

Eine kurze Einführung in das Textsatzsystem

Texte schreiben mit LaTeX

Alexander Florin

12. September 2013

Inhaltsverzeichnis

1	Willkommen beim Textsatz	2
1.1	Wofür ist LaTeX geeignet?	2
1.2	Was bedeutet „professioneller Textsatz“?	3
1.3	Wie funktioniert LaTeX?	4
1.3.1	Trennung des Inhalts und der Struktur von der Form	4
1.3.2	Kontrolle über alles dank Mark-up	5
1.3.3	Strafen für unschönes Aussehen	6
1.4	Woher bekomme ich LaTeX?	6
1.5	Wie fange ich an?	7
2	Text schreiben mit Editor und Mark-up	8
2.1	Jetzt gehts los	9
2.2	Formate	10
2.3	Zusatztexte	11
2.4	Tabellen	12
2.5	Bilder einfügen	14
2.6	Tipps beim Setzen	15
3	Text schreiben mit Lyx	15
4	Praktische Tipps	17
4.1	Layout, allgemein	17
4.2	Zeilentrennungen	17
4.3	Seitenzahlen	17
4.4	Nachschlagequellen	17
5	Abbildungsverzeichnis	18
6	Tabellenverzeichnis	19

1 Willkommen beim Textsatz

Zum Texteschreiben braucht man nicht unbedingt Word von Microsoft, es gibt je nach Anwendungsfall wesentlich geeignetere Werkzeuge. Für längere Texte, beispielsweise Hausarbeiten in Schule oder Studium, bietet sich das Textsystem \LaTeX in der Kombination mit dem Editor Lyx an. Beide sind kostenlos und liefern eine professionell und gut aussehende PDF zum Ausdrucken oder Digital-Einreichen. Dieser Beitrag stellt die ersten Schritte vor, damit man versteht, worauf man sich mit Latex einlässt und möglichst schnell ein Erfolgserlebnis vorweisen kann.

LaTeX wurde Anfang der 1980 entwickelt und steht für alle Betriebssysteme kostenlos zur Verfügung. Es wird entweder „LaTeX“ oder „LaTeX“ (mit weichem oder hartem „ch“) gesprochen – nicht wie das Fetischmaterial „Latex“. Für die bessere Lesbarkeit schreibe ich im Fließtext aber „Latex“, es ist ja klar, was gemeint ist.

Latex ist ein Textsatzsystem, es eignet sich also vorwiegend, um lange Texte zu *setzen*, wie die Schriftsetzer in den Druckereien seit dem 15. Jahrhundert. Die schöne und feingliedrige Gestaltung von Druckseiten gelingt zwar auch mit Latex ganz gut, solche Textlayout-Arbeiten sind aber bei Word oder richtigen Layout-Programmen wesentlich besser aufgehoben.

Dass derzeit erst die Latex-Version 2 aktuell ist, sollte nicht über die Reife der Software hinwegtäuschen. Das System wird kontinuierlich weiterentwickelt, und all die zahlreichen Pakete, aus denen das Gesamtsystem besteht, erhalten regelmäßig Verbesserungen und Aktualisierungen. Da man Latex nicht als Einzelelement, sondern eher in seiner Gesamtheit als „TexLive“-System einsetzt, hat die Versionsnummer an Bedeutung verloren. Seit einigen Jahren erscheint etwa Mitte jedes Jahres eine neue TexLive-Version wie „TexLive 2012“, die Mac-Version von TexLive nennt sich parallel dazu „MacTeX 2012“.

1.1 Wofür ist LaTeX geeignet?

Ursprünglich wurde Latex für den Formelsatz entwickelt, und mathematische Formeln sind auch die Stärke des Systems. Aber auch ohne Interesse an aufwändigen Formeln profitiert man von den Fähigkeiten. Mir geht es im Folgenden weniger um den Formelsatz als vielmehr um den Satz von lesbaren Texten und die Grundlagen von Latex. Wer diese gemeistert hat, ist auf den Formelsatz bestens vorbereitet.

Latex eignet sich für Studienarbeiten, Aufsätze und Essays, Romane und Kurzgeschichten, Anleitungen und Dokumentationen, lange und kurze Texte mit Tausenden Seiten – eben alles, was sich drucken lässt, um es danach zu lesen. Latex’ Stärken sind die Automatismen:

- automatisches Inhalts- und andere Verzeichnisse
- gute Platzierung von Bildern/Grafiken im Textfluss
- zuverlässige Fußnotenintegration
- Tabellen über mehrere Seiten
- Indexe, Listen, Verzeichnisse
- Integration einer Bibliografie (mittel BibTeX)
- effektive Gestaltung von mathematischen Formeln
- stabiles Seitenlayout mit automatischen Kopf- und Fußzeilen

Beginnt man eine Latex-Datei, liefert die Standard-Einstellung bereits sehr gute Ergebnisse. Bei Bedarf kann man aber – fast – alles manuell anpassen und umgestalten.

Latex ist besonders für alle geeignet, die auf den Inhalt ihres Textes Wert legen. Denn beim Schreiben sieht man nur seinen Text und dessen Struktur. So arbeitet man fokussiert und kann sich besser auf das Wesentliche (den Inhalt) konzentrieren. Die Form gelingt dank der durchdachten Standards dann von selbst und erfordert kaum Nacharbeit.

Word macht zwar vieles auch im Textsatz recht einfach, ist aber insgesamt gesehen komplexer und aufwändiger zu bedienen, jedenfalls wenn man etwas mehr Ansprüche an seinen Textsatz stellt. Die Formatvorlagen für Absätze, Zeichen und Tabellen zu bändigen, kann da frustrierend sein. Achja, und Word zickt schnell mal rum, wenn man zu viele Fußnoten, Abbildungen oder Automatismen in einem Dokument nutzt. Die Nacharbeiten im zweiten, dritten und vierten Korrekturlauf sind eine Katastrophe, wenn die Form sich partout nicht den eigenen Ansprüchen beugen will. Da landen Fußnoten auf der falschen Seite, Abbildungen verschwinden hinter dem Seitenrand, oder Word stürzt auf der 31. Seite ab.

1.2 Was bedeutet „professioneller Textsatz“?

Es gibt zahlreiche Regeln, die den Profi vom Laien unterscheiden. Mit Latex schwinden die Unterschiede, denn Latex kümmert sich darum, dass alle Texte gut aussehen. Bagetellen wie Mediävalziffern (die mit Unter- und Oberlängen gestaltet sind, was sich harmonischer in den Textfluss integriert) sind da nur kleine Sahnehäubchen:

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum 1234567890 dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Abbildung 1: Mediävalziffern fügen sich mit ihrer variablen Zeilenorientierung harmonisch ins Schriftbild.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum 1234567890 dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Abbildung 2: Standardziffern wirken im Fließtext immer etwas aufdringlich, etwa wie Wörter in Großbuchstaben.

Über die Jahrhunderte haben sich Standards für guten Textsatz etabliert, an die sich Latex hält und beispielsweise die Abstände zwischen einzelnen Textelementen optimal

gestaltet – jedenfalls solange man sie nicht manuell undefiniert. Aber warum sollte man das tun, wenn alles schon im Normalzustand nach etablierten Standards eingestellt ist? Mit den Anpassungsmöglichkeiten ist gewährleistet, dass die Dokumente je nach Stil-Vorgaben gesetzt werden, beispielsweise in Bezug auf Schriftgröße, Zeilenabstand, Seitenränder, Kopf- und Fußzeile.

