

Eine kleine Latex Einführung

1. Einleitung

Diese Seite soll keine vollständige Latex Einführung sein. Dafür gibt es bessere [Dokumente](#) (siehe auch [diese deutsche Latex Einführung](#)). Auch soll diese Seite nicht ideologisch für oder wider Office Pakete argumentieren. Office Pakete sind nützliche Werkzeuge zur Erledigung von Büroaufgaben. Dazu gehört sicher das Verfassen von Briefen, Kurzberichten und so weiter. Sicher keine alltägliche Büroaufgabe ist das Verfassen einer wissenschaftlichen Arbeit. Hier sollte man spezialisierte Werkzeuge benutzen, die all die Besonderheiten des Buchdrucks unterstützen. Latex ist ein solches Werkzeug. Es gibt sicher noch viele andere Werkzeuge. Das Besondere an Latex ist, dass es kostenlos für jeden zur Verfügung steht. Und, um gleich mit noch einem Vorurteil aufzuräumen, Latex gibt es nicht nur für Unix. Im Gegenteil, das wohl komfortabelste Latex Paket gibt es für [MS Windows](#).

Um Latex auszuprobieren braucht man also nicht erst den Umgang mit Unix zu lernen, sondern kann gleich in der gewohnten Windows Umgebung mit der Arbeit beginnen.

[Zum Seitenanfang...](#)

2. Benötigte Software

Eine erste Einstiegshürde in die Latexwelt besteht darin, die benötigte Software zu installieren. Denn Latex besteht aus vielen kleinen Werkzeugen, die zusammen ein mächtiges Werkzeug ergeben. Man benötigt (unter MS Windows):

- ein Latex Paket wie [MikTeX](#) (die kleinste Variante reicht für den Anfang)
- Einen Editor zum Verfassen der Texte. Ein universaler Editor für Windows ist [PSPad](#). Das [TeXnicCenter](#) hingegen ist eine komplette Latexumgebung mit integrierter Rechtschreibprüfung, Verwaltungsfunktionen und vielen anderen netten Kleinigkeiten. Gerade Anfängern erleichtert solch eine Umgebung den Umgang mit Latex. Ansonsten kann man natürlich immer auf die Editorgesteine [Vim](#) und [Emacs](#) zurückgreifen. Neben all diesen frei zugänglichen Programmen gibt es auch eine große Anzahl von kommerziellen Programmen. Wie dem auch sei, man braucht lediglich **einen** Editor und nicht mehrere!
- Man sollte [Ghostscript](#) installieren, damit man sich sein fertiges Dokument anschauen kann.
- Mittels Latex kann man natürlich auch PDF Dateien erzeugen, aber den entsprechenden Viewer (Acrobat Reader) sollte ja jeder haben.

Unix bringt meist die oben genannten Werkzeuge mit. Deshalb braucht man dort nicht extra Software zu installieren. Will man aber eine spezialisierte Latexoberfläche haben, dann sollte man mal einen Blick auf [Kile](#) werfen.

Ich möchte an dieser Stelle nicht die Installation beschreiben, denn für alle oben genannten Pakete gibt es ausführliche Installationsanleitungen.

Einleitung

Benötigte Software

Unterschiede Latex und Office Tools

Grundlagen

Umlaute und deutsche Sprache

Grafiken

Fußnoten und Verweise

Deckblatt

Zur Organisation

Literaturverzeichnis

Stichwortverzeichnis

Rechtschreibprüfung

Makefile für Unix

Zweispaltiges Layout

Und doch ein paar Bücher

Abschließende Worte

 del.icio.us

 WONG!

 BOOKMARK

[Zum Seitenanfang...](#)

3. Unterschiede Latex und Office Tools

Worin besteht nun der Unterschied zwischen Latex und Office Werkzeugen? Nun, in Latex konzentriert man sich lediglich auf den Inhalt des Textes und nicht auf die Formatierung. Die Formatierung übernimmt Latex. Deshalb wirkt eine Latexbedienoberfläche auch oft sehr einfach, denn man gibt hauptsächlich den Text seines Werkes ein. Damit Latex den Text richtig formatieren kann, muss es natürlich ein paar Hinweise bekommen. Latex kann nicht erraten, ob es sich bei einer Überschrift auch wirklich um eine Überschrift handelt oder ob es ein normaler Satz ist, bei dem lediglich das abschließende Satzzeichen vergessen wurde. Deshalb muss man alles kennzeichnen, was nicht normaler Text ist. Dazu gehört z. B.:

- Überschriften, z. B. von Kapiteln oder Abschnitten
- hervorgehobene Worte, z. B. in **Fettdruck**
- wenn man z. B. ein Bild, eine Tabelle oder eine mathematische Formel einfügen möchte
- und natürlich gibt es noch viele Ausnahmen ;-)

