

LaTeX – Textverarbeitung auf elegante Art

Eine Einführung von ubiwahn und Warg

L^AT_EX

Textbearbeitungsprogramme auf Rechnern gibt es zur Genüge! Durchgesetzt haben sich im alltäglichen Gebrauch die sogenannten WYSIWYG-Programme, wie MS Word oder OpenOffice.org. Doch gerade im wissenschaftlichen Bereich zeigen diese Programme Schwächen auf, bzw. verlangsamen die Arbeit durch das umständliche Rumgeklicke in Formeditoren. Außerdem wurden die Dokumente, zumindest in den älteren Versionen, ab einer stattliche Größe etwas instabil. Als Student im naturwissenschaftlichen Spektrum angesiedelt, trifft man aber schnell auf Abhilfe: LaTeX. Beim Lesen dieses Wortes denken natürlich nur die wenigsten an ein Textverarbeitungsprogramm. Es wird allerdings auch nicht wie der allseits beliebte Kunststoff ausgesprochen.

Ein wenig Geschichte

LaTeX beruht selbst auf dem Textsatzsystem TeX, welches von Donald E. Knuth entwickelt wurde. Ausgesprochen wird TeX wie beispielsweise „Technik“, da das X das griechische „Chi“ (χ) symbolisiert. „T“ und „e“ stehen für die griechischen Buchstaben „tau“ (τ) und „epsilon“ (ϵ), die aber der deutschen bzw. lateinischen Aussprache ähnlich sind. Nachdem wir diese Feinheiten geklärt hätten, die euch in Gesprächen nicht als blutigen Anfänger auflaufen lassen, weiter im Text.

LaTeX ist eigentlich „nur“ ein Satz von Makros, die die Bedienung und Dokumenterstellung mit TeX vereinfachen. Entwickelt wurde dieses System von Leslie Lamport, daher auch der Name: Lamport TeX also LaTeX.

TeX und LaTeX gelten als abgeschlossene Programme. Es werden daher nur noch Fehlerkorrekturen vorgenommen. Von nun an begegnet man einer ganzen Reihe von Hilfsprogrammen, Editoren, etc Wir wollen im Folgenden eine kleine Einführung geben, mit der ihr fähig sein solltet, kleine Dokumente zu erstellen. Wer sich tiefer mit der Materie beschäftigen will, der findet problemlos Literatur im Internet oder im Buchhandel. Eine kleine Übersicht empfehlenswerter Literatur findet ihr am Ende des Artikels.

Die Software

Als Erstes braucht man eine LaTeX Distribution, die die Compiler und eine Menge von Zusatzpaketen beinhaltet. Unter Windows ist MikTeX am häufigsten (www.miktex.org), daher wahrscheinlich auch das bekannteste Projekt.

TeX ist die Distribution, die unter Linux am häufigsten angetroffen wird. Diese wird auch bei der Mandriva Linux Installation in der Regel automatisch mitinstalliert, bzw. findet man sie in den main-Quellen der FTP-Server. Unbedingt sollte auch die TeTeX-Dokumentation installiert werden, da diese neben der Beschreibung der Pakete oft auch Vorlagen oder Beispiele zu diesen beinhaltet und uns somit das Leben in vielen Fällen erleichtert.

Von Nutzen für diesen Artikel ist auch noch das RPM-Paket LaTeX-Beamer, worauf wir später nochmal gezielt eingehen. Dieses befindet sich in den Contrib-Quellen.

Die Wahl des Editors ist das A und O, wenn man mit LaTeX arbeitet. Hier gibt es einige empfehlenswerte Programme, kommerzielle oder Freeware. Unter Windows sind dabei die wichtigsten Vertreter Texnicenter (www.toolscenter.org) oder LaTeX-editor (www.latexeditor.org). Winedt ist ein kommerzieller Editor für Windows (www.winedt.com). Unter Linux lässt sich TeX-Code in jedem beliebigen Editor (Kate, Kwrite, Gedit, emacs, etc ...) schreiben. Der bequemste und für unsere Zwecke auch programmierte Editor nennt sich Kile (<http://kile.sourceforge.net/> oder ebenfalls aus den contrib-Quellen installierbar). Ihr seht schon, alleine mit der Mandriva Free DVD lässt sich direkt mit TeX loslegen!

Arbeiten mit Kile



Wie schon erwähnt, kann LaTeX mit einem einfachen Texteditor geschrieben werden, und mittels Übersetzungs-Befehlen nach verschiedenen Ausgabeformaten wie PDF oder DVI übersetzt werden. Sobald eine Datei mit der Endung *.tex versehen wird, unterstützen die meisten Texteditoren farbliche Syntax-Hervorhebung.

Frontends wie Kile (oder auch Texmaker) übernehmen einen gewissen Teil an Arbeit, und erleichtern somit den Schreibvorgang entscheidend. Grundsätzlich dient das Frontend nur als „grafische“ Oberfläche und bedient sich dahinter der LaTeX-Module.

Ganz ohne LaTeX-Kenntnisse ein Dokument zu schreiben, ist selbst mit einem Frontend schwierig. Allerdings vereinfacht eine grafische Oberfläche den Einstieg ungemein, und bietet einen hervorragenden Ausgangspunkt, sich mit der Materie näher zu befassen.