Ein Beispiel für guten Textsatz sind die Einzüge bei Absätzen. Die erste Zeile jedes Absatzes beginnt nicht ganz links, sondern leicht nach rechts versetzt. So sind die Absatzanfänge auf einer Seite immer schnell zu finden. Allerdings beginnt der erste Absatz nach einer Überschrift ohne Einzug ganz links. Bei LaTeX ist das automatisch so, daran muss man nicht denken.

Ausgeglichener Textsatz bedeutet, dass eine bedruckte Seite aus der Ferne betrachtet eine einheitliche Graufäche bildet, da gibt es keine Zeilen mit ganz schmalen und andere mit ganz breiten Leerzeichen, sondern die Leerzeichen sind alle ungefähr gleich breit – keine triviale Aufgabe.

Latex kommt zwar nur mit seinen eigenen Schriftsätzen (oder speziell aufbereiteten) gut zurecht, diese sind aber sehr sauber gearbeitet. Word dagegen errechnet aus den Grundformen der Schrift die Fett- und Kursiv-Variante – dieser Automatismus treibt Typografen die Zornesröte ins Gesicht, denn Automatismen gehen viel zu grobschlächtig und universell vor.



Abbildung 3: Die Einzüge bei Absätzen gelingen automatisch.

1.3 Wie funktioniert LaTeX?

Am besten erkennt man die Unterschiede im Vergleich zu anderen Textverarbeitungen. Word beispielsweise muss direkt beim Schreiben berechnen, wie die fertige Seite aussieht (siehe Kapitel 9 „Der Apple Faktor“). Latex dagegen nimmt sich Zeit, um zu berechnen, wie die fertigen Seiten aussehen. Word organisiert den Zeilenumbruch nur nach der aktuellen Zeile: Passt der Text drauf oder muss getrennt bzw. auf die nächste Zeile umgebrochen werden? Latex dagegen berechnet für den gesamten Absatz das optimale Aussehen: so wenig Trennungen wie möglich, eine harmonische Verteilung der Leerzeichen, Integration von anderen Elementen etc.

1.3.1 Trennung des Inhalts und der Struktur von der Form

Das Arbeitsschema für Word sieht in etwa so aus:

- Text schreiben (Inhalt = Autor)
- Struktur vergeben (Autor)
- Darstellung auf dem Monitor des fertigen Druckbildes (Form = Berechnung)

Alles verläuft parallel – so wie der Autor sich quasi gleichzeitig für Inhalt, Struktur und Form interessieren muss, so kümmert sich Word auch gleichzeitig um alles bzw. weil

Word alles gleichzeitig anbietet, verleitet es den Autor dazu, sich um mehr zu kümmern als nur um seine Inhalte.

Bei Latex sind die Arbeitsschritte stärker getrennt:

1. Text schreiben (Inhalt = Autor)
2. Struktur zuweisen (halb-parallel zum Schreiben = Autor)
3. Errechnung des Druckbildes (Form = Berechnung)

Da die Form vom Inhalt und von der Strukturierung strikt getrennt berechnet wird, kann die Form auch sehr viel einfacher angepasst werden. So ist es ohne Aufwand möglich, eine Din-A4-Druckversion eines Textes zu erstellen und parallel dazu eine im Din-A5-Querformat, die man im Web veröffentlichen möchte und die deshalb möglichst viele der Elemente auch verlinkt (Inhaltsverzeichnis, Index-Einträge etc.) und die Bilder und Tabellen in anderer Weise darstellt. In Word wäre eine solche parallele Ausgabe des selben Ursprungstextes in unterschiedliche Zielformen nur mit viel manueller Nacharbeit zu bewerkstelligen. Zugegeben, das skizzierte Szenario ist eher selten.

Aber ein Gedanke ist enorm wichtig: Inhalt und Struktur gehören zusammen, und die Form wird dann erst später dazu errechnet. Beim Schreiben einer wissenschaftlichen Hausarbeit ist es beispielsweise unerheblich, ob eine Kapitelüberschrift nun in 14, 16 oder 17 Punkt Größe ausgegeben wird, ob sie fett ist oder kursiv. Beim Schreiben ist nur wichtig, dass die strukturelle Information da ist: Hier steht eine Kapitelüberschrift mit folgendem Text. Erst wenn die Texterstellung im Wesentlichen abgeschlossen ist, braucht man sich Gedanken darüber zu machen, wie die Kapitelüberschrift konkret aussehen soll. Dabei nutzt man dann entweder die Standard-Version oder passt diese den eigenen Vorstellungen an – genauso wie man bei Word die Formatvorlagen bearbeitet (und sowieso immer mit diesen arbeiten sollte!). Realistischerweise wird man allerdings bereits eine Denkpause zwischendurch dazu nutzen, die Formate schon mal anzuschauen und ggf. anzupassen.

Statt des behaupteten Wysiwyg („What you see is what you get“) von Word arbeitet man bei Latex mit einer Struktur, die eher Wysiwym („What you see is what you mean“) entspricht.

1.3.2 Kontrolle über alles dank Mark-up

Wer sich mit dem Internet auskennt, weiß, dass Internetseiten nur aus HTML- und CSS-Dateien bestehen – erst der Browser berechnet aus diesen dann die Seite, die dargestellt wird. Genauso berechnet Latex aus den .tex-Dateien dann die PDF, die gedruckt werden kann. Womit die HTML-Dateien erstellt werden, ist dem Browser egal, Hauptsache die Dateien sind korrekt. Ebenso ist es Latex egal, mit welchem Editor man seine .tex-Dateien erstellt. Man kann sie mit dem Notepad-Programm, Emacs oder sogar Word erstellen. Erst wenn man sie an Latex übergibt, beginnt die Berechnung des Aussehens, und eine PDF-Datei entsteht. Für die einfachere Bearbeitung kann man natürlich einen speziellen Latex-Editor verwenden, der die Texterstellung vereinfacht und die entstehende Datei automatisch an Latex übergibt, sodass man sich zwischendurch während einer Denkpause unkompliziert schon mal anschauen kann, wie der Text am Ende aussehen wird. Zwei stelle ich weiter unten ausführlich vor (TexShop: Seite 8, Lyx: Seite 15).

Latex ist eine sogenannte Mark-up-Sprache. Grundsätzlich ist erst mal alles Text. Durch spezielle Befehle („Mark-up“) kann man Zeilen den Status von Titel, Überschrift, Zwischen-titel oder Zitat zuweisen. Eine Textpassage kann als Fußnote (`\footnote{Fußnotentext}`) markiert werden. Mit einem Anfangs- und End-Befehl (z.B. `\begin{description}` ...

`\end{description}`) gibt man an, wo beispielsweise Aufzählungen oder Listen beginnen und enden.

Das klingt erst mal ungewohnt, ist in der praktischen Arbeit aber bald genauso schnell wie das Anklicken der passenden Symbole in Word. Für Kontrollfreaks ist es allerdings die einzig wahre Art der Textbearbeitung, denn nur so haben sie wirklich alles im Griff; bei Word müssen sie sich darauf verlassen, dass im Hintergrund keine falschen Formatzuweisungen oder Einstellungen mehr mitlaufen. Bilder, Tabellen und andere Elemente fügt man über spezielle Mark-up-Befehle ebenfalls unkompliziert ein. Begriffe fügt man ruckzuck über einen Befehl in einen Index ein, Verweise auf andere Kapitel entstehen dynamisch und flexibel. Nicht abschrecken lassen – das meiste benötigt man am Anfang noch nicht.

