

**YaLT**  
*Yet another  $\LaTeX$  Tutorial*

Autor: Steve Moser  
E-Mail: [smoser@anno74.de](mailto:smoser@anno74.de)  
Datum: Mai 2004  
Semester: SS 2004  
Fach: Open Source

# Inhaltsverzeichnis

1.1	Ein paar Worte . . . . .	3
1.2	Einleitung . . . . .	3
1.3	$\LaTeX$ . . . . .	3
1.4	Was brauche ich alles? . . . . .	5
1.5	1.Beispiel . . . . .	5
1.5.1	Ich will endlich was sehen! . . . . .	6
1.5.2	Was war das nochmal? . . . . .	7
1.6	Grundsätzliches zur $\LaTeX$ Syntax . . . . .	9
1.6.1	Kapitel, Überschriften . . . . .	10
1.6.2	Schriftarten . . . . .	10
1.6.3	Aufzählungen und Nummerierungen . . . . .	11
1.6.4	Schriftgrößen . . . . .	12
1.6.5	Abstände . . . . .	12
1.6.6	Einzüge . . . . .	13
1.6.7	Text ausrichten . . . . .	13
1.6.8	Seitenformatierung, Zeilen und Seitenumbrüche . . . . .	14
1.6.9	Randnotiz . . . . .	15
1.6.10	Tabellen . . . . .	15
1.6.11	Überschrift und Inhaltsverzeichnis . . . . .	17
1.6.12	Kopf-, und Fußzeilen Layout . . . . .	18
1.6.13	Fußnoten . . . . .	19
1.6.14	Nützliche Befehle . . . . .	19

---

1.7	Mathematik . . . . .	20
1.8	Grafiken, Bilder und Co . . . . .	22
1.9	Präsentationen mit dem Packet slides . . . . .	25
1.10	Noch Fragen, dann ... . . . .	31

## 1.1 Ein paar Worte

Yalt ist ein Miniprojekt, dass ich mir zu Anfang als Gedächtnisstütze geschrieben habe. Beim Schreiben ist mir jedoch aufgefallen, dass es doch ein wenig in Arbeit ausgeartet ist. Hinzu kam, dass eine richtige Versionierung nach "Pi mal Daumen" nicht funktionieren kann. Letztendlich habe ich dann RCS (Revision Control System) gefunden, und somit eine vernünftige Lösung für eine Versionierung dieses kleinen Projektes eingeführt. Nun ist es auf der einen Seite sehr übersichtlich geworden und auf der anderen Seite kann ich nun mit gutem Gewissen jeden dazu einladen seine Ideen, Vorschläge und Erweiterungen einfließen zu lassen, damit neue Versionen entstehen können. Natürlich gibt es eine Fülle an L<sup>A</sup>T<sub>E</sub>X Büchern, jedoch ist das Ziel dieses Tutoriums, Interessierte schnell an ein akzeptables Ergebnis zu führen, und vielleicht sogar Studenten zu animieren aus dem "Lock in" Mechanismus proprietärer Software zu entfliehen (wenn das die Schule schon nicht tut) um sich eine eigene Meinung bilden zu können.

## 1.2 Einleitung

YaLT ist ein weiteres Tutorial für Schnelleinsteiger, die nur mal reinschnuppern und sich einen groben Überblick verschaffen wollen, was denn L<sup>A</sup>T<sub>E</sub>X eigentlich ist, und ob man es wirklich brauchen kann für seinen eigenen Zwecke. Vielleicht findet der eine oder andere etwas, was er noch nicht wusste, oder man benutzt es als kleines Nachschlagewerk. Ich hoffe es ist für jeden etwas dabei.

Viel Spass damit.

Steve Moser

## 1.3 L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X (sprich Lahtech) ist ein Textsatzsystem, das zum Beispiel mit Word verglichen werden kann. Es kann ein komplettes Dokument mit Hilfe eines Texteditors und dem Latexpaket erstellt werden. Der große Vorteil liegt dabei in der Plattformunabhängigkeit. Das heißt, es kann sowohl unter Unix/Linux als auch unter Windows damit gearbeitet werden. L<sup>A</sup>T<sub>E</sub>X ist weitgehend mit einer Art Programmiersprache zu vergleichen. Sie besitzt ihre eigene Syntax und ihren eigenen Aufbau. Ein weiterer Vorteil von L<sup>A</sup>T<sub>E</sub>X liegt darin, dass die Dokumentdateien in reiner Textform erstellt werden; somit kann genau nachvollzogen werden, was denn wirklich in diesem Dokument steht (nicht wie andere Textsatzsysteme, die Dateien derart verunstalten, dass man sie nur mit dem Erstellungsprogramm öffnen kann).

Es ist weiterhin möglich und üblich, sein Dokument in mehrere unabhängige Teile zu zerlegen. Nehmen wir beispielsweise an, jemand hat seine Diplomarbeit mit Hilfe von Microsoft Word verfasst und diese umfasst 150 Seiten, derjenige hat sich ein Inhaltsverzeichnis erstellen lassen und ihm fällt ein, dass es vielleicht doch besser wäre, die Gliederung zu ändern. Sprich er möchte einen der letzten Unterabschnitte mit allen dazugehörigen Textnoten weiter vorne haben. Er muss nun erst das entsprechende Kapitel markieren (bei 20 Seiten ein Akt der Geschicklichkeit) und es weiter vorne einfügen. Dann muss er das Inhaltsverzeichnis neu schreiben bzw. abändern lassen und wenn er Glück hat, werden die Fußnoten auch richtig gesetzt. (Am Rande sollte noch erwähnt werden dass u.a. Word ernsthafte Probleme mit großen Dateien haben kann).

In L<sup>A</sup>T<sub>E</sub>X kann man auch alles in ein Dokument schreiben, was aber nicht sinnvoll ist. Im Gegenteil, man teilt sein Dokument in eigene Abschnitte ein, sogenannte "sections" (z.B. die Einleitung). Diese kann man wiederum auf verschiedene Dateien aufteilen. Man hat dann eine Datei, die latex-einleitung.tex heisst, eine weitere, hauptteil.tex, und eine letzte, die sich schluss.tex nennt. Der Vorteil liegt darin, dass u.U. mehrere Personen an verschiedenen Teilen der Dokumentation schreiben können und dass die Dateien nicht allzugroß werden, denn das gesamte Dokument wird erst am Schluss mittels dem L<sup>A</sup>T<sub>E</sub>X Compiler erstellt. Dieser generiert automatisch ein Inhaltsverzeichnis, falls es gewünscht ist. Auch wenn die eine oder andere Datei nicht mehr lesbar sein sollte, wieso auch immer, so ist nur ein Teil des gesamten Dokuments neu zu verfassen und nicht das komplette Dokument von Anfang an.

