

# Herbert Voß

## Die wissenschaftliche Arbeit mit L<sup>A</sup>T<sub>E</sub>X

unter Verwendung von Lua<sub>T</sub><sub>E</sub>X,  
KOM<sub>A</sub>-Script und Biber/Bib<sub>L</sub>A<sub>T</sub><sub>E</sub>X



```
\begin{titlepage}
\parbox{0.5\linewidth}{\sffamily
\includegraphics[width=\linewidth]{FU2.png}\
Fachbereich Philosophie
und Geisteswissenschaften\
Institut für Romanische Philologie\
Frankreichstudien}\par
\vspace{4cm}\Large\bfseries
\begin{center}
\begin{tabular}{c}Die Feminisierung\
der Berufsbezeichnungen in\
frankophonen Ländern\ im Vergleich
\end{tabular}\par\normalsize\normalfont
\vspace*{2cm}%
\begin{tabular}{c}
Diplomarbeit\ [10pt]
März 2006\ [20pt]
\large\textbf{Jana Voß}
\end{tabular}/\par\vspace*{\fill}
Gutachter:
\begin{tabular}{t}{1}
Prof.\,Dr. Jürgen Trabant\
Prof.\,Dr. Thomas Kotschi
\end{tabular}
\end{center}
\end{titlepage}
```

# Die wissenschaftliche Arbeit mit L<sup>A</sup>T<sub>E</sub>X

unter Verwendung von LuaT<sub>E</sub>X,  
KOMA-Script und Biber/BibL<sup>A</sup>T<sub>E</sub>X

Für meinen Vater (8. 9. 1927 – 18. 1. 2018)  
und  
Johan (geb. 2017)

# Die wissenschaftliche Arbeit mit L<sup>A</sup>T<sub>E</sub>X

unter Verwendung von LuaT<sub>E</sub>X,  
KOMA-Script und Biber/BibL<sup>A</sup>T<sub>E</sub>X

Herbert Voß

*Berlin*

dante

lehmanns   
media

Alle in diesem Buch enthaltenen Programme, Darstellungen und Informationen wurden nach bestem Wissen erstellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund ist das in dem vorliegenden Buch enthaltene Programm-Material mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autoren und Herausgeber übernehmen infolgedessen keine Verantwortung und werden keine Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieses Programm-Materials oder Teilen davon oder durch Rechtsverletzungen Dritter entsteht.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann verwendet werden dürften.

Alle Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt und sind möglicherweise eingetragene Warenzeichen. Autoren und Herausgeber richten sich im Wesentlichen nach den Schreibweisen der Hersteller. Andere hier genannte Produkte können Warenzeichen des jeweiligen Herstellers sein.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdrucks und der Vervielfältigung des Buches, oder Teilen daraus, sind vorbehalten.

© 2018 Herbert Voß, Berlin

Erste Auflage

ISBN 978-3-86541-946-0

Umschlag: Herbert Voß

Karikaturen: Duane Bibby, mit Genehmigung der TUG

Satz: L<sup>A</sup>T<sub>E</sub>X

Verlag: Lehmanns Media, Berlin (<http://www.lehmanns.de>)

Druck: Dimograf – Bielsko-Biała – Polen

---

# Inhaltsverzeichnis

<b>Vorwort</b>	<b>i</b>
<b>1 Einführung</b>	<b>1</b>
1.1 Warum $\text{\LaTeX}$ und nicht die Textverarbeitung $xy$ ?	1
1.2 Warum $\text{\LuaTeX}$ und nicht $\text{\pdfLaTeX}$ ?	7
1.3 Das Prinzip	8
1.4 Externe $\text{\LaTeX}$ -Anwendungen	10
<b>2 Grafische Entwicklungsumgebungen</b>	<b>11</b>
2.1 $\text{\TeX}$ studio	11
2.2 Online-Editoren	16
<b>3 Erste Schritte mit <math>\text{\LaTeX}</math></b>	<b>17</b>
3.1 Struktur eines $\text{\LaTeX}$ -Dokuments	17
3.2 Die Dokumentenklasse	19
3.3 Globale Optionen	20
3.4 Der Satzspiegel	20
3.5 Nationale Besonderheiten	27
<b>4 Schriften</b>	<b>29</b>
4.1 Einführung	29
4.2 Schriftauswahl	30
4.3 Schriftnamen und Dateinamen	36
4.4 Kombination von Schriften	38
4.5 Was man wissen sollte	40
<b>5 Sprachunterstützung</b>	<b>43</b>
5.1 Einführung	43
5.2 Das Sprachpaket <code>babel</code>	43
5.3 Trennungen	45
5.4 Anführungszeichen	49
5.5 Rechtschreibkontrolle	52

<b>6</b>	<b>Textelemente</b>	<b>55</b>
6.1	Allgemeine Festlegungen . . . . .	55
6.2	Der Zeilenabstand . . . . .	58
6.3	Die Titelseite . . . . .	60
6.4	Der Seitenstil . . . . .	64
6.5	Ausrichten von Text . . . . .	70
6.6	Auszeichnung von Text . . . . .	77
6.7	Hoch- und tiefgestellter Text . . . . .	80
6.8	Fußnoten . . . . .	80
6.9	Endnoten . . . . .	84
6.10	Randbemerkungen . . . . .	85
6.11	Textmarken und -referenzen . . . . .	88
6.12	Hyperlinks . . . . .	92
6.13	Umbruch . . . . .	99
6.14	Horizontale und vertikale Abstände . . . . .	105
6.15	Mehrspaltiger Text . . . . .	108
6.16	Kritische Editionen . . . . .	112
6.17	Große Dokumente organisieren . . . . .	113
<b>7</b>	<b>Gliederung</b>	<b>117</b>
7.1	Die Hauptebenen . . . . .	117
7.2	Nichtnummerierte Ebenen . . . . .	120
7.3	Definition neuer Gliederungsebenen . . . . .	122
<b>8</b>	<b>Listen</b>	<b>127</b>
8.1	Standardlisten . . . . .	127
8.2	Das Paket <code>enumitem</code> . . . . .	134
<b>9</b>	<b>Tabellen</b>	<b>139</b>
9.1	Einführung . . . . .	139
9.2	Die tabular-Umgebung . . . . .	140
9.3	Pakete zur Tabellenformatierung ohne Seitenumbruch . . . . .	143
9.4	Pakete zur Tabellenformatierung mit Seitenumbruch . . . . .	174
9.5	Farbe in Tabellen . . . . .	185
9.6	Beispiele . . . . .	188
<b>10</b>	<b>Abbildungen einfügen</b>	<b>195</b>
10.1	Einführung . . . . .	195
10.2	Anwendung . . . . .	197
10.3	PDF-Dokumente . . . . .	199
<b>11</b>	<b>Gleitumgebungen</b>	<b>201</b>
11.1	Definition . . . . .	202
11.2	Platzierung . . . . .	203
11.3	Parameter . . . . .	205
11.4	Über- oder Unterschriften . . . . .	208
11.5	Seitliche Beschriftungen . . . . .	214
11.6	Die Formatierung der Beschriftung . . . . .	217

11.7	Nicht-gleitende Abbildungen und Tabellen . . . . .	222
11.8	Mehrere Objekte in einer Gleitumgebung . . . . .	226
11.9	Textumflossene Tabellen und Abbildungen . . . . .	232
11.10	T <sub>E</sub> Xnisches . . . . .	233
<b>12</b>	<b>Verzeichnisse</b>	<b>235</b>
12.1	Standardverzeichnisse . . . . .	235
12.2	Eigene Verzeichnisse erstellen . . . . .	242
12.3	Stichwortverzeichnis . . . . .	244
12.4	Glossar . . . . .	252
<b>13</b>	<b>Bibliografie</b>	<b>263</b>
13.1	Einführung . . . . .	263
13.2	Bibliografie-Datenbanken . . . . .	265
13.3	Bibliografien mit BibL <sup>A</sup> T <sub>E</sub> X und Biber . . . . .	280
<b>14</b>	<b>Mathematik</b>	<b>301</b>
14.1	Typografische Regeln . . . . .	302
14.2	Das Paket amsmath . . . . .	303
14.3	Der Zeilenmodus . . . . .	305
14.4	Mathematik im abgesetzten Modus . . . . .	308
14.5	Matrix-Umgebungen . . . . .	319
14.6	Gleichungsnummern . . . . .	321
14.7	Marken (Label) . . . . .	323
14.8	Weitere mathematische Elemente . . . . .	325
<b>15</b>	<b>Grafiken erstellen</b>	<b>343</b>
15.1	Koordinaten . . . . .	344
15.2	Einfache Pfade . . . . .	345
15.3	Füllen von Pfaden . . . . .	348
15.4	Pfeile . . . . .	349
15.5	Knoten . . . . .	350
15.6	Bibliotheken . . . . .	351
15.7	Pakete, die auf TikZ aufbauen . . . . .	353
<b>16</b>	<b>Fragen und Antworten</b>	<b>357</b>
16.1	L <sup>A</sup> T <sub>E</sub> X . . . . .	357
16.2	Kodierung . . . . .	360
<b>A</b>	<b>Anhang</b>	<b>361</b>
A.1	Längen . . . . .	361
A.2	T <sub>E</sub> XLive-Installation . . . . .	362
A.3	LuaT <sub>E</sub> X . . . . .	363
A.4	Tastaturkürzel . . . . .	364
	<b>Bibliografie</b>	<b>373</b>
	<b>Index der Befehle und Begriffe</b>	<b>377</b>
	<b>Personen</b>	<b>404</b>



# Vorwort

Die Anwendung von L<sup>A</sup>T<sub>E</sub>X im nicht-wissenschaftlichen Bereich kann man faktisch vernachlässigen. Dennoch gibt es Unterschiede bei der Erstellung einer Bachelor-, Master-, Doktorarbeit u. ä. oder einer umfangreichen Monografie zum Thema Linguistik. Bei wissenschaftlichen Abschlussarbeiten müssen formale Kriterien eingehalten werden. So sollte beispielsweise jede Literaturangabe auch einen entsprechenden Verweis im Text haben, da sonst der Bezug der angegebenen Literatur zum Inhalt der Arbeit nicht eindeutig ist. Damit ist aber die Frage »Wie verweist man korrekt auf eine Literatur«, noch zu klären.