Die Trennung in Arbeitsdatei (.tex) mit den Mark-up-Befehlen und fertige Druckdatei (.pdf) ist auch stabiler als die Kombination in einer Datei wie bei normalen Textverarbeitungen. Während Word sich beispielsweise am installierten Standarddrucker orientiert und dadurch je nach Computer die selbe Datei anders umbricht, arbeitet Latex immer mit dem vorgegebenen Seitenformat. Die Ergebnisse von Latex-Dateien sind dadurch zuverlässiger und leichter anpassbar als Word-Dateien, bei denen man eben keinen Zugriff auf die Mark-up-Informationen hat; selbst wenn man eine Word-Datei in einem Editor öffnet, kann man mit den kryptischen Steuerzeichen meist wenig anstellen.

1.3.3 Strafen für unschönes Aussehen

Hat man seinen Text mit den Mark-up-Befehlen strukturiert, beginnt mit Klick auf „Setzen“ das Zauberwerk von Latex. Dabei werden für jeden Absatz sämtliche Varianten durchgerechnet. Für jede Zeilentrennung vergibt Latex Strafpunkte, für jede Abweichung vom Super-Ideal gibt es Strafpunkte. Manche Bereiche werden mehrere Dutzend Mal durchgerechnet, bis eine optimale Version entstanden ist – nämlich eine, die besonders wenig Strafpunkte aufweist.

Über die Anzahl der Strafpunkte pro Abweichung vom Ideal wird das Verhalten gesteuert, so erhält eine falsch platzierte Fußnote eine unakzeptabel hohe Strafe, während eine lapidare Zeilentrennung das Ergebnis nur leicht verschlechtert. Als Profi kann man dann später die Strafpunkteverteilung sogar anpassen und so bestimmte Aspekte nachsteuern.

1.4 Woher bekomme ich LaTeX?

Latex erhält man kostenlos zum Download als TexLive-Version für die verschiedenen Systeme. Das TexLive-System bietet zahlreiche Pakete für so ziemlich alle Anwendungsfälle und Besonderheiten. Gelingt ein bestimmtes Detail (noch) nicht, liegt es meist daran, dass das richtige Paket noch nicht für das Dokument aktiviert wurde. TexLive bringt einen Updater mit, der die einzelnen Komponenten stets aktuell hält, neue Pakete automatisch installiert und Paketabhängigkeiten beachtet. So hat man stets ein funktionierendes Textsatz-System auf dem aktuellsten Stand.

Zu Anfang ist ein großes Paket (ca. 2 GB) herunterzuladen:

TexLive für Windows www.dante.de/tex/tl-install-windows/download.html – deutsche Anleitung

MacTex für Mac www.tug.org/mactex/ – Download, Installieren, Loslegen

TexLive für Linux <http://tug.org/texlive/quickinstall.html> – englische Anleitung

Empfehlenswert ist es, einen speziellen Latex-Editor zu nutzen, auch diese sind kostenlos. Ich favorisiere TexShop, den es allerdings nur für MacOS gibt. TexWorks ist diesem recht

ähnlich und auch für Windows verfügbar, auch TexMaker kann überzeugen. Jeder findet bei Latex einen Editor, der seinem Geschmack und seiner Arbeitsweise entspricht. Lyx als besonderer Editor (deshalb erhält er ein eigenes Kapitel, Seite 15) ist besonders für Nicht-Techniker zu empfehlen. Denn dort arbeitet man in einem visuellen Editor, muss die Mark-up-Befehle also nicht selbst setzen, sondern kann mit Einträgen in Einstellungen, Assistenten und Eigenschaften diese setzen lassen, sodass man nur seinen eigenen Text bar aller Mark-up-Befehle sieht. Das ist ein sehr komfortables Arbeiten, das sich für Autoren oder Studenten, die einfach nur strukturierten Text schreiben wollen, sehr bewährt.

Ein anständiges PDF-Programm ist zusätzlich sehr hilfreich. Bei Macs genügt das Vorschau-Programm. Andere sollten zumindest den Adobe Reader installiert haben oder schauen, ob sie irgendwo eine Acrobat-Version nutzen können. Das ist dann besonders nützlich, wenn man andere ums Korrekturlesen bittet oder gebeten wird. Dann kann man direkt in die PDF-Datei hineinkommentieren. Mit Vorschau oder Acrobat kann man auch mal eine Seite austauschen (beispielsweise die Standard-Titelseite gegen eine selbst gestaltete) oder Seiten löschen.

1.5 Wie fange ich an?

Mehr als die genannte Software braucht es nicht. Einen Duden hat man entweder in Reichweite oder die Duden-Website geöffnet. Kluge Gedanken sind im Kopf genug vorhanden, und einen Roten Faden für den Textaufbau hat man bereits ansatzweise beisammen.

Man legt sich für sein aktuelles Textprojekt einen Ordner an, in dem man später auch seine Datei speichern wird – und wo man später die errechnete PDF-Datei findet. In diesem Ordner kann man sich zwei Unterordner anlegen: „Material“ und „Bilder“. Im ersteren sammelt man alles, was man während der Texterstellung benötigen könnte: PDFs, die man heruntergeladen hat, Bilder, die man einbauen möchte, Textbausteine, die man (natürlich mit Quellenangabe!!) verwenden möchte usw. usf. Im Ordner „Bilder“ landen nur die Dateien, die man in seinen Text tatsächlich integrieren möchte, also die bearbeiteten Bilder, möglichst mit aussagekräftigen Dateinamen. Empfehlenswert ist bei Bildern übrigens das Dateiformat .png. Das ist besser als .jpg, da ja für die Einbindung in den Text jedes Bild angefasst und ggf. umgewandelt wird, sodass man sich hässliche .jpg-Artefakte erspart. Technische Zeichnungen können als .eps- oder .svg-Dateien vorliegen.

Man startet den Editor seiner Wahl, schreibt los und speichert die Datei in den Hauptordner des Projekts. Nicht wundern: Latex legt sich dort auch einige Hilfsdateien an, die es benötigt, beispielsweise für Verzeichnisse oder Indexe.

Wenn man technisch nicht so versiert ist, springt man jetzt einfach zum Kapitel „Text schreiben mit Lyx“ (Seite 15). Wer sich etwas intensiver mit Latex beschäftigen und die wichtigsten Mark-up-Befehle kennenlernen möchte, liest kontinuierlich weiter.

2 Text schreiben mit Editor und Mark-up

Mein Lieblingseditor TexShop ist einfach zu bedienen, flott und unterstützt mit seiner Farbmarkierung des Textes das Arbeiten: blau sind Befehle, rot sind Kommentare (durch %-Zeichen vom restlichen Absatz abgetrennt, gerade im Textkopf sehr nützlich), schwarz ist der eigentliche Text. Man stellt sich – unabhängig davon, wie der fertige Text später aussehen soll – eine Schrift für das Schreiben ein, so groß oder klein man möchte und kann gut fokussiert arbeiten. Mit `<cmd><+>` bzw. `<cmd><->` kann man die Schrift vergrößern/verkleinern. Die Mac-eingebaute Rechtschreibkorrektur leistet gute Dienste, alles geht flüssig von der Hand.