L<sup>A</sup>T<sub>E</sub>X kann laufend erweitert werden. Beispielsweise können viele Zusatzpakete in das Dokument mit eingebunden werden. So können umfassende Mathematische Formeln erstellt, wie auch Grafiken eingebunden werden. Es gibt sogar ein Paket dass die Brailleschrift (Blindenschrift) unterstützt. Den größten Vorteil sehe ich persönlich darin, dass das erstellte Dokument innerhalb von Sekunden als HTML oder PDF Datei dargestellt werden kann ohne Wissen über deren Syntax.

Es gibt natuerlich nicht nur Vorteile. So gilt es jedem Neuling die spezielle Syntax der L<sup>A</sup>T<sub>E</sub>X Sprache näher zu bringen. Es bedarf eines guten Gedächtnises um sich die grundlegendsten Befehle zu merken, oder einem guten Frontend dass diese bereitstellt. Hier zu erwähnen wäre z.B. Kile fuer Linux, oder LyX. Mit diesen ist der Einstieg ein leichtes und für den ambitionierten Anwender empfiehlt sich natürlich emacs im L<sup>A</sup>T<sub>E</sub>X Modus, oder vim. (siehe Links)

Wer sich ein wenig Zeit nimmt und dieses Tutorial als Grundlage verwendet hat in kurzer Zeit seine ersten persönlichen Erfolge und den nötigen Überblick um in dieses Themengebiet tiefer einzusteigen.

Und nun gehts endlich los. :-)

Das wichtigste habe ich vergessen: LATEX IST FÜR ALLE SYSTEME KOSTENLOS!!!

## 1.4 Was brauche ich alles?

Wie gesagt kann man Latex in vielen Betriebssystemen nutzen. Wir brauchen zwei Dinge, den Latex Compiler und einen beliebigen Texteditor (z.b. gvim, vi, emacs, joey, unter Windows notepad, ultraedit ... wobei die Linuxeditoren auch schon für Windows portiert wurden). Hier die wichtigsten Tools für Linux und Windows:

- Windows: Unter Windows kenne ich zwei Möglichkeiten Latex zu verwenden. Eine Möglichkeit ist es sich Miktex herunter zu laden. (zu finden unter : <http://www.miktex.org/setup.html> ). Eine andere Lösung wäre sich Cygwin herunter zu laden. (zu finden unter: <http://www.cygwin.com/> ). Beim installieren einfach das Latex bzw. Tex packet mitinstallieren. Cygwin simuliert eine Unixumgebung, daher kann dort auch das portierte Tex/Latex installiert werden.
- Linux: Unter Linux wird z.B über Yast die Pakete Tex und Latex nachinstalliert. Diese sind standardmäßig auf der Distributoren CD enthalten.

Und nun zu unserem 1.Beispiel.

## 1.5 1.Beispiel

Wir öffnen einen beliebigen Editor und schreiben folgendes hinein:

```
\documentclass[a4paper,12pt]{article}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage{german}
\begin{document}
\section{Einleitung}
Hier steht dann ein fettes ''foobar''. Der Text den man
eben schreiben will.
\end{document}
```

Die Datei wird nun unter einem Namen abgespeichert, sinnvoll wäre z.B. "beispiel1.tex". Sie muss zwar nicht zwingendermaßen auf .tex enden, doch gerade für Windowsbenutzer ist dies die sinnvollere Variante.

Nun tippen wir in der Konsole "latex beispiel1.tex" ein und setzen den Befehl mit Enter ab. Eine Konsole ist unter Windows ein DOS Fenster (Eingabeaufforderung). Sollte der Befehl latex nicht gefunden werden, so setzt man entweder den Pfad in der PATH variablen oder man begibt sich direkt in das Verzeichniss, in dem sich latex.exe befindet. Dort "uegibt man den kompletten Pfad. Z.B. latex.exe "Verzeichnis\Unterverzeichnis\beispiel1.tex. Es kommen nun ein paar Compilermeldungen die uns aber noch nicht interessieren sollen. Der Compiler hat nun folgende Dateien erstellt.

1. beispiel1.log: Hier stehen die Meldungen die der Compiler erzeugt hat.
2. beispiel1.aux: Das ist eine Hilfsdatei für Querverweise.
3. beispiel1.toc: In der Regel enthält diese das Inhaltsverzeichnis.
4. beispiel1.lot: Enthält das Tabellenverzeichnis.
5. beispiel1.dvi: Und hier die wichtigste. Das ist unser erstes Beispiel und kann nun angesehen werden.

Am Rande bemerkt, je nach Gestaltung des Dokuments müssen nicht alle Dateien vorhanden sein, aber mindestens .dvi .log. Um das Inhaltsverzeichnis (.toc) zu erzeugen muss das Diokument zweimal kompiliert werden. Auch um Startprobleme aus dem Weg zu gehen sollte man sicher sein das der Latex Befehl dem System bekannt ist, oder man sich in dessen Verzeichniss befindet wenn man in aufruft ;-). An die Windowsbenutzer ->Pfade setzen nicht vergessen!.

### 1.5.1 Ich will endlich was sehen!

Um nun die Datei beispiel1.dvi ansehen zu können wird ein dvi Viewer benötigt (dvi=device independent).

Unter Linux gibt es z.B xdvi. Also in der Konsole nachsehen ob das Paket installiert ist. Z.B mit " which xdvi", dann sollte etwa sowas erscheinen wie "/usr/bin/xdvi", oder mit "locate xdvi", wobei die erste Variante die elegantere mit weniger Text ist. Sollte es nicht installiert sein; nachinstallieren über das Web oder Installations CD.

Nun geben wir in der Konsole ein

"xdvi beispiel1.dvi"

und siehe da... .

Eine andere Methode das Ergebnis zu betrachten ist es, die .dvi Datei mit Hilfe von

```
"dvipdf beispiel1.dvi"
```

in eine pdf Datei zu verwandeln (beispiel1.pdf). Diese kann mit dem AcrobatReader betrachtet werden. "acroread", "xpdf", "gv" wären hier zu nennen.

Auch ist es möglich, die .dvi Datei in eine PostScript Datei mit dem Kommando

```
"dvips beispiel1.dvi"
```

zu formatieren. Diese Datei kann entweder an den Drucker gesendet werden oder man kann sie mit GhostView ("gv beispiel1.ps") vorher ansehen.

