

Einführung in Textsatz mit \LaTeX

Andreas Dähn

25.10.2006

Inhaltsverzeichnis

1	Einleitung	4
2	LaTeX, erster Kontakt	4
2.1	Was ist LaTeX	4
2.2	Was unterscheidet LaTeX von einer Textverarbeitung?	4
2.3	Bezug von LaTeX, Installation	4
2.3.1	Windows	4
2.3.2	Linux	5
2.3.3	BSD	5
2.3.4	MacOS X	5
2.3.5	Installation erfolgreich?	5
3	Der erste LaTeX-Lauf	5
3.1	Dokument erstellen	5
3.2	Von der Eingabe zur Ausgabe	6
4	Der Grundaufbau eines LaTeX-Dokumentes	6
4.1	Präambel	7
4.2	Inhalt	7
5	LaTeX-Syntax	7
5.1	Befehle	7
5.2	Umgebungen	8
5.3	Sonstige Anweisungen	8
5.3.1	Kommentare	8
5.3.2	Einbinden von weiteren Dateien	8
5.3.3	Maskierung	8
5.3.4	Weglassen von Klammern	8
6	LaTeX in der Anwendung	9
6.1	Probleme	9
6.1.1	Geschütztes Leerzeichen	9
6.1.2	Anführungszeichen	9
6.1.3	Umlaute und Kodierung	9
6.2	Gliederungsbefehle	10
6.2.1	Inhaltsverzeichnis	10
6.2.2	Gliederung erstellen	10
6.2.3	Titelseite	10
6.3	Mathesatz	11
6.3.1	Mathe-Umgebungen	11
6.3.2	Grundlegende Konstruktionen	11
6.3.3	Griechische Buchstaben	12
6.3.4	Mathematische Sonderzeichen	12

6.3.5	Funktionsnamen	14
6.4	Vielgenutzte Umgebungen	14
6.4.1	itemize oder: „Listen, unnumeriert“	14
6.4.2	description oder: „Liste, eigene Überschrift“	15
6.4.3	enumerate oder: „Listen, numeriert“	15
6.4.4	verbatim oder: „Nur-Text“	15
6.4.5	tabular oder: „Tabelle, simpel“	15
6.4.6	lstlisting oder: Programmlistings einbinden	16
6.4.7	minipage oder: Seite in der Seite	16
6.4.8	Literaturverzeichnis	17
6.5	Graphiken im eps-Format einbinden	17
6.6	Fußnoten	18
6.7	Textformatierung	18
6.7.1	Textteile hervorheben	18
6.7.2	Schriften	18
6.7.3	Schriftanschnitt und -größe	19
6.8	Zeilen- und Seitenumbruch	19
7	DVI, Postscript und PDF	19
7.1	Was ist PostScript?	20
7.2	Was ist PDF?	20
8	Fehlersuche	20
9	Quellen	21
9.1	Internet	21
10	Changelog	22

1 Einleitung

Dies ist die ausführliche und ausformulierte Version des Vortrags „Einführung in LaTeX“, die im Rahmen des Vorlesungsseminars zum Thema „Medien und Gestaltung“¹ am 25.10.2006 gehalten wurde.

2 LaTeX, erster Kontakt

2.1 Was ist LaTeX

LaTeX ist eine Sammlung von Makros für das Textsatzsystem TeX. Diese Sammlung wurde von Leslie Lamport im Jahr 1984 begonnen und LaTeX steht für „Lampports TeX“.