Dieses Buch setzt konsequent auf drei Dinge: Die Anwendung von

- LuaL<sup>A</sup>T<sub>E</sub>X, beziehungsweise das darunterliegende LuaT<sub>E</sub>X, als Programm zur Dokumentenerstellung,
- Dokumentenklassen aus dem KOMA-Script-Paket und
- Biber/BibL<sup>A</sup>T<sub>E</sub>X als Programm/Paket zum Verarbeiten von Literaturdatenbanken und Erstellung einer oder mehrerer Bibliografien.

Mit LuaT<sub>E</sub>X wird der zur Zeit leistungsfähigste T<sub>E</sub>X-Compiler verwendet. Er bietet den Vorteil, dass Unicode als Standardkodierung der Zeichen benutzt wird, womit faktisch alle Sprachen der Welt in einem Dokument verwendet werden könnten. Voraussetzung ist lediglich, dass entsprechende Schriften in den Formaten OpenType oder TrueType zur Verfügung stehen. Sobald man das Programm LuaL<sup>A</sup>T<sub>E</sub>X aufruft, wird, so wie auch in der Vergangenheit üblich, das Makro-Paket L<sup>A</sup>T<sub>E</sub>X geladen und das Ergebnis in Form einer pdf-Datei durch Aufruf des T<sub>E</sub>X-Compilers LuaT<sub>E</sub>X erzeugt. Für Anwender, die bereits bisher L<sup>A</sup>T<sub>E</sub>X benutzten, bedeutet dies also kaum eine Umstellung in der Vorgehensweise.

Die vielfältigen neuen Möglichkeiten von LuaT<sub>E</sub>X kommen allerdings mit einer Einbuße bei der Übersetzungsgeschwindigkeit daher. Für wissenschaftliche Arbeiten mit weniger als tausend Seiten stellt dies jedoch kein wirkliches Problem dar.

Die Dokumentenklassen von KOMA-Script bieten vielfältige Möglichkeiten zum Anpassen des formalen Layouts an externe oder eigene Vorgaben. Dies beginnt bereits mit der Titelseite, die oft den Richtlinien der wissenschaftlichen Einrichtung folgen muss.

Das bekannte Programm Bib<sub>l</sub>TeX nicht zu verwechseln mit dem gleichnamigen Dateiformat, wird hier nicht mehr behandelt. Das Programm hat erhebliche Mängel, wenn Bibliografien erstellt werden sollen, die nicht nur die englische Sprache verwenden. Mit der Kombination Biber/BibLaTeX kann den Anforderungen an unterschiedliche Ausgabeformen sowohl beim Literaturverweis im laufenden Text als auch bei der eigentlichen Bibliografie entsprochen werden. Die Zeiten, in denen Umlaute in der Form `Tr{"a}gheitsd{"u}se` eingegeben werden mussten, gehören dann der Vergangenheit an.

Wieder war Rolf Niepraschk bereit, bei dem einen oder anderen Problem zu helfen. Ein großer Dank geht an Elke Schubert, die jede noch so kleine Ungereimtheit im Zusammenhang mit KOMA-Script aufgespürt hat. Alles, was jetzt noch an Fehlern vorhanden ist, geht ausschließlich zu meinen Lasten.

Alle Beispiele dieses Buches findet man wie üblich als lauffähige TeX-Dokumente zum Herunterladen auf <http://archiv.dante.de/~herbert/Books/>.

Berlin, im Januar 2018

Herbert Voß

## Einführung

1.1 Warum $\text{\LaTeX}$ und nicht die Textverarbeitung $xy$ ? . . . . .	1
1.2 Warum $\text{\LuaTeX}$ und nicht $\text{\pdfLaTeX}$ ? . . . . .	7
1.3 Das Prinzip . . . . .	8
1.4 Externe $\text{\LaTeX}$ -Anwendungen . . . . .	10

Die formale Ausführung einer wissenschaftlichen Arbeit kann man in zahlreichen Publikationen nachlesen, beispielsweise in (Balzert u. a. 2008; Toraman 2016; Wytrzens u. a. 2012). Daneben gibt es dann noch diverse Veröffentlichungen, die ein bestimmtes Textsystem als Voraussetzung für die Erstellung der Arbeit haben, beispielsweise (Farke 1997; Koch 2010; Seimert 2011) Für den Einsatz von  $\text{\LaTeX}$  zur Erstellung einer wissenschaftlichen Arbeit gibt es unter anderem die Veröffentlichungen von (Machert 2013; J. Schlosser 2016; P. Schlosser und Thibud 2007; Würth und Gallmann 2018), wobei nur (J. Schlosser 2016; Würth und Gallmann 2018) als aktuell bezeichnet werden kann. Mit der hier vorliegenden Arbeit wird konsequent auf die neuesten Entwicklungen von  $\text{\TeX}$  gesetzt: Als Standardprogramm wird  $\text{\LuaTeX}$  verwendet, als Dokumentenklasse KOMA-Script und als Bibliografieprogramm Biber.

Die verwendete  $\text{\TeX}$ -Distribution war  $\text{\TeX}Live$  (<http://www.tug.org/texlive/>), was aber im Prinzip für die eigentliche Arbeit mit  $\text{\LaTeX}$  unerheblich ist; mit der Distribution  $\text{\MikTeX}$  (<http://www.miktex.org>) wird dasselbe Ergebnis erreicht.<sup>1</sup>

### 1.1 Warum $\text{\LaTeX}$ und nicht die Textverarbeitung $xy$ ?

Die Anwendung von  $\text{\LaTeX}$  zur Erstellung einer wissenschaftlichen Arbeit kann man für den Bereich Mathematik als »flächendeckend« bezeichnen. Es besteht aber heutzutage

---

<sup>1</sup>Kommerzielle Distributionen werden hier nicht behandelt.

eigentlich kein direkter Zusammenhang mehr zwischen Mathematik und  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . Auch für Sprachwissenschaftler, die beispielsweise linguistische Bäume erstellen müssen oder andere Autoren, die sehr große Dokumente erstellen wollen, haben durch die Anwendung von  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  viele Vorteile. Im folgenden Abschnitt wird gezeigt, dass auch eine relativ kurze wissenschaftliche Arbeit durch die Anwendung von  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  mindestens ästhetische Vorteile gegenüber den allgemein bekannten Textverarbeitungssystemen hat.

Die folgende Abbildung 1.1 zeigt den typischen Umbruch einer Textverarbeitung. Um den Unterschied zu einem Satzprogramm zu verdeutlichen, wurde die Trennung von Wörtern nicht zugelassen. Textverarbeitungen beziehen den Umbruch auf eine einzelne Zeile, indem diese beendet wird, wenn ein weiteres Wort nicht mehr in diese passt. Dies führt im folgenden Beispiel dazu, dass die dritte Zeile mit einem sehr großen Wortabstand gesetzt wird, was optisch immer sehr zweifelhaft erscheint.

Neithardt von Gneisenau, der Kommandant der Festung Kolberg, deren ruhmreiche Verteidigung einen der interessantesten Abschnitte dieses Buches (von <http://www.gutenberg.org/files/23333/23333-8.txt>) bildet, 1760 geboren, hatte schon in einem zu Erfurt garnisonierenden österreichischen und danach in einem der Regimenter des Markgrafen von Ansbach-Bayreuth gedient, die in englischem Solde in und gegen Amerika kämpften, als Friedrich der Große ihn 1786 als Premierleutnant in die preußische Armee aufnahm.

Abbildung 1.1: Zeilenumbruch einer Textverarbeitung (LibreOffice).

Bei einer Textverarbeitung erfolgt *nach* einem Zeilenumbruch unabhängig von den zukünftigen Zeilenumbrüchen keine Änderung mehr. Diese Tatsache führt immer dann zu ungünstigen Umbrüchen, wenn die Zeile Bestandteile enthält, die nicht getrennt werden sollen oder können. In dem angegebenen Beispiel ist es eine lange Internetadresse, die ohne externen Eingriff nicht getrennt wird. Vergleicht man dazu Abbildung 1.2, welches die Ausgabe von  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  darstellt, so fällt sofort auf, dass sich die Größe der Lücken verringert hat.

*Ungünstige  
Zeilenumbrüche*

Neithardt von Gneisenau, der Kommandant der Festung Kolberg, deren ruhmreiche Verteidigung einen der interessantesten Abschnitte dieses Buches (von <http://www.gutenberg.org/files/23333/23333-8.txt>) bildet, 1760 geboren, hatte schon in einem zu Erfurt garnisonierenden österreichischen und danach in einem der Regimenter des Markgrafen von Ansbach-Bayreuth gedient, die in englischem Solde in und gegen Amerika kämpften, als Friedrich der Große ihn 1786 als Premierleutnant in die preußische Armee aufnahm.

