



L^AT_EX-Vorlage für die Fakultät IMN

Von: Dominik Först
Matrikelnr.: 42135
Am: 26. September 2011
Studienrichtung: Medieninformatik
Professor: Prof. Dr. rer. nat. habil. Michael Frank
Fakultät: Informatik, Mathematik und Naturwissenschaften
der Hochschule für Technik, Wirtschaft und Kultur (HTWK) Leipzig

Tabellenverzeichnis

5.1	Tabelle mit Zellenausrichtung	20
5.2	Tabelle mit Linien	20
5.3	Einfache Liste	27
5.4	Nummerierte Liste	27
5.5	Liste mit Bezeichner	27
5.6	minipage-Umgebung	27

Abbildungsverzeichnis

3.1	MiKTeX Package-Manager	5
3.2	TeXnicCenter	6
3.3	Texmaker	6
3.4	Zotero	8
3.5	Citavi	9
3.6	calc2latex	11
5.1	Parametereinträge für Makeindex	22

Inhaltsverzeichnis

Tabellenverzeichnis	i
Abbildungsverzeichnis	ii
1 Einleitung und Intension	1
2 Allgemeines zu \LaTeX	2
3 Was wird benötigt zum Arbeiten mit \LaTeX	3
3.1 Das \LaTeX Grundsystem	3
3.2 MiKTeX	4
3.3 Editoren	4
3.4 Helfer für Quellen und Zitate	7
3.5 Helfer zur Tabellenerstellung	10
4 Arbeitsumgebung	12
4.1 Dateistruktur	12
4.2 Formatdateien	13
4.3 Masterdatei	13
5 Vorgaben des Fachbereiches IMN	15
5.1 Schrift	15
5.2 Überschriften	16
5.3 Absätze	17
5.4 Abbildungen	18
5.5 Tabellen	19
5.6 Abkürzungen und Glossar	20
5.7 Zitate	23
5.8 Fußnoten	24
5.9 Seitenzahlen	24

Inhaltsverzeichnis

5.10 Aufzählungen und Nummerierungen (Listen)	24
5.11 Ränder	25
5.12 Eingebettete Objekte	25
6 Aussicht	28

1 Einleitung und Intension

Das Anfertigen von Graduierungsarbeiten ist mit vielen Vorgaben zu Inhalt und Layout verbunden. Diese Vorgaben variieren zwischen den Meinungen und Vorstellungen der Hochschulen und der jeweiligen Professoren.

Es existieren zwar entsprechende Normen, um eine grobe Vereinheitlichung zu erreichen, jedoch beinhalten diese immer noch gewisse Freiheitsgrade, welche dann unterschiedlich interpretiert werden. Um für den Fachbereich IMN der HTWK eine Vereinheitlichung zu bestimmen, wurde von Herrn *Mathias Thomann* Meinungen von mehreren Professoren erfasst und in seiner Arbeit *Gedanken zur Gestaltung von wissenschaftlichen Abschlussarbeiten an der Fakultät Informatik, Mathematik und Naturwissenschaften der HTWK Leipzig* zusammengetragen. Diese Arbeit orientiert sich an *Microsoft Word* und enthält deshalb vor allem Vorschläge, die mit demselben umgesetzt werden. Da in naturwissenschaftlichen Bereichen häufig komplexe Formeln, Diagramme oder Quellcode erstellt werden müssen, bietet es sich an, das Textsatzsystem $\text{T}_{\text{E}}\text{X}$ zu benutzen. Dieses bietet eine Vielfalt an Formatierungsmöglichkeiten und vorgefertigten Umgebungen für diese Einsatzgebiete.

Die folgende Arbeit enthält eine nach den Vorgaben der Fakultät erstellte Vorlage, mit welcher der Einstieg zur Arbeit mit $\text{T}_{\text{E}}\text{X}$ und der $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Umgebung erleichtert werden soll.

Die Arbeit selbst beschreibt zu Beginn einige Grundlagen zu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (Kapitel 2) und was zum Arbeiten benötigt wird (Kapitel 3), später den Aufbau der erstellten Vorlage (Kapitel 4) und danach einige Vorgaben der Fakultät IMN und der damit verbundenen Formatierungsmöglichkeiten (Kapitel 5). Sollte man noch keine Erfahrungen mit $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ haben, dann empfiehlt sich eine Einführung [1, 2] oder der Onlinekurs der Universität Stuttgart [3].

¹<http://www.meta-x.de/faq/LaTeX-Einfuehrung.html>

²<http://lefti.amigager.de/latex/>

³http://mo.mathematik.uni-stuttgart.de/kurse/kurs44/index_full.html

2 Allgemeines zu L^AT_EX

Zwischen 1977 und 1986 entwickelte *Donald E. Knuth* das Textsatzsystem T_EX inklusive einer eingebauten Makrosprache. Grundsätzlich ist T_EX für alle Arten von Texten geeignet, jedoch wird es vorwiegend für umfangreiche Texte und wissenschaftliche Arbeiten benutzt. Als Vorteile werden meist der mathematische Formelsatz und das Schriftbild genannt. Mit ca. 300 fest integrierten Befehlen und einem Mechanismus zur Definition eigener Makros arbeitet T_EX die von den Autoren erstellten Skripte ab. Dabei wird absatzweise der Textsatz mit einem Optimierungsverfahren erstellt.¹ Basierend auf T_EX erstellte in den 1980er Jahren *Leslie Lamport* eine umfangreiche Sammlung von Makros. Diese Sammlung erhielt den Namen L^AT_EX als Kürzel von Lamport T_EX.²

L^AT_EX ermöglicht eine Objektivierung und Standardisierung von Formatanweisungen. So werden in T_EX beispielsweise zentral Anweisungen zu Schriftgröße und Schriftart für Überschriften hinterlegt. In L^AT_EX wird die Überschrift dann unter dem zuvor in T_EX definierten Schlüsselbegriff verwendet (z.B. `\section{Beispiel--Überschrift}`). So werden Format-Anweisungen oder Änderungen nur einmal für alle Überschriften des Typs `\section` erstellt und sind im gesamten Dokument gültig.

Im Gegensatz zu Textverarbeitungssystemen wie *Microsoft-Word* oder *Open-Office*, welche als *WYSIWYG* (what-you-see-is-what-you-get) bezeichnet werden, arbeitet man in einem reinen Textmodus, welcher Format-Anweisungen wie **Fett** oder *Kursiv* nicht im Ergebnis anzeigt sondern mit dem entsprechenden Befehl `\textbf{Fett}` für fettgedruckte Schrift oder `\textit{Kursiv}` für kursive Schrift. Das formatierte Ergebnis sieht man erst nach der Verarbeitung dieses Textdokumentes durch das L^AT_EX-System. Dieses Verfahren wird mit *WYGIWYM* (what-you-get-is-what-you-mean) bezeichnet, da Inhalt und Layout komplett voneinander getrennt sind. Für Einsteiger gibt es auch Editoren für L^AT_EX, welche nach dem WYSIWYG-Prinzip arbeiten oder beide Ansichten bieten (siehe Kapitel 3.3).

¹<http://projekte.dante.de/DanteFAQ/WebHome>

²<http://research.microsoft.com/en-us/um/people/lamport/pubs/pubs.html#latex>

3 Was wird benötigt zum Arbeiten mit \LaTeX

Bei WYSIWYG-Programmen benötigt man zum Arbeiten lediglich eine installierte Version. Diese enthält sowohl den Editor als auch den verarbeitenden Interpreter im Hintergrund. Da \LaTeX aus vielen Einzelpaketen besteht, werden nur eine Sammlung an Grundpaketen und einige Helfer-Programme installiert. Den Editor kann man dann selbst auswählen. Den Umfang der Sammlung an Grundpaketen kann man beim Herunterladen auswählen. Im Folgenden wird die Einrichtungsroutine erläutert. Anschließend werden einige Editoren vorgestellt und es wird auf Paketsammlungen eingegangen. Weiterhin werden Helferprogramme vorgestellt, welche das Arbeiten mit Quellen und Tabellen vereinfachen.

