

Herbert Voß

Die wissenschaftliche Arbeit mit L^AT_EX

unter Verwendung von Lua_TE_X,
KOMA-Script und Biber/BibL_AT_EX

2. Auflage



```
\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\
März 2006\
\large\textbf{Jana Voß}
\end{tabular}/par\vspace*{\fill}
Gutachter:
\begin{tabular}[t]{l}
Prof.\,Dr. Jürgen Trabant\
Prof.\,Dr. Thomas Kotschi
\end{tabular}
\end{center}
\end{titlepage}
```

Für Raphael,
der das jetzt schon lesen kann.

Die wissenschaftliche Arbeit mit L^AT_EX

unter Verwendung von LuaT_EX,
KOMA-Script und Biber/BibL^AT_EX

Zweite, überarbeitete und erweiterte Auflage

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.

© 2021 Herbert Voß, Berlin

Zweite Auflage

ISBN 978-3-96543-217-8

Umschlag: Herbert Voß

Karikaturen: Duane Bibby, mit Genehmigung der TUG

Satz: L^AT_EX mit LuahbT_EX

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

Druck: Dimograf – Bielsko-Biała – Polen

Inhaltsverzeichnis

Vorwort	1
1 Einführung	1
1.1 Warum L ^A T _E X und nicht die Textverarbeitung xy?	2
1.2 Warum LuaL ^A T _E X und nicht pdfL ^A T _E X?	7
1.3 Das Prinzip	9
1.4 Externe L ^A T _E X-Anwendungen	10
2 Grafische Entwicklungsumgebungen	13
2.1 T _E Xstudio	13
2.2 Online-Editoren	18
3 Erste Schritte mit L^AT_EX	23
3.1 Struktur eines L ^A T _E X-Dokuments	23
3.2 Die Dokumentenklasse	25
3.3 Globale Optionen	26
3.4 Der Satzspiegel	26
3.5 Nationale Besonderheiten	33
4 Schriften	37
4.1 Einführung	37
4.2 Schriftauswahl	38
4.3 Schriftnamen und Dateinamen	46
4.4 Kombination von Schriften	49
4.5 Was man wissen sollte	51
5 Sprachunterstützung	55
5.1 Einführung	55
5.2 Das Sprachpaket babel	55
5.3 Trennungen	57
5.4 Anführungszeichen	62
5.5 Rechtschreibkontrolle	65

6	Textelemente	67
6.1	Allgemeine Festlegungen	67
6.2	Der Zeilenabstand	70
6.3	Die Titelseite	72
6.4	Der Seitenstil	76
6.5	Ausrichten von Text.	82
6.6	Auszeichnung von Text	89
6.7	Hoch- und tiefgestellter Text.	92
6.8	Fußnoten.	92
6.9	Endnoten.	96
6.10	Randbemerkungen	97
6.11	Textmarken und -referenzen	101
6.12	Hyperlinks.	104
6.13	Umbruch.	111
6.14	Horizontale und vertikale Abstände	118
6.15	Mehrspaltiger Text	121
6.16	Kritische Editionen	125
6.17	Große Dokumente organisieren	126
7	Gliederung	131
7.1	Die Hauptebenen	131
7.2	Nichtnummerierte Ebenen	134
7.3	Definition neuer Gliederungsebenen	135
8	Listen	141
8.1	Standardlisten	141
8.2	Das Paket enumitem.	148
9	Tabellen	153
9.1	Einführung	153
9.2	Die tabular-Umgebung	154
9.3	Pakete zur Tabellenformatierung ohne Seitenumbruch	157
9.4	Pakete zur Tabellenformatierung mit Seitenumbruch	188
9.5	Farbe in Tabellen	199
9.6	Beispiele	202
10	Abbildungen einfügen	209
10.1	Einführung	209
10.2	Anwendung	211
10.3	PDF-Dokumente.	213
11	Gleitumgebungen	215
11.1	Definition	216
11.2	Platzierung	217
11.3	Parameter	219
11.4	Über- oder Unterschriften	222
11.5	Seitliche Beschriftungen	228
11.6	Die Formatierung der Beschriftung	230

11.7	Nicht-gleitende Abbildungen und Tabellen	236
11.8	Mehrere Objekte in einer Gleitumgebung	239
11.9	Ganz- und mehrseitige Objekte mit dem Paket hvfloat	246
11.10	Textumflossene Tabellen und Abbildungen	259
11.11	TeXnisches	260
12	Verzeichnisse	263
12.1	Standardverzeichnisse	263
12.2	Eigene Verzeichnisse erstellen	270
12.3	Stichwortverzeichnis	272
12.4	Glossar	282
13	Bibliografie	293
13.1	Einführung	293
13.2	Bibliografie-Datenbanken	295
13.3	Bibliografien mit BibL ^A T _E X und Biber	310
14	Mathematik	333
14.1	Typografische Regeln	334
14.2	Das Paket amsmath	335
14.3	Der Zeilenmodus	336
14.4	Mathematik im abgesetzten Modus	340
14.5	Matrix-Umgebungen	351
14.6	Gleichungsnummern	353
14.7	Marken (Label)	355
14.8	Weitere mathematische Elemente	357
15	Grafiken erstellen	375
15.1	Koordinaten	376
15.2	Einfache Pfade	377
15.3	Füllen von Pfaden	379
15.4	Pfeile	380
15.5	Knoten	381
15.6	Bibliotheken	383
15.7	Pakete, die auf TikZ aufbauen	385
A	Anhang	393
A.1	L ^A T _E X3	393
A.2	Längen	395
A.3	TeXLive- Installation	396
A.4	Lua _T E _X	397
A.5	Tastaturkürzel	399
A.6	Fragen und Antworten	399
	Bibliografie	411
	Index der Befehle und Begriffe	415
	Personen	436



Vorwort

Die Anwendung von L^AT_EX 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^AT_EX, beziehungsweise das darunterliegende LuaT_EX, als Programm zur Dokumentenerstellung,
- Dokumentenklassen aus dem KOMA-Script-Paket und
- Biber/BibL^AT_EX als Programm/Paket zum Verarbeiten von Literaturdatenbanken und Erstellung einer oder mehrerer Bibliografien.

Mit LuaT_EX wird der zur Zeit leistungsfähigste T_EX-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^AT_EX aufruft, wird, so wie auch in der Vergangenheit üblich, das Makro-Paket L^AT_EX geladen und das Ergebnis in Form einer pdf-Datei durch Aufruf des T_EX-Compilers LuaT_EX erzeugt. Für Anwender, die bereits bisher L^AT_EX benutzten, bedeutet dies also kaum eine Umstellung in der Vorgehensweise.

Die vielfältigen neuen Möglichkeiten von LuaT_EX 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 BibTeX, 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

Vorwort zur zweiten Auflage

Die erste Auflage wurde in relativ großer Stückzahl gedruckt, um einen günstigen Verkaufspreis zu erreichen, den der Verlag »Lehmanns Media« immer anstrebt. Nach drei Jahren ist diese nun vergriffen und es lohnt sich eine Neuauflage, da sich einige wichtige Dinge im Umfeld von LaTeX geändert haben. Erweiterungen gibt es insbesondere im Bereich der Gleitumgebungen, wo jetzt auch die Ausgabe von doppelseitigen Objekten behandelt wird. Auch wurde die Bedeutung von Biber/BibLaTeX und insbesondere von der Verwendung von HarfBuzz hervorgehoben. Daneben gibt es viele kleinere Veränderungen. Das Laden von Schriften im Format OpenType oder TrueType wird mittlerweile durch mehrere Pakete unterstützt, was dem Anwender viel Arbeit abnehmen kann.

Auf Fehler in der ersten Auflage hat mich insbesondere Lukas C. Bossert hingewiesen. Dank geht auch an Lucas Bierwirth, Gunnar Gewiss, Ralf Mispelhorn und Michael Rösch für ihre Hinweise zu Ungereimtheiten im Text.

Alle Beispiele dieses Buches findet man als lauffähige LaTeX-Dokumente zum Herunterladen auf <http://hvoss.org/Books/>.

Berlin, im April 2021

Herbert Voß

Kapitel 1

Einführung

1.1 Warum \LaTeX und nicht die Textverarbeitung xy ?	2
1.2 Warum \LuaTeX und nicht \pdfLaTeX ?	7
1.3 Das Prinzip	9
1.4 Externe \LaTeX -Anwendungen	10

Die formale Ausführung einer wissenschaftlichen Arbeit kann man in zahlreichen Publikationen nachlesen, beispielsweise in Balzert u. a. (2008), Toraman (2016) und 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) und Seimert (2011). Für den Einsatz von \LaTeX zur Erstellung einer wissenschaftlichen Arbeit gibt es unter anderem die Veröffentlichungen von Machert (2013), J. Schlosser (2016), P. Schlosser u. a. (2007) und Würth u. a. (2018), wobei nur Würth u. a. (2018) als aktuell bezeichnet werden kann. Mit der hier vorliegenden Arbeit wird konsequent auf die neuesten Entwicklungen von \TeX gesetzt: Als Standardprogramm wird \LuaTeX verwendet, als Dokumentenklasse KOMA-Script und als Bibliografieprogramm Biber mit dem korrespondierenden \BibLaTeX .

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



¹Die Distribution \MacTeX ist eine spezielle Anpassung an $\text{\TeX}Live$ und \proTeX eine spezielle Version von \MiKTeX .

1.1 Warum L^AT_EX und nicht die Textverarbeitung xy?

Die Anwendung von L^AT_EX zur Erstellung einer wissenschaftlichen Arbeit kann man für den Bereich Mathematik als »flächendeckend« bezeichnen. Es besteht aber heutzutage eigentlich kein direkter Zusammenhang mehr zwischen Mathematik und L^AT_EX. Auch für Sprachwissenschaftler, die beispielsweise linguistische Bäume erstellen müssen oder andere Autoren, die sehr große Dokumente erstellen wollen, haben durch die Anwendung von L^AT_EX viele Vorteile. Im folgenden Abschnitt wird gezeigt, dass auch eine relativ kurze wissenschaftliche Arbeit durch die Anwendung von L^AT_EX mindestens ästhetische Vorteile gegenüber den allgemein bekannten Textverarbeitungsprogrammen 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 absichtlich 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 Lua_TE_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 (Lua_TE_X).

Das Programm Lua_TE_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 Lua_TE_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 Lua_TE_X 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 Lua_TE_X 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

Für jeden Absatz gilt: Zwei aufeinander folgende Zeilen müssen *gleiche* oder *benachbarte* Formatierungskriterien erfüllen!

Dadurch ist es bei Lua_TE_X 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 Lua_TE_X 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^AT_EX).

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 minimalen Abstand und kommt ohne Trennungen aus. Lua_T_EX 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_EX-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_T_EX 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 117).
 - 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) und Voß (2007) und insbesondere Knuth und Plass (1981) verwiesen. Hier sollte nur gezeigt werden, welchen Aufwand T_EX im Gegensatz zu herkömmlichen Textverarbeitungsprogrammen treibt, um einen optimierten Umbruch zu erreichen.

Was man jetzt schon wissen sollte ...

Bei der Texterstellung mit L^AT_EX 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
\AddToHook{begindocument}{\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_EX die erste und letzte Zeile über den Rand! Es bleibt natürlich die Frage, wieso kann das jede Textverarbeitung vermeiden, aber LuaT_EX nicht. Im Gegensatz zu einer Textverarbeitung, die Zwischenräume in ihrer Größe nicht begrenzt, erlaubt LuaT_EX per *Definition* nur einen maximalen Wortzwischenraum von 3.75 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. Kann aber bei schmalen Textspalten, beispielsweise in Tabellen, durchaus häufiger auftreten, wenn die zugrundeliegende Sprache relativ lange Silben aufweist.

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
\AddToHook{begindocument}{\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 sein wird:

- 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 62).