Man sieht hier schön dass wir nur einen Text in L<sup>A</sup>T<sub>E</sub>X geschrieben haben und sehr schnell drei verschiedene Dateiformate erstellen konnten (.dvi, .ps, .pdf!!). Es ist auch möglich eine Webseite aus unserem beispiel1.tex zu machen. Richtig gehört, der textohml compiler tth erstellt aus dem tex File eine HTML Seite. Der Compiler ist erhältlich unter ' 'http://hutchinson.belmont.ma.us/tth/' '. Für die Windowsbenutzer heissen die Kommandos:

1. latex beispiel1.tex
2. latex beispiel1.tex
3. dvips .-Ppdf beispiel1.dvi <erzeugt eine postscriptdatei>
4. epstopdf beispiel1.ps <pdf datei wird erzeugt>

Und um das dvi File anzusehen gibt es 'yap' (Yet another previewer). Das Kommando lautet dann unter der Konsole "yap deinFile.dvi".

## 1.5.2 Was war das nochmal?

Kommen wir auf unser 1. Beispiel nochmal zurück.

Ein Dokument wird vorerst in zwei Teile geteilt. Das Präambel und das Dokument selbst. Der Präambel ist sozusagen die Umgebung die der Compiler dem Dokument zuordnet und folgt dem

```
\documentclass
```



Umgebung können Aussehen, Stile, Schriftgröße, und benötigte Pakete includiert werden. Es gibt eine Reihe von Präambel Befehlen die ich noch näher erklären werde. Danach kommt das eigentliche Dokument selbst. Dies wird gekennzeichnet durch

```
\begin{document} und \end{document}
```

Gehen wir nun im einzelnen das schon geschriebene durch.

- `\documentclass[a4paper,12pt]{article}`:

Die Dokumentenklasse braucht mindestens zwei Parameter. Einmal die Größe und die Art des Dokumentes. Es gibt folgende wichtige Größen:

1. `a4paper`: Das Dokument wird für DIN A4 formatiert (210 x 297 mm)
2. `a5paper`: Das Dokument wird für DIN A5 formatiert (148 x 210 mm)
3. `b5paper`: Das Dokument wird für DIN B5 formatiert (176 x 250 mm)

- Folgende Standardgrößen sind verfügbar:

1. `10pt`: Die Schriftgröße beträgt 10 Punkte. Dies ist Standard und wird bei fehlender Angabe verwendet.
2. `11pt`: na was wohl..
3. `12pt`: hmm...

- Nun wenden wir uns dem Typ des Dokuments zu:

1. **article**: Diese Klasse dient vor allem für kurze technische Artikel. Die oberste Stufe eines Gliederungspunktes wird mit `\section` eingeleitet. Es werden keine gesonderten Titelseiten gedruckt, wie bei "report".
2. **report**: wird für längere technische Artikel benutzt. Zusätzlich steht der Befehl `\chapter` zur Verfügung. Die Überschrift eines neuen Gliederungspunktes erscheint auf einer neuen Seite.
3. **letter**: Wie der Name schon sagt dient diese Klasse zum Erstellen von Briefen, wobei die Vorlage auf Amerikanischen Briefen basiert. Für Deutsche Briefe sollte man eher **dinbrief** benutzen.

4. **book:** Diese Klasse dient der Erstellung von Büchern. Der Inhalt wird in drei Bereiche eingeteilt. "front matter" "main matter" und "back matter" eingeteilt. Wobei

**\frontmatter**

das Inhaltsverzeichnis, Widmung und das Abbildungsverzeichnis enthält. Die Seitenzahlen sind Römisch. \mainmatter enthält den eigentlichen Inhalt. Die Seitenzahlen sind arabisch. Zu guter letzt beschreibt

**\backmatter**

beipieelsweise Index, Nachwort, Anhänge. Seitenzahlen sind ebenfalls arabisch.

- Es gibt noch weitere Klassen, aber die wichtigsten sind genannt worden.

- **\usepackage [T1 ]{fontenc}**

\usepackage wir verwendet um weitere Pakete im Präambel einzubinden. Die Syntax lautet \usepackage [optionale Angabe]{zwingender Parameter}. [T1]steht für einen bestimmten Zeichensatz. {fontenc} erlaubt die Benutzung von deutschen Umlauten, wie ä, ö, ü. Es wird standartmäßig mit dem Packet \usepackage {german} verwendet. Auch kann als Parameter [latin1 ]und {inputenc} verwendet werden. Dies bildet wiederum auch Zeichen im Dokument auf interne L<sup>A</sup>T<sub>E</sub>X-Codes ab. Somit können auch die deutschen Umlaute direkt benutzt werden. Wird dieses Packet nicht eingebunden so müssen die Umlaute mit " im Text eingebunden werden. Ein ä wird dann in L<sup>A</sup>T<sub>E</sub>Xso geschrieben; "a oder \"a.

Später gebe ich noch eine Liste der gängigsten Pakete an, die man in L<sup>A</sup>T<sub>E</sub>Xbrauchen kann.

## 1.6 Grundsätzliches zur L<sup>A</sup>T<sub>E</sub>XSyntax

Den Präambel-Teil haben nun erstmal hinter uns. Kommen wir nun zum Dokument selbst. Das eigentliche Dokument steht zwischen \begin{document} und \end{document}. Generell ist es wichtig, da's die Kommandos richtig hingeschrieben werden. Wenn ich z.B ein Text halbfett machen möchte, dann heißt das Kommando \textbf{Text}. Ergebnis: **Text**. Sollte man nun aus Versehen ein Leerzeichen zwischen \und textbf haben so wird der Compiler eine Fehlermeldung bringen und dann geht die Suche erst richtig los. Auch ist es wichtig, daß die geschweiften Klammern direkt anschließen und nicht mit einem Leerzeichen dazwischen. Einfach eine Sache der Korrektheit.

## 1.6.1 Kapitel, Überschriften

Kapitel sind in Unterkapitel und Unterpunkte aufgeteilt. In L<sup>A</sup>T<sub>E</sub>X wird dies bei {book}, {report} in vier Bereiche eingeteilt. Bei {article} ist die oberste Stufe \section.