Aufgrund der diversen Angebote für Frontends sollte zwecks Kompatibilität zu TeX und LaTeX darauf geachtet werden, dass softwareseitig eine saubere Syntax verwendet wird. Es gibt viele Editoren, die LaTeX-Scripte einbinden und mit eigenen Ergänzungen erweitern (LyX, ...). Dies ist allerdings nicht im Sinne des Erfinders, und andererseits selten oder meist gar nicht mit LaTeX kompatibel.

Der Einfachheit halber wird am Beispiel von Kile eine kleine Übersicht der Möglichkeiten eines Frontends geboten. Im erweiterten Rahmen gelten diese Dinge aber auch für andere grafische Editoren.

Kile sollte vor dem Erstgebrauch nach den Bedürfnissen des Benutzers eingerichtet werden. Es können Verhaltensweisen von Kile definiert, das Aussehen, die Darstellung des Textes, deren farblisches Highlighting nach Syntax angepasst, Auswahlmodus mit der Maus/Tastatur, Kurzbefehle erweitert oder abgeändert, ausgewählt welcher PDF-Viewer benutzt sowie ein Projektpfad definiert werden, wo man die eigenen Dokumente auch wieder findet, und dieselben beim nächsten Start des Frontends geladen werden. Es gibt allerdings zahlreiche Möglichkeiten, mit denen der Anfänger durchaus auch überfordert sein kann. Am besten, man schaut sich diese Möglichkeiten einmal durch, ergänzt oder ändert ein wenig und schaut nach einiger Erfahrung und Angewöhnen des Frontends nochmals bei den Einstellungen vorbei.

Unter -->Einstellungen -->Systemüberprüfung kann die korrekte Installation und das Vorhandensein der benötigten Module für LaTeX und Kile automatisch überprüft werden. Es erscheint ein Fenster, mit farblicher Ausgabe der Module. In grün wurde der Systemtest bestanden, orange sind nicht-kritische Fehler, falls rote Einträge erscheinen, ist irgendetwas schief gelaufen. Bei Installationen über die Mandriva-Softwaretools sollten sich allerdings keine Fehler einschleichen.

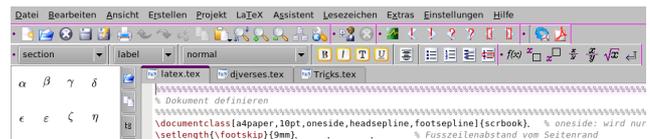
In den diversen Pulldown-Menüs befinden sich weitere Einstellmöglichkeiten, die bis zum Exzess ausgenutzt werden können, so kann beispielsweise der Einrückungsstil nach verschiedenen Programmiersprachen angepasst werden...

Unter -->Datei -->Neu kann aus Vorlagen ausgewählt werden. Es sind dies hauptsächlich Bücher, Briefe, Artikel, Reports und Folienvorlagen. Dieselben können angepasst oder erweitert werden (Briefe mit eigenem Briefkopf, Standarddokumente einer Firma, Folien mit den jeweiligen Headers etc). Mit einer zusätzlich aktivierbaren Hilfe im gleichen Dialog lassen sich auch zu verwendende Pakete per Häkchen anwählen und somit automatisch in die Masterdatei einfügen.

Werkzeugleisten können per Knopfdruck die verschiedensten Setz-Befehle in den Text integrieren. Sie ähneln dabei den Werkzeugleisten von Office-Suiten und können dem jeweiligen Geschmack angepasst werden.

Es sind dies zum Beispiel:

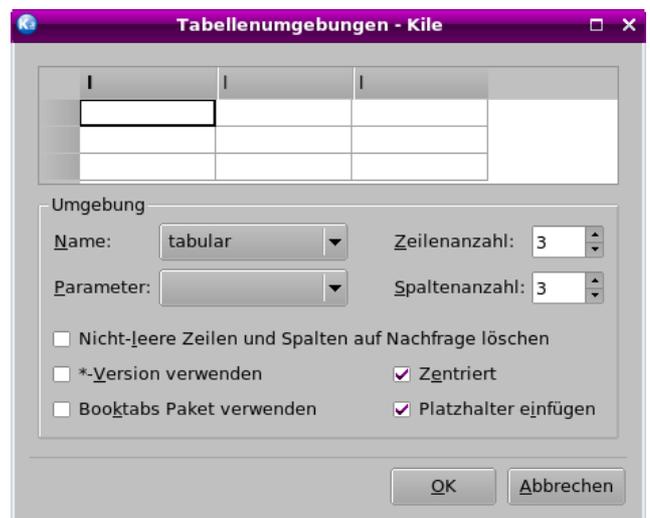
- Editieren: Kopieren, Ausschneiden, Fettschrift, Schrägschrift, Unterstreichen
- Einfügen von Script-Körpern für Aufzählung, Nummerierung, Beschreibung
- Schriftgröße
- Einsetzen von Verweisen, Fußnoten, Zitate, Indizes, Referenzen auf Stellen oder Bilder/Tabellen im Text
- Mathematische Formeln
- Zeilen- oder Seitenumbruch



Alle genannten Beispiele lassen sich auch von Hand tippen, jedoch vervollständigen die Knopfbedienungen jeweils den ganzen Scriptkörper, sie sollten demnach keine Syntaxfehler erzeugen. Falls von Hand ein Syntaxfehler geschrieben wurde, ermöglicht es ein Debug-Assistent, den teilweise kryptischen Fehlermeldungen oder Warnhinweisen zu folgen.