3.1 Das \LaTeX Grundsystem

Um schnell und unkompliziert mit dem Arbeiten zu beginnen, würde es genügen auf die *proTeXt*-Distribution¹ hinzuweisen. Diese enthält alle essentiellen Teile, um mit \LaTeX arbeiten zu können. Darunter das im folgenden beschriebene *MiKTeX*, das *TeXnicCenter* (siehe 3.3) und *Ghostscript*², einem Interpreter für Pdf und Postscript. Neben *proTeXt* gibt es noch *TeXLive*³, welches als Gesamtpaket auf DVD vertrieben wird. Einige Editoren können jedoch nicht mit der Pfadstruktur im *TeXLive* arbeiten. Deshalb, und um den Umfang der Arbeit in Grenzen zu halten, wird in dieser Arbeit nicht weiter auf *TeXLive* eingegangen. Jedoch ist zu beachten das unter *MacOsX* zu verwendende *MacTeX*⁴ auf *TeXLive* basiert. Um ein besseres Verständnis für die Arbeitsweise von \LaTeX zu bekommen, wird kurz das *MiKTeX*-Paketsystem beschrieben.

¹<http://www.tug.org/protext/>

²<http://www.ghostscript.com/>

³<http://www.tug.org/texlive/>

⁴<http://www.tug.org/mactex>

3.2 MiKTeX

Wie vorangegangen schon beschrieben, besteht das L^AT_EX-System aus einer Sammlung von Einzelpaketen. Für verschiedene Aufgaben (Einbinden von Bildern, Tabellen, Benutzen von Farben,...) werden zusätzliche Pakete benötigt. Diese müssen im Quelltext eingebunden werden. Alle auf dem Computer vorzuhalten, wäre sehr speicherintensiv. Um nur benötigte Teile zu nutzen, bietet MiKTeX die Möglichkeit ein Basis-System⁵ zu installieren, welches mittels eines Paketmanagers die fehlenden Teile bei Bedarf automatisch nachlädt. Das Nachladen geschieht demzufolge, wenn der Interpreter den Quelltext durcharbeitet und erkennt, dass ein Paket benötigt wird. Durch die Angabe eines speziellen Paketnamens im Kopf der Arbeit mit dem Befehl `\usepackage['Paketname']` werden ebenso Pakete eingebunden, die während des Erstellens heruntergeladen werden. Eine dritte Möglichkeit besteht darin, gewünschte Elemente im Paketmanager (Abbildung 3.1 auf Seite 5) auszuwählen. So sind auch ohne Internetverbindung alle Teile vorhanden. Sollte man häufig offline arbeiten, empfiehlt es sich das gesamte MiKTeX herunterzuladen. Hierfür wählt man die entsprechende Option beim Ausführen des MiKTeX-*Net Installer*.

Um nun mit dem MiKTeX-Grundsystem Texte zu erstellen, bedarf es eines Editors. Im Folgenden werden einige Editoren kurz vorgestellt.

3.3 Editoren

Mit einem MiKTeX-Grundsystem kann man mit Hilfe eines Konsolen-Fensters, dem entsprechenden Konsolen-Befehl und des erstellten Quelltextes ein Dokument erstellen. Da dies aber für den Laien recht umständlich ist, gibt es Editoren, welche das Arbeiten mit L^AT_EX erleichtern und das Erstellen des Dokumentes übernehmen.

Nachfolgend werden einige Editoren kurz benannt und ihre Vorteile beschrieben. Ein Vergleich der benannten sowie weiterer Programme findet man unter [6].

TeXnicCenter Einer der meistverwendeten Editoren ist *TeXnicCenter*⁷ (Abbildung 3.2). Er ist frei verfügbar aber nur für Windows erhältlich. Durch Tastaturkommandos und anwählbare Buttons können häufig verwendete Formatierungsanweisungen direkt in den Text eingearbeitet werden. Man arbeitet direkt im L^AT_EX-Quellcode. Anweisungen

⁵<http://miktex.org/2.9/setup>

⁶http://en.wikipedia.org/wiki/Comparison_of_TeX_editors

⁷<http://www.texniccenter.org/>

3 Was wird benötigt zum Arbeiten mit L^AT_EX

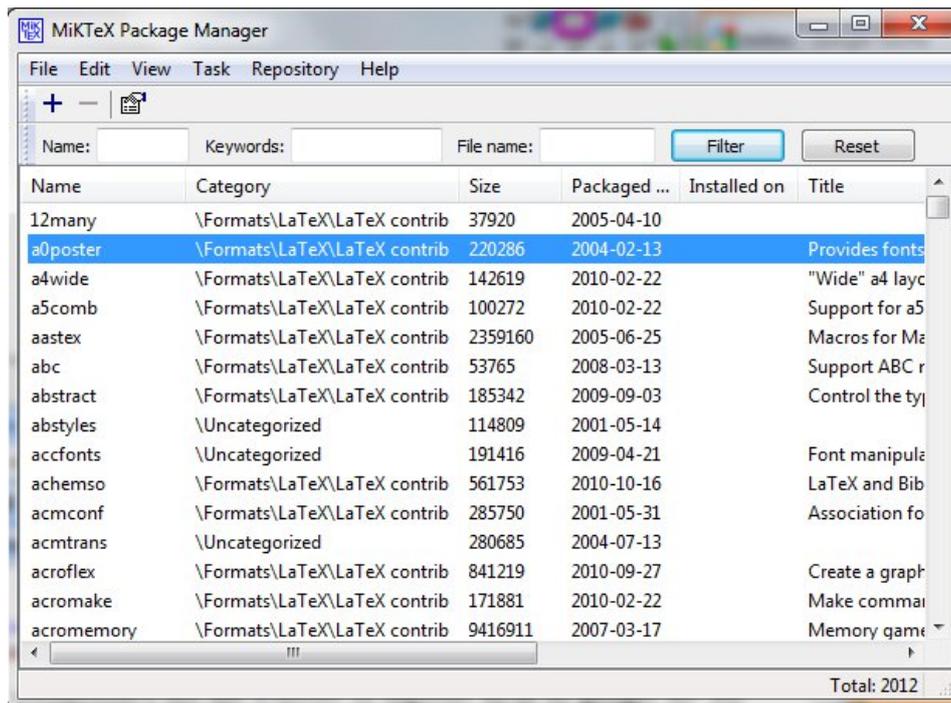


ABBILDUNG 3.1: Mit dem Package-Manager hat man Überblick über die verfügbaren und installierten Pakete.

werden farblich hervorgehoben und bei der Eingabe vorgeschlagen. Derzeit ist TeXnic-Center in der Version 2.0 Alpha 3 erhältlich.

Texmaker *Texmaker*⁸ (Abbildung 3.3) ist ein Editor, der mehrere Fenster zum Editieren oder zur Voransicht bietet. Dies vereinfacht den Schreib- und Voransichtprozess. Texmaker ist für Windows, Linux und MacOSX erhältlich.

Eclipse Ist man mit der Programmierung unter *Eclipse*⁹ vertraut, kann man das Plugin *TeXlipse*¹⁰ integrieren. Der Vorteil von TeXlipse ist eine gute Autovervollständigung und eine einfache Einbindung von Quellen und Zitaten. Eclipse ist für Windows, Linux und MacOSX erhältlich.

Kile *Kile*¹¹ ist ein unter Linuxanwendern weit verbreiteter Editor. Er bietet eine hervorragende Autovervollständigung, einfache Einbindung von Objekten, die dem Projekt

⁸<http://www.xmlmath.net/texmaker/>

⁹<http://www.eclipse.org/>

¹⁰<http://texlipse.sourceforge.net/>

¹¹<http://kile.sourceforge.net>

3 Was wird benötigt zum Arbeiten mit L^AT_EX

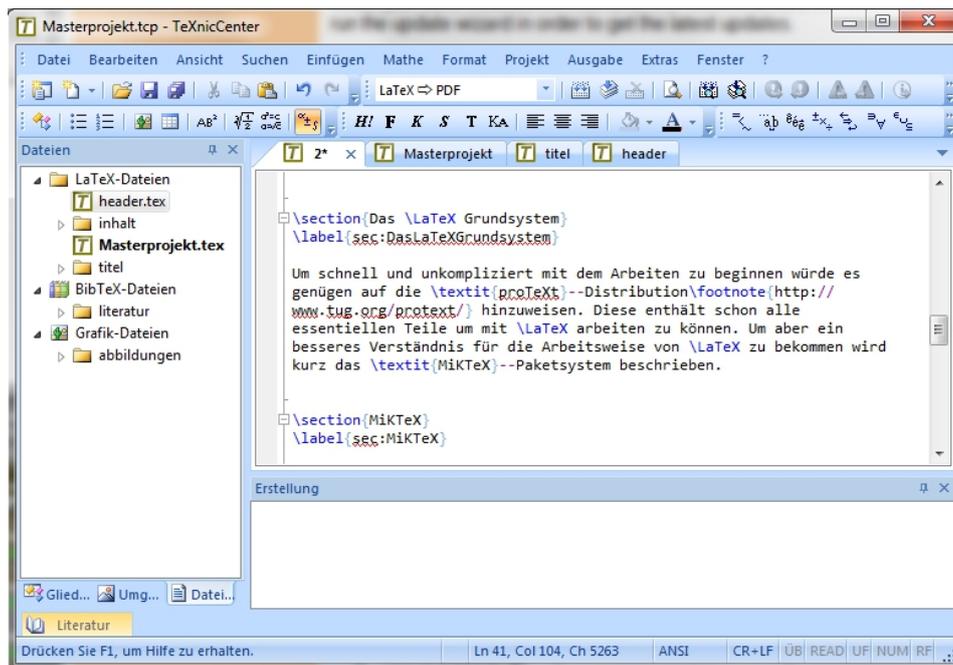


ABBILDUNG 3.2: Das TeXnicCenter ist einer der meist verwendeten Editoren.