Man führt also keine Formatierung durch. Das bedeutet auch, dass man die Kapitel nicht nummeriert. Dies tut Latex automatisch. Aus den Überschriften kann natürlich auch ein Inhaltsverzeichnis automatisch erzeugt werden. Man kann komfortabel Querverweise in den Text einfügen. Eine ganz große Stärke von Latex ist weiterhin die Verwaltung von Literaturverzeichnissen. Man gibt lediglich an einer zentralen Stelle die Daten der Quelle (also z. B. eines Buches) ein und fügt dann an den entsprechenden Stellen im Text lediglich einen Querverweis auf das Werk ein. Latex erstellt dann automatisch das komplette Literaturverzeichnis (nach den entsprechenden DIN Normen). Weiterhin kann man relativ komfortabel ein Glossar oder ein Stichwortverzeichnis erstellen. Die Verwaltung von Fußnoten ist ebenfalls kinderleicht. Weiterhin kann man sich für seine Grafiken und Tabellen ein entsprechendes Abbildungs- bzw. Tabellenverzeichnis automatisch erstellen lassen.

Latex bietet somit sehr viel Funktionalität. Natürlich muss man auch etwas dafür tun, nämlich sich mit Latex beschäftigen. Ich glaube, dies ist das größte Hindernis für Latex. Viele Computernutzer sind an die Grundfunktionen von Office Paketen gewöhnt und sind lieber bereit sich mit diesen durch viele Seiten Text zu quälen, als sich die Zeit zu nehmen und ein besseres Werkzeug (für lange Texte) kennen zu lernen. Man kann sicher davon ausgehen, dass man die in die Latexerlernung investierte Zeit sehr schnell durch höhere Produktivität wieder herausholen kann. Andererseits macht es keinen Sinn zum Schreiben eines Briefes erst Latex zu lernen.

Wer bis hier gelesen hat, wird vielleicht wirklich mit dem Gedanken spielen, Latex zu erlernen. Deshalb Schluss mit der Vorrede, fangen wir mit einigen Beispielen an!

[Zum Seitenanfang...](#)

4. Grundlagen

Beispiel 1

Erstellen wir also eine ganz einfache Datei ([beispiel01.tex](#)). Die Latexquelldateien haben immer die Endung tex.

```
% das Papierformat zuerst
\documentclass[a4paper, 11pt]{article}
```

```
% hier beginnt das Dokument
\begin{document}
```

Dies ist ein Satz.

```
% das ist wohl jetzt das Ende des Dokumentes
\end{document}
```

Man erkennt, dass es sich bei allen Zeilen, die mit einem Prozentzeichen (%) beginnen, um Kommentare handelt. Die erste Zeile (*documentclass*) legt fest, dass wir die Papiergröße A4 haben wollen unter Verwendung einer Standardschrift mit 11 Punkten. Weiterhin sagen wir, dass es sich bei der vorliegenden Arbeit um einen Artikel handelt. Es gibt noch andere Formen wie Buch, aber das soll nicht weiter interessieren.

Nun beginnt das eigentliche Dokument. Es wird von *begin* und *end* eingeschlossen. Unser Dokument enthält im Moment nur einen Satz. Man muss die Quelldatei an Latex übergeben und es erzeugt dann die entsprechende Ausgabe, z. B. als PDF. An dieser Stelle verweise ich auf die Dokumentation der verwendeten Werkzeuge. Das fertige Ergebnis kann [hier](#) bestaunt werden.

Beispiel 2

Nun hat uns dieses Beispiel sicher nicht besonders beeindruckt. Deshalb gehen wir gleich einen Schritt weiter ([beispiel02.tex](#)):

```
% das Papierformat zuerst
\documentclass[a4paper, 11pt]{article}

% hier beginnt das Dokument
\begin{document}

% Inhaltsverzeichnis anzeigen
\tableofcontents

% Abschnitt soll auf naechster Seite beginnen
\newpage

% Abschnittsueberschrift
\section{Einleitung}

% Ueberschrift eines Abschnittes
\subsection{Motivation}

Dies ist ein Satz.

% Ueberschrift eines Abschnittes
\subsection{Abgrenzung}

Dies ist ein Satz.

% Abschnitt soll auf naechster Seite beginnen
\newpage

% Abschnittsueberschrift
\section{Theoretische Betrachtungen}

Dies ist ein Satz.

% das ist wohl jetzt das Ende des Dokumentes
\end{document}
```

Das [Ergebnis](#) sieht nun schon wesentlich interessanter aus. Wir haben insgesamt 3 Seiten mit 2 Abschnitten und einem Inhaltsverzeichnis. Das Inhaltsverzeichnis wird mit dem

Befehl `tableofcontents` erzeugt und an der Stelle in das Dokument eingefügt. Der Befehl `newpage` bewirkt, dass eine neue Seite angefangen wird. Wir wollen ja schließlich nicht, dass Inhaltsverzeichnis und Abschnitt 1 auf der gleichen Seite stehen.

Es wurde ja bereits darauf hingewiesen, dass Latex nicht wissen kann, wann es sich bei einem Satz um eine Überschrift handelt. Deshalb müssen Überschriften mit den Befehlen `section`, `subsection` und `subsubsection` gekennzeichnet werden. Dabei ergibt sich die Hierarchie der einzelnen Überschriften. Man kann dies sehr schön am [erzeugten Dokument](#) nachvollziehen.