Um Text beispielsweise fett darzustellen, wird wie in einer Office-Suite der betreffende Abschnitt mit der Maus markiert und auf den B Knopf gedrückt. Der entsprechende Text wird in den geschweiften Klammern von `\textbf{}` eingekapselt und erscheint, wie nicht anders zu erwarten, im Enddokument (PDF) dann fett gedruckt.

Ebenso stehen in den diversen Pulldown-Menüs Assistenten für das Erstellen von Tabellen, Arrays oder die Einbettung von Grafiken zur Verfügung. Ein Assistent hilft, eine Tabelle zu definieren. Einerseits, von Hand geschrieben, bräuchte dies ein Mehrfaches an Zeit, und andererseits kann man sich bei nicht alltäglichem Gebrauch von LaTeX kaum alle Kapselungen wie Tabellen etc merken.



In die Felder werden die Werte eingetragen. In diesem Beispiel soll eine Tabelle mit Name, Alter und Größe erstellt werden. Im Quelltext kann das Ganze noch von Hand angepasst werden und sieht dann so aus:

```
\begin{center}
\begin{tabular}{|l|r|r|}
\hline
\textbf{Name}&\textbf{Alter}&\textbf{Grösse[m]}
}
\\hline
Fritz      & 15  & 1.50 \\hline
Johannes  & 39  & 1.99 \\hline
Priska    & 27  & 1.74 \\hline
\end{tabular}
\end{center}
```

Die Tabelle kann des Weiteren noch beschriftet (Label) und mit einer Referenz versehen werden. Die Referenz dient dazu (sofern das gewünscht wird), dass beispielsweise in einem Tabellenverzeichnis automatisch ein Eintrag gemacht wird oder im Text darauf verwiesen werden kann.

Etwas anders sieht es mit den mathematischen Formeln aus. Diese sind im Verhältnis zu Tabellen relativ kompliziert (abhängig von der Komplexität der Formel selbst) zu schreiben und sollten auf jeden Fall im Enddokument auf Korrektheit überprüft werden.

Was sich ebenfalls als sehr praktisch herausstellt, ist die automatische Textvervollständigung. Ein selbstlernendes Wörterbuch speichert schon geschriebene Worte und bietet sie (nach einer gewissen Anzahl getippter Buchstaben eines Wortes) zur Auswahl an. Mittels Pfeil- und Eingabetaste kann ein Wort aus der Liste gewählt und somit ergänzt werden.

Kile bietet ebenfalls die Möglichkeit einer Baumstruktur. Script-Körper können auf- oder zugeklappt werden. Beispielsweise die oben gezeigte Tabelle erhält, nachdem sie fertig geschrieben wurde, vor dem ersten `\begin` ein `-` (minus). Will man die Übersicht in einem längeren Text behalten und die Lesbarkeit erhöhen, kann auf dieses minus geklickt werden, und die ganze Tabelle wird „versteckt“, anstelle des Minus erscheint nun ein `+` und es wird eine horizontale Linie dargestellt. Daraufgeklickt wird der Text wieder im Editor eingeblendet. Es empfiehlt sich daher, vor einer Tabelle einen Kommentar zu platzieren, welcher in wenigen Worten den folgenden Scriptkörper beschreibt. Zusammengeklappt steht dann im Text der Kommentar, darunter die gestrichelte Linie. So können Bilder, Tabellen, etc ausgeblendet werden, in der Hierarchie immer höher, bis zum einzelnen Kapitel.

Weiters können gleichzeitig mehrere LaTeX-Dateien geöffnet und bearbeitet werden. Die verschiedenen Dateien werden, wie schon aus Browsern bekannt, in Tabs geöffnet. Dazwischen kann hin und her gewechselt und bearbeitet werden.

Beim Kompilieren (Übersetzung des Quelltextes in ein PDF) wird der Benutzer feststellen, dass teilweise automatisch zweimal kompiliert wird. Dies geschieht, damit Referenzen korrekt gesetzt werden können. Wenn also beim ersten Mal Übersetzungsfehler auftreten, können diese beim zweitenmal weg sein.

Was hat es mit `\usepackage{}` auf sich?

Nach dem langen Vorspiel wollen wir uns nun endlich ins kalte Wasser stürzen! Die Struktur des einfachsten TeX-Dokuments beinhaltet 3 Zeilen. Unser Text lautet klassisch „Hallo Welt!“. Grundsätzlich werden LaTeX spezifische Befehle mit „`\`“ begonnen. Das „`%`“-Zeichen gibt die Möglichkeit, Kommentare in den Quelltext einzubauen, die im Dokument natürlich nicht erscheinen.

```
\documentclass[10pt,a4paper]{article}
%Die sogenannte Präambel
\begin{document} %Beginn unseres Dokuments
Hallo Welt!
\end{document}
```

Wie Ihr merkt, die Befehle sind meistens selbsterklärend. In der ersten Zeile wird die Dokumentklasse definiert. Den Begriff zwischen den geschweiften Klammern könnt ihr z.B. je nach Bedarf mit „report“, „book“, etc. tauschen. Auf die unterschiedlichen Dokumentklassen werden wir hier nicht tiefer eingehen. In den eckigen Klammern können Optionen (so wie hier Schrift- und Papiergröße) angegeben werden. Alles was sich zwischen „`\begin{document}`“ und „`\end{document}`“ befindet, wird nachher auch im Dokument angezeigt. Mit dieser Struktur kann man schon einiges machen. Als deutschsprachiger Benutzer wird man aber sehr schnell ein paar Unbequemlichkeiten begegnen. Insbesondere wenn man Umlaute im Text verwendet, die in der deutschen Sprache nun mal nicht allzu selten auftauchen. Abhilfe schaffen da Erweiterungen, sogenannte packages, die mit dem Befehl `\usepackage{}` in der Präambel in das Dokument eingebunden werden.