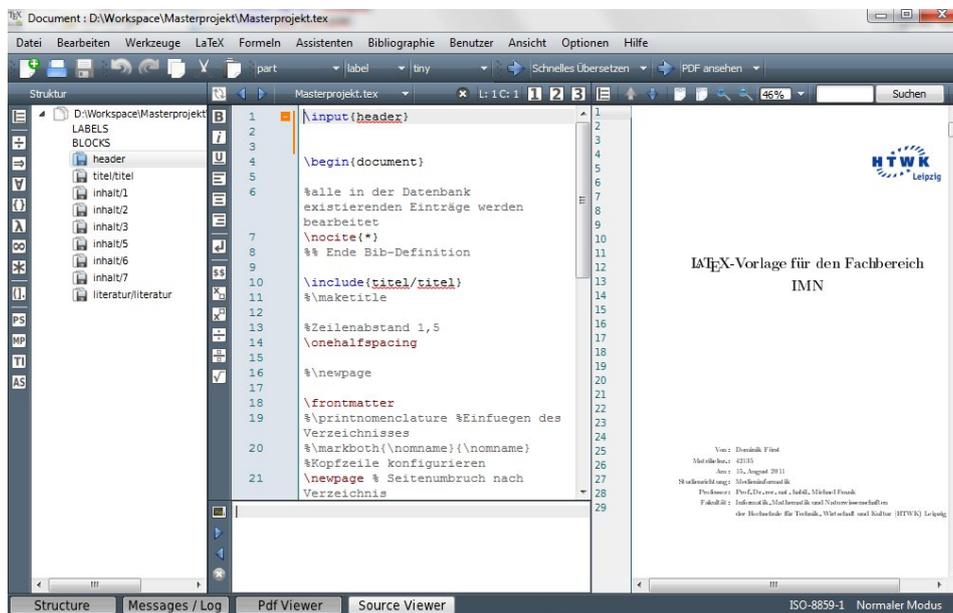


ABBILDUNG 3.3: Texmaker bietet eine Quellcodeansicht und ein Lesefenster für das erstellte Dokument

zugehören, und besitzt ein umfangreiches Archiv an Symbolen, welche per Mausklick hinzugefügt werden können.

TeXShop *TeXShop*¹² ist ein für MacOS entwickelter Editor, der durch die ins System integrierte PDF-Unterstützung direkte Wechsel zwischen Quellcode- und PDF-Ansicht ermöglicht. So erscheint im PDF stets die Position, an der sich der Cursor im Quellcode befindet und umgekehrt.

LyX Ein Editor, der auf WYSIWYG-Basis arbeitet ist *LyX*¹³. LyX vereint die Vorteile eines Textsatzsystems mit der Editierbarkeit eines Textverarbeitungsprogramms. Weiterhin ist in LyX eine umfangreiche Bibliothek von Vorlagen integriert und er ist für Windows, Linux und MacOSX verfügbar.

3.4 Helfer für Quellen und Zitate

L^AT_EX zeichnet sich auch durch die Einbindung von Quellen aus. Mit einem erstellten Kürzel für eine erstellte Quelle kann man diese im Text einfach mit dem Befehl `\cite{'Quelle'}` aufrufen. Danach erscheint sowohl im Text die Quellenangabe im gewünschten, zuvor definierten Format, als auch der Eintrag im Quellenverzeichnis. Dazu verwendet L^AT_EX das Format *BibTeX*, welches die Quellenangaben in einer separaten Datei verwaltet. Um umfangreiche Quellensammlungen zu verwalten, gibt es verschiedene Programme, welche das Sammeln erleichtern und die spezifischen Angaben nach Quellentypen kategorisieren. Außerdem erleichtern sie das Editieren der Angaben. Im Folgenden werden einige dieser Programme kurz beschrieben.

Zotero Um Quellen aus dem Internet samt Zusatzinformationen in einen Datensatz zu konvertieren, bietet sich *Zotero*¹⁴ (Abbildung 3.4 auf Seite 8) an. Das Programm arbeitet als Plugin im *Firefox* oder als eigenständiges Programm. Von vielen Bibliothekswebseiten oder auch von namhaften Internetquellen findet Zotero automatisch Angaben wie ISBN-Nummer, Autor oder Erscheinungsjahr. Die Einträge lassen sich händisch vervollständigen und mit einem Plugin in *BibTeX* exportieren.

¹²<http://www.apfelwiki.de/Main/TeXShop>

¹³<http://www.lyx.org/WebDe.Home>

¹⁴<http://www.zotero.org/>

3 Was wird benötigt zum Arbeiten mit L^AT_EX

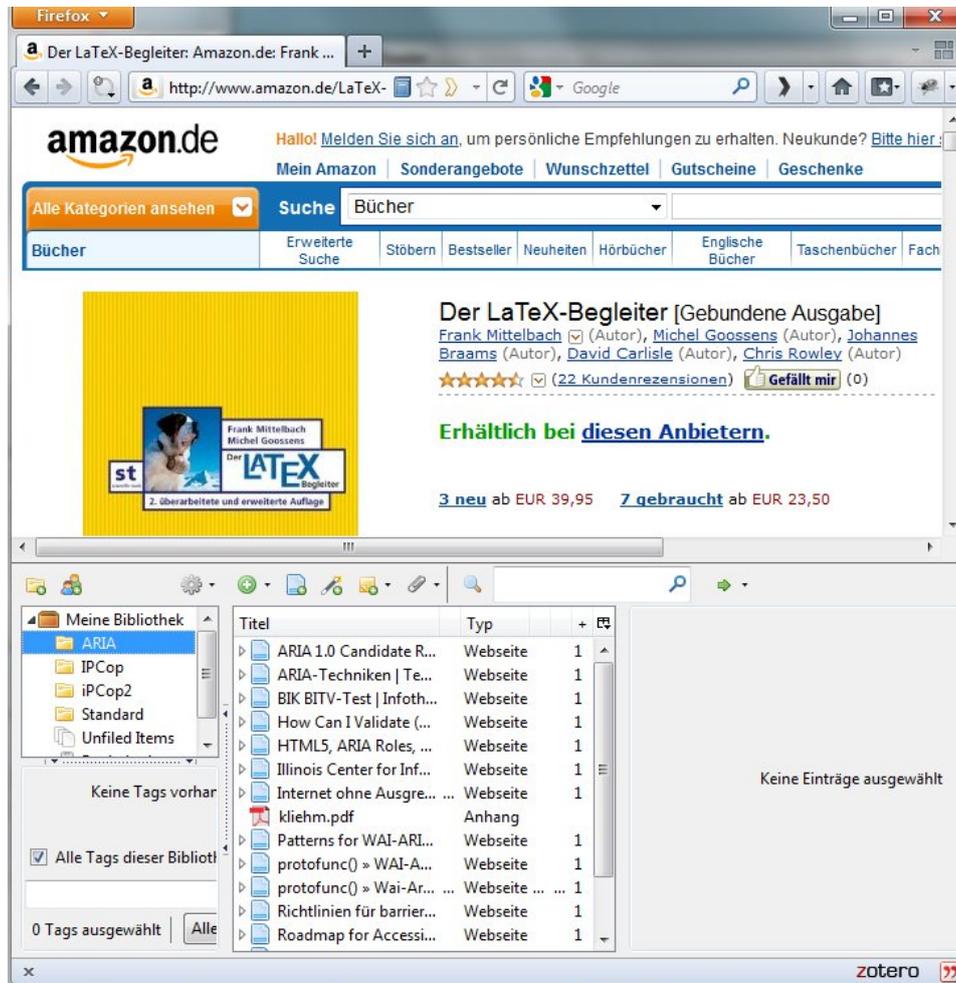


ABBILDUNG 3.4: Zotero ist ein Plugin für den Browser und kann auf einigen Webseiten auch Literaturangaben extrahieren

3 Was wird benötigt zum Arbeiten mit L^AT_EX

JabRef *JabRef*¹⁵ ist ein Literaturverwaltungsprogramm auf Basis von *BibTeX*. Es ermöglicht den Import von Daten aus wissenschaftlichen Online-Datenbanken wie *EndNote* oder *MEDLINE* und ein übersichtliches Editieren von Datensätzen. Außerdem bietet es Exportschnittstellen zu vielen Formaten. Durch die Implementierung in Java läuft es auf Windows, Linux und MacOSX.