2.2 Was unterscheidet LaTeX von einer Textverarbeitung?

Mit einer Textverarbeitung werden die meisten Leser bereits einmal in Kontakt gekommen sein. Textverarbeitungen gehen nach einem radikal anderen Konzept vor als LaTeX dies tut. Das Textverarbeitungssystem versucht, die Eingabe des Benutzers bereits auf dem Bildschirm während der Eingabe so darzustellen, wie es einmal auf dem Papier aussehen soll. LaTeX hat dieses Ziel nicht. Hier wird eine Quelltextdatei geschrieben; um eine Druckvorschau in gesetzter Form zu erhalten ist es notwendig, einen LaTeX-Lauf durchzuführen. Dafür stellt das Erstellen von LaTeX-Dokumenten wesentlich geringere Voraussetzungen an den Computer, auf dem es erfolgt. Hierzu ist nur die Rechenleistung zur Darstellung eines Editors notwendig, wenn der LaTeX-Lauf auf eine leistungsfähigere Maschine ausgelagert wird. Ferner fällt auf, daß LaTeX mit Textdateien als Eingabe arbeitet, wohingegen die meisten Textverarbeitungen ein binäres Format zur Speicherung von Daten verwenden (in dem dann auch eingebettete Objekte und weitere Metainformationen Platz finden).

2.3 Bezug von LaTeX, Installation

2.3.1 Windows

LaTeX kann man kostenlos herunterladen. Für die Windows gibt es die Distribution „MiKTeX“. Die Distribution kann unter <http://www.miktex.de> heruntergeladen werden. Mit dem „kleinen“ Setup-Programm werden dann die einzelnen Pakete heruntergeladen; in einem zweiten Durchlauf wird das Programm dann auf dem Rechner installiert. Danach stehen in der Windows-Shell die zentralen (hier besprochenen) Kommandos „latex“, „dvips“ und „ps2pdf“ sowie „psnup“ und „pdflatex“ zur Verfügung. Wichtig: Anstelle von „xdvi“ heißt der DVI-Viewer „yap“².

¹angeboten von Fr. Prof. Schumann

²Dies steht für „yet another previewer“; im folgenden wird dennoch stets von „xdvi“ die Rede sein.

2.3.2 Linux

Linux-Benutzer müssen sich an ihrer Distribution orientieren - die meisten Distributionen bieten ein Paket namens „tetex“, welches installiert werden sollte. Wie genau das vor sich geht, sollte sich unter zuhilfenahme der Distributionsdokumentation des jeweiligen Linux klären lassen.

2.3.3 BSD

Entweder LaTeX aus den Ports selbst kompilieren (/usr/ports/print/texTeX) oder als binäres Paket installieren.

2.3.4 MacOS X

Benutzer von Macintoshs MacOS X können LaTeX mittels der DarwinPorts oder Fink installieren, ausserdem gib es das Paket „MacTeX“, welches einen graphischen Installer bietet.

2.3.5 Installation erfolgreich?

Um zu überprüfen, ob die Installation von LaTeX erfolgreich verlaufen ist, gebe man in einer Shell³ den Befehl „latex“ ein. Wenn keine Fehlermeldung sondern ein Text im Stile von „This is *TeX. Version ****“ ausgegeben wird, so war die Installation erfolgreich.

3 Der erste LaTeX-Lauf

3.1 Dokument erstellen

Das Dokument wird zunächst als Quelltext erstellt – also ohne jede Formatierung. Hierzu kann ein beliebiger Editor, von „Notepad“ bis „Notepad2“, von „KWrite“ bis „vi“ benutzt werden. Wichtig ist, daß eine Datei erzeugt wird, die selbst keine Formatierung sondern nur den eingegeben Text enthält. Diese wird mit einem Dateinamen wie „test.tex“ erzeugt⁴.

Um zu beginnen, erstelle nun ein Dokument, „eins.tex“. Was erwartet LaTeX von uns? Zunächst einmal benötigt LaTeX eine Information darüber, was für eine Art Dokument in der gegebenen Datei enthalten ist. Hier soll einmal ein Artikel angedeutet werden. Nach dieser Information kann der eigentliche Inhalt des Dokumentes beginnen. Im Code sieht das so aus:

```
1 \documentclass{article}
2 \begin{document}
3 Hier ist mein supertoller Text.
4 \end{document}
```

³Windows-Benutzer starten unter Windows 2000 und XP eine „cmd“; unter Windows 95, 98, 98SE, ME eine „command.com“, wer ein Linux oder Unix benutzt nimmt die Shell seiner Wahl.

⁴Dem LaTeX-Interpreter selbst ist es ziemlich egal, welche Extension die Datei hat; gebräuchlich sind „.tex“ und „.latex“.