Im ersten Beispiel kann LaTeX Umlaute kompilieren, allerdings mit einer etwas umständlichen Syntax: Für das „ä“ müssten wir „a“ schreiben.

```
\documentclass[10pt,a4paper]{article}
\usepackage[T1]{fontenc}
%oder
\usepackage[latin1]{inputenc}
\begin{document}
Hätten wir das doch gleich gewusst!
\end{document}
```

An diesem Beispiel sieht man auch, dass Packages manchmal Optionen brauchen. Diese werden dann in den eckigen Klammern angegeben.

Nun aber los, mein erstes Dokument

Nachdem wir in den letzten Zeilen die sogenannte Präambel kennengelernt haben, gehen wir einen wichtigen Schritt weiter. Wir wollen nun lernen wie wir einen einfachen Artikel erstellen. Dazu verwenden wir passenderweise die Dokumentklasse „article“. In die eckigen Klammern können noch zusätzliche Optionen definiert werden, A4 und Schriftgröße 10pt reichen aber für unser Beispiel allemal. Nun liegt es an euch die Packages zu wählen. Damit z.B. Inhaltsverzeichnisse oder Bilder einen deutschen Titel erhalten, müsst ihr das Paket „babel“ mit der Option „german“ einfügen. Sinnvoll, wenn ihr Bilder an Bord habt, ist es, die Packages „eso-pic“ und „graphicx“ einzubinden. Für eine etwaige Bibliographie sollte das Paket „bibgerm“ hinzugefügt werden. Für diejenigen unter euch, die mathematische Formeln oder Ausdrücke benutzen möchten, empfehle ich „amsmath“, „amsthm“, „amssymb“. Das Präfix „ams“ steht für *American Mathematical Society*. Ihr seht, dass in den Naturwissenschaften LaTeX durchaus ein Standard ist. Vor kurzem wohnte ich einem Vortrag einer Nature-Editorin bei, in dem sie ausdrücklich darauf hinwies, dass der Redaktion eingesandte Artikel in LaTeX am liebsten sind.

Jeder Artikel will gut strukturiert werden. Dazu teilen wir das Dokument in Abschnitte ein. Der Befehl `\section{Abschnitt nr.1}` teilt dem Dokument mit, dass hier solch ein Abschnitt beginnt! Benötigt man eine Unterteilung benutzt man den Befehl `\subsection{unterabschnitt 1}`. Ein Inhaltsverzeichnis wird übrigens, wenn gewollt, automatisch erstellt. Dazu gibt man in der „document“ Umgebung den Befehl `\tableofcontents` ein. In der Dokumentklasse „book“ stehen natürlich noch die Kapitel (`\chapter{}`) zur Verfügung, die über den Sektionen stehen. Es sei noch erwähnt, wenn der Titel des Abschnitts für das Inhaltsverzeichnis zu lang ist, besteht die Möglichkeit, dem Programm eine Kurzversion mitzuteilen, die in eckige Klammern vor die geschweiften gesetzt werden. Also dementsprechend:

`\section[Kurzversion]{Abschnitt}`

Auch das Hervorheben von Textabschnitten (z.B. Fett, kursiv, etc...) geschieht mit Backslash und geschweiften Klammern. Wie immer ist der hier aufgezeigte Weg nicht der einzige, der nach Rom führt. Fettigen Text setzt man mit der Option `\textbf`, kursive Buchstaben über die Option `\textit` und unterstrichenen Text mit `\underline`. Den Teil des Textes, der derart dargestellt werden soll, grenzt man mit geschweiften Klammern ab. Also z.B.: das ist `\textbf{fetter Text}`.

Mathematische Umgebungen werden entweder mit der Umgebung „displaymath“ oder noch einfacher mit „\$“ gekennzeichnet. Wobei sich letzteres fließend in den Text einbinden lässt, was insbesondere dann nötig ist, wenn man griechische Buchstaben im Text benötigt oder mal eine Größe indizieren muss. Weiter werden wir aber hier nicht auf die mathematische Umgebung eingehen, da der Artikel sonst zu lang werden würde und diese nicht für jeden von Nutzen ist. Außerdem findet ihr die nötigen Befehle oft auch im Editor oder in Dokumentationen.

Natürlich darf das wichtigste nicht fehlen bevor wir endlich loslegen können: Der Titel! Frei gestalten kann man den Titel, wenn man in der „document“ Umgebung eine „titlepage“ Umgebung definiert. Dort kann man sich nun austoben wie man möchte. Allerdings besteht die Möglichkeit, auch vordefinierte Titel zu benutzen, mit dem Vorteil, dass dies schnell und sauber geschieht. Mit der eigenen Titelseiten-Umgebung muss halt etwas gespielt werden bis sie zu 100% sitzt. Titel, Autoren, Datum werden anhand der Optionen `\title{}`, `\author{}` und `\date{}` dem Dokument mitgeteilt. In der Hauptumgebung genügt dann das Einfügen eines lapidaren `\maketitle`.