Abbildung 1.2: Absatzumbruch eines Satzprogramms ( $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ).

Das Programm  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  ist keine Textverarbeitung im eigentlichen Sinne. Dies wird schon daran deutlich, dass es völlig unerheblich ist, mit welchem System der Quelltext für das *Programm*  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  erstellt wird. Der Anwender programmiert, d. h., er erstellt einen Programmcode, welcher als Ausgabe einen formatierten Text erzeugt. Es erleichtert

Anfängern den Umgang mit dem Gesamtsystem, wenn man sich auch als (*Text-*)*Programmierer* versteht. Dies bedeutet, dass es völlig unwichtig ist, welche Formatierung der Quelltext aufweist, solange man die Regeln der dem System zugrundeliegenden Befehlssyntax beachtet. Dies betrifft primär den Zeilenumbruch; ein Satzprogramm umbricht grundsätzlich Absätze in Zeilen, wohingegen alle Textverarbeitungsprogramme nur einzelne Zeilen umbrechen.

Da bei  $\text{\LuaTeX}$  *immer* der gesamte Absatz für den Umbruch der einzelnen Zeilen herangezogen wird, kann das Aufeinandertreffen von sehr eng und sehr weit gesetzten Zeilen vermieden werden; jeder Umbruch einer Zeile steht in Bezug zu den vorhergehenden und nachfolgenden Zeilen. Vereinfacht formuliert kann man sagen, dass  $\text{\LuaTeX}$  vier optische Kriterien für den Satz von Zeilen kennt: *eng*, *weniger eng*, *weit* und *sehr weit*. *Absatz-  
umbruch*

Zeilen können

- eng gesetzt sein:

Dies ist eine sinnlose Zeile mit etwas Text

- weniger eng gesetzt sein:

Dies ist eine sinnlose Zeile mit etwas Text

- weit gesetzt sein:

Dies ist eine sinnlose Zeile mit etwas Text

- sehr weit gesetzt sein:

Dies ist eine sinnlose Zeile mit etwas Text

#### Umbruchkriterium

Zwei aufeinander folgende Zeilen müssen  
*gleiche* oder *benachbarte* Kriterien erfüllen!

Dadurch ist es bei  $\text{\LuaTeX}$  unmöglich, dass eine sehr weit gesetzte Zeile einer eng gesetzten folgt, was ansonsten zu den unangenehmen Lücken aus Abbildung 1.1 auf der vorherigen Seite führt. Erlaubt man zusätzlich Worttrennungen einschließlich eines automatischen Umbruchs der Internetadresse, so kann das optische Bild bei LibreOffice nur etwas verbessert werden (Abbildung 1.3). Erst bei der Anwendung von  $\text{\LuaTeX}$  ergibt sich mit Abbildung 1.4 auf der nächsten Seite ein ansprechendes Ergebnis.

Neithardt von Gneisenau, der Kommandant der Festung Kolberg, deren ruhmreiche Verteidigung einen der interessantesten Abschnitte dieses Buches (von <http://www.gutenberg.org/files/23333/23333-8.txt>) bildet, 1760 geboren, hatte schon in einem zu Erfurt garnisonierenden österreichischen und danach in einem der Regimenter des Markgrafen von Ansbach-Bayreuth gedient, die in englischem Solde in und gegen Amerika kämpften, als Friedrich der Große ihn 1786 als Premierleutnant in die preußische Armee aufnahm.

Abbildung 1.3: Zeilenumbruch mit Worttrennungen (LibreOffice).

Neithardt von Gneisenau, der Kommandant der Festung Kolberg, deren ruhmreiche Verteidigung einen der interessantesten Abschnitte dieses Buches (von <http://www.gutenberg.org/files/23333/23333-8.txt>) bildet, 1760 geboren, hatte schon in einem zu Erfurt garnisonierenden österreichischen und danach in einem der Regimenter des Markgrafen von Ansbach-Bayreuth gedient, die in englischem Solde in und gegen Amerika kämpften, als Friedrich der Große ihn 1786 als Premierleutnant in die preußische Armee aufnahm.

Abbildung 1.4: Absatzumbruch mit Trennungen und Linkumbruch (L<sup>A</sup>T<sub>E</sub>X).

## Etwas Theorie

Der Umbruch für den Absatz wird nach dem sogenannten Best-Fit-Algorithmus vorgenommen; von allen möglich Umbrüchen eines Absatzes wird letztlich derjenige ausgewählt, der dem idealen Umbruch am nächsten kommt. Der ideale Umbruch hat zwischen allen Wörtern denselben Abstand und kommt ohne Trennungen aus. Lua<sub>T</sub>E<sub>X</sub> geht hierbei in maximal drei Stufen vor:

1. Versuch, einen optimalen Umbruch ohne Trennungen zu erhalten.
  - Genügt das Ergebnis den Vorgaben, wird der Umbruch akzeptiert und als abgeschlossen betrachtet.
  - In der T<sub>E</sub>X-Terminologie bedeutet dies, dass die Dehnungspunkte (`\badness` – Abstände zwischen den Wörtern) *jeder* umbrochenen Zeile den Wert von `\pretolerance=100` nicht überschreiten.
  - Wegen nicht vorhandener Trennstellen benötigt der 1. Durchgang sehr wenig Rechenzeit!
2. Versuch, einen optimalen Umbruch mit Trennungen zu erhalten.
  - Es kommen als sprachenspezifische Trennstellen die Textstellen in Frage, die Lua<sub>T</sub>E<sub>X</sub> durch seinen Trennalgorithmus findet.
  - Der Umbruch wird akzeptiert, falls alternativ
    - die Dehnungspunkte (`\badness`) jeder umbrochenen Zeile den Wert von `\tolerance=200` nicht überschreiten und `\emergencystretch` (hier gleich `30.0pt`) einen positiven Wert aufweist;
    - falls `\tolerance` den Wert `10000` hat; dann kommen zusätzlich `\hfuzz=0.1pt` und `\hbadness=1000` zum Tragen.
3. und letzter Versuch, durch zusätzliche Zwischenräume.
  - Der intern vorgegebene oder vom Anwender geänderte Wert von `\emergencystretch (30.0pt)` wird zur erlaubten Dehnungsmöglichkeit jeder Zeile addiert (siehe auch Abschnitt 6.13.4 auf Seite 104).
  - Durch diese Addition werden die zulässigen Dehnungspunkte jeder Zeile verringert.
  - Abschließende Entscheidung mithilfe der Register `\hfuzz=0.1pt` und `\hbadness=1000`.

Hierbei wird, wie oben erwähnt, die Berechnung sofort nach der ersten oder zweiten Stufe beendet, falls der Umbruch den internen Vorgaben entspricht. Auf den sehr aufwändigen Algorithmus der einzelnen Stufen soll hier nicht weiter eingegangen werden;

es wird auf die ausführliche Beschreibung in (Römer und Voß 2010; Voß 2007) und insbesondere (Knuth und Plass 1981) verwiesen. Hier sollte nur gezeigt werden, welchen Aufwand T<sub>E</sub>X im Gegensatz zu herkömmlichen Textverarbeitungsprogrammen treibt, um einen optimierten Umbruch zu erreichen.

### Was man jetzt schon wissen sollte ...

Bei der Texterstellung mit L<sup>A</sup>T<sub>E</sub>X kann etwas auftreten, was bei keiner Textverarbeitung passiert und bei Anfängern zu Irritationen führen kann, weil das Prinzip oft zuerst nicht verstanden wird. Zur besseren Erklärung wird für die folgende Demonstration die Trennmöglichkeit abgeschaltet (`\langwohyphens` → language without hyphens). Das Beispiel zeigt einen Text mit relativ schmaler Textbreite und den für die deutsche Sprache typischen langen Wörtern.

```
\newlanguage\langwohyphens
\AtBeginDocument{\language\langwohyphens} % Trennungen jetzt nicht möglich
```

Der Begriff Transzendentalphilosophie umfasst philosophische Systeme und Ansätze, welche Grundstrukturen des allgemeinen Seins nicht durch eine Ontologie (Theorie des Seienden), sondern im Rahmen des Entstehens und Begründens von Wissen über das Sein beschreiben.

01-01-1

Der Begriff Transzendentalphilosophie umfasst philosophische Systeme und Ansätze, welche Grundstrukturen des allgemeinen Seins nicht durch eine Ontologie (Theorie des Seienden), sondern im Rahmen des Entstehens und Begründens von Wissen über das Sein beschreiben.

Wie zu sehen ist, schreibt LuaT<sub>E</sub>X die erste und letzte Zeile über den Rand! Es bleibt natürlich die Frage, wieso kann das jede Textverarbeitung vermeiden, aber LuaT<sub>E</sub>X nicht. Im Gegensatz zu einer Textverarbeitung, die Zwischenräume in ihrer Größe nicht begrenzt, erlaubt LuaT<sub>E</sub>X per Definition nur einen maximalen Wortzwischenraum von 3.10501 pt, was gerade einmal 1,4 mm entspricht. Reicht dieser Wert nicht aus, um eine Zeile vernünftig zu umbrechen, so wird automatisch das letzte Wort über den Rand hinaus geschrieben. Dieser Fall ist unter anderem sprachenabhängig; die allgemeine Wortlänge und Trennungsmöglichkeiten der zugrunde liegenden Sprache spielen eine große Rolle. Man sollte diesen Effekt nicht als Makel, sondern als Hinweis darauf auffassen, dass der vorgegebene Text keinen Umbruch bei den strengen Vorgaben erlaubt. Er wird ohnehin in Texten mit »normaler« Textbreite und Trennmöglichkeiten selten zu beobachten sein.