Um ein Gefühl für die Häufigkeit zu bekommen, sei darauf hingewiesen, dass für dieses Buch die Anwendung der Umgebung `sloppypar` nur für den Absatz mit langen nicht-trennbaren Makronamen nötig war (siehe auf Seite 164). 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 (siehe Beispiel 01-01-3). 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.

01-01-3

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.

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^AT_EX und nicht pdf^AT_EX?

Man kann zur Zeit noch davon ausgehen, dass der größere Teil der Nutzer von ^AT_EX seine Dokumente mit pdf^AT_EX erstellen dürfte. Das Hauptargument dafür ist schlicht und ergreifend »never change a running system«. Mit der Anwendung von pdf^AT_EX benutzt man faktisch ein abgekapseltes System, welches Quelldokumente verarbeitet, die völlig unabhängig vom Betriebssystem und der zugrundeliegenden ^AT_EX-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^AT_EX teilweise erheblich. Mit Lua^AT_EX, 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

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

دای ایترنج فوراستسیونگ است، داس مان این سکریت بینوتست، دای اوتش دیز زیتشن دارستین کان. فی دیسم تقع نانورلیش این آرایسش سکریت.

Durch die Verwendung von Schriften im Format TrueType oder OpenType ist dies im Allgemeinen kein Problem mehr. So wurde in obigem Beispiel temporär für Japanisch auf die Schrift Kozuka Gothic Pro², für Griechisch auf Lucida Sans Unicode³ und für Arabisch auf die TrueType-Schrift ArabicTypesetting⁴ umgeschaltet. Die Definitionen lauten:

TrueType
OpenType

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

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

³https://en.wikipedia.org/wiki/Lucida_Sans_Unicode

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

1.2.1 LuaTeX oder Lua^ATeX?

Im Umgang mit $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ wird der Anfänger mit einer Vielzahl an Programmnamen konfrontiert, deren Einordnung nicht einfach ist. Auch geht es begrifflich ziemlich durcheinander: Es wird immer vom Programm $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, beziehungsweise $\text{LuaL}^{\text{A}}\text{T}_{\text{E}}\text{X}$ gesprochen, obwohl es im eigentlichen Sinne keine eigenständigen Programme sind. Das zugrundeliegende Programm ist meistens $\text{T}_{\text{E}}\text{X}$, $\text{pdfT}_{\text{E}}\text{X}$ oder $\text{LuaHbT}_{\text{E}}\text{X}$, wobei es noch weitere Varianten gibt. Allen ist gemeinsam, dass sie auf dem originalen $\text{T}_{\text{E}}\text{X}$ von Donald Knuth, einschließlich der Erweiterung durch $\text{eT}_{\text{E}}\text{X}$, basieren, welches mit jeder $\text{T}_{\text{E}}\text{X}$ -Distribution zur Verfügung steht. Das von Taco Hoekwater und anderen entwickelte $\text{LuaHbT}_{\text{E}}\text{X}$ ist eine Weiterentwicklung von $\text{T}_{\text{E}}\text{X}$, welches die vielen Beschränkungen von $\text{T}_{\text{E}}\text{X}$ aufhebt und zudem die Integration der Skriptsprache Lua in $\text{T}_{\text{E}}\text{X}$ -Dokumenten erlaubt.

This is LuaHBTeX, Version 1.14.0 (TeX Live 2021)

Execute 'luahbtex --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 2021 Taco Hoekwater and the LuaTeX Team.

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ wiederum ist nichts anderes als eine Sammlung (Bibliothek) von Makros (Befehlen), die in die jeweilige $\text{T}_{\text{E}}\text{X}$ -Variante fest eingebunden sind. Unter Linux ist $\text{LuaL}^{\text{A}}\text{T}_{\text{E}}\text{X}$ daher nichts anderes als ein Link (Verweis) auf $\text{LuaHbT}_{\text{E}}\text{X}$. Da Windows dieses Verlinken nicht unterstützt, ist hier $\text{LuaL}^{\text{A}}\text{T}_{\text{E}}\text{X}$ im Prinzip eine Kopie von $\text{LuaHbT}_{\text{E}}\text{X}$ mit eingebundenen $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Makros. Hierbei steht »hb« für HarfBuzz⁵ (<https://de.wikipedia.org/wiki/HarfBuzz>, <https://harfbuzz.github.io/>), welches eine freie Grafikbibliothek für Schriften ausschließlich im Format OpenType ist und nun standardmäßig in $\text{LuaT}_{\text{E}}\text{X}$ eingebunden ist. Unterstützt werden vorrangig indische, orientalische und Thai-Sprachen. Aktiviert werden muss die HarfBuzz-Engine über das optionale Argument beim Laden einer Schrift, was im folgenden Beispiel für Burmesisch und Arabisch gezeigt wird. Im ersten Fall ist das Wort jeweils ohne HarfBuzz und im zweiten Fall dann mit HarfBuzz gesetzt worden. Man erkennt deutlich die Anwendung der Ligaturen bei Benutzung von HarfBuzz.

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$



```
\usepackage{fontspec}
\newfontfamily\Arab{ArabicTypesetting}[Scale=3, Renderer=Basic, Script={Arabic}]
\newfontfamily\ArabHB{ArabicTypesetting}[Scale=3, Renderer=Harfbuzz, Script={Arabic}]
\newfontfamily\Burmese{NotoSerifMyanmar-Regular.ttf}[Renderer=Basic, Script=Myanmar, Scale=1.5]
\newfontfamily\BurmeseHB{NotoSerifMyanmar-Regular.ttf}[Renderer=Harfbuzz, Script=Myanmar, Scale=1.5]
\newenvironment{RTL}% für Arabisch
  {\par \pardir TRT\textdir TRT\bodydir TRT}
  {\par}

\Arab\begin{RTL}ديضنتل لال عري\end{RTL}

\ArabHB\begin{RTL}ديضنتل لال عري\end{RTL}

\Burmeseမုမတီခင်းလင်းကိူ
\BurmeseHBမုမတီခင်းလင်းကိူ
```

⁵Persisch für OpenType

التنضيد العربي
التنضيد العربي

မတီခင်းလင်းကြီး
မတီခင်းလင်းကြီး

1.2.2 Die T_EX-Distribution

Das T_EX-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_EXLive⁶ und MiK_TE_X⁷. T_EXLive unterstützt ebenso wie MiK_TE_X alle bekannten Betriebssysteme, wie Windows, macOS, Linux in fast allen Varianten. 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 auf der nächsten Seite zeigt den Ablauf eines Übersetzungsvorganges zur Erzeugung eines Dokumentes, wie es der augenblickliche Standard ist. Die dicke rote Linie kennzeichnet den in dieser Veröffentlichung beschriebenen Ablauf, der ebenso auch für dieses Buch selbst der Standard war. Der Quelltext muss wie jeder andere Quelltext eines beliebigen Programmes, beispielsweise C++ oder Perl, syntaktisch korrekt sein, wenn eine fehlerfreie Ausgabe erfolgen soll. Je nach der Komplexität des Dokumentes kann es notwendig sein, mehrere Durchläufe mit dem T_EX-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^AT_EX arbeitet. Die gesamte

⁶<http://www.tug.org/texlive>

⁷<http://www.miktex.org>

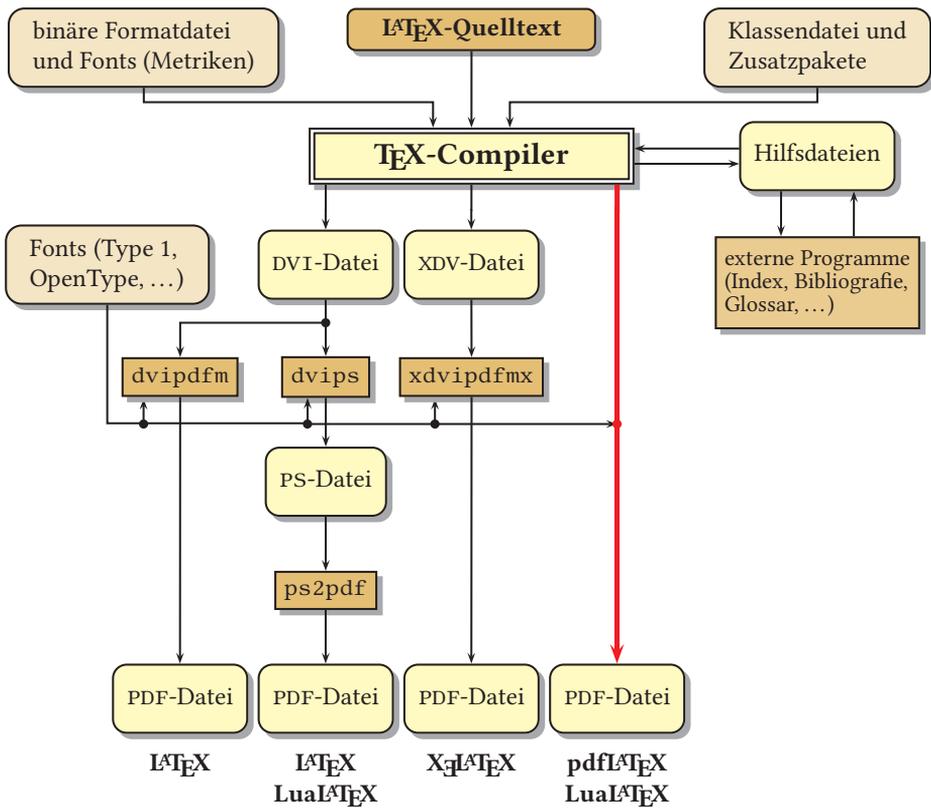


Abbildung 1.5: »Von der Quelle bis zur Mündung ...« Der Aufbau eines TeX-Systems mit pdfTeX, XeTeX oder LuaTeX als Compiler, die eine Ausgabe im PDF-Format erlauben.

Schrifteinbindung, die in der Abbildung 1.5 dargestellt ist, erfolgt vollautomatisch und muss den Anwender nur in seltenen Fällen interessieren.

1.4 Externe $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Anwendungen

Mit Lua $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 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 der Installation einer TeX-Distribution zur Verfügung stehen.

Tabelle 1.1: Auswahl einiger Programme, die mit jeder TeX-Distribution installiert werden.

Name	Bedeutung
tex	Das Original ...
latex	TeX mit dem Format $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.
luatex	Das »neueste« TeX

<i>Name</i>	<i>Bedeutung</i>
luahbtex	Das »neueste« T _E X mit Unterstützung für HarfBuzz
lualatex	LuahbT _E X mit dem Format L ^A T _E X.
dviluatex	LuahbT _E X mit einer Ausgabe im DVI-Format.
dvilualatex	Lual ^A T _E X mit einer Ausgabe im DVI-Format.
biber	Das Programm zum Verarbeiten von Bibliografieverweisen im Zusammenhang mit einer B _I B _T E _X -kompatiblen Datenbank.
bibtex	Veraltetes Programm zur Erzeugung einer Bibliografie.
makeindex	Ein Programm zum Verarbeiten von Indexeinträgen.
xindex	Ein neueres unicodefähiges Programm zum Verarbeiten von Indexeinträgen.
xindy	Ein Programm zum Verarbeiten von Indexeinträgen.
makeglossaries	Das Programm zum Verarbeiten von Glossareinträgen.
texcount	Zählen der Wörter eines T _E X-Dokumentes.
texdoc	Anzeigen von Paket- und Dokumentenklassendokumentationen.
tex4ebook	Konvertieren eines L ^A T _E X-Dokumentes in ein EBook-Format.
pdfcrop	»Croppen« (Entfernen der weißen Ränder) eines PDF-Dokuments.
pdfbook	Drucken eines PDF-Dokuments im Buchformat (zweiseitig).
texhash	Aktualisieren der Datei-Datenbank eines T _E X-Systems (File Name Data Base).
arara	Skript zum automatisierten Verarbeiten eines L ^A T _E X-Dokuments.
latexmk	Skript zum automatisierten Verarbeiten eines L ^A T _E X-Dokuments.
exeltex	Skript zum Exportieren von Excel-Tabellen nach L ^A T _E X.
asy	Skript zum Ausführen von Quellcode für das Grafiksystem Asymptote.
kpsewhich	Bestimmung des Speicherortes (Verzeichnisses) einer Datei, die Teil des T _E X-Systems ist.