Damit seid ihr nun für einen ersten Artikel gerüstet! Wem das Ganze nun zu lang, zu kompliziert oder einfach zu trocken war, für den haben wir natürlich ein kleines Beispiel parat:

```
\documentclass[a4paper,10pt]{article}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage[german]{babel}
\usepackage{eso-pic}
\usepackage{graphicx}
\usepackage{bibgerm}
\usepackage{amsmath, amsthm, amssymb}
\title{Der Titel thront über alles}
\author{Ubi Wahn & Warg}
\date{\today oder sonst irgendwann}
\begin{document}
\maketitle
\tableofcontents
\section{Erster Abschnitt}
toll, \textbf{toll}, \textit{toll},
\underline{\textbf{toll}} !
\section{Zweiter Abschnitt}
\subsection{Erster Aspekt}

So weit wäre das geklärt
\subsection{Zweiter Aspekt}
Und damit ist die Welt wieder rosig!

\end{document}
```

Wie hier zu sehen, ist die Verschachtelung von Befehlen auch kein Problem. Wer einen Seitenbruch nach dem Inhaltsverzeichnis wünscht (oder zu einem anderen Zeitpunkt), kann dies mit dem Befehl `\pagebreak` erreichen.

Damit seid ihr nun in der Lage, einen eigenen Bericht oder ein eigenes Buch zu verfassen.

Bilder und Graphiken

Eine Warnung vorab: Bilder und Graphiken einfügen ist an sich simpel, Die gewünschte Positionierung kann aber mitunter einige Nerven kosten, besonders am Anfang, wenn noch etwas die Erfahrung mit LaTeX fehlt. Schuld daran ist hauptsächlich die sogenannte fließende Umgebung, die es LaTeX ermöglicht den eingefügten Körper dorthin zu platzieren, wo es dem Programm am besten passt. Eigentlich ganz praktisch, nur ist der Ort, den das Programm für die Graphik am besten findet, manchmal nicht der Ort, den der Benutzer haben möchte. Und dann fängt die nervenaufreibende Spielerei mit den Parametern an.

Der Basisbefehl, um eine Graphik einzufügen, lautet: `\includegraphics[]{}{}`

In die geschweiften Klammern kommt der Dateiname. Dieser sollte auch im selben Ordner wie eure *.tex Datei sein. Bei größeren Dokumenten habe ich mir irgendwann angewöhnt einen Unterdner bilder/ einzurichten. Damit bleiben die Ordner übersichtlich. Auf Groß- und Kleinschreibung solltet ihr auch achten. Unser Befehl sieht also nun derart aus:

```
\includegraphics[]{\bilder\bild1}
```

Das Format, das ihr benutzen dürft, hängt von eurer Ausgabe ab. Wenn euer Dokument als pdf ausgespuckt wird, müsst ihr Dateien des Formats *.jpg, *.png, etc... verwenden. Ist eure Ausgabe ein Postscript, so akzeptiert TeX so gut wie nur *.eps Dateien. In die eckigen Klammern kann nun eine Menge an Optionen reingepackt werden. LaTeX skaliert aber die Bilder automatisch, so dass man in der Regel nur eine Option angeben muss, um die Graphik in einer vernünftigen Größe angezeigt zu bekommen. Ich habe mir angewöhnt die gewünschte Breite in cm anzugeben, z.B.:

```
\includegraphics[width=5cm]{bilder\bild1}
```

Natürlich kann auch gleichzeitig z.B. die Höhe (height), die Pixelzahl (px), etc... angegeben werden.

Das Ganze muss man nun nicht in eine fließende Umgebung packen, bietet aber insbesondere später doch Vorteile, wenn man die Graphik indizieren will oder dem Bild Untertitel oder Beschreibungen hinzufügen möchte. Die fließende Umgebung wird einfach mit `begin` und `end {figure}` gesetzt. Hinter `\begin{figure}` kommen in eckigen Klammern nun Optionen, die uns das Leben vereinfachen werden.

```
\begin{figure}[htbp!]  
\includegraphics[width=5cm]{bilder\bild1}  
\end{figure}
```

Das h steht für „here“ und teilt dem Programm mit, dass man das Bild gerne an dieser Stelle hätte. Das ist dem Compiler aber oft ziemlich egal. Um ihn dazu zu zwingen, die gewünschte Graphik an dieser von uns vorgegebenen Stelle einzufügen, benutzt man das Ausrufezeichen. „t“ und „b“ bedeuten top und bottom. Sie teilen dem Programm mit, dass die Fließumgebung eher am oberen, bzw. unteren Rand der Seite orientiert wird. Wählt man z.B. „t“, so wird versucht, das Objekt auf einer Seite oben zu platzieren. „p“ erstellt eine eigene Seite mit Fließumgebungen, auf der diese platziert werden.

Manchmal passierte es mir, dass LaTeX ein Bild in ein anderes Kapitel setzte, weil es vom Platz her für ihn so am besten war. Mit den Optionen „htbp!“ war dem leider auch nicht beizukommen. Für diesen Fall gibt es ein Paket mit dem Namen „placeins“. Dieses beinhaltet den Befehl `\FloatBarrier`. Dieser zwingt das Programm, alle fließenden Objekte, die im Quelltext vor ihm vorkommen, auch im Dokument vor dieser Stelle auszugeben.