3.2 Von der Eingabe zur Ausgabe

Um diese Datei nun in eine „ansehbare“ Form zu bringen, ist sie einem LaTeX-Lauf zu unterwerfen. Hierzu wird die Datei dem Interpreter übergeben, der sie dann weiterverarbeitet. In diesem Beispiel wäre der Befehl dazu „`latex eins.tex`“. Daraufhin erscheinen einige Ausgaben des Programmes, die hier der Kürze halber nicht behandelt werden sollen. Wichtig ist fürs erste Folgendes: entweder der Prozess bleibt mit einem Fehler hängen (er schreibt dann eine mehrzeilige Fehlermeldung und wartet mit einem „?“ darauf, daß man ihm weitere Anweisungen gibt (es ist allerdings in den meisten Fällen zweckmäßig, diesen „Debug-Modus“ durch Eingabe von „q“ abzubrechen)). Wenn hingegen die LaTeX-Datei eine korrekte Syntax besitzt, erhält man eine Ausgabe, wieviele Seiten erstellt wurden (eine Information unter vielen Anderen). Der LaTeX-Lauf erzeugt eine handvoll Dateien im aktuellen Verzeichnis. Eine Testausgabe:

```
ad@ATP:~/latex$ ll kualax.*
-rw-r--r--  1 ad ad  395 2005-10-20 18:31 kualax.aux
-rw-r--r--  1 ad ad 5704 2005-10-20 18:31 kualax.dvi
-rw-r--r--  1 ad ad 4066 2005-10-20 18:31 kualax.tex
-rw-r--r--  1 ad ad 5311 2005-10-20 18:31 kualax.log
-rw-r--r--  1 ad ad  315 2005-10-20 18:31 kualax.toc
ad@ATP:~/latex$
```

Zur Erklärung: In der Datei `kualax.aux` befinden sich ersteinmal nicht weiter relevante Hilfsdaten; in der Datei `kualax.log` befinden sich die auf dem Bildschirm erschienenen Meldungen. Die Datei `kualax.toc` ist für ein automatisch erstelltes Inhaltsverzeichnis von Bedeutung. Nun bleibt noch die Datei `kualax.dvi`. Dies ist das Ziel der Bemühungen gewesen. Die Datei kann man sich nun als Ergebnis ansehen: „`xdvi eins.dvi`“. Zur weiteren Verarbeitung äußere ich mich zum Schluß noch einmal - hier also erst einmal soviel: Dies ist der generelle Ablauf, um eine LaTeX-Datei zur Anzeige zu bringen. Alternativ kann auch anstelle von „`latex`“ „`pdflatex`“ als Interpreter benutzt werden, in dem Falle wird gleich eine pdf-Datei erstellt.

4 Der Grundaufbau eines LaTeX-Dokumentes

Was im letzten Kapitel an einem Beispiel demonstriert wurde, soll hier nun theoretisch unterfüttert werden. Hierzu soll zunächst die Testdatei nocheinmal in abstrahierter Form angesehen werden:

```
1 \documentclass [<opts >]{<klasse >}
2 \usepackage [<opts >]{<paket >}
3 \begin{document}
4 <Inhalt >
5 \end{document}
```

4.1 Präambel

Die Zeilen 1 und 2 bilden den Teil, der als Präambel bezeichnet wird. An dieser Stelle ist noch nichts vom Inhalt geschrieben, noch wird LaTeX auf alles vorbereitet, was kommt. Hier können Optionen für das gesamte Dokument gesetzt werden, wie zum Beispiel die Papiergröße oder auch Regeln zur Trennung von Worten ergänzt werden. Die erste Zeile ist unabdingbar, hier wird LaTeX vorgegeben, um was für eine Art von Dokument es sich im Folgenden handelt. In der Folge wird das Grundlayout entsprechend für einen Brief oder ein Buch gesetzt. Der Dokumentenklasse können optionale Argumente mitgegeben werden, wie beispielsweise die Papiergröße. Die Standarddokumentklassen sind im amerikanischen Layout gehalten, es gibt eine Alternative, die den europäischen Gewohnheiten angepasste Dokumentklassen bietet: Das KOMA-Script. Die entsprechenden Klassen heißen hier „scrartcl“, „scrbook“ usw.