Jetzt haben wir gesehen, wie einfach es ist Überschriften zu erzeugen und ein Inhaltsverzeichnis zu erstellen. Damit sind die wesentlichen Sachen bereits geklärt. Wir können nun zu den Spezialfällen übergehen ;-)

[Zum Seitenanfang...](#)

5. Umlaute und deutsche Sprachunterstützung

Vielleicht ist es ja aufgefallen, im Beispiel 2 hatte das Inhaltsverzeichnis die englische Überschrift "Contents". Dies muss natürlich geändert werden. Weiterhin wurde in keinem Beispiel bis jetzt ein Umlaut oder das ß verwendet. Auch das wollen wir nun ändern. Dazu betrachte man das nächste Beispiel ([beispiel03.tex](#)):

```
% das Papierformat zuerst
\documentclass[a4paper, 11pt]{article}

% deutsche Silbentrennung
\usepackage[ngerman]{babel}

% wegen deutschen Umlauten
\usepackage[ansinew]{inputenc}

% wir wollen auf jeder Seite eine Ueberschrift
\pagestyle{headings}

% hat was mit Abstaenden zu tun
\frenchspacing

% hier beginnt das Dokument
\begin{document}

% Inhaltsverzeichnis anzeigen
\tableofcontents

% Abschnitt soll auf naechster Seite beginnen
\newpage

% Abschnittsueberschrift
\section{Einleitung}

% Ueberschrift eines Abschnittes
\subsection{Motivation}

Ein bekannter Stolperstein bei Latex sind die deutschen Umlaute wie: ä, ö, ü,
Ä, Ö, Ü und natürlich auch das ß.

% Abschnitt soll auf naechster Seite beginnen
\newpage

% Abschnittsueberschrift
\section{Theoretische Betrachtungen}
```

Dies ist ein Satz.

```
% das ist wohl jetzt das Ende des Dokumentes
\end{document}
```

Nach dem obligatorischen Latexlauf (meist 3mal wiederholt) erhält man [folgendes Ergebnis](#). Es fällt auf, dass das Inhaltsverzeichnis nun auch wirklich "Inhaltsverzeichnis" heißt. Die Umlaute und das ß werden ebenfalls richtig dargestellt und jede Seite weist nun die Abschnittsüberschrift auf. Die Unterstützung der deutschen Sprache erhalten wir aus den Werkzeugpaketen *babel* und *inputenc*. Der Befehl *usepackage* sagt Latex, dass wir diese Werkzeugpakete nutzen wollen. Die Pakete werden am Anfang eingebunden. Die Seitenbeschriftung ist durch den Befehl *pagestyle* gesetzt wurden. Falls einem die Seitenbeschriftung nicht gefällt, kann man natürlich auch einen anderen Seitenstil wählen. Weiterhin wurde der Befehl *frenchspacing* aufgenommen. Dieser bewirkt, dass der Abstand zwischen Worten sowie zwischen Satzzeichen und nächstem Wort identisch ist. Das ist wohl so üblich im europäischen Sprachraum.

[Zum Seitenanfang...](#)

6. Grafiken

Auch die Einbindung von Grafiken ist denkbar einfach. Man sollte dabei immer versuchen, Vektorgrafiken einzubinden. Die meisten gängigen Vektorzeichenprogramme wie CorelDraw, [Dia](#) oder [Inkscape](#) beherrschen den Export ihrer Grafiken in das Postscriptformat (ps oder eps). Es ist auch möglich in Latex normale Pixelgrafiken wie JPG zu nutzen, dies ist aber nicht empfehlenswert, da diese Grafiken nur unter Qualitätsverlust skaliert werden können und man generell ein schlechteres Druckergebnis erhält. Bevor die Einbindung von Grafiken genau erläutert wird, zunächst das Beispiel ([beispiel04.tex](#)):

```
% das Papierformat zuerst
\documentclass[a4paper, 11pt]{article}

% deutsche Silbentrennung
\usepackage[ngerman]{babel}

% wegen deutschen Umlauten
\usepackage[ansinew]{inputenc}

% Grafikpaket laden
\usepackage{graphicx}

% hier beginnt das Dokument
\begin{document}

% das Abbildungsverzeichnis
\listoffigures

% Grafik auf neuer Seite anzeigen
\newpage

Jetzt soll eine Grafik angezeigt werden. Dabei werden wir feststellen, dass die
Grafik über diesem Satz angezeigt wird, obwohl wir den Befehl für die Grafik
erst in den nächsten Zeilen geben.
\begin{figure}
  \centering
  \includegraphics{grafik}
  \caption{eine Grafik ohne Sinn und Verstand}
  \label{img:grafik-dummy}
\end{figure}

% eine weitere Seite
\newpage
```

Weiterhin wollen wir an dieser Stelle Bezug auf die Grafik `\ref{img:grafik-dummy}` auf Seite `\pageref{img:grafik-dummy}` nehmen, was uns hiermit gelungen sein dürfte. Latex passt die Seitenzahl aber auch die Nummer der Grafik automatisch an, wir müssen uns um nichts kümmern.