\chapter	z.B Punkt 1
\section	z.B Punkt 1.1
\subsection	z.B Punkt 1.1.1
\subsubsection	z.B Punkt 1.1.1.1

## 1.6.2 Schriftarten

Sehen wir uns folgendes an:

\textrm{Serifenschrift}	Serifenschrift
\textsf{Serifenlose Schrift}	Serifenlose Schrift
\texttt{Maschinenschrift}	Maschinenschrift
\textmd{Medium}	Medium
\textbf{Halbfett}	<b>Halbfett</b>
\textup{Aufrecht}	Aufrecht
\textit{Kursiv}	<i>Kursiv</i>
\textsl{Geneigt}	<i>Geneigt</i>
\textsc{Kapitälchen}	KAPITÄLCHEN
\emph{Hervorgehoben}	<i>Hervorgehoben</i>
\textnormal{Standardschrift des Dokumentes}	Standardschrift des Dokumentes

Es gibt eine Reihe von Befehlen die die Syntax haben:

```
\begin{befehl}
.
.
.
.
\end{befehl}
```

Möchte ich z.B. eine Aufzählung oder Nummerierung machen, dann benutze ich die folgenden Befehle.

### 1.6.3 Aufzählungen und Nummerierungen

Möchte ich eine einfache punktuelle Aufzählung so geht das mit:

```
\begin{itemize}
\item{text} oder einfach nur \item text
\item text
\end{itemize}
```

Das sieht dann so aus:

- Ein Aufzählungspunkt
- Ein weiterer Aufzählungspunkt

Möchte ich nun Nummerierungen davor haben, mache ich das gleiche mit:

```
\begin{enumerate}
\item{text} oder einfach nur \item text
\item text
\end{enumerate}
```

Sieht natürlich so aus:

1. Ein Aufzählungspunkt
2. Ein weiterer Aufzählungspunkt

Und was passiert wenn ich folgendes eingebe:

```
\begin{itemize}
\item Eine Aufzählung -- default $\bullet$
\item [$\circ$] etwas anders
\item und der letzte
\end{itemize}
```

Mal sehen was dabei herauskommt:

- Eine Aufzählung – default •
- etwas anders
- und der letzte

## 1.6.4 Schriftgrößen

Natürlich muss man auch die Schriftgröße für einzelne Wörter verändern können. Hier die wichtigsten Befehle:

<code>\tiny</code> klein	klein
<code>\scriptsize</code> skriptgröße	scriptgröße
<code>\footnotesize</code> blabla	blabla
<code>\small</code> klein eben	klein eben
<code>\normalsize</code> supernormal	supernormal
<code>\large</code> größer	größer
<code>\Large</code> noch größer	noch größer
<code>\LARGE</code> aufwachen	aufwachen
<code>\huge</code> big	big
<code>\Huge</code> BIG	BIG

Ein wichtiger Hinweis! Wenn die Schriftgröße einmal umgestellt wurde, wird der nachfolgende Text solange in diese Schriftgröße weitergeführt bis wieder ein `\normalsize` kommt.

## 1.6.5 Abstände

Es soll auch Leute geben die gerne mit Abständen (Absatzabständen) spielen wollen. Dann können wir das auch. Um einen Abstand zwischen Worten zu erzwingen, muss dies explizit angegeben werden. Nicht vergessen, sowie es im Editor aussieht wird es garantiert nicht aussehen wenn man es compiliert. Außer der Editor ist ein spezieller L<sup>A</sup>T<sub>E</sub>X WYSIWYG Editor (**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et). Wie immer natürlich die wichtigsten und bekanntesten:

Die Familie der Abstände. Es war einmal ein kleiner Abstand `\smallskip`.

Der hatte einen größeren Bruder, der nannte sich `\medskip`.

Aber dieser war nicht der letzte. Der ganz große Bruder war ein wirklicher Abstand, ein richtiger Männerabstand und nannte sich `\bigskip`.

Die Mutter aller Abstände war die Herrin der Abstände, sie konnte Abstände so groß machen wie sie wollte. Sie sagte nur `\setlength{\parskip}{Abstand in cm}` und schon wuchs ihr Abstand auf cm.

Um diesen Abstand zu sehen, muss natürlich noch ein Text folgen, damit man den hier auch wirklich sieht. Und auch hier natürlich nicht wieder vergessen einen Standardabstand einzustellen. Defaultmäßig ist der Wert 0cm eingestellt.

Um diese Absätze zu sehen reicht es, wenn im Editor auch ein Absatz eingefügt wird. Sprich eine Leerzeile. Danach darf aber kein "neue Zeile Befehl erfolgen" (`\`).

### 1.6.6 Einzüge

Um Einzüge braucht man sich eigentlich nicht zu kümmern, wer diese dennoch verändern möchte, kann dies mit ein paar einfachen Befehlen bewerkstelligen. Standardmäßig wird die erste Zeile eines Absatzes um `\parindent` eingerückt.

Auch hier ist der Standardwert 0pt (wird also gar nicht eingerückt). Daher setzen wir ihn mit `\setlength{\parindent} {Abstand in cm oder pt}` auf einen größeren Wert.

Der Folgende Text ist zwei cm eingerückt. Nun schalte ich ihn wieder aus mit `\setlength{\parindent} {0cm}`.

Für den aktuellen Absatz habe ich `\indent` benutzt. D.h. der Erstzeileneinzug wurde für diesen Absatz eingeschaltet. Dieser Befehl gilt nur für den aktuellen Absatz.

Um diesen Absatz ohne Einzug darzustellen benutzt man `\noindent`. Dieser und der vorherige Befehl müssen jeweils am Anfang eines Absatzes stehen.

#### Wichtig!

Der `indent` und `noindent` Befehl funktionieren natürlich nur wenn vorher irgendwann im Dokument der Einzug  $> 0$  gesetzt wurde. Normalerweise macht man dies nur einmal und zwar im Präambel (vor dem `\begin{document}`). Dort kann man einfacher halber schreiben:

```
\parskip 0.2cm %Absatzabstand  
\parindent 0.5cm %Einzug bei einem Abstand
```

### 1.6.7 Text ausrichten

L<sup>A</sup>T<sub>E</sub>X nimmt an daß alles im Blocksatz geschrieben wird. Aber auch hier gibt es die Möglichkeit den Text zu positionieren.

Es gibt drei Arten den Text zu plazieren. Linksbündig, rechtsbündig und zentriert. Diese leiten wiederum gleichzeitig einen neuen Absatz ein.