Briefe schreiben mit LaTeX

Briefklassen gibt es zu Genüge:

- letter, die als Standard beigefügte Briefklasse,
- scr ltr und scr ltr2 sind die Klassen, die das KOMA-Skript mitbringt,
- dinbrief,
- lettre,

g-brief sind ebenfalls Pakete mit denen sich einfach Briefe erstellen lassen.

Wenn ihr auf die Suche geht, findet ihr sicherlich noch mehr. Ich bevorzuge in der Regel die Briefe aus den Paketen, da sie oft besser aussehen und aus dem ganz einfachen und praktischen Grund, dass in der Dokumentation eine Vorlage beiliegt, die man einfach nur kopieren muss. Optisch am ansprechendsten finde ich das Paket g-brief, welches sich auch noch sehr einfach bedienen lässt. Die Vorlage zu g-Brief findet ihr unter: `/usr/share/texmf/doc/latex/g-brief`. Dort findet ihr die Dateien „beispiel1.tex“ und „beispiel2.tex“. Wenn ihr „beispiel2.tex“ kompiliert (entweder mit kile oder über die Konsole mit `pdflatex beispiel2.tex`) solltet ihr das Ergebnis in der Abbildung auf der nächsten Seite erhalten.

Ihr seht, das Verhältnis Aufwand gegen Ergebnis ist sehr gut! Natürlich könnt ihr einige Sachen ändern und hinzufügen (einige Optionen sind noch auskommentiert). Ich habe z.B. für dieses Beispiel den Befehl `\trennlinien` aktiviert. Einer der besten Teile dieses Pakets ist aber die Fußzeile.

OTTO RAFFZAHN

OTTO RAFFZAHN · Finsterweg 13 · Tiefeschlucht

E I N S C H R E I B E N

Frau
Else Mittellos
Finsterweg 12

93456 Tiefeschlucht

IHR ZEICHEN
EM

IHR SCHREIBEN VOM

MEIN ZEICHEN
OR

DATUM
2. Februar 2008

Zahlung Ihrer Miete

Sehr geehrte Frau Mittellos,

seit der siebten Mieterhöhung im laufenden Jahr konnte ich keinen Zahlungseingang von Ihnen mehr verbuchen. Ich bitte Sie Ihren Verpflichtungen schnellstens nachzukommen, da ich mich sonst gezwungen sehe, Ihre Miete ein weiteres mal anzuheben.

Mit freundlichen Grüßen

Otto Raffzahn

Otto Raffzahn

Mietwucherer
Menschenfreund
& Gierschlund

Adresse:
Wissenacker 25a
99533 Weitewelt

Inselblick 12
23213 Am Meer

Telefon:
Festnetz: +49 000 0000000
Freecall: +49 800 0000000
Mobil: +49 171 00000000
Fax: +49 0000 000-0000

Internet:
raffzahn@example.com
info@example.com
office@example.com
http://www.example.com
http://www.example.net

Bankverbindung:
Bankhaus Skrupellos & Co.
BLZ: 000.000.00
Konto: 000.000.000

Die Vorlagen der anderen Pakete befinden sich ebenfalls unter: /usr/share/texmf/doc/latex/. Vielleicht findet ihr darin auch noch andere Briefpakete, die euch noch besser gefallen! Hier kann ich nur sagen, viel Spaß beim Ausprobieren! ;)

Beamer: Vorträge mit TeX

Eine meiner Lieblingsanwendungen von LaTeX ist das Paket Beamer, das es einem ermöglicht, Vorträge mit LaTeX zu erstellen. Beamer wurde von Till Tantau (TU Berlin) für die Verteidigung seiner Doktorarbeit entwickelt. Es ist nicht das einzige Paket, mit dem man Vorträge erstellen kann (Es seien hier prosper und seminar genannt), aber der Code ist relativ einfach, mittlerweile das geläufigste und, da von einem deutschen entwickelt, besitzt es auch teilweise deutsche Dokumentationen. Persönlich finde ich, dass es auch optisch einen sehr guten Eindruck macht, die eine oder andere Spielerei mitbringt, diese aber elegant in die Folien einbaut, und der wichtigste Punkt: Der Platz auf einer Folie ist limitiert. Es gibt nichts Schlimmeres als vollgeklatschte Folien, in denen man sich nicht entscheiden kann, ob man dem Vortragenden folgt oder den Text auf der Leinwand liest. Nun aber wieder ran an die Arbeit!

Dazu benötigen wir zunächst das Paket LaTeX-Beamer aus den Contrib-Quellen. Mit Beamer zu arbeiten bedarf am Anfang noch etwas Umgewöhnung. Schnell akklimatisiert man sich aber. Die Präambel beginnt mit `\documentclass[{beamer}]`. Hier möchte ich nochmal betonen, dass es mit LaTeX praktisch zu jeder Zeit möglich ist, irgendwelche Sachen persönlich zu konfigurieren. Das kann manchmal einfach sein, manchmal erfordert es auch sehr gute Kenntnisse in TeX. Da es sich hier um eine Einführung handelt, werde ich den Weg zu einem einfachen Dokument zeigen. Ich komme zu diesem Punkt, weil schon an dieser Stelle (in die eckigen Klammern) diverse Optionen eingestellt werden können, die Einfluss auf das Dokument haben können. Ich möchte hier nur 2 nennen:

- **red**: Damit wird die Farbe des Dokuments auf rot gesetzt.