Der Einstellung des Dokumenttyps folgt der Import von benötigten Paketen, die mittels „\usepackage“ eingebunden werden. Pakete erweitern den Befehlsumfang um jeweils ein Spezialgebiet (z.B. bindet „amsmath“ Schriftarten für den Mathesatz ein, „pstricks“ erlaubt das Plotten von Funktionen und „ngerman“ stellt Trennungsregelung und Überschriften auf Deutsches Layout um.). Pakete bieten Modularisierung und sind ein Grund für die Beliebtheit von LaTeX. Es gibt kaum ein wissenschaftliches Gebiet, für daß es kein spezielles Paket gibt⁵.

4.2 Inhalt

Auf die Präambel folgt der eigentliche Inhalt des Dokumentes (ab Zeile 3). Hier ist der Text, hier werden die aus den Paketen importierten Funktionalitäten genutzt.

5 LaTeX-Syntax

Da es sich bei LaTeX um eine Programmiersprache handelt, besitzt sie auch Regeln für die Syntax. Diese sind einzuhalten, ansonsten gibt es beim LaTeX-Lauf Probleme.

5.1 Befehle

Generell ist zu unterscheiden zwischen einfachem Text (der allerdings auch immer im Rahmen eines Befehls erscheint) und Befehlen. Befehle sind dadurch gekennzeichnet, daß sie mit „\“ beginnen. Befehle können obligatorische Parameter besitzen (zum Beispiel ist der Befehl `\documentclass` ohne Angabe einer Klasse unsinnig), die in geschweiften Klammern übergeben werden. Ferner sind optionale Parameter (wie schon erwähnte Angabe zum Papierformat) möglich; diese werden in eckigen Klammern übergeben. Ein Beispiel für einen Befehl mit einem obligatorischen und einem optionalen Parameter wäre etwa „`\documentclass[a4paper]{article}`“. Der Pflichtparameter teilt mit, daß der Dokumenttyp „article“ sein soll und der Optionale, daß das Dokument passend für Papier im Format A4 gesetzt werden soll.

⁵Musik, sprich Notensatz, eingeschlossen.

5.2 Umgebungen

Umgebungen sind die Antwort auf die Frage, was passieren soll, wenn ein Befehl für einen längeren Text gelten soll. Eine Umgebung wird begonnen, dem folgt alles, was durch die Umgebung beeinflusst werden soll und schließlich wird sie wieder beendet. Auch Umgebungen können sowohl obligatorische wie auch optionale Parameter besitzen. Eine Umgebung wird mit dem Befehl „`\begin{<name>}`“ begonnen und durch „`\end{<name>}`“ wieder beendet. Außerdem ist die nichtüberlappende Schachtelung von Umgebungen erlaubt.

5.3 Sonstige Anweisungen

5.3.1 Kommentare

Wie jede größere Programmiersprache bietet auch LaTeX die Möglichkeit, Zeilen auszukommentieren. Dies ist beispielsweise sinnvoll, wenn man ein Paket testweise auskommentieren möchte (weil man es beispielsweise verdächtigt, für seltsame Ausgabe verantwortlich zu sein). Eine Zeile wird durch voranstellen eines `%` auskommentiert.

5.3.2 Einbinden von weiteren Dateien

Für große Projekte ist es sinnvoll, den Inhalt auf mehrere einzelne Dateien aufzuteilen (z.B. kapitelweise). Dies ist insbesondere sinnvoll, wenn mit mehreren Personen gearbeitet wird, weil Konflikte durch gleichzeitiges Zugreifen auf eine Datei vermieden werden⁶. Hierzu finden sich die Befehle „`\input{<datei>}`“ sowie „`\include{<datei>}`“. Beiden Befehlen wird der Dateiname einer weiteren LaTeX-Datei gegeben. In Dieser darf allerdings keine Präambel sowie kein eigener „document“-Teil definiert sein, da die Datei so wie sie ist anstelle des Befehls eingesetzt wird. Der Unterschied zwischen den beiden Befehlen liegt darin, daß „`\include`“ automatisch einen Seitenumbruch einfügt.