Kapitel 2

Grafische Entwicklungsumgebungen

2.1 T _E Xstudio	13
2.2 Online-Editoren	18

Grafische Benutzerumgebungen (GUI – Grafical User Interface) für T_EX 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_EXstudio (<http://www.texstudio.org>) oder Emacs (<http://www.gnu.org/software/emacs>) mit dem AUCT_EX-Zusatz jedoch nur zwei bekanntere Varianten, die für unterschiedliche Plattformen wie Linux, macOS und Windows im nahezu gleichen Layout zur Verfügung stehen. Im Folgenden wird ausschließlich auf den Editor T_EXstudio 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_EXstudio

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

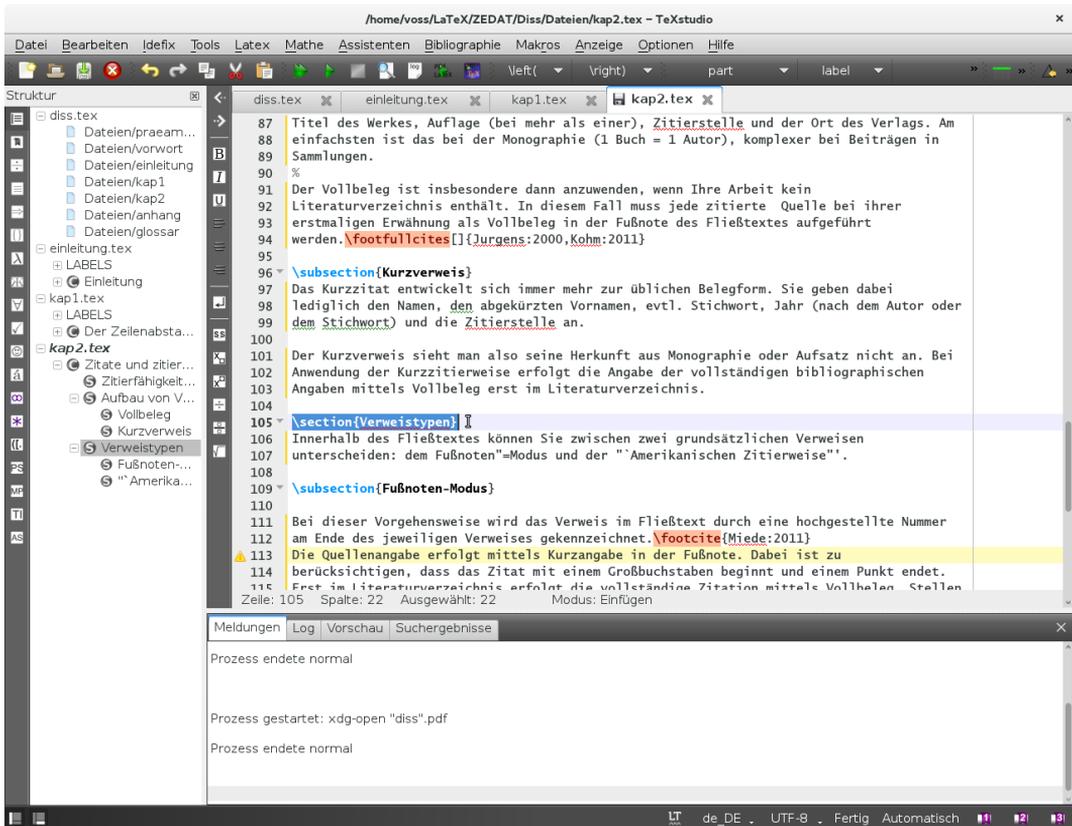


Abbildung 2.1: Beispielansicht der Entwicklungsumgebung TeXstudio für Linux

Dokumenten 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 16 zeigt das entsprechende Menü, wobei als erstes die Einstellung für die Zeilennummern durch einen gestrichelten roten Rahmen 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 daher 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.

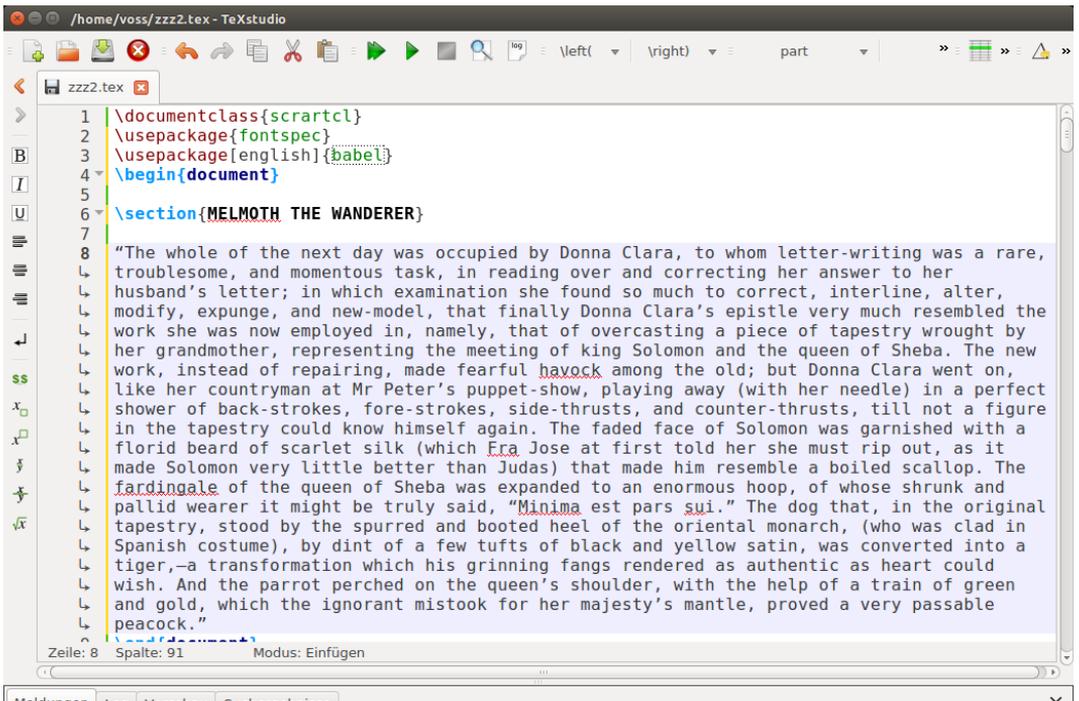


Abbildung 2.2: Fortlaufende Texteingabe ohne Betätigung der Enter-Taste.

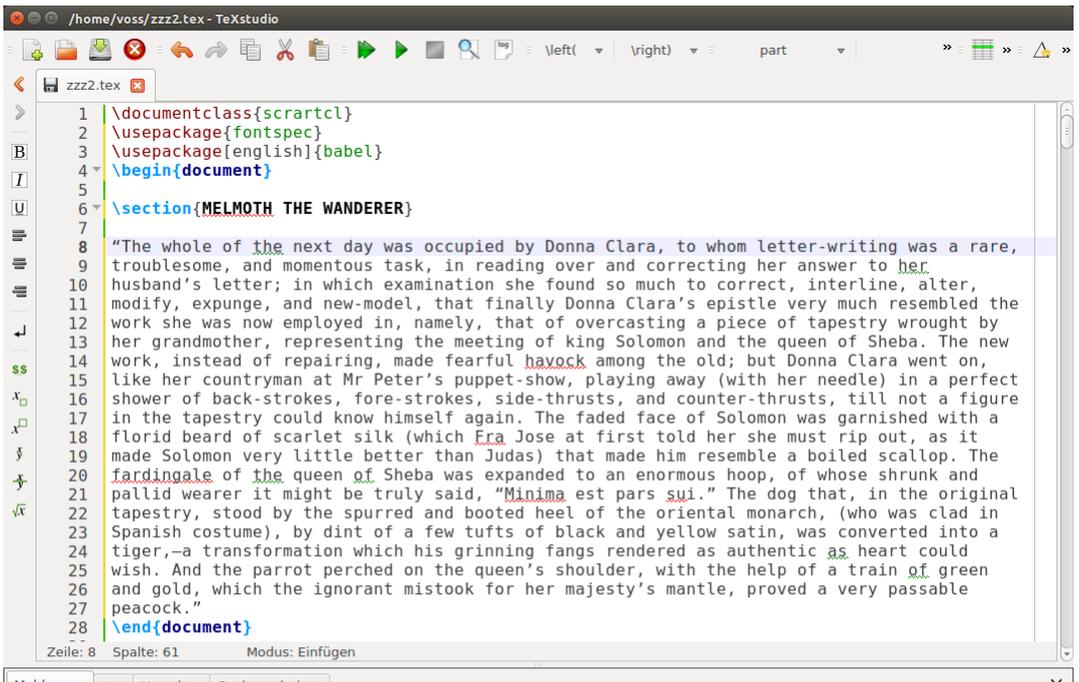
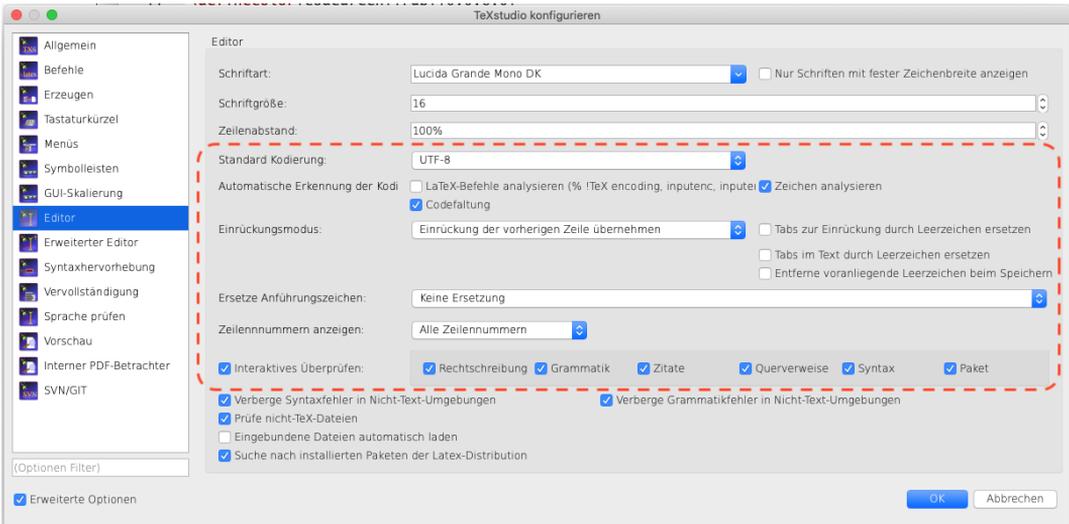
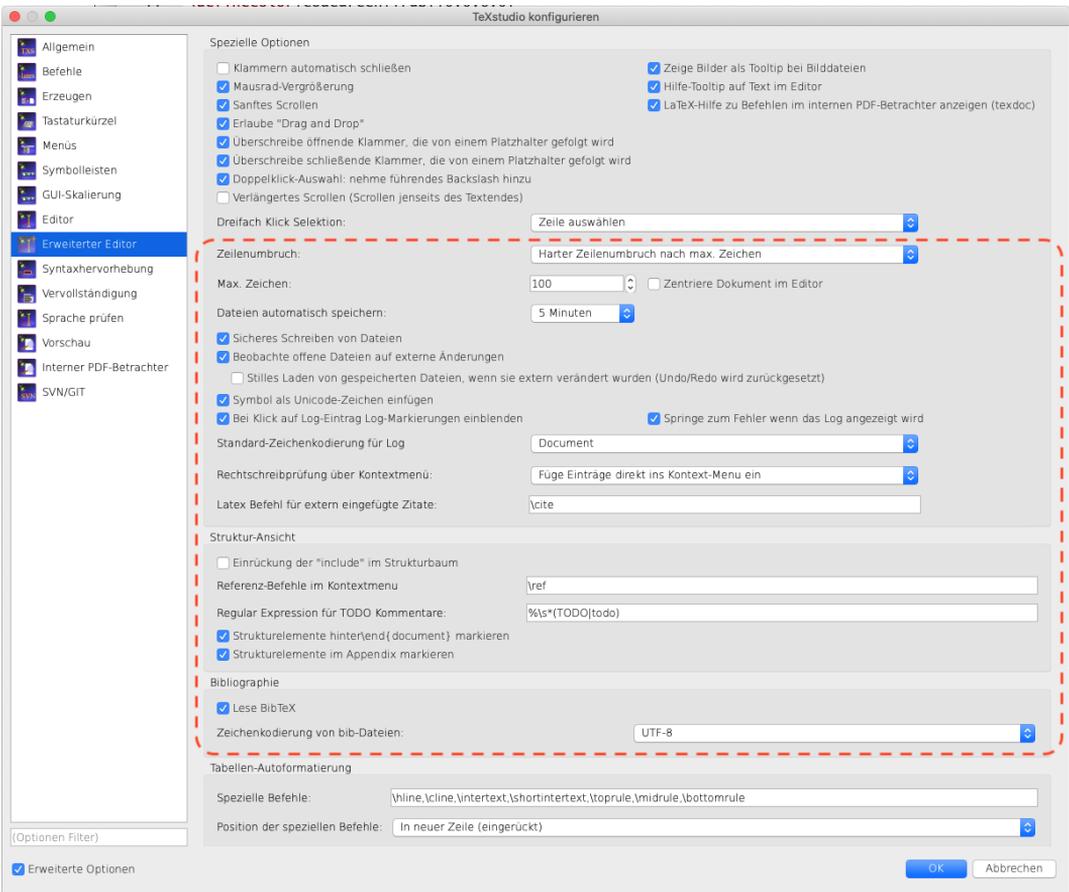