Citavi Mit *Citavi*¹⁶ (Abbildung 3.5) ist es möglich direkt in wissenschaftlichen Online-Bibliotheken, Bibliotheks- und Buchhandelskatalogen zu recherchieren. Weiterhin kann es PDF- und Worddateien importieren und die Metadaten auslesen. Das mitinstallierte Browserplugin erkennt ISBN-Nummern auf Webseiten und kann die zugehörigen Daten bei Anwahl importieren. Außerdem kann es Textstellen und Bilder aus Dokumenten als Zitate verwalten und diese automatisch mit entsprechenden Daten versehen. Das Programm ist in der kostenlosen Version limitiert auf 100 Einträge pro Projekt.

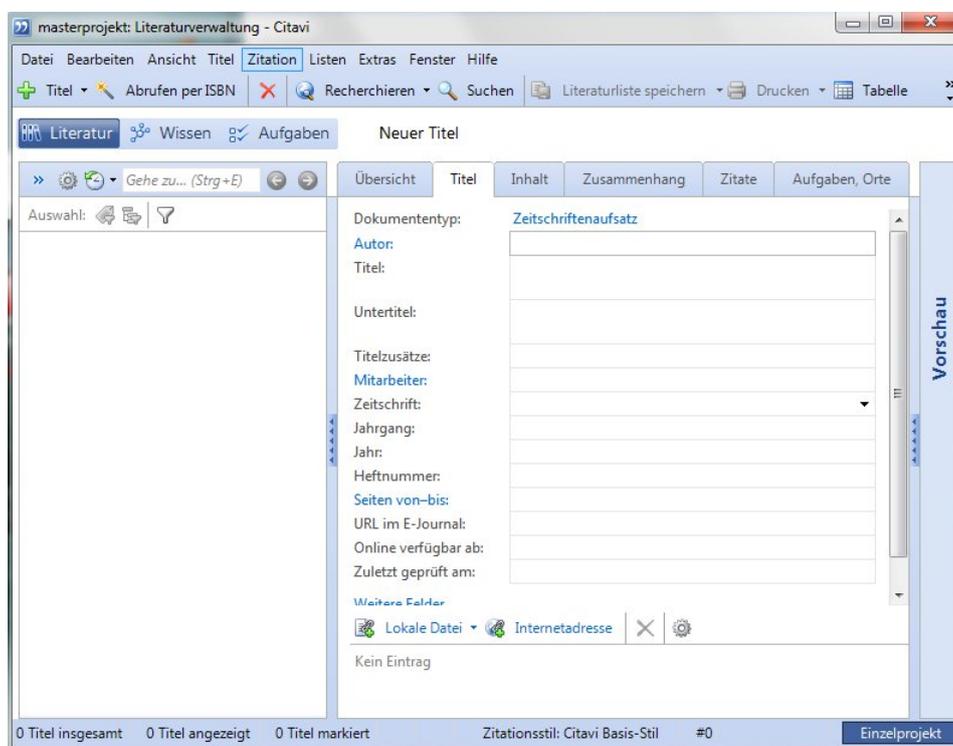


ABBILDUNG 3.5: Citavi ermöglicht die direkte Recherche in Bibliotheken

¹⁵<http://jabref.sourceforge.net/>

¹⁶<http://www.citavi.com/de/>

3.5 Helfer zur Tabellenerstellung

Da das Erstellen von Tabellen sehr komplex werden kann, bieten sich Programme an, welche den Quellcode erzeugen. Folgend werden einige dieser benannt.

Online Tabellengenerator Unter [17] findet man einen komfortablen Generator für L^AT_EX–Tabellen. Durch einfache Anwahl der gewünschten Eigenschaften und das Eintragen der Daten erzeugt dieser eine einfache Tabelle, welche dann im Quelltext ausgegeben wird.

excel2latex Um aus *Microsoft Excel* Daten in eine L^AT_EX–Tabelle zu überführen, empfiehlt sich *excel2latex*¹⁸. Dabei handelt es sich um ein Plugin, welches nach dem Markieren der zu überführenden Daten per Button ein Exportfenster öffnet. Wahlweise kann der entstandene Tabellencode kopiert werden oder als T_EX–Datei abgespeichert werden.

calc2latex Für *Open Office Calc* und *Libre Office Calc* empfiehlt sich *calc2latex*¹⁹ (Abbildung 3.6) für das Überführen der Daten in eine L^AT_EX–Tabelle. Es arbeitet als Makro und erzeugt aus den im Programm markierten Daten und einem Menü für das Layout den entsprechenden L^AT_EX–Code, welcher dann kopiert werden kann.

¹⁷<http://truben.no/latex/table/>

¹⁸<http://selinap.com/2008/05/create-latex-table-easily/>

¹⁹<http://calc2latex.sourceforge.net/>

3 Was wird benötigt zum Arbeiten mit $\text{L}^{\text{T}}\text{E}^{\text{X}}$

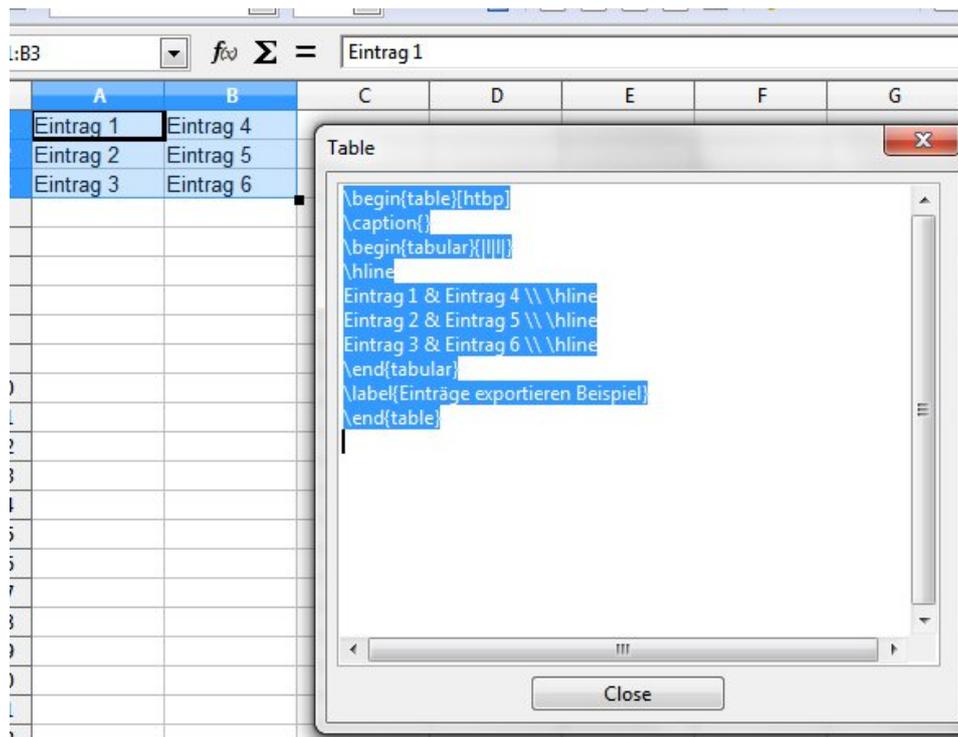


ABBILDUNG 3.6: Mit dem Macro `calc2latex` kann man markierte Tabellenteile in $\text{L}^{\text{T}}\text{E}^{\text{X}}$ -Code exportieren.

