlene TeXworks.

[1] Die Installation von Emacs unter Windows finden Sie im Tutorial Joachim Schlosser. *LaTeX – ein komplettes System unter Windows*. 2016. `http://schlosser.info/latexsystem`.

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

Selbst wenn Sie schon eine Weile mit LaTeX arbeiten, sollten Sie sich nicht scheuen, den Editor zu wechseln, wenn Sie einen besseren finden. »Besser« ist dabei natürlich von den einzelnen Bedürfnissen abhängig. Der eine arbeitet lieber mit der Maus und entsprechenden Buttons, der andere hat lieber Tastaturkürzel und eine integrierte Voransicht. Für jeden Geschmack existiert ein passender Editor!

In diesem Buch gehe ich nur ein Stück weit auf den Editor TeXworks 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. Unter macOS nutze ich TeXShop.

Um Ihr erstes LaTeX-Dokument erstellen zu können, öffnen Sie also zunächst einmal Ihren Editor. Sofern Sie TeXworks verwenden, sehen Sie jetzt in etwa Abbildung 3.1 vor sich.

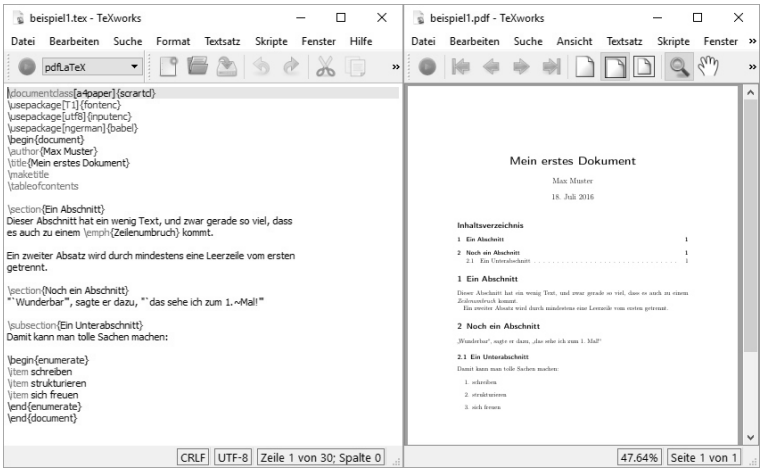


Abb. 3.1: TeXworks-Hauptfenster und Ausgabefenster

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 Website `http://www.latexbuch.de/beispiele` herunterladen und in TeXworks kopieren.

```
1 \documentclass{scrartcl}
2 \begin{document}
```

```
\section{Ein Abschnitt}
Das ist mein erstes Dokument.
\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 35. 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 LaTeX-eigenes Format –, das 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 LaTeX und pdfLaTeX), die teilweise unterschiedlich arbeiten und vor allem unterschiedliche Grafikformate verarbeiten, worauf später in Abschnitt 8.1 auf Seite 135 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 LaTeX-Lauf genannt). Genaueres hierzu finden Sie in Abschnitt 15.2 auf Seite 291.

Oft ist es notwendig, zwei oder mehr Übersetzungsvorgänge durchzuführen, um Dinge, die LaTeX nicht gleich beim ersten Mal komplett erledigen kann (z. B. das Indizieren von Überschriften), vollständig abzuarbeiten.

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

TeXworks Wählen Sie in der Buttonleiste den grünen Pfeil neben »pdfLaTeX«, wenn Sie die Ausgabedatei erzeugen und auch gleich ansehen möchten.

Kommandozeile Ganz allgemein, also ohne an einen bestimmten Editor gebunden zu sein, können Sie auch Folgendes tun: Öffnen Sie eine Kommandozeile unter Windows mit einem rechten Mausklick auf das Windows-Logo, dann auf 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 `beispiel0.ps` bzw. `beispiel0.pdf` können wir mit GSview bzw. Adobe Reader betrachten, je nachdem, welches Ausgabeformat gewählt wurde.

Auch das Zwischenformat der DVI-Dateien kann mittels eines Programms angesehen werden. 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[utf8]{inputenc}
4  \usepackage[ngerman]{babel}
5  \usepackage{lmodern}
6  \begin{document}
7  \author{Max Muster}
8  \title{Mein erstes Dokument}
9  \maketitle
10 \tableofcontents
11
12 \section{Ein Abschnitt}
13 Dieser Abschnitt hat ein wenig Text, und zwar gerade so viel, dass