Besteht keine Möglichkeit den Text zu modifizieren, so setzt man den entsprechenden Absatz »schlampig«. Dazu kann man die Umgebung `sloppypar` oder den Schalter `\sloppy` verwenden. Theoretisch könnte man den Befehl `\sloppy` bereits in der Präambel verwenden, was jedoch nicht zu empfehlen ist. Der Umbruch wird dadurch qualitativ schlechter; die Wortzwischenräume sind dann allgemein größer. Der Vollständigkeit halber sei erwähnt, dass `\fussy` die Umkehrung von `\sloppy` ist, also der Standard. Obiges Beispiel mit der Umgebung `sloppypar` gesetzt, ergibt ein besseres Erscheinungsbild:

```
\newlanguage\langwohyphens
\AtBeginDocument{\language\langwohyphens} % Trennung nicht mehr möglich
```

```
\begin{sloppypar}
```

Der Begriff Transzendentalphilosophie umfasst philosophische Systeme und Ansätze, welche Grundstrukturen des allgemeinen Seins nicht durch eine Ontologie (Theorie des Seienden), sondern im Rahmen des Entstehens und Begründens von Wissen über das Sein beschreiben.

```
\end{sloppypar}
```

01-01-2

Der Begriff Transzendentalphilosophie umfasst philosophische Systeme und Ansätze, welche Grundstrukturen des allgemeinen Seins nicht durch eine Ontologie (Theorie des Seienden), sondern im Rahmen des Entstehens und Begründens von Wissen über das Sein beschreiben.

Grundsätzlich kann man davon ausgehen, dass die Anwendung von `sloppypar` oder `\sloppy` eher selten ist und meistens nur für die folgenden Fälle notwendig:

- Sehr schmale Textspalten, die einen Umbruch grundsätzlich erschweren.
- Längere Wörter, die aus bestimmten Gründen nicht getrennt werden sollen, beispielsweise Funktionsnamen von Programmen.
- Ungünstige URLs, die nach Vorgabe im Allgemeinen nur bei den Sonderzeichen umbrochen werden.
- Einsatz falscher Trennmuster, weil beispielsweise vergessen wurde, für ein fremdsprachiges Textzitat auch in diese Fremdsprache zu wechseln:  
`\foreignquote{Sprache}{Text}` (siehe dazu Abschnitt 5.4 auf Seite 49).



Um ein Gefühl für die Häufigkeit zu bekommen, sei darauf hingewiesen, dass für dieses Buch die Anwendung der Umgebung `sloppypar` nur für den Absatz nach dem arabischen Schriftbeispiel auf der nächsten Seite wegen der drei URLs notwendig war. Festzustellen ist zudem, dass obiges Beispiel völlig unproblematisch ist, wenn die Trennungen erlaubt sind, was standardmäßig ohnehin der Fall wäre. Allerdings ist der Absatz aus  $\TeX$ -Sicht typografisch ungünstig, denn eine Trennung in der vorletzten Zeile wird grundsätzlich als schlecht bewertet. Eine Alternative wäre aber nur das Erweitern der Wortzwischenräume, was eine noch schlechtere Wertung hätte.

```
\usepackage[ngerman]{babel}
```

Der Begriff Transzendentalphilosophie umfasst philosophische Systeme und Ansätze, welche Grundstrukturen des allgemeinen Seins nicht durch eine Ontologie (Theorie des Seienden), sondern im Rahmen des Entstehens und Begründens von Wissen über das Sein beschreiben.

Der Begriff Transzendentalphilosophie umfasst philosophische Systeme und Ansätze, welche Grundstrukturen des allgemeinen Seins nicht durch eine Ontologie (Theorie des Seienden), sondern im Rahmen des Entstehens und Begründens von Wissen über das Sein beschreiben.

01-01-3

Mit den Methoden der Mikrotypografie durch das Laden des Paketes `microtype` von Robert Schlicht kann man das Layout verbessern:

```
\usepackage{microtype}
\usepackage[ngerman]{babel}
```

Der Begriff Transzendentalphilosophie umfasst philosophische Systeme und Ansätze, welche Grundstrukturen des allgemeinen Seins nicht durch eine Ontologie (Theorie des Seienden), sondern im Rahmen des Entstehens und Begründens von Wissen über das Sein beschreiben.

01-01-4

Der Begriff Transzendentalphilosophie umfasst philosophische Systeme und Ansätze, welche Grundstrukturen des allgemeinen Seins nicht durch eine Ontologie (Theorie des Seienden), sondern im Rahmen des Entstehens und Begründens von Wissen über das Sein beschreiben.

## 1.2 Warum Lua $\text{\LaTeX}$ und nicht pdf $\text{\LaTeX}$ ?

Man kann zur Zeit noch davon ausgehen, dass der größere Teil der Nutzer von  $\text{\LaTeX}$  seine Dokumente mit pdf $\text{\LaTeX}$  erstellen dürfte. Das Hauptargument dafür ist schlicht und ergreifend »never change a running system«. Mit der Anwendung von pdf $\text{\LaTeX}$  benutzt man faktisch ein abgekapseltes System, welches Quelldokumente verarbeitet, die völlig unabhängig vom Betriebssystem und der zugrundeliegenden  $\text{\TeX}$ -Distribution sind. Dieser Vorteil ist auch gleichzeitig ein Nachteil: Sobald man in seinem Dokument sprachenspezifische Dinge, wie Kodierung und Sortierung von fremdsprachlichen Zeichenketten beachten muss, steigt der Aufwand mit pdf $\text{\LaTeX}$  teilweise erheblich. Mit Lua $\text{\LaTeX}$ , was eine vollwertige Unicode-Unterstützung ermöglicht, muss darauf faktisch keine Rücksicht genommen werden. Hat man eine Schrift, die die entsprechenden Zeichen enthält, können fremdsprachliche Textzitate eingefügt werden, ohne dass man irgendwelche speziellen Pakete laden muss. Auch die Änderung der Schreibrichtung ist möglich, wenn eine der RTL-Sprachen (Right-To-Left) verwendet wird.

。たしまきでん学はに時、に前年の本日、ていでん住にンリルベは私

Ὡστόσο, ποτέ δεν έμαθαν ελληνικά και γνωρίζω περισσότερα γράμματα μόνο και μόνο επειδή είχα ανάγκη αυτό για τα μαθηματικά.

دای اینیج فوراستسیونغ است، داس مان این سکریفٹ بینوتست، دای اوتش دییز زیتشن دارستین کان.  
فی دیسم تقع ناتورلیش این آرایسش سکریفٹ.

Durch die Verwendung von Schriften im Format OpenType ist dies im Allgemeinen kein Problem mehr. So wurde in obigem Beispiel temporär für Japanisch auf die Schrift Kozuka Gothic Pro<sup>2</sup>, für Griechisch auf Lucida Sans Unicode<sup>3</sup> und für Arabisch auf die TrueType-Schrift ArabicTypesetting<sup>4</sup> umgeschaltet. Die Definitionen lauten:

```
\newfontfamily\LucidaUnicode[Scale=0.9]{LucidaSansUnicode}
\newfontfamily\KozGo[Scale=0.88]{KozGoPro-Regular}
\newfontfamily\Arabic[Scale=MatchUppercase,Script={Arabic}]{ArabicTypesetting}
```

<sup>2</sup><http://www.adobe.com/products/type/font-designers/masahiko-kozuka.html>

<sup>3</sup>[https://en.wikipedia.org/wiki/Lucida\\_Sans\\_Unicode](https://en.wikipedia.org/wiki/Lucida_Sans_Unicode)

<sup>4</sup><https://www.microsoft.com/typography/fonts/font.aspx?FMID=1704>

### 1.2.1 LuaTeX oder Lua<sup>A</sup>TeX?

Im Umgang mit L<sup>A</sup>TeX wird der Anfänger mit einer Vielzahl an Programmnamen konfrontiert, deren Einordnung nicht immer ganz einfach ist. Auch geht es begrifflich ziemlich durcheinander: Es wird immer vom Programm L<sup>A</sup>TeX, beziehungsweise Lua<sup>A</sup>TeX gesprochen, obwohl es im eigentlichen Sinne keine eigenständigen Programme sind. Das zugrundeliegende Programm ist meistens T<sub>E</sub>X, pdfT<sub>E</sub>X oder LuaT<sub>E</sub>X, wobei es noch reichlich andere Varianten gibt. Allen ist gemeinsam, dass sie auf dem originalen T<sub>E</sub>X von Donald Knuth basieren, welches mit jeder T<sub>E</sub>X-Distribution zur Verfügung steht. Das von Taco Hoekwater und anderen entwickelte LuaT<sub>E</sub>X ist eine Weiterentwicklung von T<sub>E</sub>X, welches die vielen Beschränkungen von T<sub>E</sub>X aufhebt und zudem die Integration der Skriptsprache Lua in T<sub>E</sub>X-Dokumente erlaubt.

This is LuaTeX, Version 1.07.0 (TeX Live 2018/dev)

Execute 'luatex --credits' for credits and version details.

There is NO warranty. Redistribution of this software is covered by the terms of the GNU General Public License, version 2 or (at your option) any later version. For more information about these matters, see the file named COPYING and the LuaTeX source.

LuaTeX is Copyright 2017 Taco Hoekwater and the LuaTeX Team.