Wie so vieles wird der Text in eine `\begin{ }` und `\end{ }` Umgebung eingebunden. Es unterscheiden sich lediglich die Inhalte innerhalb der geschweiften Klammern.

```
\begin{center}
```

```
.  
.zentriert
```

```
.  
\end{center}
```

```
\begin{flushleft}
```

```
.  
.linksbündig
```

```
.  
\end{flushleft}
```

```
\begin{flushright}
```

```
.  
.rechtsbündig
```

```
.  
\end{flushright}
```

### 1.6.8 Seitenformatierung, Zeilen und Seitenumbrüche

Da man im Editor zwar eine neue Zeile beginnt, heisst das noch lange nicht, dass dies auch der Compiler tut. L<sup>A</sup>T<sub>E</sub>X erwartet hier ein explizites Kommando, das dem Compiler mitteilt, dass er eine neue Zeile oder eine neue Seite beginnen soll. Will man nun im Text eine neue Zeile beginnen, so fügt man am Ende des Satzes folgendes ein: `\\`

Will man eine neue Seite beginnen, so macht man dies mit dem Kommando: `\newpage`

Es gibt noch zusätzliche Befehle, die der Formatierung dienen. Hier zu nennen sind:

- **`\newline`** : Beginnt eine neue Zeile.
- **`\linebreak [0-4]`** : Standardmäßig ist eine Priorität von 4 eingestellt. Die Verringerung der Zahl setzt die Priorität für einen Zeilenumbruch herab.
- **`\pagebreak [0..4]`** : Wie `\linebreak`...
- **`\enlargethispage { länge in cm }`** : Verlängert die aktuelle Seite einmalig um x-cm.

- **\textheight** : Legt die Höhe des Textblockes fest, ohne Kopf und Fußzeile. Wird im Präambel festgelegt. Z.B. **\textheight 20cm** .
- **\textwidth** : Legt die Breite des Textblockes fest.
- **\pageheight** und **\pagewidth** wird über die Dokumentclassoptionen schon vorgelegt (z.B. über a4,report..).
- **\headheight** : Legt die Höhe der Kopfzeile fest.

### 1.6.9 Randnotiz

Natürlich gibt es Gründe, hin und wieder eine Randnotiz einzufügen. Das wird mit dem Befehl

**\marginpar{text}**

bewerkstelligt. Diese wird automatisch auf der rechten Seite (ungeraden) rechts vom Text *RANDNOTIZ* dargestellt und bei linken Seiten (gerade) links.

### 1.6.10 Tabellen

Nun, es gibt natürlich auch Mittel und Wege eine Tabelle darzustellen.

Eine Tabelle kann z.B. so aussehen:

	Anzahl	Preis
Windows	1	200-? Euro
Linux	1	0 Euro
Windows Server	1	400-?
Linux Server	10	0

oder eine Tabelle wie man sie einfachhalber aus Excel kennt:

Artikel	Anzahl	Preis
Bananen	10	10 Euro
Äpfel	20	10 Euro
Karotten	30	5 Euro
Lebenshaltungskosten	=	je nach Eurostimmung

Nun erst mal die Syntax:

```
\begin{tabular}{|l|cr|}\hline
```



```

& Anzahl & Preis\\ \cline{2-3}
Windows & 1 & 200-? Euro\\ \hline
Linux & 1 & 0 Euro\\ \hline
Windows Server& 1 & 400-?\\ \hline
Linux Server& 10 & 0\\ \hline

\end{tabular}

\begin{tabular}{|l|c|r|}\hline

Artikel & Anzahl & Preis\\ \hline
Bananen & 10 & 10 Euro\\ \hline
"Apfel & 20 & 10 Euro\\ \hline
Karotten & 30 & 5 Euro\\ \hline
Lebenshaltungskosten & = & je nach Eurostimmung\\ \hline

\end{tabular}\\

```

Eine Tabelle wird eingeleitet mit **\begin {tabular}** und beendet mit **\end {tabular}**. In der zweiten Klammerumgebung wird die Anzahl der Spalten festgelegt über die Anzahl der Formatierungsbuchstaben. **l** steht für left (linke Ausrichtung der Spalte), **c** für center (zentriertes Ausrichten der Spalte) und **r** für right (rechts Ausrichtung der Spalte). Im Beispiel 2 wird also eine dreispaltige Tabelle eingeleitet, wobei die erste Spalte links, die mittlere zentriert und die 3.Spalte rechts ausgerichtet wird. Ein | zwischen diesen Formatierungszeichen bedeutet, dass ein vertikaler Strich gezogen werden soll, zwischen den Spalten.

Eine Tabelle mit 10 Spalten würde dann mit {lcccccccr } gestartet, wobei die erste Spalte links, die anderen zentriert und die letzte Spalte rechts ausgerichtet wird.

Nun wird das erste Element in der Spalte/Zeile (1/1) erwartet. Im Beispiel 2 ist dies das Wort "Artikel". Gefolgt von dem kaufmännischen Und (&) wird in die nächste Spalte gewechselt. Am Ende der 3.Spalte wird eine neue Zeile eingefügt mit \\. Das **\hline** (horizontal line) unterstreicht die aktuelle Zeile.

Im Beispiel 1 wurde nach der ersten Zeile ein **\cline {2-3}** (center line) eingefügt. Dies ermöglicht ein einzelnes Unterstreichen von einzelnen oder zusammengefassten Spalten. Als Parameter erwartet der Compiler die Nummern der Spalten. Hier wurde eine Linie unter "Anzahl" und "Preis" gemacht. Das erste Feld wurde einfach leer gelassen. Für den einfachen Gebrauch von Tabellen reichen diese Befehle aus.

### 1.6.11 Überschrift und Inhaltsverzeichnis

Die Überschrift sowie Inhaltsverzeichnis werden im Präambel erledigt. Die Überschrift kann z.B Titel und Autor enthalten. Um automatisch ein Inhaltsverzeichnis generieren zu lassen muss man zwei Dinge erledigen. Zum einen gehört in den Präambel der Befehl `\tableofcontents` und zum anderen muss das fertige Dokument zweimal compiliert werden. Ich erledige diese Dinge in einer eigenen Datei, die ich `index.tex` nenne. Dort schreibe ich alles nötige rein, was die Formatierung und grundsätzliches Aussehen betrifft. Hier also ein Beispiel wie man es machen könnte, aber das ist Geschmackssache.

```
\documentclass[12pt,a4paper]{report}

\title{YaLT\Yet another \LaTeX~Tutorial}
\author{Steve Moser\< \textit{\textbf{smoser@anno74.de}} >}
%\date{16. November, 2002}
%the current date at compile time will be used if the above line is comr

\pagenumbering{arabic}
\pagestyle{plain}
\textwidth 15cm
\textheight 21cm
\topmargin 0.2cm
\oddsidemargin 1cm
\evensidemargin 1cm
\parskip 0.2cm %Absatzabstand
\parindent 0cm %Einzug bei einem Absatz
\setcounter{chapter}{1}
\usepackage{german,times}
\usepackage{enumerate}
\usepackage[T1]{fontenc} %wurde hinzugefügt
\usepackage{wrapfig}

\begin{document}

\maketitle
\tableofcontents
\newpage

\include{einleitung} %d.h. die datei
```

```
``einleitung.tex`` oder ``einleitung`` wird eingebunden
\include{hauptteil}
\include{schluss}
\include{grafik}
\include{slides}
\end{document}
```