14 es auch zu einem \emph{Zeilenumbruch} kommt.
15
16 Ein zweiter Absatz wird durch mindestens eine Leerzeile vom ersten
17 getrennt.
18
19 \section{Noch ein Abschnitt}
20 "`Wunderbar"', sagte er dazu, "`das sehe ich zum 1.~Mal!"'
21
22 \subsection{Ein Unterabschnitt}
23 Damit kann man tolle Sachen machen:
24
25 \begin{enumerate}
26 \item schreiben
27 \item strukturieren
28 \item sich freuen
29 \end{enumerate}
30 \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 LaTeX 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 35.

Umgebungen beginnen immer mit `\begin{umgebung}`[2] 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 25 bis 29, die eine Aufzählung beschreibt.

Mehr dazu finden Sie ebenfalls in Kapitel 4 auf Seite 35.

Präambel und Textteil bilden die zwei Teile eines LaTeX-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 verwenden eventuell einige weitere Befehle.

Klasse Üblicherweise in der ersten Zeile der Datei wird die Klasse und damit die Dokumentart angegeben.

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 Beispieltext soll das Layout eines Artikels haben und DIN-A4-Format aufweisen.

Neben der Klasse der Artikels stehen Klassen bereit 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 89.

[2] Oft wird in der Informatik in Syntaxbeschreibungen bei variablen Anteilen mit spitzen Klammern gearbeitet, sodass man `<umgebung>` 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 zugutekommt. Das inputenc-Paket mit der Option `utf8` 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 105.

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 41.

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 14.

Anführungszeichen kommen in vielen Variationen vor; die deutschen Anführungszeichen können mit `"‘` (Gänsefüßchen Akzent rechts) und `"’` (Gänsefüßchen Apostroph) erzeugt werden, die französischen mit `">` und `"<`.

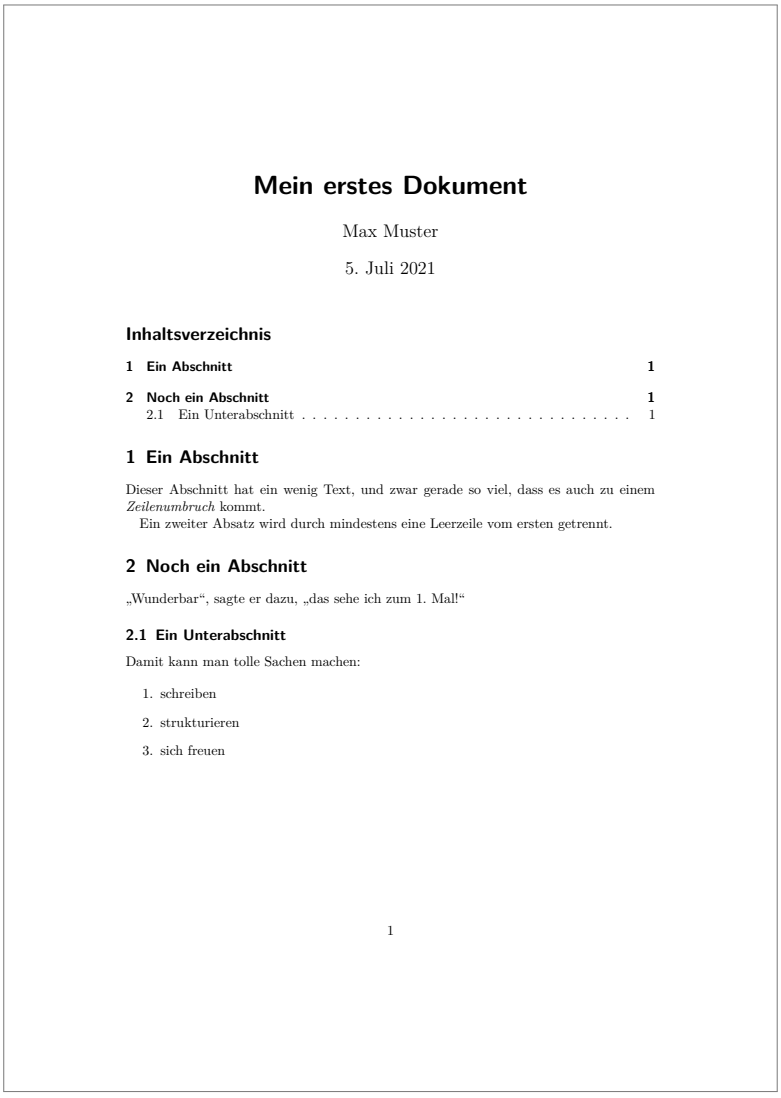
Mehr dazu finden Sie in Abschnitt 4.2.4 auf Seite 46.

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

Möchten Sie einen bestimmten Wortzwischenraum, so können Sie dies angeben. Das Zeichen ~ in Zeile 20 erzeugt einen sogenannten *geschützten Leerraum*, das heißt, die beiden dadurch verbundenen Wörter stehen immer in einer Zeile. Weitere Möglichkeiten in diesem Umfeld werden später in Abschnitt 4.3 auf Seite 48 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 lesen Sie in Abschnitt 4.4 auf Seite 53.

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 Abbildung 3.2 sehen.



Mein erstes Dokument

Max Muster

5. Juli 2021

Inhaltsverzeichnis

1 Ein Abschnitt

Dieser Abschnitt hat ein wenig Text, und zwar gerade so viel, dass es auch zu einem *Zeilenumbruch* kommt.
Ein zweiter Absatz wird durch mindestens eine Leerzeile vom ersten getrennt.

2 Noch ein Abschnitt

„Wunderbar“, sagte er dazu, „das sehe ich zum 1. Mal!“

2.1 Ein Unterabschnitt

Damit kann man tolle Sachen machen:

1. schreiben
2. strukturieren
3. sich freuen

1

Abb. 3.2: Ausgabe für Beispieldokument 1