L<sup>A</sup>TeX wiederum ist nichts anderes als eine Sammlung (Bibliothek) von Makros (Befehlen), die in die jeweilige T<sub>E</sub>X-Variante fest eingebunden sind. Unter Linux ist Lua<sup>A</sup>TeX daher nichts anderes als ein Link (Verweis) auf LuaT<sub>E</sub>X. Da Windows dieses Verlinken nicht unterstützt, ist hier Lua<sup>A</sup>TeX im Prinzip eine Kopie von LuaT<sub>E</sub>X mit eingebunden L<sup>A</sup>TeX-Makros.

### 1.2.2 Die T<sub>E</sub>X-Distribution

Das T<sub>E</sub>X-System als Ganzes ist eine ziemlich komplexe Angelegenheit, sodass man für eine Installation auf eine sogenannte Distribution zurückgreift, die sich »um alles kümmert«. Es gibt im Prinzip nur zwei freie Distributionen: T<sub>E</sub>XLive (<http://www.tug.org/texlive>) und MiK<sub>T</sub>EX (<http://www.miktex.org>). T<sub>E</sub>XLive unterstützt alle bekannten Betriebssysteme (Windows, MAC OSX, Linux in fast allen Varianten), MiK<sub>T</sub>EX dagegen nur Windows. Für welche der beiden man sich entscheidet, ist unerheblich; das Ergebnis, die erzeugte PDF-Datei wird identisch sein. Daneben gibt es noch einige kommerzielle Varianten, auf die hier jedoch nicht weiter eingegangen wird.

## 1.3 Das Prinzip

Es wurde bereits darauf hingewiesen, dass man sich bei der Erstellung des Textes eher als *Programmierer* denn als *Texteingabe* fühlen sollte. Dies erleichtert das Verständnis für das Prinzip bei der Texteingabe. Bei einer optimalen Situation kann man bei Auswahl einer bestimmten Dokumentenklasse erreichen, dass man lediglich die Sprachauswahl,

Eingabekodierung und Fontkodierung vorgeben muss. Sämtliche anderen Vorgaben, die das Layout des Dokumentes betreffen, sind bereits in der gewählten Dokumentenklasse vorgegeben, sodass der sogenannte Übersetzungsvorgang ein fertig formatiertes Dokument höchster Qualität liefert.

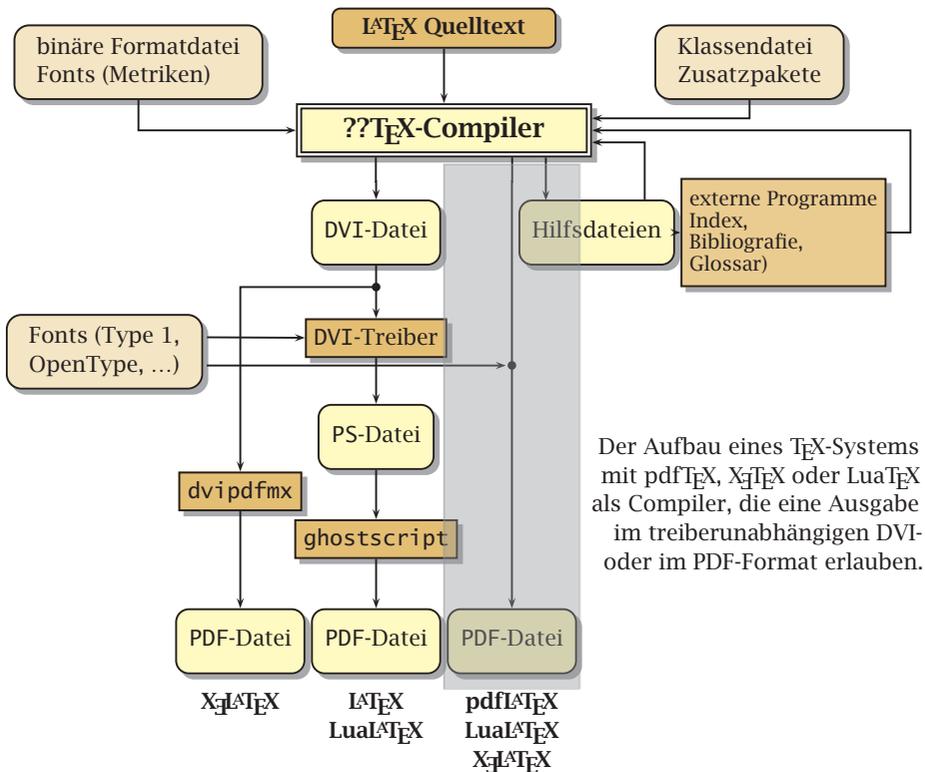


Abbildung 1.5: »Von der Quelle bis zur Mündung ...«

Abbildung 1.5 zeigt den Ablauf eines Übersetzungsvorganges zur Erzeugung eines Dokumentes, wie es der augenblickliche Standard ist. Der grau hinterlegte Bereich ist der in dieser Veröffentlichung beschriebene Ablauf, der ebenso auch für dieses Buch selbst der Standard war. Der Quelltext muss wie jeder andere Quelltext eines beliebigen Programmes, beispielsweise C<sup>++</sup> oder Perl, syntaktisch korrekt sein, wenn eine Ausgabe erfolgen soll. Je nach der Komplexität des Dokumentes kann es notwendig sein, mehrere Durchläufe mit dem T<sub>E</sub>X-Compiler und anderer externer Programme vorzunehmen. Zu diesen externen Programmen zählen insbesondere solche zur Erstellung einer Bibliografie, eines Indexes oder Glossars. Die Anforderungen an den Anwender sind hierbei gering, selbst für den Fall, dass er nicht mit einem Makefile oder einer grafischen Entwicklungsumgebung (GUI) für L<sup>A</sup>T<sub>E</sub>X arbeitet. Die gesamte Fonteinbindung, die in der Abbildung 1.5 dargestellt ist, erfolgt vollautomatisch und muss den Anwender nur in seltenen Ausnahmefällen interessieren.

## 1.4 Externe L<sup>A</sup>T<sub>E</sub>X-Anwendungen

Mit LuaL<sup>A</sup>T<sub>E</sub>X wird das Dokument erstellt, welches den reinen Text mit allen Querverweisen enthält. Ohne weitere externe Programme können aber zusätzliche Elemente, wie Bibliografie, Index, Glossar, usw. im Allgemeinen nicht erstellt werden. Die Tabelle 1.1 zeigt eine Zusammenstellung einiger Programme, die automatisch bei Installation einer T<sub>E</sub>X-Distribution zur Verfügung stehen.

Tabelle 1.1: Auswahl einiger Programme, die mit jeder T<sub>E</sub>X-Distribution installiert werden.

<i>Name</i>	<i>Bedeutung</i>
tex	Das Original ...
latex	T <sub>E</sub> X mit dem Format L <sup>A</sup> T <sub>E</sub> X.
luatex	Das »neueste« T <sub>E</sub> X
lualatex	LuaT <sub>E</sub> X mit dem Format L <sup>A</sup> T <sub>E</sub> X.
biber	Das Programm zum Verarbeiten von Bibliografieverweisen im Zusammenhang mit einer BibT <sub>E</sub> X-kompatiblen Datenbank.
makeindex	Das Programm zum Verarbeiten von Indexeinträgen.
xindy	Ein neueres Programm zum Verarbeiten von Indexeinträgen.
makeglossaries	Das Programm zum Verarbeiten von Glossareinträgen.
texcount	Zählen der Wörter eines T <sub>E</sub> X-Dokumentes.
texdoc	Anzeigen von Paket- und Dokumentenklassendokumentationen.
tex4ebook	Konvertieren eines L <sup>A</sup> T <sub>E</sub> X-Dokumentes in ein EBook-Format.
pdfcrop	»Croppen« (Entfernen der weißen Ränder) eines PDF-Dokumentes.
pdfbook	Drucken eines PDF-Dokumentes im Buchformat (zweiseitig).
texhash	Aktualisieren der Datei-Datenbank eines T <sub>E</sub> X-Systems.
arara	Skript zum automatisierten Verarbeiten eines L <sup>A</sup> T <sub>E</sub> X-Dokumentes.
latexmk	Skript zum automatisierten Verarbeiten eines L <sup>A</sup> T <sub>E</sub> X-Dokumentes.
exeltex	Skript zum Exportieren von Excel-Tabellen nach L <sup>A</sup> T <sub>E</sub> X.
kpsewhich	Bestimmung des Speicherortes (Verzeichnisses) einer Datei, die Teil des T <sub>E</sub> X-Systems ist.

# Grafische Entwicklungsumgebungen

2.1 T <sub>E</sub> Xstudio . . . . .	11
2.2 Online-Editoren . . . . .	16

Grafische Benutzerumgebungen GUI – Graphical User Interface für T<sub>E</sub>X werden häufig in Anlehnung an normale Programmiersprachen auch als Integrated Development Environment (IDE) bezeichnet. Mittlerweile gibt es deren reichlich und auch für jedes bekannte Betriebssystem. Es gibt mit T<sub>E</sub>Xstudio (<http://www.texstudio.org>) oder Emacs (<http://www.gnu.org/software/emacs>) mit dem AUCT<sub>E</sub>X-Zusatz jedoch nur zwei bekanntere Variante, die für unterschiedliche Plattformen wie Linux, MacOSX und Windows im nahezu gleichen Layout zur Verfügung stehen. Im Folgenden wird ausschließlich auf den Editor T<sub>E</sub>Xstudio Bezug genommen (Abbildung 2.1 auf der nächsten Seite). Dies ist keine wesentliche Einschränkung; bis auf Emacs arbeiten fast alle Editoren nach demselben Prinzip.