Bis jetzt wurde es noch nicht erwähnt, aber man fügt einen neuen Absatz im Text ein, indem man eine Leerzeile lässt, wie hier gerade gezeigt wurde.

```
% das ist wohl jetzt das Ende des Dokumentes
\end{document}
```

Das Ergebnis ist [hier](#) zu finden. Die Grafik wurde mit [Dia](#) erstellt und findet sich [hier als Dia-Datei](#) sowie [hier als Postscript-Datei](#).

Die Werkzeuge zum Umgang mit Grafiken sind im Paket *graphicx* gesammelt. Dieses Paket müssen wir natürlich am Anfang einbinden. Weiterhin soll am Anfang des Dokumentes ein Abbildungsverzeichnis stehen. Dieses wird durch den Befehl *listoffigures* erstellt und angezeigt.

Eine Grafik wird in eine so genannte Umgebung (*figure*) eingebettet. Damit wird Latex verständlich gemacht, dass es sich bei der Grafik sowie der entsprechenden Bildunterschrift um ein unzertrennbares Objekt handelt. Bei der Grafik wurde als Ausrichtung zentriert (*centering*) gewählt. Als Dateiname wurde [grafik](#) angegeben und es wurde eine Bildunterschrift (*caption*) gewählt. Man kann die Dateierweiterung der Grafikdatei weglassen, da automatisch *.eps* bei der DVI Erstellung angehängt wird. Weiterhin wurde der Grafik ein eindeutiger Bezeichner (`img:grafik-dummy`) mittels des *label* Befehls zugewiesen.

Latex nimmt sich die Freiheit heraus, Umgebungen (*figure*) im Text so hin und her zu schieben, dass sie möglichst optimal verteilt sind. So betrachtet es Latex als vorteilhaft, wenn die Grafik am Seitenanfang steht. Deshalb wird in den meisten Fällen die Grafik nicht an der Position angezeigt, wo sie in der Quelldatei eingefügt wurde. Dies ist sicherlich am Anfang etwas verwirrend und nicht unbedingt einsichtig. Bei größeren Dokumenten lernt man dies dann aber zu schätzen. Es soll erwähnt werden, dass die Positionierung beeinflusst werden kann. Dazu sollte man die Latexdokumentation studieren.

Es ist natürlich möglich einen Bezug auf die Grafik herzustellen. Dazu dient der eindeutige Bezeichner. Mittels der Befehle *ref* und *pageref* kann man dann ganz einfach den Bezug herstellen, wie weiter unten im Beispiel gezeigt wurde. Es ist zu beachten, dass man in Latex niemals den Unterstrich (`_`) z. B. für Dateinamen oder Bezeichner verwenden sollte, da es dann zu Problemen kommt.

[Zum Seitenanfang...](#)

7. Fußnoten und Verweise

In diesem Beispiel soll gezeigt werden, wie man Fußnoten verwendet und auf Textstellen verweisen kann. Im vorherigen Beispiel wurde ja bereits gezeigt, wie man einen Verweis auf ein Bild erstellt. Zunächst das Beispiel ([beispiel05.tex](#)):

```
% das Papierformat zuerst
\documentclass[a4paper, 11pt]{article}

% deutsche Silbentrennung
\usepackage[ngerman]{babel}

% wegen deutschen Umlauten
\usepackage[ansinew]{inputenc}

% hier beginnt das Dokument
```

```
\begin{document}
```

```
\section{Abschnitt}
\label{sec:abschnitt}
```

```
\subsection{Unterabschnitt}
\label{subsec:unterabschnitt}
```

Einzelne Worte kann man `\emph{hervorheben}`. Man sollte dieses Mittel aber sehr sparsam einsetzen. In diesem Beispiel soll es um Fußnoten`\footnote{\label{foot:1}}`Dies ist jetzt eine Fußnote.) gehen.

```
% neue Seite
\newpage
```

```
\section{Weiterer Abschnitt}
```

```
\subsection{Weiterer Unterabschnitt}
```

Weiterhin soll in diesem Beispiel erwähnt sein, wie man Verweise auf andere Abschnitte erstellt, also z.~B. auf Abschnitt `\ref{sec:abschnitt}` auf Seite `\pageref{sec:abschnitt}`. Dabei ist es natürlich wichtig, dass man die Labels richtig setzt. So gibt es natürlich auch noch Abschnitt`\footnote{\label{foot:2}}`Und dies ist natürlich auch eine Fußnote.) `\ref{subsec:unterabschnitt}` auf Seite `\pageref{subsec:unterabschnitt}`. Man kann natürlich auch auf Fußnoten wie Fußnote `\ref{foot:1}` auf Seite `\pageref{foot:1}` und Fußnote `\ref{foot:2}` auf Seite `\pageref{foot:2}` verweisen.

```
% das ist wohl jetzt das Ende des Dokumentes
\end{document}
```

Hier findet man das [Ergebnis](#). Die Erstellung einer Fußnote ist denkbar einfach. An der gewünschten Stelle den Befehl `footnote` mit dem Fußnotentext aufrufen.