Abbildung 2.3: Fortlaufende Texteingabe ohne Betätigung der Enter-Taste und konfiguriertem harten Zeilenumbruch nach 91 Zeichen (siehe Abbildung 2.4 auf der nächsten Seite).



(a) Editoreinstellungen für TeXstudio



(b) Erweiterte Editoreinstellungen für TeXstudio

Abbildung 2.4: Menü für Einstellungen des Editors, welches für andere TeXstudio -Versionen anders aussehen kann (macOS).

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 Seite 15). Wo genau die Entertaste betätigt wird, ist für \TeX 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 2.4b auf der vorherigen Seite. An der oberen Kante des roten Rahmens der Abbildung ist die notwendige Einstellung für den Zeilenumbruch zu erkennen: »harter Zeilenumbruch nach max. Zeichen«. Darunter ist die maximale Zeichenzahl mit 100 angegeben, was im Allgemeinen in Abhängigkeit der vorhandenen Bildschirmauflösung willkürlich festgelegt wird. Als letztes markiert Abbildung 2.4b auf der vorherigen Seite noch die notwendige Einstellung für das Einlesen von UTF-8-kodierten Bib \TeX -Datenbanken, sodass man im Editor bei der Anwendung eines der Verweisbefehle entsprechende Einträge aus der Datenbank auswählen kann. Die allgemeine Kodierung ist in Abbildung 2.4a auf der vorherigen Seite im oberen Bereich der Markierung mit UTF-8 festgelegt und sollte nur geändert werden, wenn man mit einem alten Betriebssystem arbeitet.

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 \LaTeX -Durchlauf überschrieben werden kann. SumatraPDF dagegen sieht keine Sperrung vor und lädt eine geänderte Datei automatisch neu.

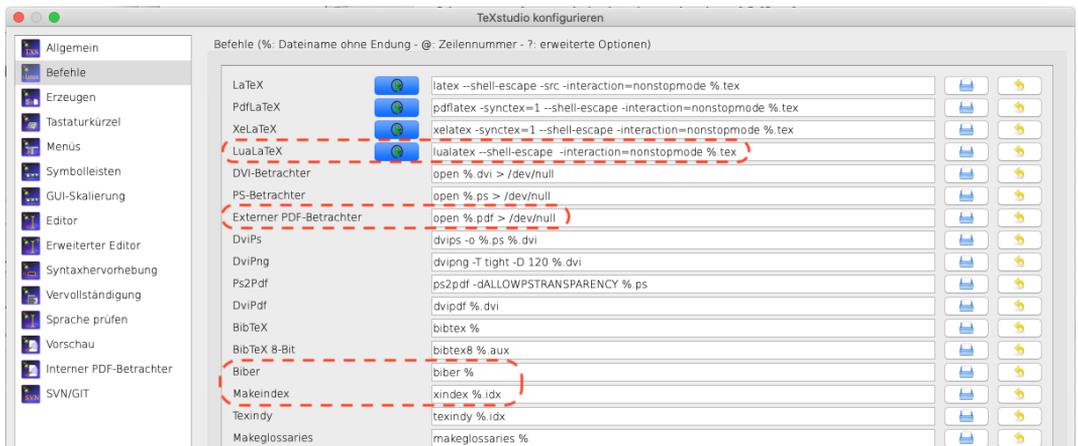


Abbildung 2.5: Befehlseinstellungen für \TeX Studio (macOS)

Abbildung 2.5 zeigt für \LaTeX die eingefügte Option `--shell-escape`, die für bestimmte Anwendungsfälle benötigt wird. Die Option birgt ein gewisses Sicherheitsrisiko, da dann aus dem \TeX -Dokument heraus das Ausführen beliebiger externer Programme ermöglicht wird. Es ist daher ratsam, bei Aktivierung dieser Option keine fremden

TeX-Dokumente laufen zu lassen, die man nicht vorher kontrolliert hat. Für eigene Dokumente kann diese Option bedenkenlos benutzt werden.

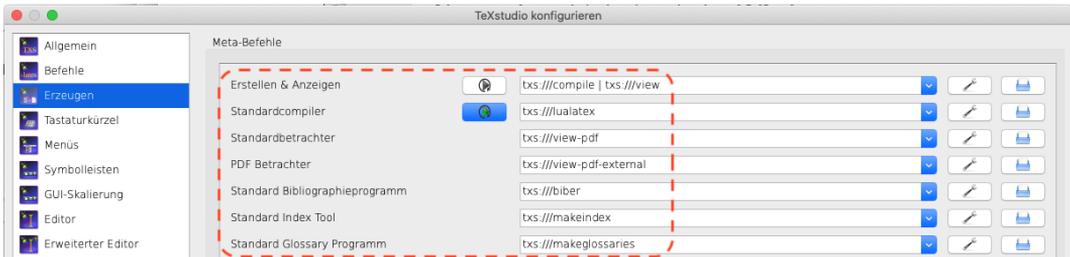


Abbildung 2.6: Standardprogramme für TeXstudio festlegen (macOS).

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. Dann kann das Fenster auf dem zweiten Bildschirm verschoben werden. Gleiches gilt für LapTops, bei denen man das Ausgabefenster hinter den Editor schieben kann, sodass genug Platz für ein sinnvolles Arbeiten mit dem Editor zur Verfügung steht.

2.1.3 Rechtschreibüberprüfung

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

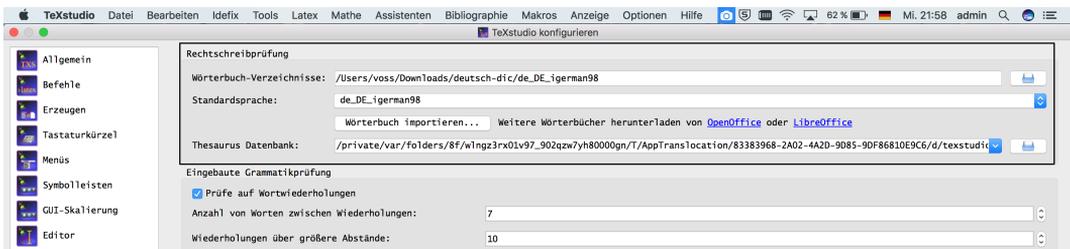


Abbildung 2.7: Installation der Wörterbücher für macOS.

2.2 Online-Editoren



Seit einiger Zeit gibt es immer wieder den Versuch, Online-Editoren mit gleichzeitiger Integration eines TeX-Systems der Allgemeinheit zur Verfügung zu stellen. Der letzte, erstmalig erfolgreiche Versuch ist Overleaf (<http://www.overleaf.com>).

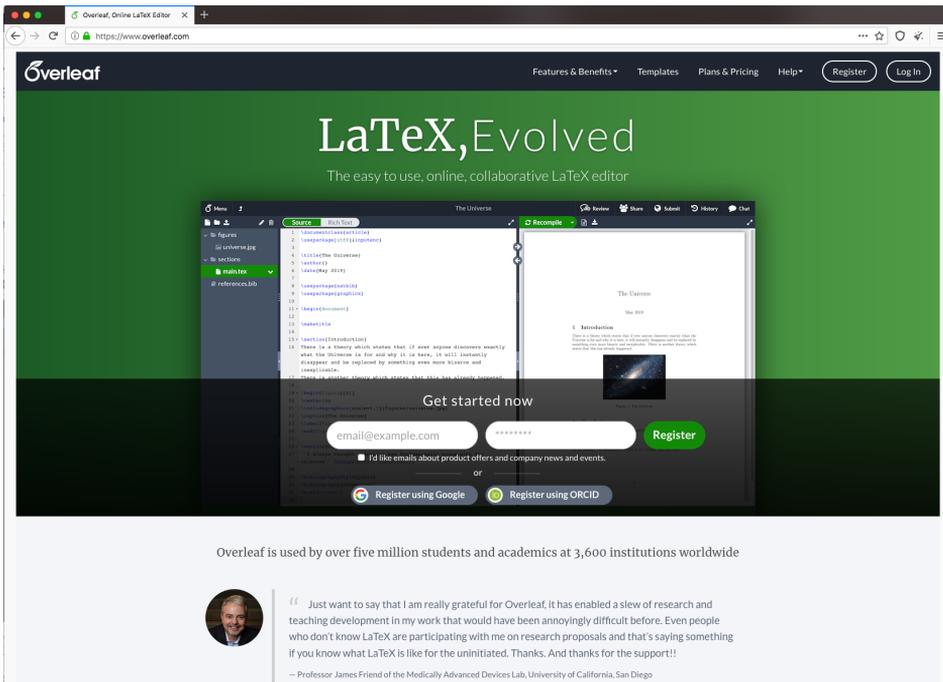


Abbildung 2.8: Startseite des Online-Editors Overleaf.

Das System eignet sich auch zum kollaborativen Schreiben.¹ 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. Abbildung 2.8 zeigt die Startseite von Overleaf.

Die einzelne private Nutzung von Overleaf ist frei und man kann nach Registrierung das System nutzen. Grundsätzlich arbeitet Overleaf im Hintergrund mit der jeweils aktuellen Version von $\text{T}_{\text{E}}\text{X}$ Live, die allerdings immer etwas verzögert zum offiziellen Erscheinungsdatum erscheint. Das Nachinstallieren von Paketen ist nicht möglich. Allerdings kann man spezielle Pakete, die nicht Teil von $\text{T}_{\text{E}}\text{X}$ Live sind, hochladen, sodass sie im Dokumentenverzeichnis gespeichert werden und damit wieder zur Verfügung stehen.