Es gibt auch die Möglichkeit, anstatt `\include` `\input` zu nehmen. Der Unterschied ist, dass ersteres das erste Kapitel der jeweiligen Datei (section), auf einer eigenen neuen Seite beginnt. Letztere Art besagt, dass die Datei in den laufenden Text miteingebunden wird. Ist z.B ein Text solange, dass er nicht auf eine Seite passt, wird er um gebrochen auf die nächste Seite. Auch wenn es nur ein Satz ist, der dann auf der zweiten Seite zu sehen ist. Der Unterschied liegt nun darin, dass `\include` eine 3. Seite eröffnet und dort mit der neuen Datei fortfährt, wohingegen `\input` auf der zweiten Seite einen Absatz einfügt und mit der neuen Datei fortfährt. Bei der `input`-variante kann u.U. zu Problemen mit Fußnoten kommen.

### 1.6.12 Kopf-, und Fußzeilen Layout

Damit das Dokument auch einen ordentlichen Rahmen erhält gibt es das Paket **fancy-headings**. Es erledigt neben Seitennummerierung die Gestaltung der Kopf und Fußzeile, und kann nach belieben angepasst werden. Das Paket wird, wie alle Pakete, im Präambel eingebunden. Ist dies erledigt, kommt die eigentliche Arbeit, die jedoch relativ gering ausfällt.

Das vorliegende Dokument hat, wie man sieht die obere und unter Fußzeile. Der "Code" sieht dazu folgendermaßen aus.

```
\pagestyle{fancy}
\lhead[Yalt - Yet another \LaTeX~Tutorial]
{\Yalt - Yet another \LaTeX~Tutorial}
\rhead[Yalt]{Yalt}
\setlength{\footrulewidth}{0.4pt}
\lfoot[] {FHA, Sommersemester 2004, Steve Moser, Informatik}
\cfoot[] {}
\rfoot[Seite \thepage]{Seite \thepage}
```

Zur Erklärung:

`\pagestyle{fancy}` legt das Aussehen generell fest. Nun kann man mit Hilfe von `\lhead` festlegen was im linken Kopfbereich stehen soll, und das für gerade und ungerade Seitenzahlen. Das gleiche gilt für `\rhead` (rechter Kopfbereich) und `\chead` (Kopf mitte). Danach werden zwei Klammerpaare angefügt, und zwar “[”]” und “{”}”. Die eckigen Klammern geben den Text für gerade Seiten an und die geschweiften Klammern den Text für ungerade Seiten. Analog zu den drei genannten verhält sich `\rfoot`, `\cfoot` und `\lfoot` für die Fußzeile.

`\thepage`, fügt die aktuelle Seite ein. In ”Der L<sup>A</sup>T<sub>E</sub>X Begleiter”, von [GMS]<sup>1</sup> wird dieses Thema ausführlicher behandelt.

### 1.6.13 Fußnoten

Ganz einfach, `\footnote{Text}`. Kann man unten schön sehen, und wird auch automatisch durchnummeriert.<sup>2</sup>

### 1.6.14 Nützliche Befehle

Nachfolgend ein paar Befehle, welche man hin und wieder braucht.

`\begin{verbatim} ... \end{verbatim}` → damit verhindert man, dass das geschriebene als Befehl interpretiert wird, wie zum Beispiel der Backslash “\”.

Möchte man nur einen einzelnen Backslash kann man dies auch mit “`\textbackslash`” bewerkstelligen.

---

<sup>1</sup>Michel Goossens, Frank Mittelbach, Alexander Samarin, Addison Wesley, 2000, ISBN: 3-8273-1689-8

<sup>2</sup><http://www.anno74.de>

## 1.7 Mathematik

Mit L<sup>A</sup>T<sub>E</sub>X ist es natürlich auch möglich, Mathematische Symbole und Formeln in den laufenden Text miteinzubinden.

Das könnte z.B. so aussehen:  $x_2 = \frac{1}{4} \cdot 8$

Wer es länger oder komplexer haben möchte, dem hilft vielleicht dies weiter:

$$x_{1,2} = \frac{2b \pm \sqrt{b^2 - 4ac}}{2ab}$$

Auch sind solche Sachen möglich:

$$\begin{aligned} a^2 - a^2 &= a^2 - a^2 \\ (a - a) \cdot (a + a) &= a \cdot (a - a) \\ a + a &= a \\ 2a &= a \\ 2 &= 1 \end{aligned}$$

Nun zur Erklärung:

Es gibt, wie gesehen zwei Möglichkeiten Mathematik einzubringen. Einmal im fließenden Text oder als Darstellung einer ganzen Formel/Rechnung. Um Mathematische Formeln im Text unterzubringen, beginnt man mit  $\$$  und endet mit  $\$$ . Hier das obige Beispiel:

```
 $\$ x_{1,2} == \frac{1}{4} \cdot 8 \$\backslash\backslash$ 
```

Es gibt auch die Möglichkeit mit `\begin{math}` und `\end{math}` eine Formel in den laufenden Text einzubinden. Möchte man wie oben eine Rechnung oder Formel auflösen, dann sieht das so aus:

```
\begin{eqnarray*}
a^2-a^2&=&a^2-a^2\\
(a-a)\cdot(a+a)&=&a\cdot(a-a)\\
a+a&=&a\\
2a&=&a\\
2&=&1
\end{eqnarray*}
```

Mit `\begin{eqnarray*}` wir eine Tabelle begonnen. Der Stern bedeutet hier eine Nicht-nummerierung der Spalten. Innerhalb dieser Tabelle muss natürlich mit `&` spaltenweise getrennt werden.

Da ich nicht viel, besser gesagt fast gar nichts mit Mathematik mache, brauche ich es ganz selten. Daher folgen nun die gängigsten Formelzeichen:

<code>a_{2}</code>	$a_2$	Der Index wird durch die Zahl innerhalb der Klammern bestimmt
<code>\frac{5}{8}</code>	$\frac{5}{8}$	Erklärt sich von selbst, oder?
<code>\cdot</code>	$\cdot$	Ein Malzeichen
<code>\div</code>	$\div$	Division
<code>\sqrt{10}</code>	$\sqrt{10}$	Wurzel
<code>\sum</code>	$\sum$	Summenzeichen
<code>e^{20}</code>	$e^{20}$	Exponentenbildung
<code>\sum\limits^i_{n-1} 5</code>	$\sum_{n-1}^i 5$	
<code>\lim\limits^i_{n-1} 5</code>	$\lim_{n-1}^i 5$	
<code>\int\limits^i_{n-1} 5</code>	$\int_{n-1}^i 5$	
...		
...		
...		