4 Arbeitsumgebung

Umfangreiche Arbeiten beinhalten meist viel Material in Form von Bildern oder externen Dokumenten, und lange Textpassagen. Um den Überblick zu behalten ist es wichtig, eine klare Struktur einzuhalten. Die Befehle `\include{Datei}` und `\input{Datei}` ermöglichen es, Textpassagen wie einzelne Kapitel, Tabellen oder umfangreiche Listen in eigene Dateien auszulagern. Erst beim Erstellen des Dokumentes werden diese wieder zusammengefügt. Ebenso können Formatpassagen aus dem Hauptdokument in eine separate Datei überführt werden, welche man bei einem anderen Projekt wiederverwenden kann. In diesem Kapitel wird zunächst auf die Datei- und Ordnerstruktur eingegangen, die der Arbeit beiliegt. Diese ist keinesfalls bindend und kann an die eigenen Bedürfnisse angepasst werden. Weiter wird noch auf Dateien eingegangen, welche Formatangaben enthalten, sowie auf die zentrale *Masterdatei*.

4.1 Dateistruktur

Wegen der Möglichkeit die Arbeit in mehrere Dateien aufteilen zu können, ist es wichtig hierbei den Überblick zu bewahren. Zu den selbst erzeugten Dateien kommen je nach verwendetem Editor automatisch erzeugte hinzu.

Jede Arbeit hat als Ursprung eine Masterdatei. Von dieser ausgehend sind sowohl Formatanweisungen als auch Textbestandteile eingebunden. Alle verwendeten Pfade gehen von dieser Datei aus. In dieser Vorlage ist die Masterdatei mit 'Arbeit.tex' benannt.

Der Übersicht halber wurden alle weiteren Dateien thematisch in folgende Ordner ausgelagert.

Abbildungen beinhaltet alle benötigten Grafiken

Anhang für alle ausgelagerten Dateien, welche in den Anhang eingebunden werden sollen

Deckblatt beinhaltet die Version fünf aus der Arbeit von Herrn Thomann; weiterhin liegt eine Vorlage dieser Arbeit bei, welche sich für Studienarbeiten oder Projekte eignet

Header enthält Dateien für Formatanweisungen, Glossar, Abkürzungen und Metadaten

Inhalt alle für die Arbeit geschriebenen Textbestandteile sind hier zentralisiert

Literatur oft wird nur eine BibTeX-Datei verwendet, aber durch Verwendung mehrerer Programme zur Erstellung sollten ab und an Versionen erstellt werden

Dies ist nur eine Variante für eine mögliche Struktur. Durch Änderungen in der Masterdatei können die Pfade jederzeit angepasst werden.

Im Folgenden werden wichtige Dateien kurz erläutert.

4.2 Formatdateien

Die Dateien im Ordner Header beinhalten unterschiedliche Anweisungen die Aussehen und Verhalten definieren. Um Formatanweisungen zu setzen, werden zuerst die entsprechenden Pakete benötigt. Um den Überblick zu behalten, wurden diese Paketaufrufe in die Datei *Pakete.tex* ausgelagert. In der Datei *Header.tex* werden dann die Anweisungen und Einstellungen vorgenommen.

In *Metadaten.tex* werden Daten zur Person und zur Arbeit an Variablen gebunden, um sie später einer Mehrfachverwendung (z.B. im Deckblatt) zugänglich zu machen.

Glossar.tex und *Abkuerzungen.tex* enthalten die Einträge, welche im Kapitel 5.6 auf Seite 20 vorgenommen werden.

Um das spätere Editieren der Formatanweisungen zu vereinfachen, wurden diese thematisch gruppiert und mit erklärenden Kommentaren versehen. Einige Kommentare enthalten zusätzlich Links zu weiterführenden Quellen im Internet.

4.3 Masterdatei

Die Erstellung des Enddokumentes wird von L^AT_EX nach dem Aufbau des Quellcodes vorgenommen. In der Masterdatei werden deshalb zuerst die Formatanweisungen eingebunden. Diesen folgt dann die Anweisung `\begin{document}`. Zwischen dieser und der Anweisung `\end{document}` wird der zu schreibende Text eingebunden. Möchte man eine Einbindung oder eine Anweisung übergangsweise außer Kraft setzen, so genügt es

4 Arbeitsumgebung

ein '%' davor zu setzen.

Welche Bedeutung die jeweiligen Aufrufe haben, wurde in einem nachstehenden Kommentar erläutert.

5 Vorgaben des Fachbereiches IMN

Um einen einheitlichen Standard für Graduierungsarbeiten im Fachbereich IMN zu schaffen, verfasste *Mathias Thomann* 2010 sein Masterprojekt zu diesem Thema. Er sammelte Meinungen der Professoren sowie Standards anderer Hochschulen und fasste diese für die Arbeit mit *Microsoft Word* zusammen.

Auf Basis dieser Zusammenfassung und den Gegebenheiten in \LaTeX wird die Umsetzung der Rahmenvorgaben folgend erläutert.

5.1 Schrift

In der Basis arbeitet \LaTeX mit dem *T1*-Schriftsatz mit der Schrift *Computer Modern*, welche auch in diesem Dokument verwendet wird. Weitere mögliche Schriften findet man unter [1].

Grundsätzlich sollte die Schrift gut lesbar sein. Dies erfüllen bei Fließtexten vor allem Serifenschriften. Als gut lesbar gelten allgemein folgende Schriftarten:

- Arial
- Computer Modern (in der vorliegenden Arbeit verwendet)
- Times New Roman (in \LaTeX nur *Times*)

Die von Herrn Thomann zusätzlich aufgeführten Schriften *Calibri* und *Cambria* sind *True-Type*-Schriften. Diese können nur durch zusätzlichen Aufwand, wie unter [2] beschrieben, verwendet werden. Ein umfangreicher Katalog von für \LaTeX verfügbaren Schriften findet man unter [3].

Um den **Schrifttyp** zu definieren, bindet man das jeweilige Paket mit

```
\usepackage{'Schrifttyp'}
```

¹<http://www.tug.dk/FontCatalogue/>

²<http://sunsite.informatik.rwth-aachen.de/ftp/pub/mirror/ctan/help/Catalogue/entries/fontools.html>

³<http://www.tug.dk/FontCatalogue/>

ein.

Will man im Dokument oft Änderungen des Schrifttyps vornehmen, empfiehlt sich die Definition eines Kommandos wie folgt:

```
\newcommand{\changeFont}[3]{
\fontfamily{#1} \fontseries{#2} \fontshape{#3} \selectfont}
```

So kann man im Text durch den Aufruf `\changeFont{Familie }{Stärke }{Form }` die Umstellung vornehmen.⁴ Für die Computer Modern (cmr) mit den Eigenschaften *Fett* (b), *langgezogen* (x) und *kursiv* (it) sieht der Befehl folgendermaßen aus:

```
\changeFont{cmr}{bx}{it}
```

Danach sieht der Text wie folgt aus:

Computer Modern Roman bold extended italic

Die **Schriftgröße** sollte bei Verwendung von Serifenschriften 12pt betragen und bei serifenlosen 11pt. Einstellen kann man dies für das gesamte Dokument im Header unter dem Punkt `\documentclass[]`, welcher die Grundvorgaben enthält. Außerhalb dieser Vorgaben verwendet man den Befehl⁵:

```
\fontsize{'Fontgröße '}{'Grundlinienabstand '} \selectfont
```

Blockatz und **Silbentrennung** ist in L^AT_EX voreingestellt. Sollte die Trennung bei einigen Wörtern versagen, kann man sie mit dem Befehl `'\-'` (z.B.: Tren\ -nung) vorgeben.

Zum Einstellen des Zeilenabstandes benötigt man das Paket *setspace*, welches dann den Zeilenabstand mit dem Befehl `\onehalfspacing` auf 1,5 fach ändert.

Ist das im Dokument an manchen Stellen nicht gewünscht, so setzt man an der Stelle den Abstand mit `\singlespacing` auf "einfach".