## 2.1 T<sub>E</sub>Xstudio

### 2.1.1 Editoreinstellungen

Wie Abbildung 2.1 auf der nächsten Seite zeigt, besteht der Editor im Prinzip aus den drei Bereichen *Navigation* (Links), *Quelltext* (Rechts) und *Kontrollfenster* (Rechts Unten). Die Navigation dient ausschließlich dem Navigieren innerhalb des Editors, hat also prinzipiell nichts mit der zu erstellenden PDF zu tun. Insbesondere bei umfangreichen Dokumenten kann das eine erhebliche Arbeitserleichterung darstellen. Bei kürzeren Dokumenten

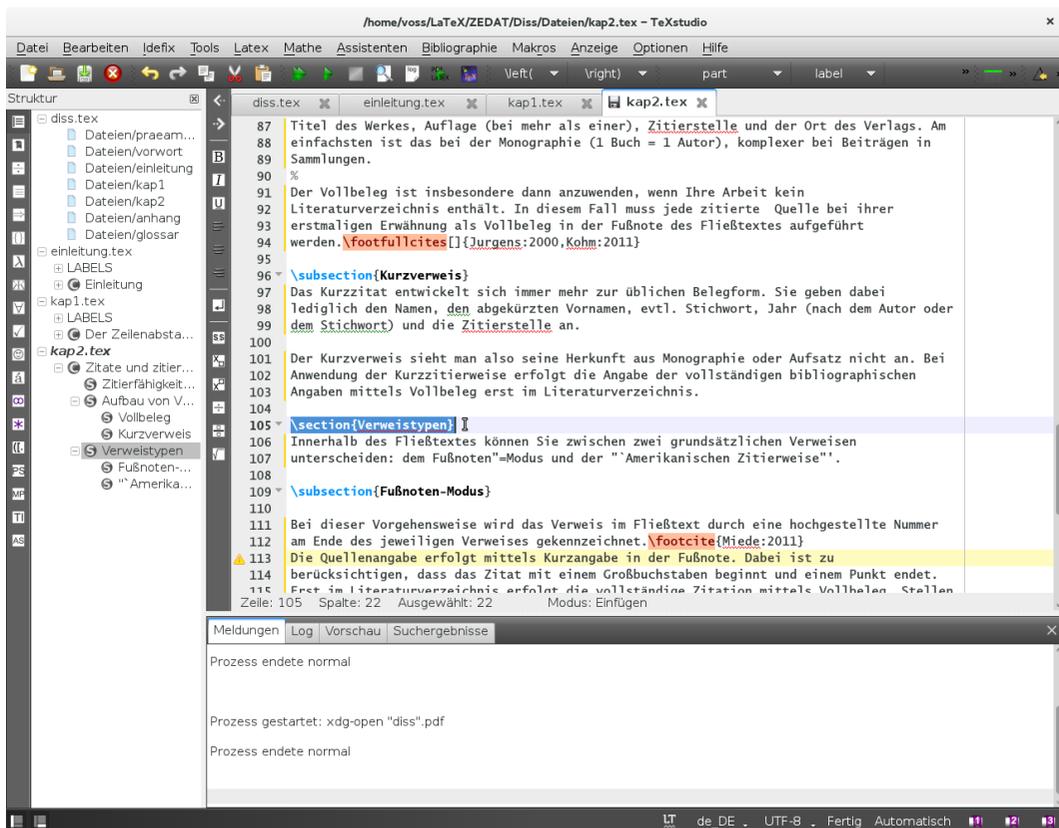


Abbildung 2.1: Beispielansicht der Entwicklungsumgebung TeXstudio für Linux

kann diese auch ausgeschaltet werden, wenn man mehr Platz für den Quelltext haben möchte. Auch das Kontrollfenster lässt sich abschalten, wobei dies nicht sonderlich sinnvoll erscheint.

Zur Erleichterung der Fehlersuche sollte bei jedem Editor grundsätzlich die Anzeige von Zeilennummern aktiviert werden (»Optionen > TeXstudio konfigurieren > Erweiterter Editor > Alle Zeilennummern«). Abbildung 2.4 auf Seite 14 zeigt das entsprechende Menü, wobei als erstes die Einstellung für die Zeilennummern durch einen schwarzen Kasten markiert ist.

Weiterhin sollte man den Text *nicht*, wie in Abbildung 2.2 auf der nächsten Seite gezeigt, eingeben. Erreicht man hier den Textrand, so schreibt der Editor automatisch in einer neuen Zeile weiter, ohne dass ein Zeilenumbruch im Quelltext eingefügt wird. Für TeX erscheint der gesamte Absatz als eine einzige Zeile.

Es ist sehr zu empfehlen, Zeilen im Quelltext immer mit einem manuellen Umbruch durch Eingabe der Enter-Taste abzuschließen. Dadurch wird die Fehlersuche erheblich vereinfacht, denn TeX gibt immer die Zeilennummer des Quelltextes aus, in der der Fehler vermutet wird. Ist diese Zeile jetzt, wie in Abbildung 2.2 auf der nächsten Seite zu sehen, sehr lang, sodass der Editor nur einen visuellen Umbruch anzeigt, dann muss der Fehler in einer sehr langen Zeile lokalisiert werden.

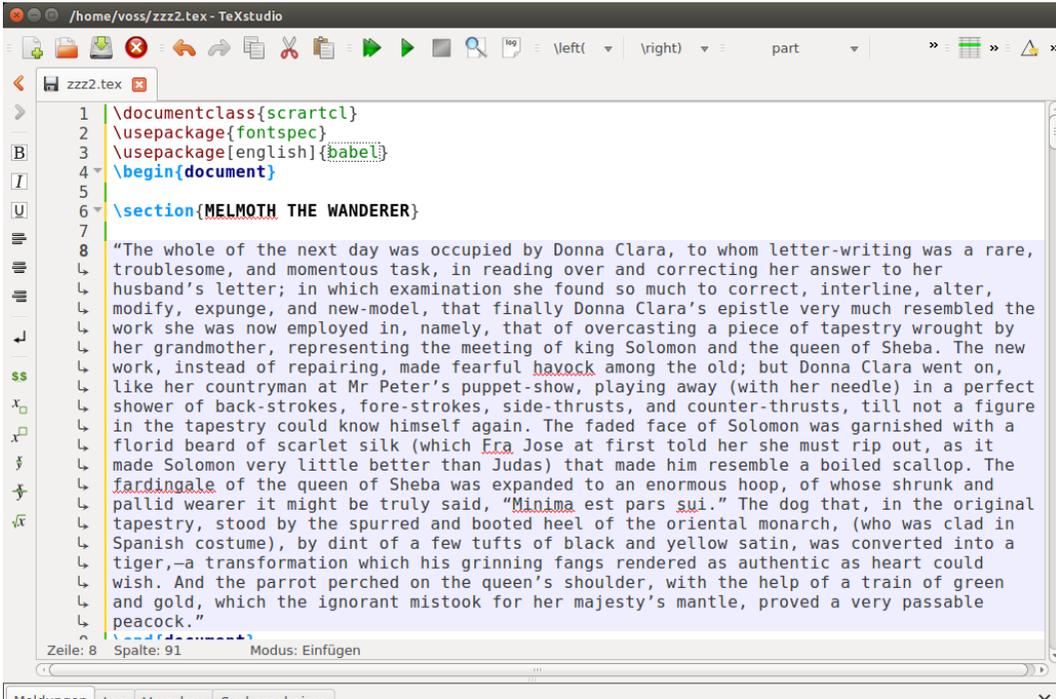


Abbildung 2.2: Fortlaufende Texteingabe ohne Betätigung der Enter-Taste.