Eine rel. gute Zusammenfassung bietet das Taschenbuch L<sup>A</sup>T<sub>E</sub>X kurz&gut, aus dem O'Reilly Verlag.

ISBN 3-89721-204-8

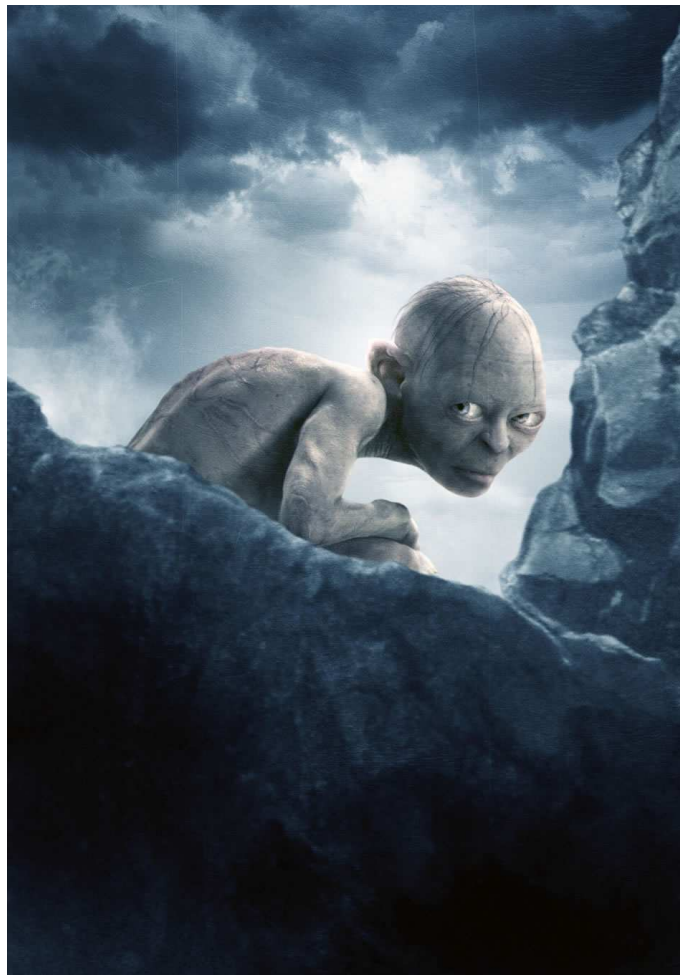
Aber auch im Internet findet man schnell etwas.

## 1.8 Grafiken, Bilder und Co

Zum Einbinden von Bildern sind mir zwei Arten bekannt.

Einmal die Einbindung der Grafik mit `\includegraphics` und `\epsfig`. Also sollte im Präambel entweder `\usepackage{dvips}{graphicx}` oder `\{epsfig}` anstatt `graphicx` stehen. Das `dvips` bedeutet, dass ein entsprechender Treiber beim Einbinden benutzt wird. (Dann kann ich das Dokument auch komplett als `.dvi` Datei ansehen, was heissen soll, die Bilder sind dann dort mitenthalten. Ansonsten sind sie es dann erst bei der Erstellung des PDF dabei.)

Fügen wir nun erstmal ein Bild ein:



Gollum aus "Der Herr der Ringe"

Nun zur Erklärung des folgenden Quelltextes:

```
\begin{center}  
\label{ Gollum }  
\epsfig{width=9cm,file=bilder/gollum2.eps}  
\caption{ Gollum aus ''Der Herr der Ringe'' }  
\end{center}
```

1. das Bild wird zentriert.
2. mit **\epsfig** weise ich an, dass ein Bild eingefügt wird.
3. width gibt die breite an, die ich auf 9 cm angegeben habe. Wenn nötig wird auf 9cm skaliert.
4. file gibt an, um welches Bild es sich handelt.
5. **\caption** nummeriert die Bilder in Abhängigkeit des Inhaltsverzeichnisses.

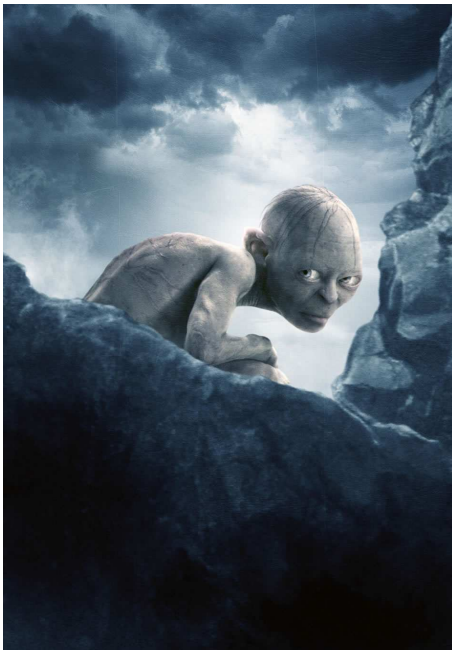


Abbildung 1.1: Gollum

Mit **\wrapfigure** lässt sich ein Text neben ein Bild setzen. Der Platz wird somit besser ausgenutzt. Achtung!!!

Bei epsfig muss die Datei eine ps (postscript) oder eps(encapsulated postscript) Datei sein. Unter Linux ist es am einfachsten, wenn man den Befehl **''convert''** in der Konsole benutzt, z.B. `convert gollum.jpg gollum.eps`. Um die Skalierung innerhalb des Dokumentes braucht man sich nicht kümmern. Unter Windows gibt es sicherlich auch genug Möglichkeiten ein Bild in .eps umzuwandeln (Gimp – gratis, ImageMagick – gratis oder Corel-Draw – teuer).

Leider kann ich nichts zu `\includegraphics` sagen, da ich es nie benutze.