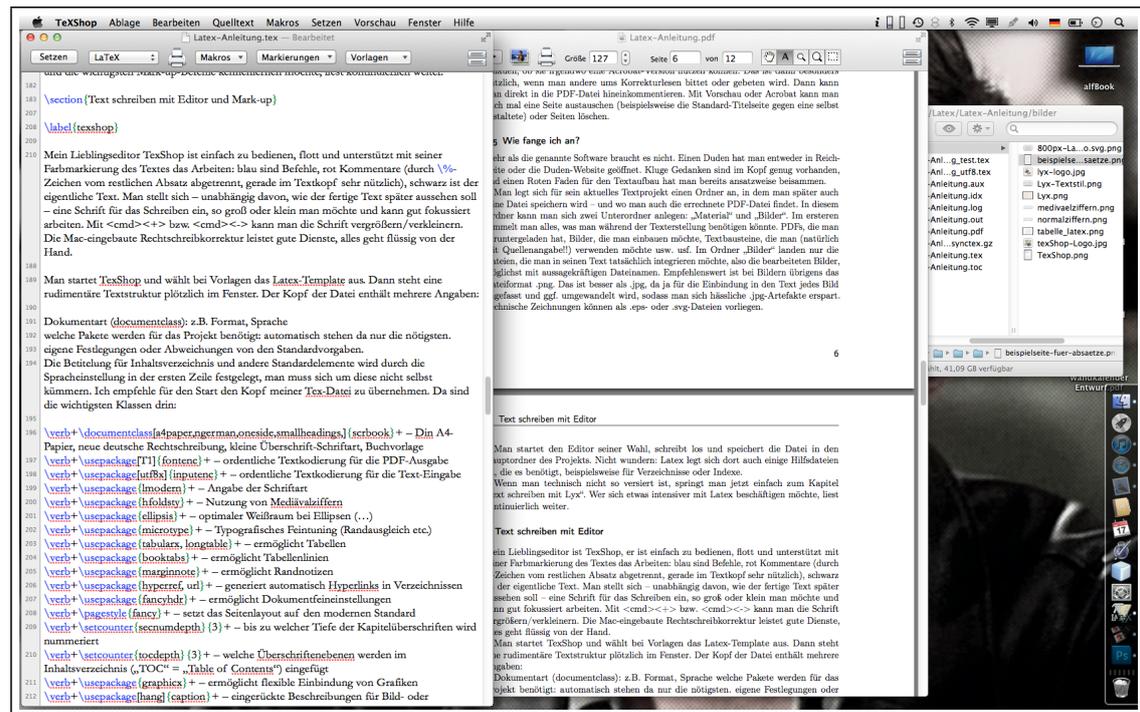


Abbildung 4: Das Programm TexShop hat ein Hauptfenster, hier links im Bild. In diesem Editor-Fenster schreibt man seinen Text und setzt die Mark-up-Befehle. Rechts daneben liegt gerade eine PDF-Ansicht; zum Testen habe ich den Text gesetzt. Im Hintergrund ist der Arbeitsordner mit den benötigten Dateien zu sehen.

Man startet TexShop und wählt bei Vorlagen das Latex-Template aus. Dann steht eine rudimentäre Textstruktur plötzlich im Fenster. Der Kopf der Datei enthält mehrere Angaben:

1. Dokumentart (documentclass): z.B. Format, Sprache
2. welche Pakete werden für das Projekt benötigt (usepackage): automatisch stehen da nur die nötigsten
3. eigene Festlegungen oder Abweichungen von den Standardvorgaben

Die Betitelung für Inhaltsverzeichnis und andere Standardelemente wird durch die Spracheinstellung in der ersten Zeile festgelegt, man muss sich um diese nicht selbst kümmern. Ich empfehle für den Start den Kopf meiner Tex-Datei zu übernehmen. Da sind die wichtigsten Klassen drin:

Tabelle 1: Die wichtigsten Angaben im tex-Dokumentkopf

<code>\documentclass[a4paper,ngerman,oneside,smallheadings]{scrbook}</code>	Din-A4-Papier, neue deutsche Rechtschreibung, kleine Überschrift-Schriftart, Dokumentart: Buchvorlage
<code>\usepackage[T1]{fontenc}</code>	ordentliche Textkodierung für die PDF-Ausgabe
<code>\usepackage[utf8x]{inputenc}</code>	ordentliche Textkodierung für die Text-Eingabe
<code>\usepackage{lmodern}</code>	Angabe der Schriftart
<code>\usepackage{microtype}</code>	Typografisches Feintuning (Randausgleich etc.)
<code>\usepackage{tabularx, longtable}</code>	ermöglicht Tabellen
<code>\usepackage{booktabs}</code>	ermöglicht Tabellenlinien
<code>\usepackage{marginnote}</code>	ermöglicht Randnotizen
<code>\usepackage{hyperref, url}</code>	generiert automatisch Hyperlinks in Verzeichnissen
<code>\usepackage{fancyhdr}</code>	ermöglicht Dokumentfeineinstellungen
<code>\pagestyle{fancy}</code>	setzt das Seitenlayout auf den modernen Standard
<code>\setcounter{secnumdepth}{3}</code>	bis zu welcher Tiefe der Kapitelüberschriften wird nummeriert
<code>\setcounter{tocdepth}{3}</code>	welche Überschriftenebenen werden im Inhaltsverzeichnis („TOC“ = „Table of Contents“) eingefügt
<code>\usepackage{graphicx}</code>	ermöglicht flexible Einbindung von Grafiken
<code>\usepackage[hang]{caption}</code>	eingrückte Beschreibungen für Bild- oder Tabellentitel
<code>\usepackage{makeidx}</code>	Index anlegen
<code>\makeindex</code>	hier wird der Index auch gleich angelegt
<code>\clubpenalty = 10000</code>	Schusterjungen und Hurenkinder (also Einzelzeilen von Absätzen am Seitenende oder -anfang) vermeiden
<code>\widowpenalty = 10000</code>	Schusterjungen und Hurenkinder vermeiden
<code>\displaywidowpenalty = 10000</code>	Schusterjungen und Hurenkinder vermeiden

Von den verwendeten Paketen hängen auch die dann zu verwendenden Mark-up-Befehle ab. Wenn man bestimmte Funktionen in Latex benötigt, muss man also nicht nur die richtigen Befehle kennen, sondern auch daran denken, die benötigten Pakete ggf. mit Zusatzparametern im Textkopf aufzurufen. In den letzten drei Zeilen habe ich die Strafpunkte für sogenannte Schusterjungen und Hurenkinder auf einen Wert gesetzt, der für Latex intolerabel ist, somit ist ausgeschlossen, dass diese auftreten; Latex muss weiterrechnen, bis es eine brauchbare Variante mit akzeptablen Strafpunkten gefunden hat.

2.1 Jetzt gehts los

Mit dem Befehl `\begin{document}` startet der eigentliche Text. Am Ende findet sich natürlich das schließende `\end{document}`. Es ist bereits sichtbar geworden: Mark-up-Befehle werden mit einem `\` eingeleitet, damit sie sich vom eigentlichen Text absetzen (auf dem Mac gibt man `\` mit `<alt><shift><7>` ein). Parameter für Befehle werden in direkt hintendrangeschrieben (ohne Leerzeichen).