Die Bearbeitung eines einfachen Dokuments zeigt Abbildung 2.9 auf der nächsten Seite, wobei hier eine Abbildung benutzt wird, die selbst Teil von $\text{T}_{\text{E}}\text{X}$ Live ist, sodass sie nicht gesondert hochgeladen werden muss. Abbildung 2.10 auf der nächsten Seite zeigt zum Vergleich ein Projekt mit mehreren Unterverzeichnissen, deren Inhalt über die Uploadfunktion hochgeladen wurden. Das Beispiel zeigt weiterhin, dass eine Rechtschreibprüfung bei einem Dokument mit vielen chemischen Bezeichnungen schwierig ist, da die Wörter alle nicht im aktuellen Wörterbuch gespeichert sind.

¹Projekte mit mehreren Autoren, die »gleichzeitig« am Text arbeiten.

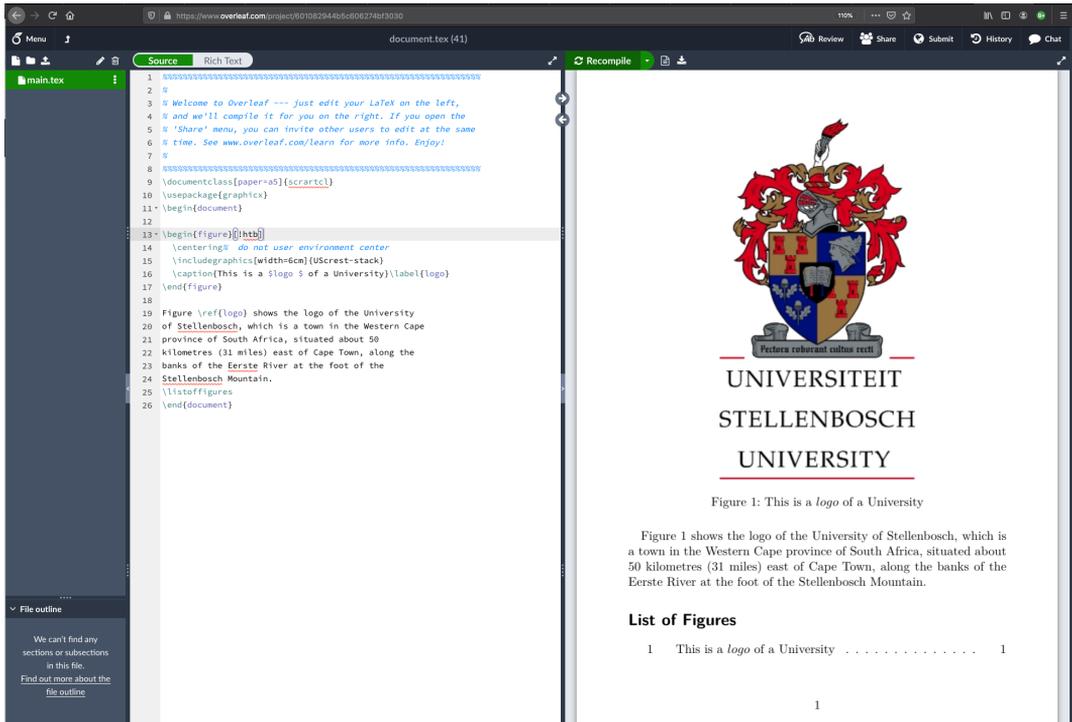


Abbildung 2.9: Editor- (links) und Ausgabefenster (rechts) bei Overleaf.

The screenshot shows the Overleaf interface with a LaTeX document titled '5410302-2.1'. The editor window on the left shows the source code, including a figure environment with a chemical reaction and a table. The output window on the right shows the rendered PDF page with the chemical reaction and table.

Cal, Weining, 5410302 Versuch Nr. 2.1 GP-OC-WS2021

Anstaltsdatum: 12.02.2021 Protokollabgabedatum: 26.02.2021

Protokoll zu Versuch 2.1

Synthese von Benzoesäuremethylester

Name: Cal, Weining Matr.-Nr.: 5410302

Studiengang: Biochemie Labor: 98, Box 1, Platz 11

Betreuerin: Carla Kirschbaum

1 Stöchiometrische Bruttoreaktionsgleichung

Abbildung 1: Synthese von Benzoesäure aus Benzoesäuremethylester.

2 Ansatz

Tabelle 1: Ansatzgrößen zur Darstellung von Benzoesäure aus Benzoesäuremethylester

Name der eingesetzten Stoffe	M [g/Mol]	ρ [g/cm ³]	$\Delta\eta$ [mmol]	n [g]	m [mL]	V [mL]
Benzoesäuremethylester	136.15	1.088	1.0	51.9	7.07	6.5
10% Natronlauge	40.00	1.110	2.8	140	55.5	50
10% Salzsäure	36.36	1.039	2.8	140	52.0	50
Benzoesäure	122.12	1.27	0.67	34.56	4.22	3.32

Abbildung 2.10: Editor- (links) und Ausgabefenster (rechts) für ein Projekt mit mehreren Unterverzeichnissen.

Ein Overleaf-Projekt kann zur Vereinfachung als komprimierte ZIP-Datei sowohl hoch- als auch heruntergeladen werden. Ein Arbeiten mit einer Dropbox ist ebenso möglich wie mit GitHub oder einem lokalen Git, welches beispielsweise durch `git clone https://git.overleaf.com/602bf293859c4e739e1483fb` erstellt werden kann, wobei letzteres die Projektnummer ist. Abbildung 2.11 zeigt den formalen Vorgang zum Anlegen eines Repositoriums für GitHub. Formale Voraussetzung ist ein eigener Account bei GitHub, der für Overleaf freigegeben werden muss.

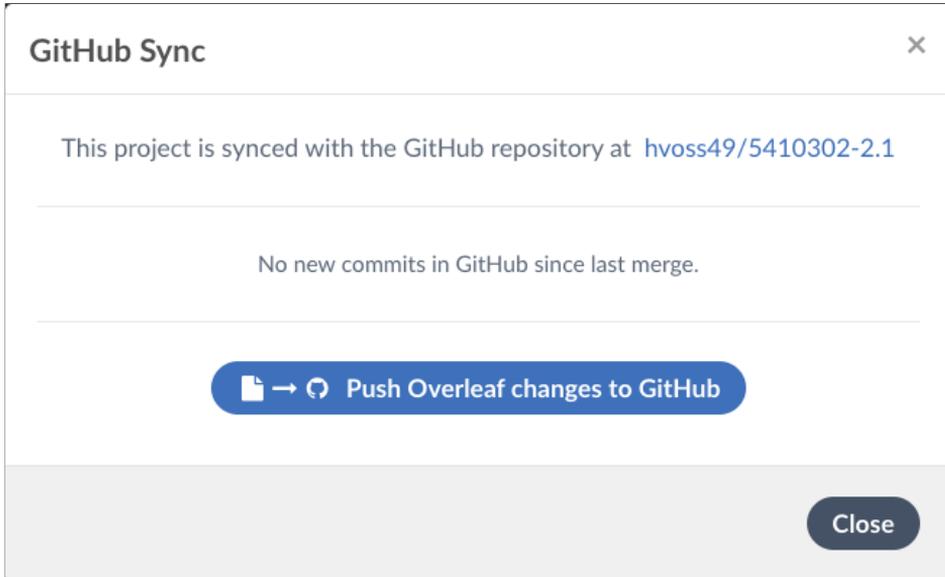


Abbildung 2.11: Vorgang zur Übernahme eines Overleaf-Projekts in ein GitHub-Repositorium.

Bibliografie-Datenbanken von Mendeley oder Zotero können direkt in ein Projekt eingebunden werden; eine eigene Konvertierung in das BIB_{TEX} -Format entfällt dadurch.

Man kann jederzeit anderen Nutzern die Möglichkeit geben, ebenfalls an den Dateien zu arbeiten und entsprechende Korrekturen oder Erweiterungen am Projekt vorzunehmen. Für eine größere Teilnehmerzahl ist allerdings eine kommerzielle Registrierung notwendig, für die eine jährliche Gebühr zu entrichten ist.



Erste Schritte mit L^AT_EX

3.1 Struktur eines L ^A T _E X-Dokuments	23
3.2 Die Dokumentenklasse	25
3.3 Globale Optionen	26
3.4 Der Satzspiegel	26
3.5 Nationale Besonderheiten.	33

3.1 Struktur eines L^AT_EX-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 Papier-
    format und Sprachunterstützung für engl ish und ngerman für die deutsche Sprache
    (»neue« – ab 1996 gültig). Diese Optionen werden sowohl von der Dokumenten-
    klasse als auch allen folgenden Paketen ausgewertet.
\usepackage{Libertinus}
    Benutze Libertinus Serif (mainfont), Libertinus Sans (sansfont) und Libertinus
    Mono (monofont), was in dem Paket festgelegt wird.
\usepackage{babel}
    Lade das Sprachpaket babel, welches die globalen Optionen engl ish und ngerman
    der Dokumentenklasse auswertet. Es werden die Trennmuster für diese bei-
    den Sprachen (Dialekte) geladen und alle sprachspezifischen Dinge direkt nach
    \begin{document} initialisiert, beispielsweise das Datumsformat.
\babelprovide[hyphenrules=ngerman-x-latest]{ngerman}
    Benutze die neuesten (experimentellen) Trennmuster für die deutsche Sprache.
```

`\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.

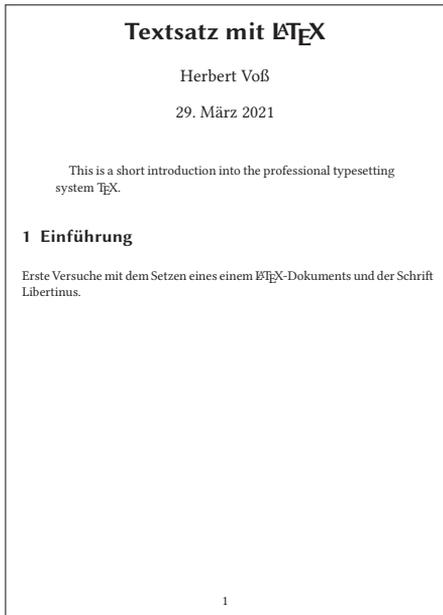
`\begin{abstract}..end{abstract}`
Fremdsprachige Zusammenfassung.

`\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{libertinus}
\usepackage{babel}
\babelprovide[hyphenrules=ngerman-x-latest]{ngerman}
% \babelprovide ist optional. Man bekommt aber
% weniger Trennfehler, weit unter 0.1%
\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 einem  $\LaTeX$ -Dokuments
und der Schrift Libertinus.
\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.