Nun zur Erklärung:

```
\begin{wrapfigure}[ n_zeilen ]{ links | rechts }{ feldbreite }  
\label{ name }  
\epsfig{width= bildbreite,bildfile }  
\caption{ Bildunterschrift }  
\end{wrapfigure}
```

n-zeilen	Anzahl der Zeilen, die für Bild inkl. Unterschrift frei bleiben sollen
links rechts	wo das Bild hin soll, l für links, r für rechts
feldbreite	Breite des freizuhaltenden Bereiches zwischen Bild und Text (z.B. in cm)

## 1.9 Präsentationen mit dem Packet slides

Es ist auch möglich mit L<sup>A</sup>T<sub>E</sub>XPräsentationen zu erstellen. Hierzu wird lediglich ein neues Packet benötigt. Dieses nennt sich "prospcr". Dieses Packet enthält Styles (Designs) wie die Folien aussehen sollen. Um Platzierungen innerhalb einer Folie muss man sich nur zweitrangig kümmern. Fangen wir einfach mal an. In meinem Präambel steht folgendes drin.

```
\documentclass[pdf,final,colorBG,slideColor,nototal,gyom]{prospcr}

\usepackage{color}
\usepackage{graphicx}
\usepackage{german}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}

\title{Test}
\subtitle{Untertitel eben}
\author{Steve Moser}
\email{smoser@anno74.de}
\institution{www.anno74.de}
&\Logo(2,5){\includegraphics[width=1cm]{gollum2.eps}} ' & ' steht für nicht

\begin{document}
\maketitle

\include{folien}

\end{document}
```

Nun erstmal zur Erklärung. Wir brauchen wie immer eine "documentclass". Diese enthält nötige Informationen wie alles zu handhaben ist, wie sich gleich aufklären wird.

final	finale Version wird ausgegeben (inkl. Grafiken)
draft	Grafiken werden ersetzt durch bounding boxes, Erstellungsdatum wird mit ausgegeben
total	die Caption enthält die aktuelle Seitennummer und die Gesamtzahl der Seiten
nototal	die Caption enthält nur die aktuelle Seitennummer
slideBW	eine eingeschränkte Menge an Farben wird benutzt, besser für Schwarz/Weiß-Ausgabe
slideColor	der volle Umfang an Farben wird benutzt
colorBG	die Farben des aktuell ausgewählten Styles werden ausgewählt
nocolorBG	die Hintergrundfarbe ist weiß
ps	kompiliertes Dokument ist optimiert für Overhead- Folien (keine Overlays)
pdf	kompiliertes Dokument ist optimiert für Beamer- Präsentation (mit Overlays)
noaccumulate	Overlays werden beibehalten
accumulate	Overlays werden unterdrückt
default	Standardfolien-Style

Zusätzlich gibt es noch folgende Styles. Ich habe oben "gyom" genommen.



### Style: contemporain

- vordefinierte Schrift
  - für den Titel
    - **Times**
    - 20,74 pt (20 pt)
  - für den Text
    - **Helvetica**
    - 14,4 pt (14 pt)
- vordefinierte Textbreite
  - 11 cm

### Style: darkblue

- vordefinierte Schrift
  - für den Titel
    - **Times**
    - 20,74 pt (20 pt)
  - für den Text
    - **Helvetica**
    - 14,4 pt (14 pt)
- vordefinierte Textbreite
  - 11 cm

### Style: frames

- vordefinierte Schrift
  - für den Titel
    - **Helvetica**
    - 14,4 pt (12 pt)
  - für den Text
    - **Helvetica**
    - 13 pt (12 pt)
- vordefinierte Textbreite
  - 11 cm

### Style: gyom

- vordefinierte Schrift
  - für den Titel
    - **Times**
    - 22 pt (20 pt)
  - für den Text
    - **Helvetica**
    - 14,4 pt (14 pt)
- vordefinierte Textbreite
  - 11 cm

### Style: lignesbleues

- vordefinierte Schrift
  - für den Titel
    - **Times**
    - 20,74 pt (20 pt)
  - für den Text
    - **Helvetica**
    - 14,4 pt (14 pt)
- vordefinierte Textbreite
  - 11 cm

### Style: nuancegris

- vordefinierte Schrift
  - für den Titel
    - **Helvetica**
    - 20,74 pt (20 pt)
  - für den Text
    - **Helvetica**
    - 14,4 pt (14 pt)
- vordefinierte Textbreite
  - 11 cm



Kommen wir nun zur Folie selbst. Das Aussehen und Verhalten haben wir festgelegt. Nun brauchen wir den eigentlichen Text. Dies ist einfach zu bewerkstelligen mit den folgenden Zeilen.

```
\begin{slide}[Übergang]{Titel}  
Inhalte  
\end{slide}
```

Mit `\begin` und `\end {slide}` wird eine Folie festgelegt. Der Übergang legt fest wie beim Folienwechsel verfahren werden soll. Ob nun die aktuelle eingeblendet wird oder langsam verschwindet, usw.. . Der Titel ist selbstredend, ok die Überschrift ist gemeint :-).

Es gibt folgende Übergänge:

1. Split
2. Blinds
3. Box
4. Wipe
5. Dissolve
6. Glitter
7. Replace (Default)

Wichtig ist hier die Großschreibung.

Das könnte dann so aussehen:

```
\begin{slide}[Wipe]{Zukunftsmusik}  
\begin{itemize}  
\item das ist meine erste folie  
\item und es sieht nicht schlecht aus  
\item oder?  
\end{itemize}  
\end{slide}
```

Nun kompilieren wir das noch und dann sollte dies dabei herauskommen:



Und nun Folie 2:

## Zukunftsmusik

- das ist meine erste folie
- und es sieht nicht schlecht aus
- oder?

Test – p.1

Das Paket prosper gibt es zum download auf meiner Homepage. Natürlich auch im Netz unter <http://prosper.sourceforge.net/>. Das Paket sollte vorher laut Installationanleitung installiert werden.

## 1.10 Noch Fragen, dann ...

- [http://www.fernuni-hagen.de/URZ/urzbib/ls\\_broschueren.html](http://www.fernuni-hagen.de/URZ/urzbib/ls_broschueren.html)  
Tutorials über L<sup>A</sup>T<sub>E</sub>X
- <http://www.miktex.org/setup.html> Win32 Version des L<sup>A</sup>T<sub>E</sub>X Compilers
- <http://hutchinson.belmont.ma.us/tth/latex2html/compiler>
- <http://www.cygwin.com> Portierung vieler Gnu-tools auf Win32
- <http://www.dante.de/> "Der Zweck des Vereins ist die Betreuung und Beratung von TeX-Benutzern im gesamten deutschsprachigen Raum"
- <http://www.dante.de/faq/de-tex-faq/> FAQ eben ;-)
- <ftp://ftp.dante.de/tex-archive/> Ftp Archiv über alle L<sup>A</sup>T<sub>E</sub>X Pakete
- <http://www.dante.de/TeX-Service/cookbook/cookbook.html>
- <http://www.prosper.sourceforge.net> alles was mit slides (folien) zu tun hat.