5.3.3 Maskierung

Um im Text ein Zeichen zu benutzen, welches ansonsten einen Befehl einleiten würde, muss es mithilfe eines Backslashes maskiert werden. So erhält man beispielsweise ein Prozentzeichen durch Eingabe von „`\%`“. Der Backslash muß anders erzeugt werden, ihn erhält man durch „`\$backslash$`“⁷.

5.3.4 Weglassen von Klammern

Wenn ein Befehl keine Parameter erhält, so ist es möglich, die geschweiften Klammern wegzulassen. Allerdings wird dann zwangsläufig ein Leerzeichen nach dem Befehl fällig, da LaTeX ansonsten nicht feststellen kann, wo der Befehl endet.

⁶was u.a. bei der Zusammenarbeit mit einem Repository, etwa svn oder cvs, von Bedeutung ist

⁷Das ist eine Matheumgebung und eigentlich nicht ganz sauber.

6 LaTeX in der Anwendung

6.1 Probleme

Wenn man beginnt, sich mit LaTeX zu befassen, wird man relativ schnell auf eine Handvoll Probleme stoßen. Ein paar davon sollen im Folgenden abgehandelt werden.

6.1.1 Geschütztes Leerzeichen

Da LaTeX automatisch im Blocksatz setzt, ist es oft notwendig, Worte zu trennen. Dies zu tun bereitet LaTeX dank der in Paketen vorgegebenen Trennungsregeln meist keine Probleme. Allerdings gibt es Leerzeichen, die nicht zum Trennen benutzt werden sollen, beispielsweise in Zahlen wie 10 000. An solchen Stellen wird anstelle des normalen Leerzeichens eine Tilde (~) eingesetzt. Solche Wortzwischenräume werden bei der Suche nach zum Zeilenumbruch geeigneten Stellen ignoriert.

6.1.2 Anführungszeichen

In Deutschland werden standardmäßig bestimmte Anführungszeichen benutzt. Diese sind ebenfalls recht einfach zu erzeugen.

Anführungsstriche (also unten „„“) erreicht man durch " ' (ein normales Anführungszeichen und einmal Akzent Gravis). Abführungsstriche (oben) sind entsprechend " ' (ein normales Anführungszeichen gefolgt von einem Apostroph).

6.1.3 Umlaute und Kodierung

Umlaute sind ein Problem. Als begonnen wurde, Computer an Landeseinstellungen bezüglich Sonderzeichen anzupassen, hat man erweiterte Codepages konstruiert. Bis zu einem ASCII⁸-Wert von 127 sind alle Zeichen weltweit genormt⁹. Darüber beginnen die Länderspezifika. Was in einem Deutschen Zeichensatz vielleicht ein Umlaut ist, ist in einem französischen Zeichensatz ein ganz anderer oder in einem Skandinavischen vielleicht gar nichts anzeigbares. Da LaTeX möglichst universell nutzbar bleiben soll(te), beherrscht es nach wie vor nur 7-Bit Eingabe, alles darüber wird ignoriert. Umlaute und diakritische Zeichen lassen sich mit einer einfachen Konstruktion erreichen: „\“ erzeugt Diakritische Zeichen; ohne Rücksicht darauf, ob dies eine sinnvolle Zusammenstellung wird: \grave{g} , \acute{g} , \grave{g} , \acute{g} , \grave{g} , \acute{g} , \grave{g} (im Quelltext `\ "g`, `\ 'g`, `\ 'g`, `\ .g`, `\ ~g`, `\ =g` und `\ r g`).

Die Eingabe läßt sich noch auf zwei Arten weiter vereinfachen: Wenn das Paket „ngerman“ (oder „german“) eingebunden ist, reicht auch schon „a“ aus. Wenn der Zeichensatz richtig eingestellt wird, können Umlaute wie gewohnt eingegeben werden. Hierzu ist es erforderlich, das Paket „inputenc“ einzubinden. Dem Paket muß lediglich die Benutzte Codierung mitgegeben werden; ein Aufruf könnte also `\usepackage[latin1]{inputenc}`¹⁰ lauten.