5.2 Überschriften

Überschriften werden in L^AT_EX nach der Strukturebene gesetzt. Je nach Ebene beeinflusst das Setzen einer Überschrift unterschiedliche Faktoren:

⁴<http://www.math.tu-dresden.de/~rudl/latex/fonts.pdf>

⁵<http://www.latex-kurs.de/fragen/schriftgroesse.html>

- `\part{}` erzeugt eine Zwischenüberschrift auf einer leeren Seite.
- `\chapter{}` steht für die Hauptkapitel und beginnt immer auf einer neuen Seite. Nach der Überschrift wird ein längerer Abstand zum folgenden Text eingefügt.
- `\section{}` erzeugt Unterkapitel erster Ordnung. Der Abstand zum nachfolgenden Text ist geringer als bei `\chapter{}`
- `\subsection{}` erzeugt Unterkapitel zweiter Ordnung.
- `\subsubsection{}` erzeugt Unterkapitel dritter Ordnung. Die Überschrift ist in Normaltextgröße und wird standardmäßig nicht mit nummeriert. Ebenso erscheint sie standardmäßig nicht im Inhaltsverzeichnis.
- `\paragraph{}` erzeugt eine Absatzüberschrift, welche ohne Zeilenumbruch und ohne Nummerierung in den Text übergeht. Vor der Überschrift werden jedoch zwei Leerzeilen eingefügt und danach zwei Leerzeichen zum nachfolgenden Text.
- `\subparagraph{}` verhält sich in der *scrbook*-Umgebung wie `\paragraph{}`

Das Inhaltsverzeichnis wird auf Basis der Überschriften an der Stelle erzeugt, an der man den Befehl `\tableofcontents` setzt. Möchte man die Nummerierung im Inhaltsverzeichnis beeinflussen, benutzt man:

```
\setcounter{tocdepth}{Gliederungstiefe}
```

Dabei sind die Gliederungstiefen ausgehend von der Hauptkapitelüberschrift von 0 bis 5 anzugeben. Möchte man z.B. bis in die `\paragraph{}`-Ebene durchnummerieren, so lautet der Befehl:

```
\setcounter{tocdepth}{4}
```

Soll eine einzelne Überschrift nicht im Inhaltsverzeichnis erscheinen, fügt man der Ebenenbezeichnung ein `*` hinzu (z.B.: `\section*{}`). Die Überschrift erhält demzufolge auch keine Nummerierung.

5.3 Absätze

Einen Absatz erzeugt man in \LaTeX , indem man eine Zeile im Quelltext frei lässt oder den Befehl `\par` einsetzt. Um die Einrückweite zu beeinflussen, benutzt man:

```
\setlength{\parindent}{'Einrückweite'}
```

Soll der Absatzabstand eine bestimmte Größe haben, so beeinflusst man diesen mit:

```
\setlength{\parskip}{'Absatzabstand'}
```

Soll der Abstand genau eine Zeile betragen, setzt man als Absatzabstand `\baselineskip` ein. Bei Einsatz des Befehls `\parskip` wird automatisch das Einrücken entfernt. ^[6]

5.4 Abbildungen

Das Einsetzen einer Grafik funktioniert in den meisten Editoren mit Hilfe eines Assistenten. Dieser erzeugt dann im Quelltext einen Eintrag:

```
\begin{figure}[htbp]
  \centering
  \includegraphics[width=0.90\textwidth]{'Pfad/Datei.jpg'}
  \caption{Bildbeschriftung}
  \label{fig:Dateiname}
\end{figure}
```

Die Umgebung `figure` sorgt für die Positionierung des Bildes. Die Angaben in `'[]'` spezifizieren die Möglichkeiten der Positionierung. Dabei steht `'h'` für die Stelle an der die Abbildung definiert wurde, `'t'` für den oberen Rand der Seite, `'b'` für den unteren Rand und `'p'` für eine eigene Seite. Bei der Angabe mehrerer Attribute wählt \LaTeX die Möglichkeit mit der besten Platzeinteilung.

Der Befehl `\centering` zentriert das Bild in Bezug zur horizontalen Textmitte. Für eine Ausrichtung auf der rechten Seite benutzt man `\flushright` und für die linke `\flushleft`.

Das eigentliche Bild wird mit dem Befehl `\includegraphics['Größe']{'Datei'}` aufgerufen. Die Bildgröße kann man entweder in der Höhe (`height`) oder der Breite (`width`) bestimmen. Der jeweils andere Wert wird automatisch angepasst.

Die Angabe `width=0.90\textwidth` bewirkt zum Beispiel, dass das Bild auf 90% der Textbreite skaliert wird.

Mit `\caption{}` bekommt das Bild einen Beschriftungstext. Diesem wird die Bezeichnung *Abbildung* vorangesetzt, gefolgt von der Nummer des Hauptkapitels und einer

⁶<http://fff2.at/drupal/content/absaetze-in-latex>

fortlaufenden Nummerierung der Bilder. Soll sich der Text im Abbildungsverzeichnis von dem unter dem Bild unterscheiden, so hängt man einen zweiten Eintrag wie folgt an:

```
\caption[Text im Abbildungsverzeichnis]{Text unter Bild}
```

Um das Bild referenzieren zu können, wird noch ein `\label{fig:Dateiname}` gesetzt. Die optionale Bezeichnung `fig:` gibt der jeweiligen Referenz den Typ des Verweises mit und ist hilfreich, wenn man mehrere verschiedenartige Objekte mit dem gleichen Label versehen will (z.B. ein Bild `LATEX` und eine Tabelle `LATEX`). Um das Bild zu referenzieren, genügt es `\ref{fig:Dateiname}` an die entsprechende Stelle zu setzen. Damit wird auch bei Änderungen am Dokument automatisch die genaue Bezeichnung ermittelt. Um die Position des Bildes im Dokument anzuzeigen, benutzt man den Befehl `\pageref{fig:Dateiname}`.

5.5 Tabellen

Das Erstellen von einfachen Tabellen in `LATEX` ist je nach Zielstellung unterschiedlich möglich. Alle Möglichkeiten der Tabellenerstellung zu erläutern, würde den Rahmen dieser Arbeit sprengen. Deshalb wird hier nur auf die geläufigste Umgebung für Tabellen eingegangen. Diese nennt sich *tabular*. Je nach Editor kann man diese automatisch erstellen. Im Falle des TeXnicCenter wird über das Menü folgendes Grundgerüst erzeugt:

```
\begin{table}[h]
    \centering
        \begin{tabular}

            \end{tabular}
        \caption{tabellenbeispiel}
        \label{tab:tabellenbeispiel}
\end{table}
```

Die Einträge für Positionierung, Beschriftung und Referenzierung sind analog zu den Abbildungen von Kapitel 5.4.

Zwischen `\begin{tabular}` und `\end{tabular}` werden die Daten eingefügt. Die Spalten werden durch ein `'&'` getrennt und am Ende der Zeile wird durch `'\\'` in die nächste umgebrochen. Die Ausrichtung der Daten in den Zellen gibt man als Argument hinter `\begin{tabular}` in geschweiften Klammern an. Tabelle 5.1 zeigt die erstellte Tabelle (rechts) und den zugrundeliegenden Quelltext (links):

```

\begin{tabular}{lcr}
    links&zentriert&rechts\\
    l&c&r\\
\end{tabular}

```

TABELLE 5.1: Tabelle mit Zellenausrichtung

Um nun der Tabelle noch Linien hinzuzufügen, setzt man für horizontale Linien ein `\hline` an die Stelle, an der sie in der fertigen Tabelle stehen soll. Für vertikale Linien fügt man zwischen den Positionierungsangaben ein `|` ein. Tabelle 5.2 zeigt ein Beispiel mit Linien.

```

\begin{tabular}{|l|c|r|}
    \hline
    links&zentriert&rechts\\
    \hline \hline
    l&c&r\\
    \hline
    1&2&3\\
    \hline
\end{tabular}

```

links	zentriert	rechts
l	c	r
1	2	3

TABELLE 5.2: Tabelle mit Linien

Zusätzliche Formatanweisungen für Tabellen sind ausführlich unter [7] beschrieben. Weitere Möglichkeiten zur Erstellung von Tabellen und den entsprechenden Umgebungen findet man unter [8].