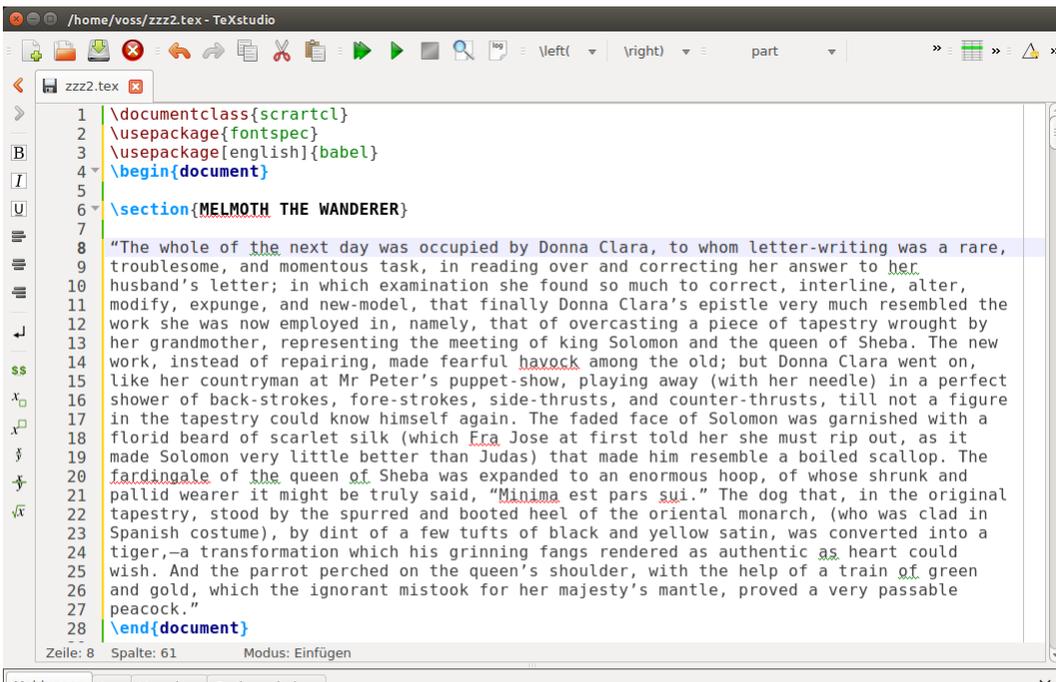
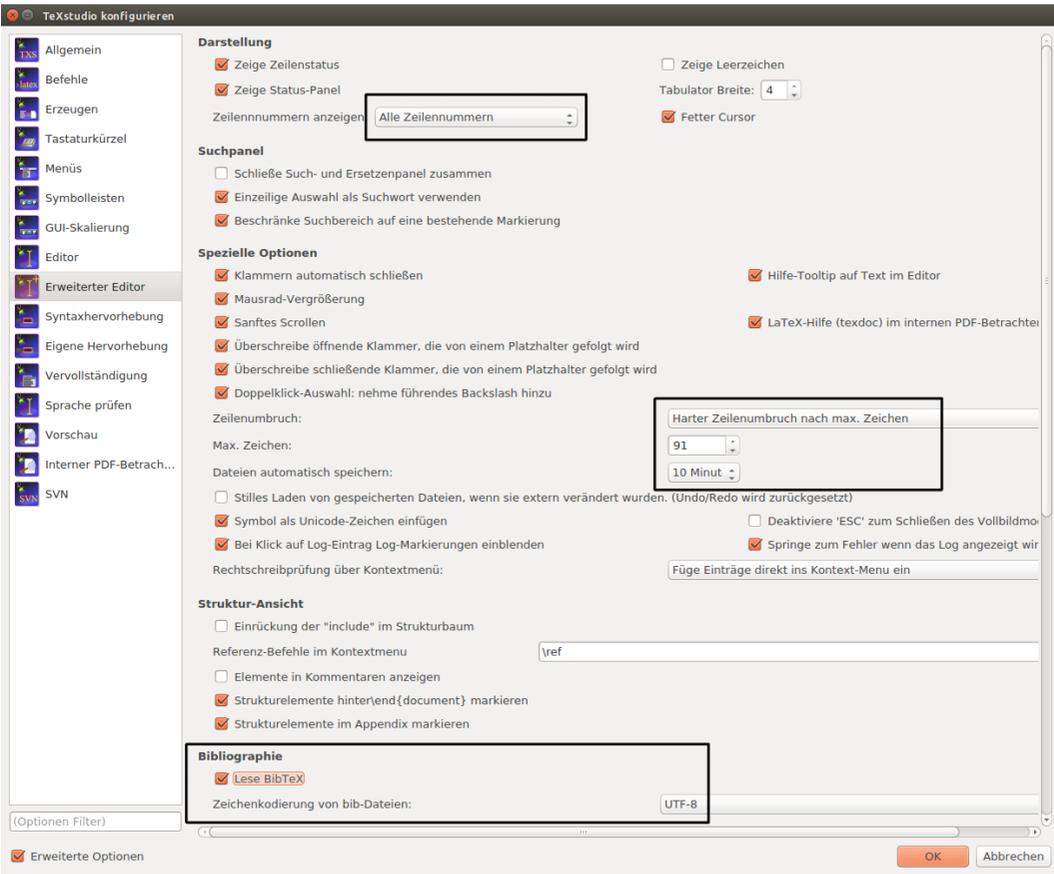


Abbildung 2.3: Fortlaufende Texteingabe ohne Betätigung der Enter-Taste und konfiguriertem harten Zeilenumbruch nach 91 Zeichen (siehe Abbildung 2.4 auf der nächsten Seite).

Abbildung 2.4: Editoreinstellungen für T<sub>E</sub>Xstudio

Der Editor lässt sich so konfigurieren, dass er explizit einen Zeilenumbruch in den Quelltext einfügt. In dem Moment, wo der Text gespeichert wird, erscheinen auch die entsprechenden Zeilennummern (Abbildung 2.3 auf der vorherigen Seite). Wo genau die Entertaste betätigt wird, ist für T<sub>E</sub>X uninteressant; es kann auch nur ein Wort in einer Zeile erscheinen. Es wird immer nur der gesamte Absatz betrachtet, nicht die einzelne Zeile. Die notwendige Einstellung für den »harten« Zeilenumbruch zeigt Abbildung 2.4. In der rechten Mitte der Abbildung ist die notwendige Einstellung durch einen Kasten hervorgehoben worden.

Als letztes markiert Abbildung 2.4 noch die notwendige Einstellung für das Einlesen von UTF-8-kodierten B<sub>I</sub>B<sub>T</sub>E<sub>X</sub>-Datenbanken, sodass man im Editor bei der Anwendung eines der Verweisbefehle entsprechende Einträge aus der Datenbank auswählen kann.

## 2.1.2 Befehlseinstellungen

Prinzipiell braucht man unter dem Menüpunkt Befehle nichts zu ändern, da die Vorgaben hinreichend sein sollten. Lediglich Windows-Nutzer sollten hier den voreingestellten AdobeReader durch SumatraPdf (<https://www.sumatrapdfreader.org/>) ersetzen. Dies ist

deswegen sinnvoll, da der AdobeReader die angezeigte PDF-Datei sperrt, sodass sie nicht bei einem erneuten L<sup>A</sup>T<sub>E</sub>X-Durchlauf überschrieben werden kann. SumatraPDF dagegen sieht keine Sperrung vor und lädt eine geänderte Datei automatisch neu.

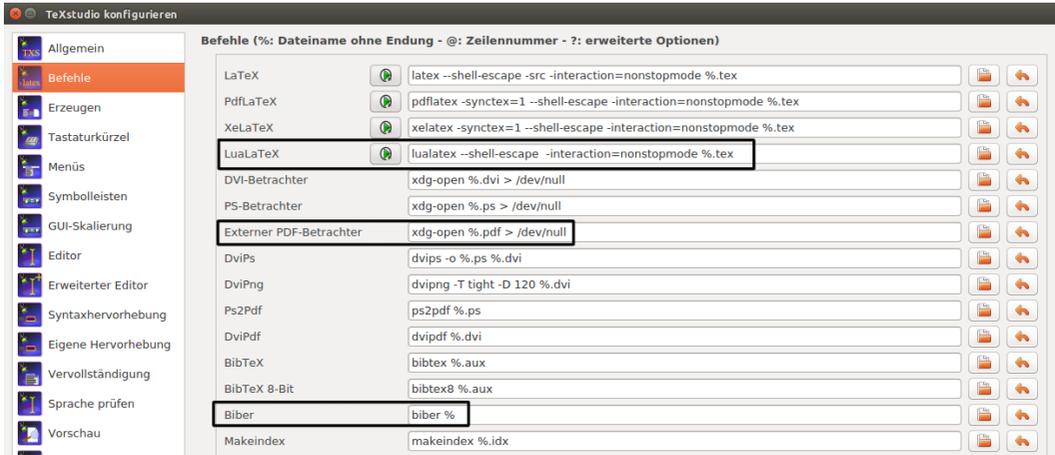


Abbildung 2.5: Befehlseinstellungen für T<sub>E</sub>Xstudio

Abbildung 2.5 zeigt für LuaL<sup>A</sup>T<sub>E</sub>X die eingefügte Option `--shell-escape`. Diese birgt ein gewisses Sicherheitsrisiko, da das Ausführen beliebiger externer Programme dadurch ermöglicht wird. Es ist daher ratsam, bei Aktivierung dieser Option keine fremden T<sub>E</sub>X-Dokumente laufen zu lassen, die man nicht vorher kontrolliert hat. Für eigene Dokumente kann diese Option bedenkenlos benutzt werden.

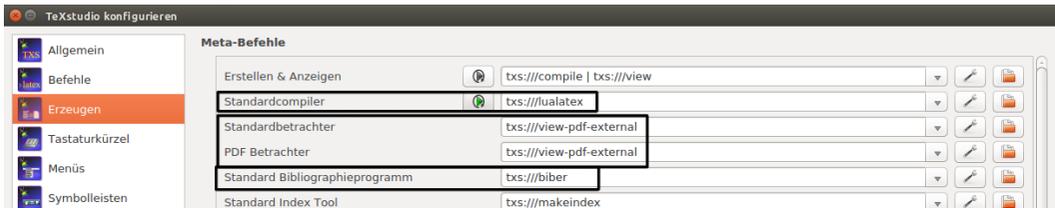


Abbildung 2.6: Standardprogramme für T<sub>E</sub>Xstudio festlegen.

Abbildung 2.6 zeigt die anzuwendenden Standardprogramme, wenn aus dem Hauptmenü heraus der einfache oder doppelte grüne Pfeil angeklickt wird. Es ist zu empfehlen, nicht den internen PDF-Viewer zu benutzen, da insbesondere beim Arbeiten mit zwei Bildschirmen das Ausgabefenster besser platziert werden kann, wenn ein externer Viewer benutzt wird.

### 2.1.3 Rechtschreibüberprüfung

Es empfiehlt sich, die Rechtschreibung gleich bei der Eingabe kontrollieren zu lassen, was nahezu alle Editoren erlauben. Lediglich beim MAC OSX kann es notwendig sein, die entsprechenden Sprachdateien selbst eintragen oder auch sogar installieren zu müssen. Abbildung 2.7 auf der nächsten Seite zeigt die Angaben, die notwendig sind, wenn die Wörterbücher von OpenOffice heruntergeladen und benutzt werden sollen.