⁸„American Standard Code for Information Interchange“, die Zahlenrepräsentation eines Zeichens

⁹wenn ich extrem ausgefallene Architekturen einmal außenvorlassen darf.

¹⁰Hierbei ist „latin1“ der Wert für Unix- und Linux-Rechner. Windows erfordert „ansinew“ und ältere Macintosh-Rechner „applemac“ [SCH05].

Ebenfalls sinnvoll kann es sein auch für die Ausgabeschrift eine Codierung vorzugeben. Dazu wird ebenfalls ein Paket eingebunden, in diesem Falle das Paket „fontenc“ mit dem Parameter „T1“, was ähnliche Konsequenzen auf anderer Ebene hat: Es geht nicht mehr um die Einsondern um die Ausgabedatei.

6.2 Gliederungsbefehle

6.2.1 Inhaltsverzeichnis

Ein Inhaltsverzeichnis kann eingefügt werden indem der Befehl `\tableofcontents` eingefügt wird. Dies hat außerhalb der Gliederung zu geschehen. Dabei ist zu beachten, daß es mehrere LaTeX-Läufe (bei einem „dummen“ LaTeX-Interpreter entspricht die Zahl der notwendigen Läufe der maximalen Gliederungstiefe...) braucht, ehe das Inhaltsverzeichnis aktuell ist. Deshalb vor Abgabe eines Textes immer noch einmal das Inhaltsverzeichnis überprüfen.

Der Grund für dieses auf den ersten Blick seltsame Verhalten liegt darin, daß LaTeX wie ein Straight-Top-Down-Compiler arbeitet: Einmal von oben nach unten, ohne Sprünge. Dass beim nächsten Lauf mehr passiert, liegt an der Datei `.toc` (Table Of Contents). Dies Problem zeigt sich bei allen Arten von Indizes, Tabellen usw.

6.2.2 Gliederung erstellen

Um eine Gliederung zu erstellen, benötigt man folgende Befehle (diese gelten in der Dokumentenumgebung „article“ - für „book“ gelten Andere; ferner halte ich sie für soweit selbsterklärend, daß ich mir eine Übersetzung spare),

- `\section{Kapitel}`
- `\subsection{Unterkapitel}`
- `\subsubsection{Unter-Unterkapitel}`

Für „book“ ist „section“ durch „chapter“ zu ersetzen. Nähere Informationen finden sich in [KOP96].

6.2.3 Titelseite

Die Titelseite soll als allem voranstehend einmal als Teil der Gliederung aufgefasst und hier behandelt werden. Die „Variablen“ für eine Titelseite werden in der Präambel festgelegt, die Titelseite wird danach innerhalb der „document“-Umgebung mithilfe des Tags `\maketitle` ausgelöst. Es folgt ein Beispiel aus dieser Datei:

```
1 \title{(Kurz-) Einf" uhrung in Textsatz mit \LaTeX{}}
2 \author{Andreas D" ahn}
3 \begin{document}
4 \maketitle
5 [...]
6 \end{document}
```

6.3 Mathesatz

Der Mathesatz ist eine der Kernkompetenzen von LaTeX, weshalb ich nicht umhinkommen werde, ihn hier zumindest anzureißen.

6.3.1 Mathe-Umgebungen

Mathe-Satz muss in einer Mathe-Umgebung vorgenommen werden. Es gibt zwei verschiedene Möglichkeiten, eine Formel in den Text einzubinden: Entweder als Formel im laufenden Text

$x_{1,2} = \frac{1}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q}$ oder als abgesetzte Formel:

$$x_{1,2} = \frac{1}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q}.$$

Man beachte: der Punkt nach der Formel steht innerhalb der Formel.

Zum Verständnis nun der ganze Absatz als Quelltext:

```
1 Es gibt zwei verschiedene Möglichkeiten, eine Formel in
2 den Text einzubinden: Entweder als Formel im laufenden Text
3 $x_{1,2}=\frac{1}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q}$ oder als
4 abgesetzte Formel:
5 \[x_{1,2}=\frac{1}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q}.\]
```

6.3.2 Grundlegende Konstruktionen

In der Auflistung werde ich als Formelzeichen a, b und ggf. c verwenden. Sollte an der betreffenden Stelle ein Ausdruck eingesetzt werden, so kann dieser einfach an derjenigen Stelle in geschweiften Klammern eingefügt werden. Es kann also (mathematisch höchst fragwürdig) zusammengefasst werden: $a = \{b + c\}$.