5.6 Abkürzungen und Glossar

Die Verwendung von Abkürzungen wurde von Herrn Thomann beschrieben. In der zur Arbeit zugehörigen Header-Datei ist die Verwendung eines Abkürzungsverzeichnisses vorbereitet. Um eine Abkürzung hinzuzufügen erzeugt man in der Datei *Header* → *Abkuerzungen.tex* wie folgt einen neuen Eintrag.

```

\newacronym{ 'Kuerzel '}{ 'Kuerzel '}{ 'ausgeschriebenes Wort ' }

```

⁷<http://www2.informatik.hu-berlin.de/~piefel/LaTeX-PS/V05-tabellen.pdf>

⁸<http://www2.informatik.hu-berlin.de/~piefel/LaTeX-PS/Archive-2004/V04-tab2/>

Für den Eintrag "Compact Disk" sieht das folgendermaßen aus:

```
\newacronym{CD}{CD}{Compact Disc}
```

Im laufenden Text muss nun der Kürzelaufruf `\gls{CD}` erfolgen und das Wort wird mit Abkürzung eingesetzt, hier wäre das *Compact Disc (CD)*.

Zur Erzeugung des Abkürzungsverzeichnisses muss vor dem Erstellen *Makeindex* ausgeführt werden. In den Einstellungen der Editoren sollte für den Punkt Makeindex folgender Parameter gesetzt werden:

```
-s %tm.ist -t %tm.alg -o %tm.acr %tm.acn
```

Beim TeXnicCenter genügt es diese Zeile einzutragen, beim Texmaker muss man noch den Programmaufruf vorausstellen:

```
makeindex -s %tm.ist -t %tm.alg -o %tm.acr %tm.acn
```

Zusätzlich zum Abkürzungsverzeichnis bietet sich ein **Glossar** an. Hierfür gibt es die Datei *Header* → *Glossar*. Einträge werden hier in folgender Form vorgenommen:

```
\newglossaryentry{glos:'Kuerzel'}{name='ausgeschriebenes Wort',  
description={'Beschreibung des Wortes'}}
```

Für den Beispielbegriff "Antwortdatei" sieht dies folgendermaßen aus:

```
\newglossaryentry{glos:AntwD}{name=Antwortdatei,  
description={Informationen zum Installieren einer Anwendung  
oder des Betriebssystems.}}
```

Im Text würde dieses Kürzel mit `\gls{glos:AntwD}` eingebunden werden. Um Makeindex zur Erzeugung zu nutzen, muss wiederum ein Eintrag in den Parametern des Programmes vorgenommen werden:

```
-s %tm.ist -t %tm.glg -o %tm.gls %tm.glo
```

Beim TeXnicCenter genügt es diese Zeile einzutragen, beim Texmaker muss man noch den Programmaufruf davor setzen:

```
makeindex -s %tm.ist -t %tm.glg -o %tm.gls %tm.glo
```

Weil sich die Einstellungen unterscheiden und in den Standardeinstellungen nur eine Einstellung vorgenommen werden kann, bietet TeXnicCenter (siehe Abbildung 5.1) unter *Ausgabe* (1) → *Ausgabeprofile definieren* → *Nachbearbeitung* (2) die Möglichkeit *Postprozessoren* (3) zu definieren. Hier erzeugt man jeweils für Abkürzung und Glossar einen neuen Eintrag (4) mit dem Pfad zu Makeindex (5) und den oben gezeigten Parametern (6). So kann Makeindex die Listen nacheinander abarbeiten. Es ist möglich weitere Glossare zu erzeugen, bspw. ein Symbolverzeichnis. Näheres hierzu findet man unter [9].

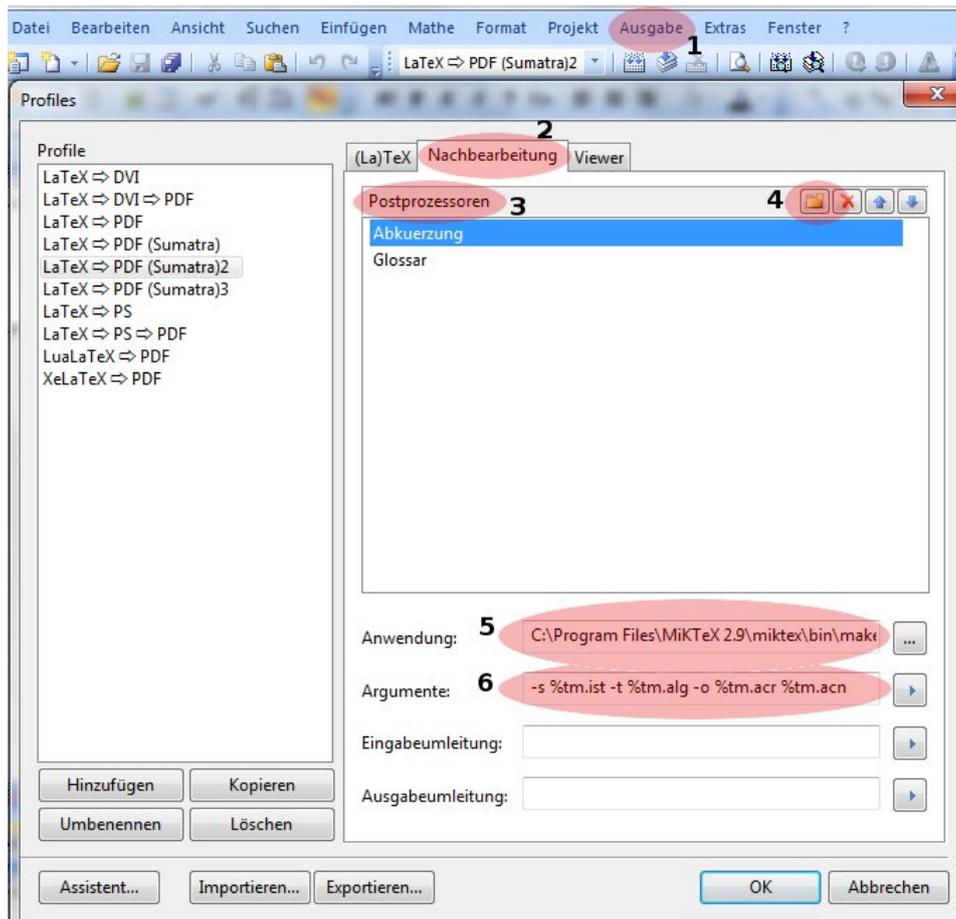


ABBILDUNG 5.1: Parametereinträge für Makeindex zur Erzeugung von Abkürzungsverzeichnis und Glossar

In der Dokumentation¹⁰ des Paketes *glossaries* findet man weitere Informationen zu Verzeichnistypen und Parametern.

⁹<http://ewus.de/tipp-1029.html>

¹⁰<http://dante.ctan.org/tex-archive/help/Catalogue/entries/glossaries.html>

5.7 Zitate

Das Einfügen von Zitaten wird in L^AT_EX je nach Länge des Zitates mit unterschiedlichen Befehlen bewerkstelligt. Für Zitate die kürzer als ein Absatz sind, benutzt man den Befehl *quote* wie folgt:

```
\begin{quote}
Ich bin ein kurzes Zitat, welches nicht länger als ein Absatz ist.
\end{quote}
```

Im Text erscheint dies folgendermaßen:

Ich bin ein kurzes Zitat, welches nicht länger als ein Absatz ist.

Ist die zitierte Textpassage länger als ein Absatz, verwendet man *quotation*.

```
\begin{quotation}
Zitat....
\end{quotation}
```

Bei *quote* wird eine Leerzeile vor dem Zitat eingefügt und der Text wird links und rechts eingerückt. Bei *quotation* wird zusätzlich der erste Satz jedes Absatzes eingerückt.

Quellenangaben für Zitate sind in L^AT_EX mit *BibTeX* möglich. Dabei wird entweder manuell eine *.bib*-Datei erzeugt oder mittels eines der unter Kapitel 3.4 genannten Programme. Im Text wird ein Quellenverweis mit `\cite{Quellenkürzel}` gesetzt.

Soll das Quellenverzeichnis einen geringen Umfang haben, besteht die Möglichkeit dies mit folgender Umgebung innerhalb des L^AT_EX-Dokuments zu realisieren:

```
\begin{thebibliography}
    \bibitem ...
    \bibitem ...
\end{thebibliography}
```

Ein Eintrag hierin würde wie folgt vorgenommen werden:

```
\bibitem[Kürzel123] Author, Verlag, etc.
```

Weitere Parameter und Beispiele findet man unter ^[1].