Tabelle 2: Wir deklarieren die Angaben für die Titelseite

<code>\subject{Eine kurze Einführung in das Textsatzsystem}</code>	Das Thema
<code>\title{Texte schreiben mit LaTeX}</code>	Der Titel des Textes
<code>\author{Alexander Florin}</code>	Der Autor
<code>\date{text}</code>	Gibt man nichts an, wird das aktuelle Datum ausgegeben

Und sorgen anschließend dafür, dass als erstes auch die Titelseite ausgegeben wird. Wir können diesen Befehl auch ans Ende des Textes setzen, aber dann erscheint die Titelseite auch am Ende des Textes: `\maketitle`

Direkt nach der Titelseite möchten wir das Inhaltsverzeichnis platzieren. Wir können den Befehl natürlich auch nach dem Vorwort und vor dem ersten eigentlichen Kapitel platzieren, dann wird das Inhaltsverzeichnis dort erscheinen: `\tableofcontents`

Jetzt schreiben wir unser Vorwort und sind guter Dinge. Übrigens muss man daran denken, dass bei Latex immer zwei `<Enter>`-Schaltungen nötig sind, um einen Absatz zu erzeugen. Dadurch entsteht am Monitor zwar immer eine Leerzeile, aber die erleichtert letztlich sogar das Arbeiten.

2.2 Formate

Tabelle 3: Für die geforderte Strukturierung unseres Textes haben wir mehrere Mark-up-Befehle

<code>\chapter{Kapiteltitle}</code>	beginnt immer auf einer neuen Seite, entspricht also einem Buchkapitel oder Teil eines Buchs, erhält eine einfache Benummerung „1 Kapiteltitle“
<code>\section{Abschnittstitle}</code>	ein Teil eines Kapitels, erhält eine zweifache Benummerung „1.x Abschnittstitle“
<code>\subsection{Unterabschnitt}</code>	quasi Zwischentitel oder Unterkapitel, erhält eine dreifache Benummerung „1.1.x Unterabschnitt“
<code>\subsubsection{Unterunterabschnitt}</code>	eine vierte Hierarchieebene innerhalb des Unterkapitels, erhält eine vierfache Benummerung (falls wir im Dateikopf vier Ebenen vorgesehen haben!) „1.1.1.x Unterunterabschnitt“

Mit diesen vier Hierarchie-Ebenen kann man seinen Text schon mal schön strukturieren und viel schaffen.

Tabelle 4: Textformate innerhalb des Textflusses

<code>\textbf{fetter Text}</code>	fetter Text
<code>\textsl{geneiger Text}</code>	<i>geneiger Text</i> – die Buchstaben werden tatsächlich nur geneigt und nicht so filigranisiert wie bei kursivem Text
<code>\textit{kursiver Text}</code>	<i>kursiver Text</i>
<code>\textsc{Text mit Kapitälchen}</code>	TEXT MIT KAPITÄLCHEN
<code>\underline{unterstrichener Text}</code>	<u>unterstrichener Text</u>
<code>\textsf{serifenlose Schrift}</code>	serifenlose Schrift

Natürlich kann man auch Schrift größer oder kleiner setzen. Dafür gibt es zwei Möglichkeiten. Entweder man gibt global an, dass die Schrift ab dem Befehl in der betreffenden Größe ausgegeben wird – solange bis eine andere Schriftgröße (meist `normalsize`) angegeben ist. Oder man definiert mit `\begin{}` und `\end{}` den betroffenen Bereich. Ich verwende meist die einfache Schreibweise, setze beispielsweise die Schrift auf `\small` und hebe das dann einfach mit `\normalsize` wieder auf. Die Größe wird immer relativ zur eingestellten Standardgröße berechnet, bei 14 Punkt Standardgröße ist `\tiny` etwas größer als bei 11 Punkt Standardgröße. Bei `large` und `huge` ist die Groß-/Kleinschreibung wichtig, da diese beiden Wörter insgesamt fünf verschiedene Schriftgrößen bezeichnen:

Tabelle 5: Schriftgrößen

<code>\tiny</code>	winzige Schrift
<code>\scriptsize</code>	gerade noch lesbare Schrift
<code>\footnotesize</code>	Schrift in Fußnotengröße
<code>\small</code>	kleine Schrift
<code>\normalsize</code>	Standardschriftgröße: im Dokumentkopf veränderbar, die anderen Schriftgrößen verändern sich dann relativ zu dieser
<code>\large</code>	große Schrift
<code>\Large</code>	größere Schrift
<code>\LARGE</code>	noch größere Schrift
<code>\huge</code>	riesige Schrift
<code>\Huge</code>	riesiger geht nicht

2.3 Zusatztexte

Für Zitate gibt es zwei Formate, diese sind jeweils auf den gesamten Absatz anzuwenden. Kürzere Zitate im Fließtext werden mit „Anführungszeichen“ oder einer anderen Schriftmethode kenntlich gemacht. Nicht vergessen: Quelle angeben, entweder in einer Fußnote oder im Text.¹

Tabelle 6: Formate für Zitat-Absätze

<code>\begin{quote}</code> Zitierter Absatz <code>\end{quote}</code>	Zitat mit einem Absatz
<code>\begin{quotation}</code> Zitierte Absätze <code>\end{quotation}</code>	Zitate mit mindestens zwei Absätzen

Fußnoten stehen auf der betreffenden Seite am unteren Seitenrand. Sie werden pro Kapitel durchgehend durchnummeriert. Fußnoten werden in einer kleineren Schriftart ausgegeben. Man fügt den Fußnotenbefehl an der Stelle im Text ein, wo dann auch die Nummer erscheinen soll: `Text Text \footnote{Text der Fußnote} Text Text`²

In langen Texten können Randnotizen oder Schlagworte am Rand die Gliederung verdeutlichen und damit die Orientierung erleichtern. Diese funktionieren genauso wie Fußnoten:

`Text Text \marginpar{Schlagwort} Text Text` – das Schlagwort wird dann in der Zeile am Rand platziert. Schlagwort

¹Mit Fußnoten gibt es keine Ausrede für vergessene Quellenangaben. Hier kann man auch gut weiterführende Gedanken loswerden, für die im eigentlichen Text der Platz fehlt oder deren Bedeutung eine Platzierung im Haupttext nicht rechtfertigt.

²Text der Fußnote

Endnoten erscheinen erst am Ende des Textes. Für diese muss das passende Package aktiviert sein. Anschließend muss man noch angeben, wo die Endnoten ausgegeben werden sollen. Den eigentlichen Endnotentext gibt man stets dort an, wo auch die Nummer im Text erscheinen soll:

Tabelle 7: Endnoten müssen im Gegensatz zu Fußnoten extra im Dokumentkopf aktiviert werden.

<code>\usepackage{endnotes}</code>	im Textkopf
<code>Text Text \endnote{Text der Endnote} Text Text</code>	Angabe der Endnote an der Stelle im Text, wo die Endnotenziffer erscheint
<code>\theendnotes</code>	an der Stelle im Dokument, wo die Endnoten ausgegeben werden Auflistungen

Es gibt drei Arten von Aufzählungen: Auflistung mit Listenpunkt, Aufzählung mit Zahlen und Auflistung von Definitionen. Der Aufbau ist bei allen drei gleich:

Tabelle 8: Aufzählungen nutzen

<code>\begin{itemize}</code>	Auflistung mit Listenpunkt:
<code>\item Eintrag 1 ...</code>	• Eintrag 1
<code>\item Eintrag 2 ...</code>	• Eintrag 2
<code>\end{itemize}</code>	
<code>\begin{enumerate}</code>	Auflistung mit Zahlen:
<code>\item Eintrag 1 ...</code>	1. Eintrag 1
<code>\item Eintrag 2 ...</code>	2. Eintrag 2
<code>\end{enumerate}</code>	
<code>\begin{description}</code>	Auflistung von Definition (ohne Listenpunkt):
<code>\item[Eintrag 1] Erläuterung ...</code>	Eintrag 1 Erläuterung
<code>\item[Eintrag 2] Erläuterung ...</code>	Eintrag 2 Erläuterung
<code>\end{description}</code>	

2.4 Tabellen

Tabellen sind recht komplex mit reinen Textzeilen und Mark-up-Befehlen zu erzeugen. Hat man aber die erste gemeistert, entsteht rasch die Lust auf mehr. Denn Latex-Tabellen sehen wirklich gut aus, und man kann sie einfach über mehrere Seiten laufen lassen. Im Drucksatz hat es sich übrigens bewährt, bei Tabellen mit Text Linien nur zur Trennung von Tabellenzeilen zu verwenden, die Spalten werden nicht mit Linien voneinander getrennt. Da Tabellen zur höheren Kunst gehören, kommentiere ich einfach mal eine meiner ersten Tabellen.