Will man auf eine Stelle verweisen, dann muss man zunächst diese Stelle markieren. Dies geschieht wieder mit dem `label` Befehl. Man gibt der Stelle einen eindeutigen Bezeichner. Man sollte sich eine kleine Systematik für die Bezeichner schaffen, damit man am Bezeichnernamen erkennen kann, ob es sich um einen Bezeichner für ein Bild, eine Fußnote, eine Tabelle, ein Abschnitt usw. handelt. Den Verweis erstellt man dann mit den Befehlen `ref` und `pageref`.

Weiterhin wurde in diesem Beispiel gezeigt, wie man beispielsweise die Abkürzung "z. B." richtig schreibt. Man setzt ein festes Leerzeichen zwischen die beiden Buchstaben. Das feste Leerzeichen wird mit einer Tilde (~) gekennzeichnet. Latex weiß dadurch, dass es an dieser Stelle keinen Zeilenumbruch einfügen darf.

[Zum Seitenanfang...](#)

8. Deckblatt

Das größte Problem stellte für mich die Gestaltung des Deckblattes dar. Für dieses Deckblatt bestanden ziemlich konkrete Vorgaben durch den Fachbereich. Hier zeigte sich dann Latex natürlich von seiner ganz schwachen Seite. Denn für solch umfangreiche eigenhändige Formatierungen muss man sehr viel Aufwand treiben und sich an die alten HTML Layouttricks erinnern. An dieser Stelle soll der Quelltext nicht gezeigt werden, da er einfach nur hässlich ist, aber es funktioniert. Bei Interesse steht [hier der Quelltext](#) bereit und [hier natürlich das Ergebnis](#).

[Zum Seitenanfang...](#)

9. Zur Organisation

Eine wissenschaftliche Arbeit besteht nun natürlich nicht nur aus wenigen Worten, sondern ist sehr umfangreich. Deshalb ist es nicht sinnvoll, die komplette Arbeit in einer einzigen Datei abzulegen. Vielmehr macht es Sinn die Arbeit auf mehrere Quelldateien zu verteilen. So kann man für jeden Abschnitt eine eigene Datei benutzen. Diese verschiedenen Dateien müssen dann natürlich noch in einem Projekt zusammengeführt werden. Dazu ist es ratsam zuerst eine Hauptdatei zu erstellen, die lediglich die ganzen zentralen Anweisungen wie Schriftgröße (*documentclass*), allgemeinen Seitenstil (*pagestyle*), die verwendeten Pakete (*usepackage*) usw. enthält. In diese Datei werden dann die einzelnen Abschnitte reingehängt. Dies geschieht mit dem *input* Befehl. In den geschweiften Klammern wird lediglich die entsprechende Latexdatei angegeben.

Für Latex macht es keinen Unterschied, ob man die Abschnitte auf einzelne Dateien auslagert oder seinen ganzen Text lediglich in einer riesigen Datei ablegt. Um aber nicht die Übersicht zu verlieren ist eine gewisse Organisation unverzichtbar.

[Zum Seitenanfang...](#)

10. Literaturverzeichnis

Jetzt wollen wir einen Schritt weiter gehen und eine Literaturliste aufbauen. Der ordentliche Literaturnachweis ist ja eine der wichtigsten aber wahrscheinlich auch zeitaufwendigsten Punkte bei einer wissenschaftlichen Arbeit. Deshalb kann jede Unterstützung nur recht sein.

In Latex wird der Literaturnachweis folgendermaßen gelöst. Man erstellt zunächst eine Datei, die für die einzelnen Werke die bibliografischen Daten (Autor, Titel, Erscheinungsjahr, Verlag, ...) enthält. Jedes Werk bekommt wiederum einen eindeutigen Bezeichner, wie bei den Verweisen. Im Text wird dann bei einem Zitat der entsprechende Bezeichner mittels eines Befehls angegeben. Weiterhin kann man das Literaturverzeichnis an einer beliebigen Stelle im Dokument anzeigen lassen.