- \int_a^b - Integral von a bis b. `\int _a^b`
- \int_a^b - Integral mit anders angeschriebenen Grenzen. `\int \limits^b_a`
- $\frac{a}{b}$ - Bruch von a durch b. `\frac a b`
- a^b - a hoch b. `a^b`
- a_b - a mit dem Index b. `a_b`
- \vec{a} - der Vektor a. `\vec a`
- \dot{a} - a abgeleitet nach der Zeit (a Punkt). `\dot a`
- \hat{a} - a Dach. `\hat a`

- $a + b$ $b - b$ $a \cdot b$ - Grundrechenarten ohne Division. `a+b` `b-b` `a \cdot b`
- ∞ - Das Unendliche. `\infty`
- \sum_a^b - Eine Summenformel. `\sum _a^b`
- \circ - Der Kringel, bekannt aus der Gruppentheorie. `\circ`
- \mathbb{N} - Die Symbole für die gegebenen Mengen. `\mathbb{N}`
- $\frac{a}{b}$ - Das altdeutsche a, geeignet für Vektoren. `\mathfrak{a}`

6.3.3 Griechische Buchstaben

Die griechischen Buchstaben werden einfach über ihren ausgeschriebenen Namen angesprochen. Beispiel: $\alpha\beta\gamma\Delta$. Um die zugehörigen Großbuchstaben zu erzeugen, genügt es, den Anfangsbuchstaben groß zu schreiben. (Das Beispiel lautete: `\alpha \beta \gamma \Delta`).

6.3.4 Mathematische Sonderzeichen

Die folgenden Tabellen habe ich nicht selbst geschriebe, sondern von <http://www.feyrer.de/OS/latex.html> entnommen. Sie sind auch in [KOP96] zu finden.

α	<code>\alpha</code>	θ	<code>\theta</code>	\circ	<code>\circ</code>	τ	<code>\tau</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	π	<code>\pi</code>	υ	<code>\upsilon</code>
γ	<code>\gamma</code>	γ	<code>\gamma</code>	ϖ	<code>\varpi</code>	ϕ	<code>\phi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>	φ	<code>\varphi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ϱ	<code>\varrho</code>	χ	<code>\chi</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ψ	<code>\psi</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>	ω	<code>\omega</code>
η	<code>\eta</code>	ξ	<code>\xi</code>				
Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		
\pm	<code>\pm</code>	\cap	<code>\cap</code>	\diamond	<code>\diamond</code>	\oplus	<code>\oplus</code>
\mp	<code>\mp</code>	\cup	<code>\cup</code>	\triangleup	<code>\triangleup</code>	\ominus	<code>\ominus</code>
\times	<code>\times</code>	\uplus	<code>\uplus</code>	\triangledown	<code>\triangledown</code>	\otimes	<code>\otimes</code>
\div	<code>\div</code>	\sqcap	<code>\sqcap</code>	\triangleleft	<code>\triangleleft</code>	\oslash	<code>\oslash</code>
$*$	<code>\ast</code>	\sqcup	<code>\sqcup</code>	\triangleright	<code>\triangleright</code>	\odot	<code>\odot</code>
\star	<code>\star</code>	\vee	<code>\vee</code>	\triangleleft^b	<code>\triangleleft^b</code>	\bigcirc	<code>\bigcirc</code>
\circ	<code>\circ</code>	\wedge	<code>\wedge</code>	\triangleright^b	<code>\triangleright^b</code>	\dagger	<code>\dagger</code>
\bullet	<code>\bullet</code>	\setminus	<code>\setminus</code>	\triangleleft^b	<code>\triangleleft^b</code>	\ddagger	<code>\ddagger</code>
\cdot	<code>\cdot</code>	\wr	<code>\wr</code>	\triangleright^b	<code>\triangleright^b</code>	\amalg	<code>\amalg</code>