- **handout**: Die Ausgabe erfolgt als Handout, welches man den Zuhörern gerne vor dem Vortrag austeilen darf.

Wenn man Einfluss auf das Aussehen seines Vortrags nehmen möchte sollten die 2 Befehle folgen:

\usetheme{} und **\usecolortheme{}**.

Wobei jedes Theme eine vordefinierte Farbe besitzt. Es gibt gut 20 solcher Themes, obgleich man bei einigen 2 Mal hinschauen muss, um Unterschiede zu erkennen. Als Beispiel möchte ich Warsaw, Antibes und Marburg nennen (die meisten Themes tragen übrigens den Namen deutscher Städte). Um alle möglichen Themes aufgelistet zu bekommen recherchiert ihr am besten im Internet oder werft einen Blick in den Beameruserguide.pdf (über Google sehr schnell zu finden, kann man aber auch in der Dokumentation kompilieren). Titel, Datum, Autoren und Packages kommen in die Präambel, genau wie beim vorher beschriebenen Artikel. Es gibt noch die Möglichkeit, die Einrichtung anzugeben, von der man kommt: **\institute{}**. Damit wären wir mit der Präambel schon am Ende.

Nun kommt das Hauptdokument, welches üblicherweise mit **\begin{document}** und **\end{document}** begrenzt wird.

Folien werden fast analog definiert:

\begin{frame} und **\end{frame}** markieren Beginn und Ende. Zum guten Ton gehört es sich, die Folien zu betiteln. Dies erfolgt mit dem Befehl **\frametitle{}**.

Und damit haben wir nun alle grundlegenden Werkzeuge, um einen Vortrag zu erstellen. Dieser könnte z.Bsp so aussehen, dabei könnt ihr gerne mit den Themes und Farben rumspielen um euch die verschiedenen Ergebnisse anzusehen:

```
\documentclass[beamer]
\usetheme{Warsaw}
\usecolortheme{default}
\usepackage[german]{babel}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{graphicx}
\usepackage{eso-pic}

\title{\LaTeX für mu.de}
\author{Nicolas Ubrig}
\institute{Mein Zuhause}
\date{\today}

\begin{document}
\begin{frame}
\maketitle %Titelseite
\end{frame}

\begin{frame}
\frametitle{Inhaltsverzeichnis}
```

```
\tableofcontents
\end{frame}

\section{Kapitel 1}
\subsection{Unterkapitel 1}
\begin{frame}
\frametitle{Gruß}
HALLO IHR DA!!
\end{frame}

\section{Kapitel 2}
\subsection{Unterkapitel 1}
\begin{frame}
\frametitle{Abschied}
Das war es auch schon
\end{frame}
\end{document}
```

Eine wichtige Option, die mit dem Paket Beamer kommt, ist, dass eine Seite in mehrere Spalten aufgeteilt werden kann. Das ist sonst natürlich auch möglich, aber mitunter nicht so trivial. Mit Beamer ist das allerdings ein Kinderspiel. Mit **\begin{columns}** und **\end{columns}** wird dabei der Bereich eingegrenzt, der mehrspaltig eingeteilt werden soll. **\column{}** setzt eine Spalte (in die geschweiften Klammern könnt ihr die Breite in cm, mm, etc...) einstellen. Alle Einträge die Folgen kommen nun in diese Spalte bis zum nächsten **\column{}** oder **\end{}** Eintrag.

```
\begin{frame}
\frametitle{Spalten}
\begin{columns}
\column{5.5cm}
Hier die erste Spalte mit dem Inhalt den ihr wollt

\column{5.5cm}
Hier ist die 2. Spalte
\end{columns}
\end{frame}
```

Ein anderer Befehl, wichtige Teile eines Vortrags optisch hervorzuheben, ist:

\begin{block}{Titel} mit **\end{block}**

Alles was zwischen diesen beiden Befehlen steht, kommt in eine farbige Umrandung, der man gerne einen Titel geben darf. Sollte nur diese farbige Box mit dem Hintergrund gewünscht werden, so kann man die geschweiften Klammern hinter **\begin{block}{}{}** so wie hier leer lassen, aber NIE-MALS (!!!!) die geschweiften Klammern weglassen. Dann könnt ihr einige Stunden mit der Fehlersuche sparen und hier spreche ich aus schmerzhafter Erfahrung!

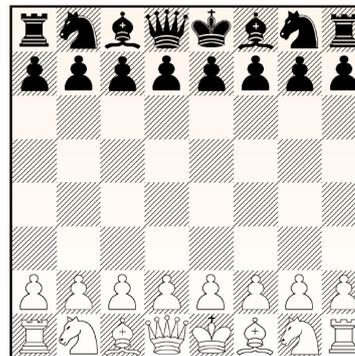
```

\begin{frame}

\frametitle{Die Box}
\begin{columns}
\column{5.5cm}
\begin{block}{Unglaublich wichtiger Inhalt}
Das hier MÜSST ihr euch merken
\end{block}

\column{5.5cm}
\begin{block}{}
Das ganze ohne Titel!
\end{block}
\end{columns}
\end{frame}