Da die Datei mit den einzelnen Werksangaben keine Latex Datei ist, muss sie nach jeder Änderung zunächst durch ein Zusatzprogramm verarbeitet werden. Dieses Programm ist *bibtex*. In den meisten guten Latexbedienoberflächen ist ein Editor für die Werksangaben integriert und auch der Aufruf von *bibtex* wird für den Nutzer gut versteckt. Trotzdem muss wahrscheinlich jeder selbst herausfinden, wie der genaue Ablauf in seinem verwendeten Latexsystem ist. Hier nun das Beispiel. Zunächst die Datei mit den Angaben zu den einzelnen Werken ([literatur.bib](#)):

```
@INBOOK{brooks2001-silver,
  TITLE="The Mythical Man-Month",
  CHAPTER="No Silver Bullet - Essence and Accident in Software Engineering",
  AUTHOR="Frederick P. Brooks",
  PAGES="177--203",
  YEAR=2001,
  PUBLISHER="Addison Wesley Longman Inc.",
  ADDRESS="New York",
  EDITION="15., Aufl."
}

@BOOK{mueller2000,
  AUTHOR="Johann-Adolf Müller",
  TITLE="Systems Engineering",
  PUBLISHER="Manz-Verlag Schulbuch (Fortis)",
  YEAR=2000,
  ADDRESS="Wien"
```

```

}

@TECHREPORT{paschel1992,
  AUTHOR="Markus Pasche",
  TITLE="Synergetik und Evolutorische Ökonomik",
  INSTITUTION="Universität Hannover Fachbereich Wirtschaftswissenschaften",
  TYPE="Diskussionspapier",
  NUMBER=179,
  MONTH=dec,
  YEAR=1992
}

```

Man erkennt, dass je nach Werk unterschiedliche Angaben notwendig sind. Auch müssen die einzelnen Werke entsprechend ihrer Art (Buch, Artikel, Dissertation, usw.) gekennzeichnet werden. Schauen wir uns nun eine Quelldatei an, die die oben aufgeführten Werke nutzt ([beispiel07.tex](#)):

```

% das Papierformat zuerst
\documentclass[a4paper, 11pt]{article}

% deutsche Silbentrennung
\usepackage[ngerman]{babel}

% wegen deutschen Umlauten
\usepackage[ansinew]{inputenc}

% fuer Zitate
\usepackage[round]{natbib}

% hier beginnt das Dokument
\begin{document}

% Festlegung Art der Zitierung - Havardmethode: Abkuerzung Autor + Jahr
\bibliographystyle{alphadin}

% Inhaltsverzeichnis erzeugen
\tableofcontents

% und nun Text mit Zitaten
\newpage
\section{Kapitel mit Zitaten}

Richtig zu zitieren ist nicht einfach. Da gibt es eine Unmenge an Regeln zu beachten. Deshalb sollte man ein entsprechendes Buch mal lesen oder im Internet schauen. Ich werde jetzt nicht erklären, wie man richtig zitiert. Hier nur ein paar Beispiele. So möchte ich z.B. mal den "Silver-Bullet"
\citep[vgl.][] {brooks2001-silver} erwähnen. Ein System\index{System} ist definiert als "eine Menge von Elementen, die miteinander durch Beziehungen verbunden sind und gemeinsam einen bestimmten Zweck zu erfüllen haben."
\citep[S.~48] {mueller2000}

\ldots

Da in der Synergetik eine Reihe von neuen Begriffen eingeführt werden, erfolgt die Darstellung an einem Beispiel \citep[nach] [S.~6] {paschel1992}. \ldots

% Literaturliste soll im Inhaltsverzeichnis auftauchen
\newpage
\addcontentsline{toc}{section}{Literatur}

% Literaturliste endgueltig anzeigen
\bibliography{literatur}

% das ist wohl jetzt das Ende des Dokumentes
\end{document}

```

Unter Windows habe ich das Beispiel nicht ausprobiert. Um in Unix das Beispiel zu verarbeiten, sind folgende Aufrufe

notwendig:

1. latex beispiel07.tex
2. bibtex beispiel07
3. latex beispiel07.tex
4. latex beispiel07.tex
5. latex beispiel07.tex

Man erhält dann folgendes [Ergebnis](#). Zunächst einmal muss das entsprechende Werkzeugpaket eingebunden werden. Ich habe bei mir *natbib* intern durch *natdin* erweitert. Dies ist eine Abwandlung der Originalversion, um die in Deutschland verbindliche DIN 1505 Teil 2 und Teil 3 umzusetzen. Das *natdin* Paket findet [man im Internet](#). Ich habe die Dateien von dieser Seite in meinen *natbib* Pfad (`/usr/share/texmf/bibtex/bst/natbib`) kopiert und danach das Programm *texhash* als Root aufgerufen. Unter MikTeX muss man zunächst das Paket *natbib* mit dem Paketmanager installieren. Um DIN konforme Literaturangaben zu erhalten, muss man sich jetzt noch die entsprechenden *.bst Dateien von oben genannter Internetseite besorgen. Die Dateien werden in das Verzeichnis `/bibtex/bst` kopiert. Der Übersichtlichkeit wegen kann auch ein Unterverzeichnis innerhalb von `bibtex/bst` angelegt werden, z. B. `/bibtex/bst/natdin` in welches dann die Dateien hineinkopiert werden. Nun müsste das obige Beispiel - nach der Installation von *natbib* und dem Herunterladen von *alphadin.bst* - funktionieren.

Mit dem Befehl *bibliographystyle* wird die Zitierart ausgewählt. Am Anfang braucht man sicher hier nichts dran rumzustellen. Nun kann man an beliebiger Stelle ein Zitat mit dem Befehl *citep* einfügen. In den geschweiften Klammern wird der Bezeichner aus der Werkdatei angegeben. Weiterhin stehen 2 eckige Klammern zur Verfügung. Mit diesen kann man den Quellenhinweis um weitere Hinweise wie Seitenzahl erweitern. Da sich sowas schlecht erklären lässt, einfach mal einen Blick in das [fertige Dokument](#) werfen!