Tabelle 9: Eine Beispieltabelle

<code>\begin{table}[h]</code>	die Tabelle beginnt
<code>\caption{Narrative Modi im Überblick}</code>	Titel der Tabelle
<code>\vspace{3mm}</code>	Abstand von drei Millimetern
<code>\centering</code>	Tabelle soll mittig auf der Seite ausgerichtet werden

<code>\begin{tabularx}{\textwidth}{@{}l>{\raggedright\arraybackslash}X>{\raggedright\arraybackslash}X>{\raggedright\arraybackslash}X>{\raggedright\arraybackslash}X@{}}</code>	das moderne tablex-Paket tritt jetzt in Kraft, im Anschluss werden die fünf Spalten definiert und der Tabelleninhalt auf linksbündig (Flattersatz rechts) gesetzt
<code>\toprule</code>	Tabellenlinie
<code>& \textsf{faktual} & \textsf{fiktional} & \textsf{phantastisch} & \textsf{inkredibel} \\ \tabularnewline</code>	Tabellenkopf (durch textsf sind die Spaltentitel hervorgehoben), Spaltenwechsel mit &-Zeichen
<code>\midrule</code>	zurückhaltende Tabellenlinie
<code>\textsf{Materie} & real & realiter & kein Wahrheitsanspruch & kein Wahrheitsanspruch \\ \tabularnewline</code>	erste Tabellenzeile mit Inhalt befüllt
<code>\addlinespace</code>	etwas Zeilendurchschuss (statt einer Tabellenlinie; find ich schöner)
<code>\textsf{Story} & real & potenziell wahr & glaubwürdig (in der Materie potenziell wahr) & kein Anspruch auf Wahrhaftigkeit \\ \tabularnewline</code>	nächste Tabellenzeile
<code>\addlinespace</code>	etwas Zeilendurchschuss
<code>\textsf{Beispiele} & Dokumentationen, wissenschaftliche Filme, Konzert-aufzeichnungen, Journalismus & Romantic Comedys (z.B. \titel{You've Got Mail}, \titel{Sleepless in Seattle}), Soap Operas & Fantasy (z.B. \titel{Lord of the Rings}), Science Fiction (z.B. \titel{Star Wars}, \titel{Star Trek}) & irrelevant \\ \tabularnewline</code>	nächste Tabellenzeile
<code>\bottomrule</code>	abschließende Tabellenlinie
<code>\end{tabularx}</code>	die Tabellenumgebung wird beendet
<code>\end{table}</code>	Tabellenende

Und so sieht die Tabelle dann in der fertigen PDF aus:

Tabelle 10: Narrative Modi im Überblick

	faktual	fiktional	phantastisch	inkredibel
Materie	real	realiter	kein Wahrheitsanspruch	kein Wahrheitsanspruch
Story	real	potenziell wahr	glaubwürdig (in der Materie potenziell wahr)	kein Anspruch auf Wahrhaftigkeit
Beispiele	Dokumentationen, wissenschaftliche Filme, Konzertaufzeichnungen, Journalismus	Romantic Comedys (z.B. <i>You've Got Mail</i> , <i>Sleepless in Seattle</i>), Soap Operas	Fantasy (z.B. <i>Lord of the Rings</i>), Science Fiction (z.B. <i>Star Wars</i> , <i>Star Trek</i>)	irrelevant

Den Mark-up-Befehl `\title` habe ich im Kopf selbst als `\textsl` definiert; dadurch konnte ich mich bis zum Ende einfach umentscheiden, falls mir doch `\textit` besser gefallen hätte, wäre eine Änderung nur an der einen Stelle im Dokumentkopf nötig gewesen):

Hätte man statt `table` den Befehl `longtable` genommen, könnte die Tabelle über mehrere Seiten fließen. Die einfache `table` muss – wie ein Bild – auf einer Seite untergebracht werden und wird mit dem ersten Parameter `[h]` platziert:

h – genau hier

b – am unteren Seitenrand

t – am oberen Seitenrand

p – auf einer eigenen Seite

Meist gibt man mehrere Parameter hintereinander an: `[hbp]`, die erste passende Möglichkeit wird umgesetzt: Ist die Tabelle an der Stelle platzierbar, landet sie hier (h); wenn nicht, prüft Latex, ob sie an den unteren Seitenrand passt (b); wenn auch das nicht gelingt, dann erhält die Tabelle ihre eigene Seite (p).

2.5 Bilder einfügen

Damit ein wenig Abwechslung in den Text kommt, kann man schnell noch ein Bild einfügen. Auch das wirkt im ersten Moment nicht allzu komfortabel, aber das Ergebnis im Textfluss überzeugt. Mit den Parametern `[htb]` kann das Bild hier, am oberen oder unteren Seitenrand platziert werden. Zu jedem Bild gehört ein Bildtitel, die Bezeichnung wird mit einer kapitelgebundenen Nummerierung wie bei der Tabelle automatisch dem Bildtitel vorangestellt.

Tabelle 11: Aufzählungen nutzen

<code>\begin{figure}[ht]</code>	Bildumgebung aufrufen
<code>\centering</code>	Bild zentriert platzieren
<code>\includegraphics[width=\linewidth]{bilder/Dateiname.png}</code>	Datei einfügen (der Dateipfad wird relativ zur Latex-Datei angegeben), der Parameter Bildbreite (<code>width</code>) ist auf Zeilenbreite gesetzt, man kann natürlich auch <code>cm</code> verwenden, dann wird das Bild passend skaliert
<code>\caption[Bildtitel]</code>	mit diesem Titel wird das Bild auch im Abbildungsverzeichnis aufgeführt
<code>\end{figure}</code>	Bildumgebung beenden

Alles in allem kein Hexenwerk. Oft kann das „Makros“-Menü helfen, denn dort sind die wichtigsten Mark-up-Befehlsfolgen verfügbar und brauchen dann nur mit den richtigen Angaben befüllt zu werden. Wer sich mit HTML-Code auskennt, sieht die Ähnlichkeiten im strukturellen Aufbau und ist die gleichen Aufwände gewohnt. Übrigens lassen sich wie bei HTML sämtliche Gestaltungselemente beeinflussen: Standardschriftgröße, Schriftschnitt von Überschriften, Abstände, Einzüge, Farben (die hab ich mal weggelassen), Formatregeln und natürlich auch neue Befehlsfolgen anlegen. Das sollte immer im Textkopf vorgenommen werden, damit es global gilt.

Mit `\newcommand{\bild}[2]{\begin{figure}[ht]\centering`

`\includegraphics[width=\linewidth]{#1}\caption[#2]\end{figure}}`.