```



Der Code dazu ist denkbar einfach:

```

\usepackage{chess}
{\font\Chess=chess20}
\board{rnbqkbnr} %Figuren 1. Reihe, schwarz
{pppppppp} % schwarze Bauern
{* * * *} % leere Felder
{* * * *} % etc
{* * * *}
{PPPPPPPP} % weisse Bauern
{RNBQKBNR} % Figuren 1. Reihe, weiss
$$\showboard$$

```

Bilder einfügen funktioniert genau so wie oben beschrieben. Für diejenigen, die den weißen Hintergrund zu Fad finden, gibt es die Möglichkeit, ein eigenes Hintergrundbild einzufügen. Ab dem Moment, ab dem das Hintergrundbild erscheinen soll, fügt ihr einfach folgenden Befehl ein:

```

\usebackgroundtemplate{\includegraphics[width\paperwidth,height=\paperheight]{DATEINAME}}

```

An der Stelle „DATEINAME“ fügt ihr natürlich den Pfad und Dateinamen des gewünschten Bildes ein. Wichtig ist nur, dass der Befehl außerhalb der frame-Umgebung steht. Weitere und weitergehende Informationen findet ihr dazu in der Dokumentation in der Datei Beameruserguide.pdf.

Einige weitere Beispielmöglichkeiten...

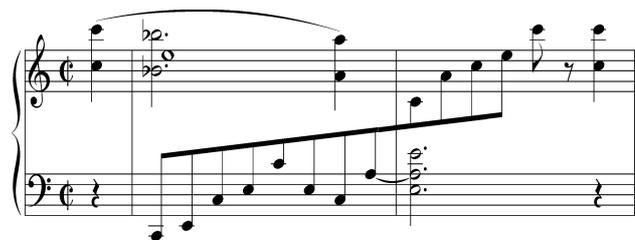
(Anmerkung: Die folgenden Beispiele wurden mit LaTeX erstellt, aus dem PDF als Bild extrahiert und in OpenOffice eingefügt. Die Bilder erscheinen hier teilweise unscharf, dies ist jedoch im Original nicht der Fall.)

Schach

Eher als Spielerei als für den täglichen Gebrauch, gibt es das Paket *chess*. Damit lassen sich Schachbretter und zugehörige Figuren zeichnen.

Musikpartitionen

Mit *musixtex* lassen sich Musiknoten schreiben. Dies gehört allerdings in die Sektion für Fortgeschrittene...



Programmiercode

Was nicht fehlen darf, ist Source-Code-Darstellung. Mit entsprechenden *listing*-Paketen wird der LateX-Umgebung mitgeteilt, um welche Sprache oder welchen „Dialekt“ es sich handelt (C# beispielsweise wird als Dialekt von C oder C++ behandelt). Der Compiler setzt automatisch Schlüsselwörter in Fettschrift. Möglich ist auch ein zusätzliches Unterscheiden mit Farben (z.B. Schlüsselwörter blau, Kommentare grün, Variablen rot etc).

```

#include "stdafx.h"
#include <iostream>
#include <conio.h>

using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    int a = 42;
    int b = 23;
    int result = 0;

    swapByValue(a,b);
    printTwoInteger(a,b);
    getch();

    swapByReference(a,b);
    printTwoInteger(a,b);
    getch();

    swapByAddress(&a,&b);
    printTwoInteger(a,b);
    getch();

    result = calcFormula(a,b);
    cout << "result="
    << result << endl;
    getch();

    return 0;
}

```

Schriften

Verschiedene Schriften sind genauso vorhanden, wie dies in den üblichen Textverarbeitungsprogrammen möglich ist, allerdings gibt es noch einige mehr (Paket *oldgerm*)

Die gotische Schriftart

Als Fraktur-Schrift bekannt

Wird Schwabacher-Schrift genannt

Verschiedene Textformatierungen

Oder halt die üblichen, ganz einfachen
Textformatierungen, **ob man** sie **braucht**
oder nicht, oder ob sie nur dazu da sind, zu
demonstrieren, dass man das mit L^AT_EX durchaus
kann. Blah blah blah, ich könnt' ewig weiter-
erzählen
doch wo führt das noch hin?

Mathematische Formeln

Formeldarstellung mit den Paketen *latexsym* und *amsmath*. Falls gewünscht, kann wie im Bild eine automatische Indexierung generiert werden, um im Text auf die Formel zu referenzieren.

$$\lim_{p \rightarrow 0} \int_a^b f_s(x) = \int_a^b f(x) \quad (1)$$

$$f_s(x) = pf(x) \sum_m \delta(x - mp) =: f(x) \text{comb} \left(\frac{x}{p} \right) \quad (2)$$

Links und nützliche Tipps

Wir hoffen, diese Einführung hat euch Lust auf mehr gegeben. Damit ihr in nächster Zeit nicht ganz alleine beim erstellen eurer Dokumente seid, haben wir hier eine kleine Linksammlung vorbereitet:

1. Sehr ausführliche LaTeX-Einführungen:

[ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a0260003.pdf](http://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a0260003.pdf)

[ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a0279510.pdf](http://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a0279510.pdf)

2. Befehls-Übersicht im Stil von man-pages:

<http://www.weinelt.de/latex/index.html>

3. Übersicht LaTeX-Beamer Themes:

<http://mike.polycat.net/gallery/beamer-themes>

4. Download zusätzlicher Pakete :

<http://www.ctan.org/>

... und natürlich zahlreiche Google-Suchtreffer zum Thema LaTeX!