Mit dem Befehl *bibliography* wird das Literaturverzeichnis erstellt und angezeigt. In der geschweiften Klammer muss die Datei mit den Werkangaben ohne Dateiendung angegeben werden.

Weiterhin wurde mit dem Befehl *addcontentsline* dafür gesorgt, dass im Inhaltsverzeichnis ein Eintrag für das Literaturverzeichnis auftaucht. Es sei noch verwiesen auf den Befehl *Idots*. Dieser fügt 3 Gedankenpunkte ein.

An dieser Stelle sei noch ein allgemeiner Hinweis gestattet. Sobald man sich Textstellen aus einem Buch notiert um sie eventuell zu zitieren oder sich zumindest auf das Buch zu beziehen, sollte man sich sofort alle nötigen bibliografischen Daten notieren. Als sehr bequem empfand ich es, das Buchcover sowie die Seite mit dem Impressum zu kopieren. Falls man dann wirklich irgendwann das Zitat verwenden möchte, hat man alle nötigen bibliografischen Daten zur Hand. Eine Alternative besteht darin, die nötigen Daten bei einem Onlinebuchhändler rauszusuchen.

[Zum Seitenanfang...](#)

11. Stichwortverzeichnis

Es ist im Rahmen einer wissenschaftlichen Arbeit nicht unbedingt erforderlich ein Stichwortverzeichnis zu erstellen. Zunächst das Beispiel ([beispiel08.tex](#)):

```
% das Papierformat zuerst
\documentclass[a4paper, 11pt]{article}

% deutsche Silbentrennung
\usepackage[ngerman]{babel}
```

```

% wegen deutschen Umlauten
\usepackage[ansinew]{inputenc}

% fuer Stichwortverzeichnis
\usepackage{makeidx}

% Stichwortverzeichnis erstellen
\makeindex

% hier beginnt das Dokument
\begin{document}

% Inhaltsverzeichnis
\tableofcontents

% eine Seite mit wichtigen Begriffen
\newpage
\section{Kapitel}

Auf dieser Seite sind die Begriffe System\index{System},
Systemzustand\index{System!Zustand}, Systemelement\index{System!Element} und
Emergenz\index{Emergenz} von Interesse.

% eine Seite mit wichtigen Begriffen
\newpage
\section{Weiteres Kapitel}

Auf dieser Seite sei nochmals was gesagt zu System\index{System} und
Systemelement\index{System!Element} sowie zu
Transformationsprozess.\index{Transformationsprozess}

% Index soll Stichwortverzeichnis heissen
\newpage
\renewcommand{\indexname}{Stichwortverzeichnis}

% Stichwortverzeichnis soll im Inhaltsverzeichnis auftauchen
\addcontentsline{toc}{section}{Stichwortverzeichnis}

% Stichwortverzeichnis endgueltig anzeigen
\printindex

% das ist wohl jetzt das Ende des Dokumentes
\end{document}

```

Das fertige Ergebnis befindet sich [hier](#). Der Index muss ebenfalls durch ein extra Programm erzeugt werden. Dafür reicht der Aufruf von "makeindex beispiel08". Allerdings ist das so erzeugte Stichwortverzeichnis nicht besonders schön. Deshalb habe ich eine [Stildatei verwendet](#), um das Stichwortverzeichnis meinen Vorstellungen entsprechend anzupassen. Bei Verwendung dieser Stildatei lautet der Aufruf dann "makeindex -g -s index.ist beispiel08". Dies kann natürlich bei anderen Latexsystemen abweichen. Die [Stildatei](#) soll an dieser Stelle nicht erläutert werden.

Die Befehle für das Stichwortverzeichnis sind relativ einfach zu verstehen. Zunächst muss das Werkzeugpaket *makeidx* eingebunden werden. Dann wird das Stichwortverzeichnis (Index) mit dem Befehl *makeindex* erstellt und an der gewünschten Stelle mit *printindex* angezeigt. Würde man keine Modifikationen vornehmen, dann hätte das Stichwortverzeichnis die Überschrift "Index". Mit dem Befehl *renewcommand* kann "Index" durch "Stichwortverzeichnis" ersetzt werden. Weiterhin wurde wiederum mit dem Befehl *addcontentsline* ein entsprechender Eintrag für das Stichwortverzeichnis in das Inhaltsverzeichnis aufgenommen.

Die Worte, die im Stichwortverzeichnis erscheinen sollen, werden auf den gewünschten Seiten mit dem Befehl *index* gekennzeichnet. Dabei muss ein entsprechender

Indexbefehl auf jeder Seite verwendet werden, die im Stichwortverzeichnis bei dem entsprechenden Wort auftauchen soll.

Am Beispiel kann man weiterhin erkennen, dass man Stichworte anderen Stichworten unterordnen kann. Im Verzeichnis wird diese Hierarchie dargestellt.