```

...
\documentclass [Optionen] {Name} [Version]
...
\begin{document}
...
\end{document}

```

} Pramabel
 } Textkorper

3.2 Die Dokumentenklasse

Die Dokumentenklasse wird ber den Befehl `\documentclass` ausgewahlt, wobei in der Regel der gewahlten 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 fhrt, wenn die verwendete Klasse alter ist. Je besser eine vorhandene Dokumentenklasse auf die eigenen Bedrfnisse zugeschnitten ist, desto weniger Aufwand wird fr Re- oder Neudefinitionen bestimmter Umgebungen oder Makros betrieben werden mssen. Standardmaig gibt es die vier Dokumentenklassen `book`, `article`, `report` und `letter` mit den folgenden standardmaigen Eigenschaften, wobei hier in Klammern die entsprechenden Namen der von Markus Kohm entwickelten KOMA-Script-Klassen angegeben sind:

book	– Titel auf einer eigenen Seite;
(scrbook)	– Seitenzahlung mit romischen und arabischen Ziffern; – Ebenen <code>\part</code> , <code>\chapter</code> , <code>\section</code> , <code>\subsection</code> , <code>\subsubsection</code> , <code>\paragraph</code> und <code>\subparagraph</code> ; – Kapitelweise Nummerierung der Abbildungen, Tabellen und Gleichungen.
report	– Wie Klasse <code>book</code> , aber nicht mit den Moglichkeiten von <code>\frontmatter</code> ,
(scrreprt)	<code>\mainmatter</code> und <code>\backmatter</code> .
article	– Titel auf keiner eigenen Seite;
(scrartcl)	– Seitenzahlung mit arabischen Ziffern; – Ebenen <code>\part</code> (<i>nicht</i> auf einer eigen Seite), <code>\section</code> , <code>\subsection</code> , <code>\subsubsection</code> , <code>\paragraph</code> und <code>\subparagraph</code> ; – Fortlaufende Nummerierung der Abbildungen, Tabellen und Gleichungen.
letter	– Keine Ebenen;
(scrletter2)	– Elemente, die ein Brief aufweist, wie Absender, Anschrift, usw.

In Vo (2019) findet sich eine Liste der mehr als 150 derzeit bekannten Dokumentenklassen, die mit jeder standardmaigen TX-Installation zur Verfgung stehen. Jede

Klasse hat ihre eigenen Optionen, die zur Anpassung der Dokumentenklasse an die persönlichen Vorgaben benutzt werden können. Für wissenschaftliche Arbeiten kann man außer der Briefklasse grundsätzlich jede der angegebenen Dokumentenklassen nehmen. Eine sinnvolle Empfehlung gibt Tabelle 3.1, wobei die Anzahl der Seiten nicht als Kriterium erscheint; es spielt inhaltlich keine Rolle, ob man eine Seite oder über Tausend Seiten erstellt. Lediglich die Zeit des Programmlaufs (Übersetzens) ist von der Seitenzahl abhängig: Längeres Dokument \triangleright längere Laufzeit.

Tabelle 3.1: Empfehlung für die Anwendung der verschiedenen KOMA-Script-Klassen.

scrbook	Bücher, längere Forschungsberichte, Dissertationen, längere Masterarbeiten
scrreprt	Forschungsberichte, Tagungsbände, Dissertationen, Masterarbeiten, längere Bachelorarbeiten
scrartcl	kürzere Masterarbeiten, Bachelorarbeiten, Seminararbeiten, Literaturlisten

3.3 Globale Optionen

Alle Dokumentenklassen erlauben die Angabe von *globalen* optionalen Argumenten (siehe Definition auf der vorherigen Seite), wobei die Angabe eines Mindestdatums eher selten auftritt. Der Dokumentenklasse können auch Optionen übergeben werden, die sie selber gar nicht auswertet, beispielsweise

```
\documentclass[ngerman,paper=a4]{scrbook}
```

Die Option `ngerman` hat für `scrbook` keine Bedeutung, `paper=a4` ist dagegen eine spezielle Option für die Papiergröße, die von KOMA-Script beziehungsweise dem Paket `typearea` ausgewertet wird, aber von anderen Paketen einfach ignoriert wird. Auch wenn die Sprache nicht von der Dokumentenklasse beachtet wird, sollte man sie hier angeben; es gibt mehrere Pakete, die `ngerman` auswerten. Man spart sich dann das wiederholte Angeben dieser Option bei den betreffenden Paketen, beispielsweise `ngerman` für die Pakete `babel` und `varioref`.

3.4 Der Satzspiegel

Abbildung 3.1 auf der nächsten Seite zeigt die traditionelle Konstruktion eines Satzspiegels. Zu diesem gehört alles, was mit Text gefüllt werden kann. Nicht dazu gehört eine einzelne Seitenzahl (Pagina), die auch als toter Kolumnentitel bezeichnet wird. Ein lebender Kolumnentitel, der zum Satzspiegel gezählt wird, ist im Allgemeinen die Kopfzeile, die durchaus eine Seitenzahl enthalten kann. Soll das Dokument nach der Fertigstellung gedruckt werden, so wird oftmals ein sogenannter Beschnitt erwartet, der mindestens 3 mm–5 mm betragen sollte.

Eine Bindekorrektur wird immer dann vorgesehen, wenn eine Arbeit geheftet wird, oder einen Umfang im Bereich von 1000 Seiten und mehr aufweist. In dem Fall kann ein derartiges Werk nicht ohne weiteres vollständig aufgeklappt werden. Abbildung 3.2 auf der nächsten Seite zeigt die Auswirkung einer Bindekorrektur auf das Seitenlayout.

03-04-1

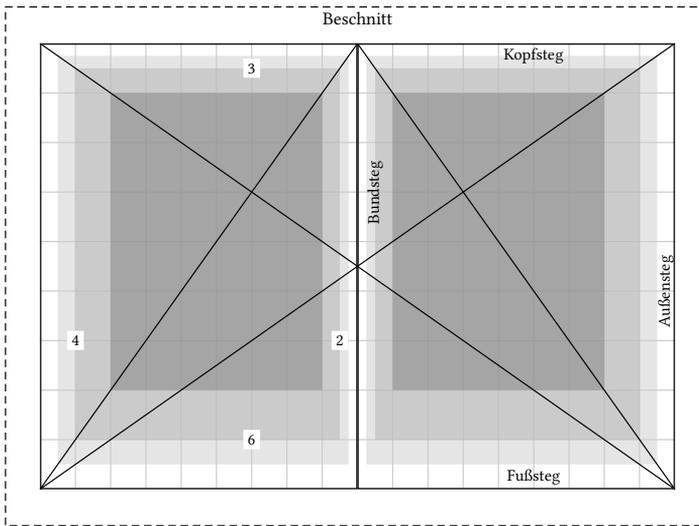


Abbildung 3.1: Der Satzspiegel: Anordnung einer Doppelseite für DIV=9, 18 und 36 ohne Berücksichtigung einer eventuellen Bindekorrektur. Die jeweiligen Ränder verhalten sich wie 2:3:4:6.

Bei einem zweiseitigen Layout bezieht sich die Angabe zu BCOR auf beide Seiten, im Gegensatz zum einseitigen Layout (`paper=a4`, `paper=landscape`). KOMA-Script unterstützt die Papierformate `letter`, `legal` und `executive`, sowie die europäischen ISO-Formate der Reihen A, B, C und D, die alle über die Option `paper` ausgewählt werden können, beispielsweise `paper=a5`. Soll das gesamte Dokument im Querformat gesetzt werden, so ist dies ebenfalls der Option `paper` zuzuweisen.

Papierformat

- a) Linke Seite für twoside b) Rechte Seite für twoside oder jede Seite für oneside

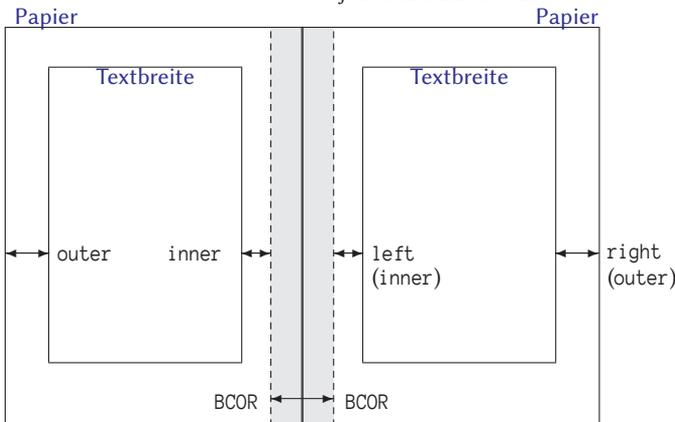


Abbildung 3.2: Die Ränder und die Bindekorrektur (BCOR) für ein- und zweiseitige Dokumente.

Es ist nicht empfehlenswert die Ränder über die internen Längenmakros zu setzen. In dem Fall ist zu beachten, dass für \TeX der interne Koordinatenursprung bei 1 Inch vom linken und oberen Rand liegt (Beispiel 03-04-3 auf Seite 29). Bei einem linken Rand von beispielsweise 0,5 Inch müsste dann `\leftsidemargin` der Wert von `-0,5` Inch zugewiesen werden (bezogen auf den Koordinatenursprung). Dieses etwas irritierende Verhalten kann vermieden werden, wenn man den Satzspiegel über eines der beiden im Folgenden beschriebenen Pakete setzt.



3.4.1 typearea

Insbesondere für die traditionellen Seitenmaße bietet sich das von KOMA-Script *standardmäßig* geladene Paket `typearea` an. Dieses wertet die Angaben zum Papierformat aus und stellt alle entsprechenden Größen ein.

```
\documentclass[paper=a4,paper=landscape,DIV=13,BCOR=10mm]{scrreprt}
```

Tabelle 3.2 zeigt die wesentlichen Optionen für die Einstellung des Satzspiegels. Sie können alle der Dokumentenklasse übergeben werden. Ist man relativ frei in der Wahl seines Satzspiegels, so kann mit der Wahl des Papierformats, des DIV-Parameters und einer eventuellen Bindekorrektur die Seite so eingestellt werden, dass sie allgemein gültigen Vorgaben folgt.

Tabelle 3.2: Zusammenstellung der Optionen für den Satzspiegel.

<i>Name</i>	<i>Werte</i>	<i>Erklärung</i>
BCOR	Länge	Angabe der Bindekorrektur.
DIV	<Zahl>	Ganzzahliges Teilungsverhältnis einer Seite. Je größer die Zahl, desto kleiner die Ränder (vergleiche Abbildung 3.1 auf der vorherigen Seite).
fontsize	Länge	Angabe der Grundschriftgröße durch Maßzahl und Einheit, normalerweise in pt (siehe auch Abschnitt A.2 auf Seite 395).
footheight	<Länge>	Höhe des Seitenfußes, wird nur ausgewertet, wenn das Paket <code>scrlayer-scrpage</code> geladen wird.
footinclude	<Ein-Aus>	Fußzeile soll Teil des Satzspiegels sein.
footlines	<Zahl>	Anzahl der Zeilen für den Fuß.
footsepline	<Ein-Aus>	Linie unter dem Fuß.
headheight	<Länge>	Höhe des Seitenkopfes.
headinclude	<Ein-Aus>	Kopfzeile soll Teil des Satzspiegels sein.
headsepline	<Ein-Aus>	Zieht eine Linie unter die Kopfzeile (nur für den Seitenstil »scrheadings«).
mpinclude		Den Rand (<code>marginpar</code>) zum Textkörper zählen.
paper	<Format>	Wähle Papierformat.
	landscape	Wähle vorhandenes Papierformat im Querformat.
	portrait	Wähle vorhandenes Papierformat im Hochformat (Vorgabe).
	<Breite>:<Höhe>	Wähle vorgegebenes Papierformat.
plainfootbotline	<Ein-Aus>	Wie <code>footbotline</code> , nur für den Seitenstil <code>plain</code> .
plainfootsepline	<Ein-Aus>	Wie <code>footsepline</code> , nur für den Seitenstil <code>plain</code> .
plainheadsepline	<Ein-Aus>	Wie <code>headsepline</code> , nur für den Seitenstil <code>plain</code> .
plainheadtopline	<Ein-Aus>	Wie <code>headtopline</code> , nur für den Seitenstil <code>plain</code> .
twocolumn	<Ein-Aus>	Zweispaltenmodus verwenden.
twoside	<Ein-Aus>	Zweiseitenmodus verwenden.
twoside	semi	Im Zweiseitenmodus den Satzspiegel wie im Einseitenmodus setzen (gleiche Ränder).
usegeometry	<Ein-Aus>	Alle notwendigen Angaben an das Paket <code>geometry</code> weiterreichen, wenn man das Seitenlayout nicht über <code>typearea</code> , sondern <code>geometry</code> festlegen möchte.

Das Papierformat an sich kann bei typearea frei gewählt werden, sodass auch eine Karteikartengröße problemlos festgelegt werden kann:

03-04-2