Und mit der Umdefinierung `\renewcommand\captionfont{\small\sf}` erscheinen bei mir die Bild- und Tabellentitel immer klein und serifenlos (sf). So Spielereien halt, wenn man zu viele Denkpausen beim Schreiben einlegt.

2.6 Tipps beim Setzen

Zur Sicherheit sollte man sich vor dem Ausdrucken das Dokument immer zweimal mit dem „Setzen“-Befehl berechnen lassen. Und dabei natürlich als Methode „Latex“ ausgewählt haben (nicht ConTeXt o.ä.). Beim ersten Durchlauf werden die Hilfsdateien für Verzeichnisse und Indexe erzeugt, die erst beim zweiten Durchlauf korrekt eingebunden werden, so können sich Seitenzahlen von Kapiteln ändern, was aber erst beim zweiten Durchlauf auch im Inhaltsverzeichnis zu sehen ist.

Mitunter erzeugen die selbstgeschriebenen Mark-up-Befehle Fehlermeldungen. Entweder hat man sich verschrieben, einen Parameter vergessen oder etwas anderes Törichtes getan. Die Fehlerkonsole kann in solchen Fällen oft Auskunft geben. Oder man fragt bei Tante Google. Nicht selten hilft es auch, die Hilfsdateien zu löschen, bevor man die Datei erneut setzt.

3 Text schreiben mit Lyx

Während die „harten Coder“ auf ihrem .tex-Format bestehen und alles per Hand eingeben wollen, wissen Autoren den Komfort einer gut eingerichteten Umgebung zu schätzen. Sie installieren Lyx, schreiben einfach los und arbeiten wie in Word: Text eingeben und einfach einer Zeile oder einem Absatz die jeweilige Formatvorlage zuweisen, beispielsweise „Abschnitt“ für Abschnittitel oder „Zitat“. Bilder und Tabellen fügen sie einfach ein und können dann mit den Eigenschaften der Elemente (im Kontextmenü, über Rechtsklick) etwas Feintuning betreiben.

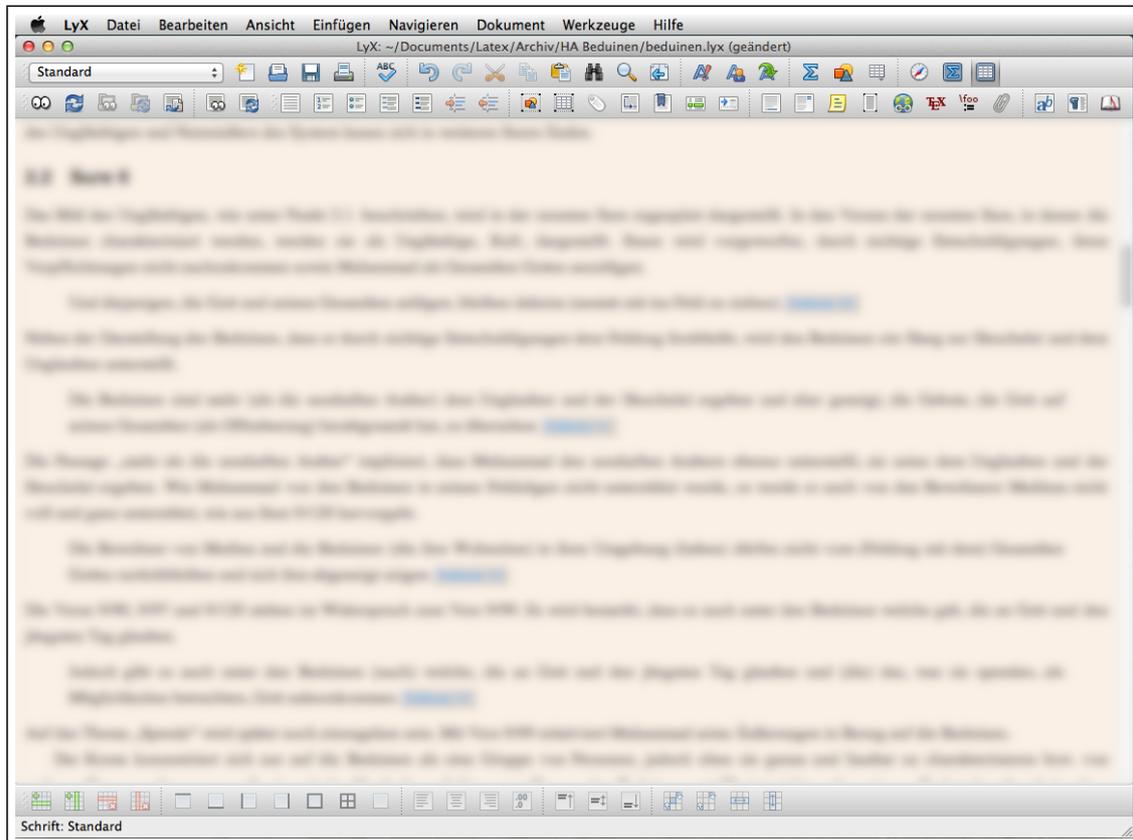


Abbildung 5: Das Arbeitsfenster von Lyx zeigt den Text und die Struktur optisch, aber im Ausdruck sieht alles viel besser aus und ist auf das Seitenformat abgestimmt.

Etwas ist aber anders als bei Word: Die Zeilen sind so breit wie das Computerfenster, und es sieht recht spartanisch und wenig schön aus, jedenfalls gemessen an Maßstäben für bedrucktes Papier. Auch in Lyx stellt man seine bevorzugte Schrift zum Schreiben ein, und erst beim Klick auf die Augen oben links wird dann das fertige Aussehen berechnet, und eine PDF-Datei entsteht.

Auch gibt es keine Symbole für „fett“ oder „kursiv“, stattdessen kann man Formate für Hervorhebung und Eigenname benutzen. Eben: Lyx stellt beim Schreiben nicht das Aussehen, sondern

die strukturelle Bedeutung in den Vordergrund. Wer trotzdem ganz bewusst ein Format ändern möchte, kann einen Textbereich markieren und im Kontextmenü (rechter Mausklick) „Textstil“ > „benutzerdefiniert“ wählen. Im erscheinenden Fensterchen gibt es dann zahlreiche Möglichkeiten, den gewählten Textbereich anders zu gestalten.

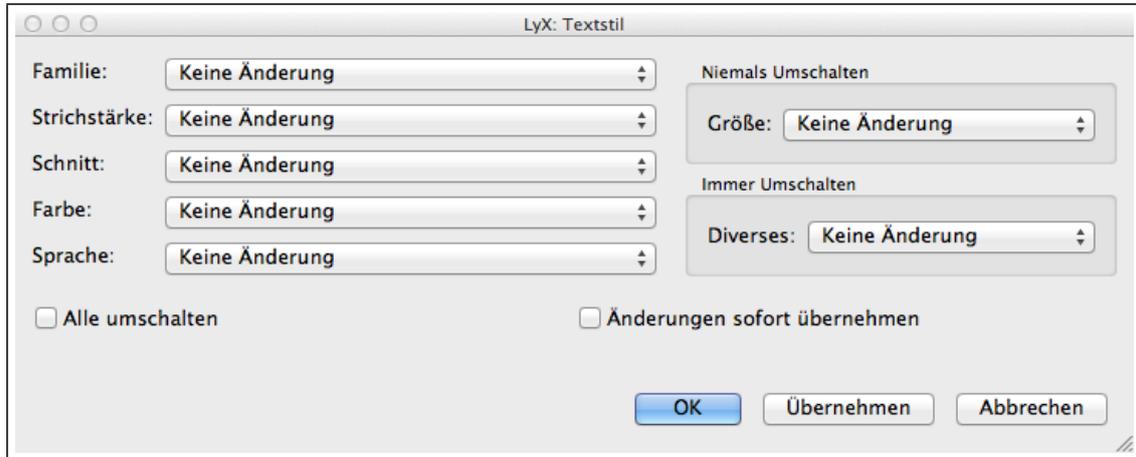


Abbildung 6: Mit dieser Textstil-Palette kann für jeden markierten Bereich das Aussehen filigran eingestellt werden. Letztlich sind so „nur“ die Mark-up-Befehle zur Textgestaltung über diese Oberfläche einfach zugänglich.