[Zum Seitenanfang...](#)

12. Rechtschreibprüfung

In vielen Latexbedienoberflächen wird die Rechtschreibprüfung bereits integriert. Dann sind meist keine weiteren Schritte notwendig. Unter Unix kann man die Rechtschreibung jeder Latexdatei mit folgendem Befehl prüfen:

```
aspell -t -x -c DATEI.tex
```

Als sehr nützlich hat sich auch erwiesen, die Rechtschreibung mit einem weiteren unabhängigen Programm gegenzuprüfen. Dazu habe ich aus der Latexdatei ein RTF mittels des Programms latex2rtf erzeugen lassen. Dieses RTF konnte ich dann in einem Officepaket prüfen lassen. Dies war sehr nützlich, da ich so noch weitere Fehler feststellen konnte. Allerdings sollte man keiner Rechtschreibprüfung voll vertrauen, denn alle Rechtschreibprüfungen haben auch neue Fehler in die Dokumente eingefügt.

[Zum Seitenanfang...](#)

13. Makefile für Unix

Für alle Unixnutzer möchte ich noch mein [Makefile](#) zur Verfügung stellen. Diese bündelt alle Kommandos und in den meisten Fällen reicht ein Aufruf von *make all* um eine aktuelle Version zu erstellen. Das Makefile ist vollständig dokumentiert und man sollte einen Blick dort hinein werfen, falls man sie nutzen möchte. Wahrscheinlich funktioniert das Makefile aber nur, wenn man GNU make einsetzt.

[Zum Seitenanfang...](#)

14. Zweispaltiges Layout

Ab und zu wird man sicher mal ein zweispaltiges Layout für ein Dokument wollen. Auch sind oft weitere Layoutvorlagen nicht schlecht, um sich ein paar Kniffe abzuschauen. Die IEEE Organisation stellt ihren Autoren eine [Vorlage zur Verfügung](#), die für Veröffentlichungen in IEEE Zeitschriften zu verwenden ist. Im Paket ist eine Beispieldatei enthalten. Diese zeigt, wie ein fertig formatiertes Dokument am Ende aussieht. Ansonsten findet man viele weitere Vorlagen bei verschiedenen Universitäten und wissenschaftlichen Verlagen.

[Zum Seitenanfang...](#)

15. Und doch ein paar Bücher

Nagut, auf vielfachen Wunsch doch ein paar Buchempfehlungen zum Thema Latex. Ich geb ja zu, ich besitze auch eins, was ich auch gar nicht schlecht finde. Wenn man sein Dokument bearbeitet, will man ab und zu mal einen Befehl nachschlagen. Dies ist nicht weiter ein Problem, wenn man über eine Standleitung mit dem Internet ständig verbunden ist - im Internet findet man sehr viele Seiten zu den einzelnen Latexbefehlen.

Eine sehr schöne Befehlsreferenz findet man in [folgendem](#)

[Buch](#), was auch den Vorteil hat preiswert zu sein:



Latex kurz & gut

Autor: Matthias K. Dalheimer
Preis: 8,00 Euro
ISBN: 3897215004
Verlag: O'Reilly

[Bestellung bei amazon.de](#)

Will man etwas mehr Geld ausgeben, kann man auch mehr Erläuterungen zum Gesamtkonzept erwarten. Es gibt aber wirklich keinen Grund sich ein Latexbuch für 70 Euro und mehr zu kaufen. Auch eine beiliegende CD-Rom rechtfertigt diesen Preis nicht. Alle Programme, die man benötigt, stehen schließlich kostenlos im Internet zur Verfügung.

Ein Buch sowohl mit Erklärungen zum Gesamtkonzept als auch zu den einzelnen Befehlen ist das hier gezeigte:



LaTeX GE-PACKT

Autor: Karsten Günther
Preis: 15,95 Euro
ISBN: 3826615883
Verlag: mitp

[Bestellung bei amazon.de](#)

[Zum Seitenanfang...](#)

16. Abschließende Worte

Irgendwie habe ich das Gefühl, dass ich was vergessen habe. Naja, es gibt sicher noch den einen oder anderen Tipp, aber ich denk man kommt schon mit dem hier Gesagtem ziemlich weit. Auch muss man auf keinen Fall am Anfang alles ausprobieren bzw. einsetzen. Es reicht erst mal Latex überhaupt zum Laufen zu bringen und mit Überschriften, Umlauten und normalen Text klar zu kommen. Der Rest kommt später.

Falls also etwas fehlt oder unklar ist, dann einfach eine Mail an mich, Adresse steht unten in der Fußzeile.

Ich habe natürlich Latex nicht aus Spaß an der Freud eingesetzt, sondern für meine Diplomarbeit. Wer das fertige Ergebnis sehen will, muss [hier schauen...](#)

[Zum Seitenanfang...](#)

[Impressum und Kontakt](#)

Letzte Änderung: 25.05.2007

webmaster-AT-hpfsc.de (-AT- ersetzen durch @)