```
\documentclass[parskip=half-,headline=false,footinclude=false,mpinclude=false,
  paper=landscape,paper=5cm:3cm,ngerman]{scrartcl}
\areaset{4cm}{2.4cm}

\subsection*{Definierte Papierformate}
\ttfamily letter, legal, executive, a0, a1, \ldots, b0, b1, \ldots,
  c0, c1, \ldots, d0, d1, \ldots\par
\rule{\linewidth}{0.4pt}
```

Definierte Papierformate

letter, legal,
executive, a0, a1,

..., b0, b1, ..., c0,
c1, ..., d0, d1, ...

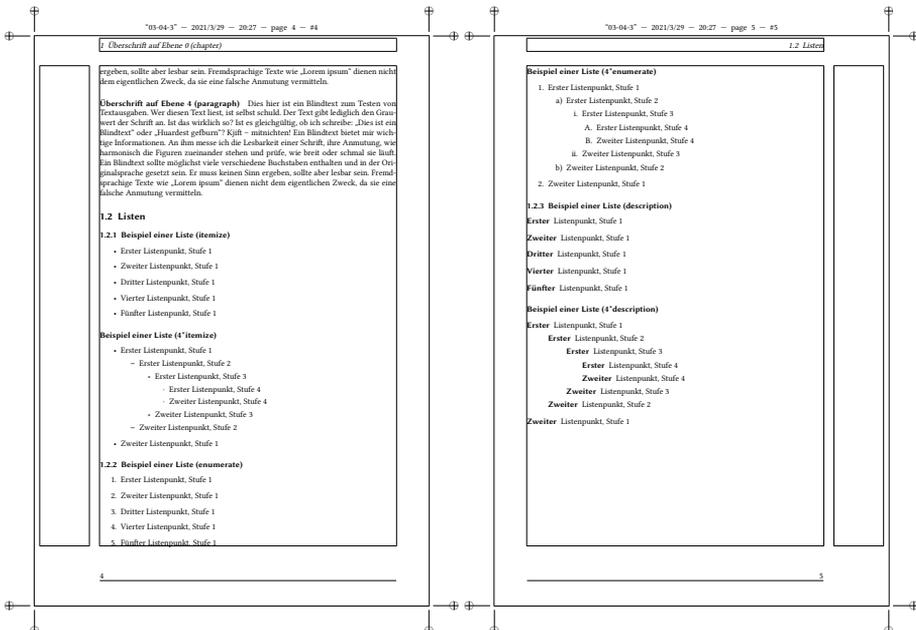
Individuelle Einstellungen der einzelnen Ränder sind mit dem Paket typearea nicht möglich. Möchte man diese vornehmen, so ist eine Verwendung des Paketes geometry anzuraten. Den KOMA-Script-Klassen wird dazu die Option usegeometry mitgegeben.

Für ein Manuskript, welches veröffentlicht und dazu gedruckt werden soll, kann mithilfe des Paketes crop für Schnittmarken gesorgt werden.

03-04-3

```
\documentclass[paper=16.5cm:24cm,ngerman]{scrbook}
\areaset{124mm}{202mm}
\usepackage[cam,a4,center]{crop}
\usepackage{showframe,blindtext}

\tableofcontents\blinddocument
```

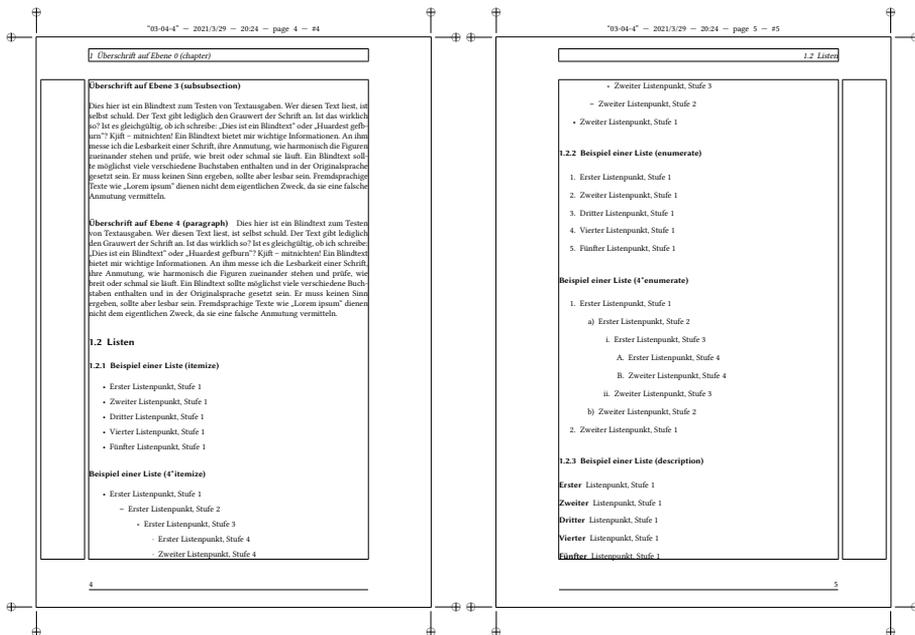


3.4.2 geometry

Eine Alternative zum Paket `typearea` ist das Paket `geometry`, welches auf einfache Weise auch willkürlich festgelegte Satzspiegel zulässt. (Umeki 2020) Um eine korrekte Übergabe der notwendigen Angaben von KOMA-Script zu `geometry` zu gewährleisten, sollte unbedingt die Option `usegeometry` benutzt werden. Das folgende Beispiel definiert den gleichen *Textbereich* wie Beispiel 03-04-3 auf der vorherigen Seite, welches für das Setzen des Satzspiegels das KOMA-Script-Paket `typearea` benutzt. Man erkennt deutlich, dass es in der horizontalen Anordnung Unterschiede gibt. Bei gleich großem Textkörper bevorzugt KOMA-Script das klassische Layout, wohingegen das Paket `geometry` durch die Vorgabe des rechten Randes mit `rmargin` den linken Rand einfach durch die anderen Werte bestimmt. Dies ist hier notwendig, da die Nummern der Beispiele in den inneren Rand gesetzt werden und dafür genügend Platz benötigen.

```
\documentclass[usegeometry,parskip=half-,twoside,ngerman]{scrbook}
\usepackage{geometry}
\geometry{paperheight=240mm,paperwidth=165mm,tmargin=5mm,textwidth=124mm,textheight=202mm,
rmargin=22mm,heightrounded,includeheadfoot,headheight=5mm,headsep=8mm,foot=13mm,
marginparsep=2mm,marginparwidth=18mm}
\usepackage[cam,a4,center]{crop}
\usepackage{showframe,blindtext}
\tableofcontents\blinddocument
```

03-04-4

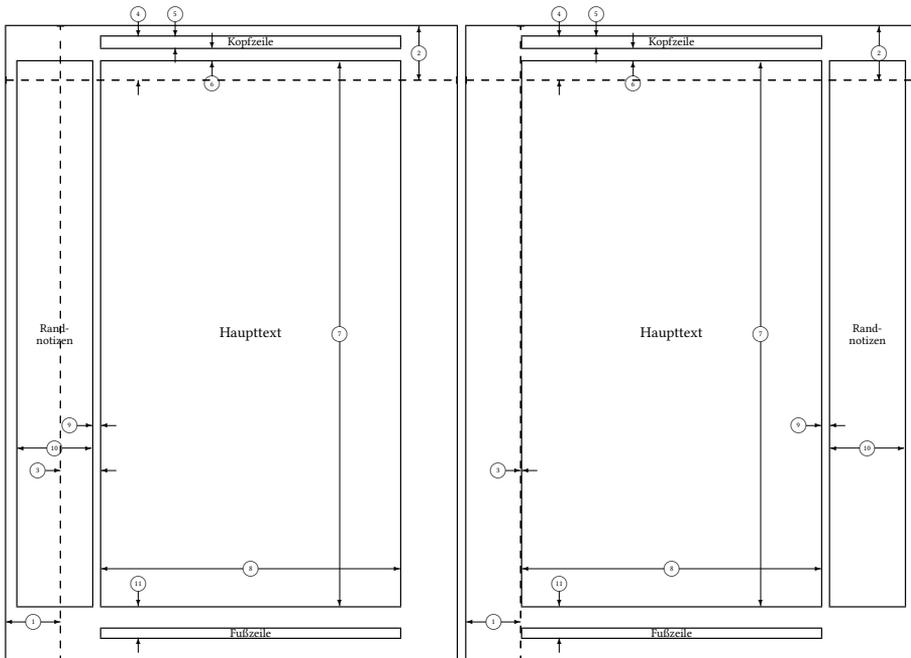


Das Paket erlaubt eine große Anzahl an möglichen Einstellungen, die nicht immer alle von Interesse sind. Insbesondere dann, wenn man frei von Vorgaben ist, reicht es, das Papierformat und die *absoluten* Ränder vorzugeben. Dazu muss die Option `includeheadfoot` benutzt werden. Anderenfalls beziehen sich die Angaben für `tmargin` und `bmargin` nur auf den Textbereich ohne Kopf- und Fußzeile. Wird kein Platz für Randbemerkungen benötigt und somit der Befehl `\marginpar` nicht benutzt, braucht auf die falschen Wert von `\marginparwidth` keine Rücksicht genommen zu werden.

Möchte man in seiner Arbeit mit Randbemerkungen arbeiten, so sollte dies bereits in den Layoutvorgaben festgelegt werden: Das folgende Beispiel zeigt zwei gegenüberliegende Seiten (gerade – ungerade).

03-04-5

```
\documentclass[paper=a4,usegeometry,parskip=half-,ngerman]{scrbook}
\usepackage{geometry}
\geometry{textwidth=14.5cm,tmargin=5mm,rmargin=5mm,bmargin=10mm,includeheadfoot,
marginpar=35mm,includemp}
\usepackage{layout-hv}
\null\newpage\layout
```



3.4.3 Beispiele für Klassenoptionen

Die folgende Zusammenstellung ist nicht vollständig, da es erheblich mehr Möglichkeiten gibt, die aber erst in den folgenden Abschnitten behandelt werden.

```
\documentclass[paper=a4,BCOR=8.25mm,twoside,parskip=half-,ngerman]{scrreprt}
\documentclass[oneside,DIV=15,fontsize=13pt,english]{sccartcl}
\documentclass[BCOR12mm,DIV=calc,twoside=on,french]{scrbook}
\documentclass[fontsize=12pt,BCOR=7mm,DIV=calc,headings=small,headsepline=on,
english,ngerman,french]{scrreprt}
\documentclass[a4paper,fontsize=8pt,headings=small,twocolumn=on,ngerman]{sccartcl}
\documentclass[fontsize=11pt,paper=a4,fleqn,listof=totoc,english]{scrbook}
\documentclass[paper=a4,fontsize=10pt,DIV=19,twoside=on,twocolumn=on,open=any,
listof=totoc,parskip=false,titlepage=on,headings=small,BCOR=12mm,spanish]{scrbook}
```

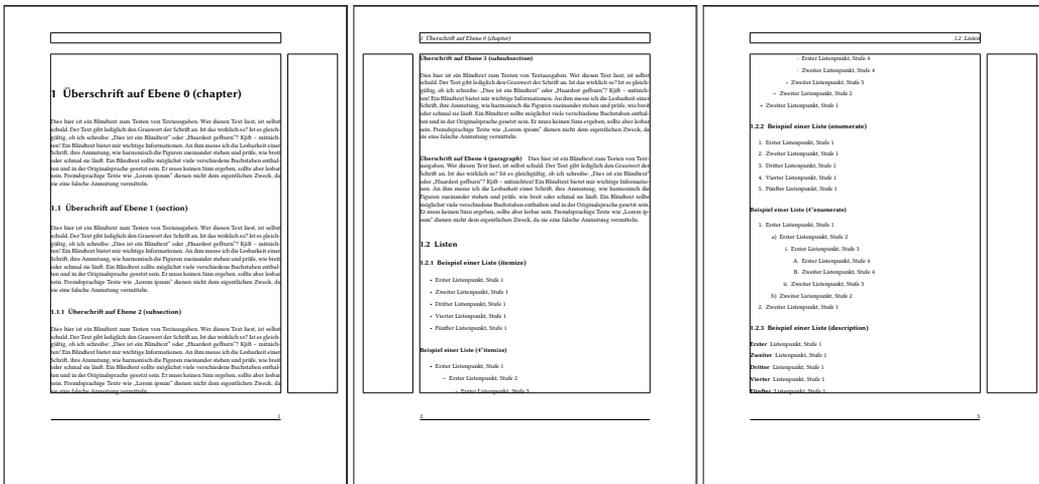
Die ersten beiden sollen zum besseren Vergleich noch einmal mit ihren ersten Seiten dargestellt werden. Mit dem Paket `showframe` oder der gleichnamigen Option des Pake-

tes geometry, lassen sich die Ränder anzeigen und blindtext liefert den notwendigen Demonstrationstext.