So gibt es bei Lyx für vieles Dialogfenster oder Assistenten, was TexShop-Besitzer mit Mark-up-Befehlen erledigen müssen. Das wichtigste Menü dürfte beim Schreiben dann das Einfügen-Menü sein, denn darüber holt man sich alles, was das Herz begehrt: Sonderzeichen, Trennmöglichkeiten, Seitenwechsel, Verweise usw. usf.

Bei Lyx muss man aufpassen, denn es gibt zwei Einstellungsmöglichkeiten. Zum einen kann man allgemeine Einstellungen für das Programm vornehmen, beispielsweise die Anzeigeschrift oder Programmsprache oder Latex-Parameter (bei denen aber lieber Finger weg, wenn man nicht genau weiß, was man tut). Dann gibt es noch Einstellungen, die nur für das aktuelle Dokument gelten. In diesen legt man fest, was man im TexShop im Dokumentkopf geregelt hat: Dokumentklasse, Schriften, Seitenränder, Verzeichnisse und kann auch eigene Latex-Befehle einfügen, sofern man dies möchte.

Kurzum, Lyx ist eine einfache und vorbehaltlos zu empfehlende Angelegenheit.

Wer es dennoch in manchen Belangen zu umständlich findet oder die gefühlte Einfachheit von Word herbeisehnt, sollte abwägen zwischen gefühltem Zusatz-Aufwand und erhaltener Zusatz-Qualität. Um lange Texte schön und ansprechend zu setzen, ist Word keine Alternative mehr, wenn man einmal Lyx-Luft geschnuppert hat.

4 Praktische Tipps

4.1 Layout, allgemein

Vor jedem Kapitel sollte man pro forma ein `\cleardoublepage` einfügen. Das sorgt dafür, dass Latex seinen Zwischenspeicher leerräumt, es werden also keine Fußnotenreste oder Bilder oder Tabellen mehr ins nächste Kapitel weitergereicht.

Mit `\newpage` erzeugt man eine neue Seite. `\newline` führt eine weiche Zeilenschaltung aus, dabei entsteht kein Absatz mit Einzug in der ersten Zeile, `\\` bewirkt das gleiche.

Abbildungsverzeichnis (`\listoffigures`) und Tabellenverzeichnis (`\listoftables`) werden automatisch generiert und an der Stelle ins Dokument integriert, wo die Befehle stehen.

4.2 Zeilentrennungen

Tabelle 12: Manuelle Trennungen angeben oder das Trennen verhindern

<code>\-</code> (Trenn\-möglich\keiten)	Angabe von Trennmöglichkeiten in einem Wort
<code>\nohyphens{Zeilentrennungverhinderung}</code>	keine Trennung; dazu wird im Dokumentkopf <code>\usepackage{hyphenat}</code> benötigt
<code>\mbox{Zeilentrennungsverhinderung}</code>	keine Trennung des Worts

4.3 Seitenzahlen

Wir können im Lauf des Textes die Seitennummerierung umschalten zwischen arabischen und römischen Zahlen. Bei längeren Vorworten oder Einleitungen hat es sich etabliert, diese römisch zu beziffern und den eigentlichen Text dann mit arabischen Seitenzahlen. Die Schreibweise wechselt ab der Seite, wo man im Text einen der folgenden Befehle platziert:

Tabelle 13: Formate für Seitenzahlen

<code>\pagenumbering{arabic}</code>	arabische Seitenzahlen: 1 2 3 4 5 6 7
<code>\pagenumbering{roman}</code>	römische Seitenzahlen: i ii iii iv v vi vii
<code>\pagenumbering{Roman}</code>	römische Seitenzahlen: I II III IV V VI VII
<code>\pagenumbering{alph}</code>	Buchstaben als Seitenzahlen: a b c d e f g
<code>\pagenumbering{Alph}</code>	Buchstaben als Seitenzahlen: A B C D E F G

4.4 Nachschlagequellen

Linupedia www.linupedia.org/opensuse/LaTeX – ausführliche Vorstellung aller Standardoptionen, ideal zum schnellen Nachschlagen

Symbole <ftp://ctan.tug.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf> – Übersicht (PDF) über Sonderzeichen und andere Symbole und wie man sie bei Latex korrekt eingibt

Dante (Deutschsprachige Anwendervereinigung Tex e.V.) www.dante.de/ – viel Hilfe, Anleitungen und Hintergrundinfos

Befehlsübersicht www.weinelt.de/latex/ – alle Mark-up-Befehle verständlich und praxisnah erklärt, zwar nicht ganz aktuell, aber immer nützlich

Google einfach „latex“ und ein bis drei Schlagwörter eingeben – die passende Antwort ist dann oft nur noch einen Mausklick entfernt

5 Abbildungsverzeichnis

1	Mediävalziffern fügen sich mit ihrer variablen Zeilenorientierung harmonisch ins Schriftbild.	3
2	Standardziffern wirken im Fließtext immer etwas aufdringlich, etwa wie Wörter in Großbuchstaben.	3
3	Die Einzüge bei Absätzen gelingen automatisch.	4
4	Das Programm TexShop hat ein Hauptfenster, hier links im Bild. In diesem Editor-Fenster schreibt man seinen Text und setzt die Mark-up-Befehle. Rechts daneben liegt gerade eine PDF-Ansicht; zum Testen habe ich den Text gesetzt. Im Hintergrund ist der Arbeitsordner mit den benötigten Dateien zu sehen.	8
5	Das Arbeitsfenster von Lyx zeigt den Text und die Struktur optisch, aber im Ausdruck sieht alles viel besser aus und ist auf das Seitenformat abgestimmt. . .	15
6	Mit dieser Textstil-Palette kann für jeden markierten Bereich das Aussehen filigran eingestellt werden. Letztlich sind so „nur“ die Mark-up-Befehle zur Textgestaltung über diese Oberfläche einfach zugänglich.	16

6 Tabellenverzeichnis

1	Die wichtigsten Angaben im tex-Dokumentkopf	9
2	Wir deklarieren die Angaben für die Titelseite	10
3	Für die geforderte Strukturierung unseres Textes haben wir mehrere Mark-up-Befehle	10
4	Textformate innerhalb des Textflusses	10
5	Schriftgrößen	11
6	Formate für Zitat-Absätze	11
7	Endnoten müssen im Gegensatz zu Fußnoten extra im Dokumentkopf aktiviert werden.	12
8	Aufzählungen nutzen	12
9	Eine Beispieltabelle	12
10	Narrative Modi im Überblick	13
11	Aufzählungen nutzen	14
12	Manuelle Trennungen angeben oder das Trennen verhindern	17
13	Formate für Seitenzahlen	17