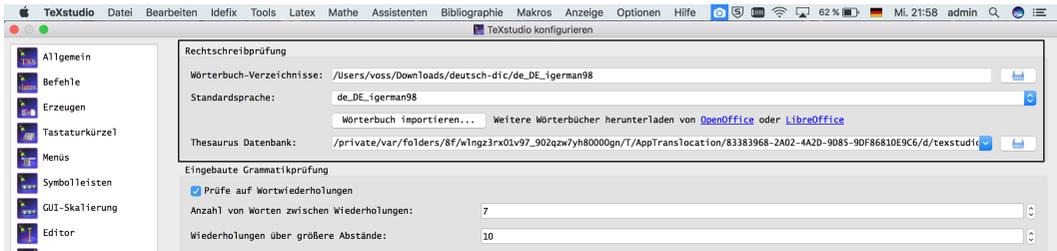


Abbildung 2.7: Installation der Wörterbücher für den MAC.

## 2.2 Online-Editoren

Seit einiger Zeit gibt es immer wieder den Versuch, Online-Editoren mit gleichzeitiger Integration eines  $\text{T}_{\text{E}}\text{X}$ -Systems der Allgemeinheit zur Verfügung zu stellen. Der letzte Versuch ist Overleaf (<http://www.overleaf.com>), welches auch zum kollaborativem Schreiben<sup>1</sup> geeignet ist. Derartige Editoren lohnen sich für wissenschaftliche Arbeiten nur dann, wenn man nicht willens oder nicht in der Lage ist, ein  $\text{T}_{\text{E}}\text{X}$ -System zu installieren. Ob es sinnvoll ist, die Quellen für eine wissenschaftliche Arbeit auf externen, allgemein nicht zugänglichen Rechnern zu speichern, die sich außerhalb der eigenen wissenschaftlichen Einrichtung befinden, muss jeder für sich selbst entscheiden.

<sup>1</sup>Projekte mit mehreren Autoren, die »gleichzeitig« am Text arbeiten.

# Erste Schritte mit L<sup>A</sup>T<sub>E</sub>X

3.1 Struktur eines L <sup>A</sup> T <sub>E</sub> X-Dokuments . . . . .	17
3.2 Die Dokumentenklasse . . . . .	19
3.3 Globale Optionen . . . . .	20
3.4 Der Satzspiegel . . . . .	20
3.5 Nationale Besonderheiten . . . . .	27

## 3.1 Struktur eines L<sup>A</sup>T<sub>E</sub>X-Dokuments

Die Minimalstruktur eines Quelltextes für die deutsche Sprache zum erfolgreichen Übersetzen in eine PDF-Datei zeigt das folgende Beispiel (Ausgabe auf der nächsten Seite):

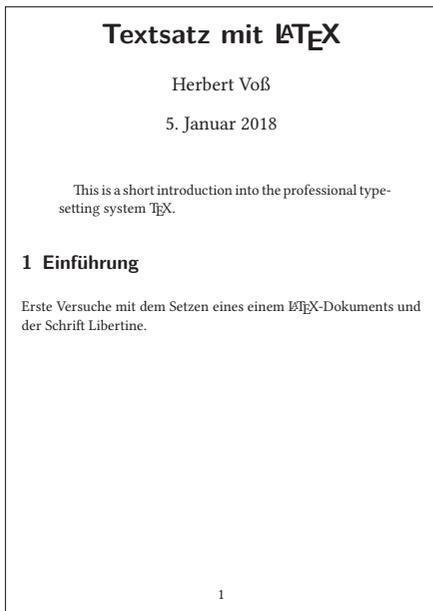
```

\documentclass[paper=a5,english,ngerman,parskip=half]{scrartcl}
    Dokumentenklasse scrartcl mit den globalen Optionen paper=a5 für das Papierformat und Sprachunterstützung für english und ngerman für die deutsche Sprache («neue» – ab 1996 gültig). Diese Optionen werden sowohl von der Dokumentenklasse als auch allen folgenden Paketen ausgewertet.
\usepackage{fontspec}
    Lade das Paket fontspec für die Schriftkodierung (font encoding), damit auch andere Schriften als die standardmäßige Latin Modern geladen werden können.
\setmainfont{LibertinusSerif}
    Benutze als Hauptschrift Libertinus Serif (Serifenschrift).
\usepackage{babel}
    Lade das Sprachpaket babel, welches die globalen Optionen english und ngerman der Dokumentenklasse auswertet. Es werden die Trennmuster für diese beiden Sprachen (Dialekte) geladen und alle sprachspezifischen Dinge direkt nach \begin{document} initialisiert, beispielsweise das Datumsformat.

```

- `\title{Textsatz mit \LaTeX}`  
Definiere den Titel.
- `\author{Herbert Voß}`  
Definiere den Autor.
- `\date{\today}`  
Definiere das Datum.
- `\begin{document}`  
Beginn des Dokumentenkörpers mit Deutsch als Standardsprache.
- `\foreignlanguage{english}{...}`  
Andere Sprache wählen, damit die korrekten Trennmuster für Englisch verwendet werden.
- `\begin{abstract}.. \end{abstract}`  
Fremdsprachige Zusammenfassung.
- `\maketitle`  
Formatierten Titel für die Angaben von `\title`, `\author` und `\date` ausgeben.
- `\section{Einführung}`  
Abschnittsüberschrift.
- `\end{document}`  
Ende des Dokuments.

03-01-1



```

\documentclass[a5paper,english,
  ngerman,parskip=half-]{scrartcl}
\usepackage{fontspec}
\setmainfont{LibertinusSerif}
\usepackage{babel}
\title{Textsatz mit \LaTeX}
\author{Herbert Voß}
\date{\today}
\begin{document}
\maketitle
\foreignlanguage{english}{%
\begin{abstract}
This is a short introduction into
the professional typesetting system \TeX.
\end{abstract}}
\section{Einführung}
Erste Versuche mit dem Setzen
eines  $\LaTeX$ -Dokuments
und der Schrift Libertine.
\end{document}

```

Der grundsätzliche Aufbau eines Dokuments teilt sich in Präambel und Dokumentenkörper auf. Die Präambel enthält prinzipiell all das, was an allgemeinen Definitionen und zusätzlichen Festlegungen für das gesamte Dokument vorgenommen wird. Dabei können zusätzlich auch Dinge überschrieben beziehungsweise neu definiert werden, die bereits in der Dokumentenklasse festgelegt wurden. Die Angabe der Dokumentenklasse muss dabei nicht zwingend die erste Zeile sein. Je nach Notwendigkeit können andere Befehle vorher erscheinen.

...	}	Präambel
<code>\documentclass [Optionen] {Name} [Version]</code>		
...	}	Textkörper
<code>\begin{document}</code>		
<code>\end{document}</code>		

### 3.2 Die Dokumentenklasse

Die Dokumentenklasse wird über den Befehl `\documentclass` ausgewählt, wobei in der Regel der gewählten Klasse Optionen mitgegeben werden, die dann *sowohl* von der Dokumentenklasse als auch von den verwendeten Paketen ausgewertet werden.

```
\documentclass [Optionen] {Name} [Version]
```

Das zweite optionale Argument (Version) wird selten benutzt, da man im Allgemeinen eine aktuelle Ausgabe verwendet. Ansonsten kann man hier ein Datum in der Form `jjjj/mm/tt` angeben, welches dann zu einer Warnung in der Logdatei führt, wenn die verwendete Klasse älter ist. Je besser eine vorhandene Dokumentenklasse auf die eigenen Bedürfnisse zugeschnitten ist, desto weniger Aufwand wird für Re- oder Neudefinitionen bestimmter Umgebungen oder Makros betrieben werden müssen. Standardmäßig gibt es die vier Dokumentenklassen `book`, `article`, `report` und `letter` mit den folgenden Eigenschaften, wobei hier in Klammern die entsprechenden Namen der von Markus Kohm entwickelten KOMA-Script-Klassen angegeben sind:

---

<b>book</b>	– Titel auf einer eigenen Seite;
<b>(scrbook)</b>	– Seitenzählung mit römischen und arabischen Ziffern;
	– Ebenen <code>\part</code> , <code>\chapter</code> , <code>\section</code> , <code>\subsection</code> , <code>\subsubsection</code> und <code>\paragraph</code> ;
	– Kapitelweise Nummerierung der Abbildungen, Tabellen und Gleichungen.

---

<b>report</b>	– Wie Klasse <code>book</code> , aber nicht mit den Möglichkeiten von <code>\frontmatter</code> ,
<b>(scrreprt)</b>	<code>\mainmatter</code> und <code>\backmatter</code> .

---

<b>article</b>	– Titel auf keiner eigenen Seite;
<b>(scrartcl)</b>	– Seitenzählung mit arabischen Ziffern;
	– Ebenen <code>\part</code> , <code>\section</code> , <code>\subsection</code> , <code>\subsubsection</code> , <code>\paragraph</code> ;
	– Fortlaufende Nummerierung der Abbildungen, Tabellen und Gleichungen.

---

<b>letter</b>	– Keine Ebenen;
<b>(scrletter2)</b>	– Elemente, die ein Brief aufweist, wie Absender, Anschrift, usw.

---

In (Voß 2014) findet sich eine Liste der mehr als 150 derzeit bekannten Dokumentenklassen, die mit jeder standardmäßigen TeX-Installation zur Verfügung stehen. Jede Klasse hat ihre eigenen Optionen, die zur Anpassung der Dokumentenklasse an die