¹<http://janeden.net/bibliographien-mit-latex>

5.8 Fußnoten

Um Informationen in Fußnoten zu setzen, trägt man die entsprechende Information in ein `\footnote{Fußnotentext}` ein. Die Nummerierung wird automatisch vorgenommen. Möchte man eine Fußnote von anderer Stelle referenzieren, so arbeitet man ein `\label{}` mit ein. Dies kann dann folgendermaßen aussehen:

```
\footnote{\label{foot:Bezeichner} Fußnotentext}
```

So kann mittels `\ref{Bezeichner}` oder `\pageref{Bezeichner}` darauf verwiesen werden.

5.9 Seitenzahlen

In der Klasse *scrbook* wird automatisch bei einseitigen Texten die Seitennummerierung am unteren Rand zentriert gesetzt. Bei zweiseitigen am jeweils äußeren Rand. Die Nummerierung zwischen Deckblatt und erster Textseite erfolgt römisch und ab der ersten Textseite arabisch. Soll nach dem Text eine andere Nummerierung vorgenommen werden, so setzt man nach der letzten Textzeile den Befehl:

```
\pagenumbering{Nummerierungstyp}
```

Der Befehl setzt automatisch den Zähler der Seitenzahl zurück. Typen können hier sein Alph (A,B,C), alph (a,b,c), Roman (I,II,III), roman (i,ii,iii) und arabic (1,2,3).

5.10 Aufzählungen und Nummerierungen (Listen)

Zum Erstellen von Listen gibt es in \LaTeX drei unterschiedliche Befehle. Um ein einfaches Zeichen(z.B. \bullet , -, +) als Listenpunkt zu setzen, benutzt man die Umgebung *itemize*. Im Standard wird hier für die erste Ebene nur ein Punkt vor den Listenpunkt gesetzt und für die zweite ein Strich. Soll ein anderes Zeichen verwendet werden, so gibt man dies mit eckigen Klammern nach dem Punkt `\item` an. Dies kann dann wie in Tabelle 5.3 auf Seite 27 aussehen.

Um eine nummerierte Liste zu erzeugen, verwendet man *enumerate*. Tabelle 5.4 auf Seite 27 zeigt ein Beispiel.

Die dritte Möglichkeit ist *description*. Hier übergibt man nach `\item` einen vorangehenden bezeichnenden String wie in Tabelle 5.5 auf Seite 27 gezeigt wird. Hinweise und

Beispiele für weitere Formatierungen von Listen findet man unter ^[12].

5.11 Ränder

Wünscht man eine andere Breite der Seitenränder, so fügt man in den Header des Dokuments den Befehl `\usepackage[...]{geometry}` hinzu. In den eckigen Klammern stellt man die Seitengröße und die Abstände des Textes ein. Dies sieht für eine A5-Seite mit 2 cm breiten Rändern wie folgt aus:

```
\usepackage[a5paper,left=2.0cm,right=2.0cm,top=2.0cm,bottom=2.0cm]
{geometry}
```

5.12 Eingebettete Objekte

Das Einbetten von Objekten, wie Bilder oder gesonderte Textpassagen, ist mit *minipage* möglich. Hiermit wird gleichzeitig eine isolierte Umgebung erzeugt. Formatangaben die innerhalb von *minipage* erzeugt werden, bleiben auch nur darin gültig. Des weiteren können Position und Größe beeinflusst und mehrere *minipages* nebeneinander positioniert werden.

```
\begin{minipage}[Position]{Breite}
    Beispieltext
\end{minipage}
```

Folgendes Beispiel zeigt zwei nebeneinander erzeugte *minipages*, eine mit Schrift und eine mit Bild. Zur Verdeutlichung werden beide mit einem Rahmen umschlossen. Das Resultat sieht man in Tabelle 5.6 auf Seite 27.

```
\begin{table}
    \fbox{
        \begin{minipage}[c]{0.4\textwidth}
            \textbf{{\LARGE Logo der HTWK Leipzig}}
        \end{minipage}
    }
    \hfill
    \fbox{
```

¹²<http://www.uni-giessen.de/hrz/tex/cookbook/ffth.html>

5 Vorgaben des Fachbereiches IMN

```
\begin{minipage}[c]{0.4\textwidth}
    \includegraphics{abbildungen/HTWK-Logo-gross}
\end{minipage}
}
\caption[minipage--Umgebung]{Zwei nebeneinander stehende minipage.}
\label{tab:minipage}
\end{table}
```

Weiterhin lassen sich minipages verschachteln. So sind auch komplexere Formatierungen möglich. Weitere Informationen findet man unter ^[13].

¹³<http://www.golatex.de/wiki/index.php?title=minipage>

<code>\begin{itemize}</code>	
<code>\item erster Listenpunkt</code>	• erster Listenpunkt
<code>\begin{itemize}</code>	– eins tiefer
<code>\item eins tiefer</code>	
<code>\end{itemize}</code>	
<code>\item[+] mit Plus</code>	+ mit Plus
<code>\item[@] mit at</code>	@ mit at
<code>\end{itemize}</code>	

TABELLE 5.3: Einfache Liste

<code>\begin{enumerate}</code>	
<code>\item erster Eintrag</code>	1. erster Eintrag
<code>\begin{enumerate}</code>	a) Untereintrag
<code>\item Untereintrag</code>	
<code>\end{enumerate}</code>	
<code>\item zweiter Eintrag</code>	2. zweiter Eintrag
<code>\item dritter Eintrag</code>	3. dritter Eintrag
<code>\end{enumerate}</code>	

TABELLE 5.4: Nummerierte Liste

<code>\begin{description}</code>	
<code>\item[erster] erster Eintrag</code>	erster erster Eintrag
<code>\item[zweiter] zweiter Eintrag</code>	zweiter zweiter Eintrag
<code>\item[dritter] dritter Eintrag</code>	dritter dritter Eintrag
<code>\end{description}</code>	

TABELLE 5.5: Liste mit Bezeichner

Logo der HTWK
Leipzig



TABELLE 5.6: Zwei nebeneinander stehende minipage.

6 Aussicht

Mit dieser Arbeit und der beiliegenden Vorlage wird ein Grundstein gelegt, um eine Arbeit in \LaTeX zu erstellen. Die von Herrn Thomann erläuterten Formatvorgaben wurden für die \LaTeX -Umgebung adaptiert und beschrieben. Jedoch werden eigene Vorstellungen und zusätzliche Vorschläge von Professoren einige Arbeit an Recherche und Versuchen bedürfen. Aufgrund der Komplexität von \LaTeX konnten nicht alle Möglichkeiten beschrieben werden, Fragestellungen in \LaTeX umzusetzen. Jedoch wurden lauffähige Vorschläge unterbreitet, um die gängigsten Vorgaben zu Layout und einzubindenden Objekten wie Abbildungen und Tabellen zu realisieren.

Weiterhin sind sowohl die \LaTeX -Pakete als auch die verwendeten Umgebungen einem ständigen Entwicklungsprozess unterworfen. Für fachspezifische Umgebungen wie mathematische Formeln, Quellcode oder \LaTeX -generierte Grafiken könnten in Zukunft weitere Ergänzungen folgen.

Außer den im Text genannten Quellen, kommen einige Bücher als Arbeitsbegleiter in Frage. Da es einen großen Fundus an Büchern zu \LaTeX gibt, verweise ich auf die Bibliothek der HTWK und auf eine Suchanfrage im Internet. Für Einsteiger eignet sich *\LaTeX -Einführung Band 1*¹ von *Helmut Kopka*. Ein sehr umfangreiches Werk ist *The LaTeX Companion*² mit detaillierten Hinweisen zu Unterschieden zwischen Paketen und ihren Eigenheiten. Als besondere Buchempfehlung möchte ich auf das Buch *Der \LaTeX -Begleiter*³ hinweisen, da es einen vollständigen und sehr detaillierten Einblick in die Arbeit mit \LaTeX und allen fachspezifischen Umgebungen darstellt. Es eignet sich sowohl für den Einstieg als auch bei der fortgeschrittenen Benutzung von \LaTeX .

¹<http://www.amazon.de/Latex-Einführung-Band-Inklusive-CD-ROM/dp/3827315573>

²<http://www.amazon.de/LaTeX-Companion-Addison-Wesley-Techniques-Computer/dp/0201362996>

³<http://www.amazon.de/LaTeX-Begleiter-Frank-Mittelbach/dp/382737166X>