```

\documentclass[paper=a4, BCOR=8.25mm, twoside, parskip=half-, ngerman]{scrreprt}
\usepackage{libertinus}
\usepackage{babel}
\usepackage{showframe, blindtext}
\usepackage{automark}{sclayer-scrpage}
\blinddocument

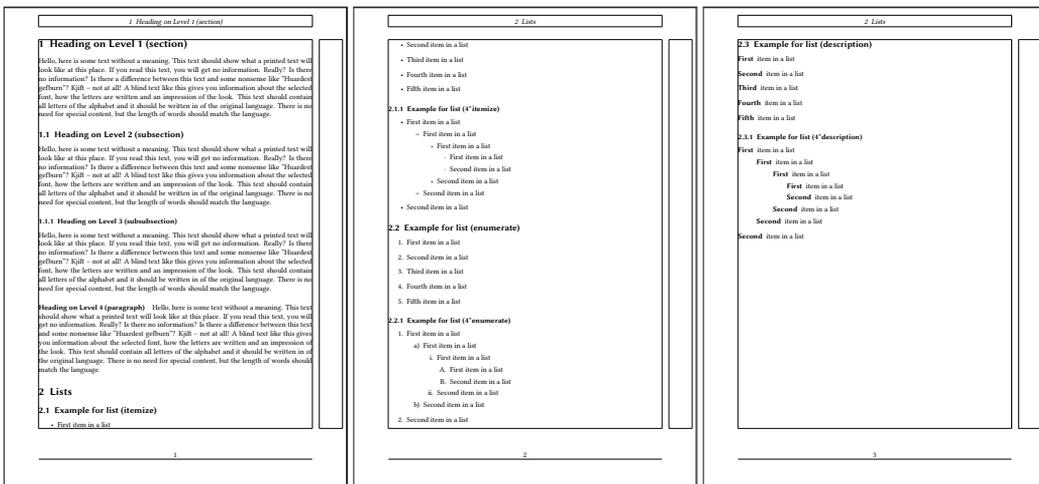
```



```

\documentclass[oneside, DIV=15, fontsize=13pt, english]{scrartcl}
\usepackage{libertinus}
\usepackage{babel}
\usepackage{showframe, blindtext}
\usepackage{automark}{sclayer-scrpage}
\blinddocument

```



Im ersten Beispiel liegt ein zweiseitiges Dokument mit der standardmäßigen Schriftgröße von 11 pt vor. Im zweiten Fall wird diese auf 13 pt gesetzt, auf die Bindekorrektur verzichtet, eine 15-er Teilung verwendet und zusätzlich zweiseitig abgewählt, was einem einseitigen Dokument entspricht.

3.4.4 Lokale Optionen

Beim Laden von Paketen können ebenfalls optionale Argumente angegeben werden, die dann nur lokal für das Paket gelten.

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

Die Angabe der Version wird eher selten verwendet, optionale Argumente sind dagegen oft die Regel. Da ein Paket immer nur einmal geladen werden kann, gibt es immer dann ein Problem, wenn das Paket ein zweites Mal mit *unterschiedlichen* optionalen Argumenten geladen werden soll. Es gibt dann die Fehlermeldung »Option clash«. Im Abschnitt A.6.1 auf Seite 408 findet man dazu eine Lösung.

Fehler
option clash

Pakete laden

Mit dem *ersten* `\usepackage` oder dem internen `\RequirePackage` wird das angegebene Paket geladen und als solches markiert, sodass ein späterer Versuch es zu laden, wirkungslos bleibt. Der dabei mögliche *Option Clash* ist ein Hinweis für den Anwender, dass ein Paket mit zwei unterschiedlichen Optionen geladen werden sollte.

3.4.5 KOMA-Script

KOMA-Script erlaubt das Setzen von Optionen auch noch nach dem Laden der Klasse oder von Paketen, welche sich direkt auf KOMA-Script beziehen.

```
\KOMAOptions{Optionenliste}  
\KOMAoption{Name}Werteliste
```

3.5 Nationale Besonderheiten

Trotz allgemeiner Globalisierung gibt es noch immer zahlreiche Besonderheiten in den jeweiligen Schriftsprachen, beispielsweise Doppelakzente im Vietnamesischen: Hàñ Thệ Thành (Entwickler des PDF-Ports für \TeX), die man auf keiner europäischen Tastatur findet. Allerdings sind die Zeiten längst vorbei, in denen deutsche Sonderzeichen in der \TeX -Notation `\"a` für den Umlaut »ä« oder `\ss` für das »ß« oder alternativ bei Verwendung des Paketes `babel` in der Form "a und "s eingegeben wurden. Man gibt heute die Zeichen so ein, wie man sie auf der Tastatur findet. Es ist letztlich nur ein Problem des Tastatur- und Bildschirmtreibers, ob man alle Zeichen auf den Bildschirm zur Anzeige bekommt. Alle neueren Betriebssysteme verwenden schon seit einigen Jahren UTF-8 als Standardkodierung. Gleiches macht $\text{Lua}\TeX$, sodass im Allgemeinen keine weiteren Maßnahmen notwendig sind, um alle speziellen Zeichen nutzen zu können. Auch sollte es heutzutage kein Problem mehr darstellen, wenn Dokumente zwischen verschiedenen Betriebssystemen ausgetauscht werden, die UTF-8-kodiert sind.

UTF-8

Ältere Dokumente kann man mit frei verfügbaren Programmen in das Kodierungsformat UTF-8 überführen. Beispielsweise `iconv`¹ gleichermaßen für Linux, macOS und Win-

¹<https://de.wikipedia.org/wiki/Iconv>

dows.² Für eine Konvertierung einer Datei mit der Kodierung Latin1 nach UTF-8, wobei Ein.tex und Aus.tex durch die richtigen Dateinamen zu ersetzen sind:

```
iconv -f LATIN1 -t UTF-8 Ein.tex > Aus.tex
```

Eine Alternative wäre das Programm recode, welches direkt die aktuelle Datei wandelt. Eine Umwandlung von der älteren Mac-Kodierung nach UTF-8 erfolgt mit:

```
recode --diacritics --touch --verbose MacCentralEurope..UTF-8 Datei.tex
```

Editor-
schriften

Grundsätzlich gilt, dass die im Editor verwendeten Schriften nichts mit denen zu tun haben müssen, die im L^AT_EX-Dokument zur Anwendung kommen. Allgemein könnte man im Editor und für das zu erstellende T_EX-Dokument dieselben Schriften verwenden. Dies wird selten gemacht; im Editor verwendet man häufig eine Schreibmaschinenschrift, um Einrückungen bei L^AT_EX-Codesequenzen für eine eventuelle Fehlersuche besser anordnen zu können. Zum einen hält sich die Zahl der vorhandenen Schreibmaschinenschriften in Grenzen und zum anderen haben die wenigsten davon eine Unterstützung für nicht-lateinische Buchstaben, beispielsweise Arabisch. Auch die hier verwendete Schreibmaschinenschrift, die Lucida Sans Typewriter, hat keine arabischen Zeichen im Gegensatz zur DejaVu Sans Mono, die auch im Editor verwendet wird (Abbildung 3.3).

T_EXstudio erkennt automatisch, dass eine sogenannte RTL-Schrift (Right-To-Left) verwendet wird und schaltet automatisch in der Ansicht auf die richtige Schreibrichtung.

```

17 \begin{document}
18
19 English text ``\textarabic{جملة عربية وسط جملة إنجليزية}`` with Arabic in
20 between. English paragraph left aligned English paragraph left aligned English
21
22 \begin{Arabic}
23 فقرة عربية محاذاة إلى اليمين فقرة عربية محاذاة إلى اليمين فقرة عربية محاذاة إلى اليمين
24 محاذاة إلى اليمين فقرة عربية محاذاة إلى اليمين فقرة عربية محاذاة إلى اليمين
25 فقرة عربية المحاذاة إلى اليمين فقرة عربية محاذاة إلى اليمين فقرة عربية محاذاة إلى
26 اليمين فقرة عربية محاذاة إلى اليمين فقرة عربية محاذاة إلى اليمين
27 {حاشية عربية}.afootnote\اليمين
28 \end{Arabic}
29
30 \end{document}

```

Abbildung 3.3: Ansicht des Editors T_EXstudio bei der Erstellung eines mehrsprachigen Textes.

Abbildung 3.3 ist ein Screenshot des Editors. Gibt man den Quellcode der Umgebung Arabic mit der Lucida Sans Typewriter aus, so fehlen viele Zeichen:

```

Lucida Sans Typewriter:
\begin{Arabic} ٠٠٠٠ ٠٠٠٠٠ ٠٠٠٠٠٠ ٠٠٠ ٠٠٠٠٠٠ ٠٠٠٠ ٠٠٠٠٠٠ ٠٠٠٠٠٠٠ ٠٠٠ ٠٠٠٠٠٠٠ \n{XXXXXX}
٠٠٠٠ ٠٠٠٠٠ ٠٠٠٠٠٠ ٠٠٠ ٠٠٠٠٠٠٠ ٠٠٠٠ ٠٠٠٠٠٠ ٠٠٠٠٠٠٠ ٠٠٠ ٠٠٠٠٠٠٠٠ ٠٠٠٠ ٠٠٠٠٠٠٠ ٠٠٠٠٠٠٠٠ ٠٠٠
٠٠٠٠٠٠٠٠ ٠٠٠٠ ٠٠٠٠٠٠٠ ٠٠٠٠٠٠٠٠ ٠٠٠٠ ٠٠٠٠٠٠٠ ٠٠٠٠٠٠٠٠ ٠٠٠ ٠٠٠٠٠٠٠٠٠ ٠٠٠٠ ٠٠٠٠٠٠٠ ٠٠٠٠٠٠٠٠٠ ٠٠٠ ٠٠٠٠٠٠٠٠٠
٠٠٠٠٠ ٠٠٠٠٠٠٠٠ ٠٠٠٠ ٠٠٠٠٠٠٠٠٠ ٠٠٠٠ ٠٠٠٠٠٠٠٠ ٠٠٠٠٠٠٠٠٠ ٠٠٠ ٠٠٠٠٠٠٠٠٠٠ ٠٠٠٠ ٠٠٠٠٠٠٠٠ ٠٠٠٠٠٠٠٠٠٠ ٠٠٠
٠٠٠٠٠٠٠٠٠. \end{Arabic}

```

Dagegen erscheinen fast alle Zeichen bei Verwendung der DejaVu Sans Mono:

²<http://www.etissimo.com/umwandlung-von-zeichenkodierungen-mittels-iconv/>