\leq	<code>\leq</code>	\geq	<code>\geq</code>	\equiv	<code>\equiv</code>	\models	<code>\models</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>	\perp	<code>\perp</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>	\mid	<code>\mid</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\asymp	<code>\asymp</code>	\parallel	<code>\parallel</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>	\bowtie	<code>\bowtie</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>	\Join^b	<code>\Join^b</code>
\sqsubset^b	<code>\sqsubset^b</code>	\sqsupset^b	<code>\sqsupset^b</code>	\neq	<code>\neq</code>	\smile	<code>\smile</code>
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\doteq	<code>\doteq</code>	\frown	<code>\frown</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\propto	<code>\propto</code>	$=$	<code>=</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	$<$	<code><</code>	$>$	<code>></code>
$:$	<code>:</code>						
$,$	<code>,</code>	$;$	<code>;</code>	$:$	<code>\colon</code>	\cdot	<code>\ldotp</code>
						\cdot	<code>\cdotp</code>
\leftarrow	<code>\leftarrow</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>		
\Leftarrow	<code>\Leftarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>	\Uparrow	<code>\Uparrow</code>		
\rightarrow	<code>\rightarrow</code>	\longrightarrow	<code>\longrightarrow</code>	\downarrow	<code>\downarrow</code>		
\Rightarrow	<code>\Rightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>	\Downarrow	<code>\Downarrow</code>		
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>	\updownarrow	<code>\updownarrow</code>		
\Leftrightarrow	<code>\Leftrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>		
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>		
\hookrightarrow	<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>	\searrow	<code>\searrow</code>		
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>	\swarrow	<code>\swarrow</code>		
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>	\nwarrow	<code>\nwarrow</code>		
\rightleftharpoons	<code>\rightleftharpoons</code>	\leadsto^b	<code>\leadsto^b</code>				
\dots	<code>\ldots</code>	\cdots	<code>\cdots</code>	\vdots	<code>\vdots</code>	\ddots	<code>\ddots</code>
\aleph	<code>\aleph</code>	\prime	<code>\prime</code>	\forall	<code>\forall</code>	∞	<code>\infty</code>
\hbar	<code>\hbar</code>	\emptyset	<code>\emptyset</code>	\exists	<code>\exists</code>	\Box^b	<code>\Box^b</code>
\imath	<code>\imath</code>	∇	<code>\nabla</code>	\neg	<code>\neg</code>	\Diamond^b	<code>\Diamond^b</code>
\jmath	<code>\jmath</code>	\surd	<code>\surd</code>	\flat	<code>\flat</code>	\triangle	<code>\triangle</code>
ℓ	<code>\ell</code>	\top	<code>\top</code>	\natural	<code>\natural</code>	\clubsuit	<code>\clubsuit</code>
\wp	<code>\wp</code>	\perp	<code>\perp</code>	\sharp	<code>\sharp</code>	\diamondsuit	<code>\diamondsuit</code>
\Re	<code>\Re</code>	\parallel	<code>\parallel</code>	\backslash	<code>\backslash</code>	\heartsuit	<code>\heartsuit</code>
\Im	<code>\Im</code>	\angle	<code>\angle</code>	∂	<code>\partial</code>	\spadesuit	<code>\spadesuit</code>
\mho^b	<code>\mho^b</code>						
\sum	<code>\sum</code>	\bigcap	<code>\bigcap</code>	\bigodot	<code>\bigodot</code>		
\prod	<code>\prod</code>	\bigcup	<code>\bigcup</code>	\bigotimes	<code>\bigotimes</code>		
\coprod	<code>\coprod</code>	\bigsqcup	<code>\bigsqcup</code>	\bigoplus	<code>\bigoplus</code>		
\int	<code>\int</code>	\bigvee	<code>\bigvee</code>	\biguplus	<code>\biguplus</code>		
\oint	<code>\oint</code>	\bigwedge	<code>\bigwedge</code>				

<code>\arccos</code>	<code>\cos</code>	<code>\csc</code>	<code>\exp</code>	<code>\ker</code>	<code>\limsup</code>	<code>\min</code>	<code>\sinh</code>
<code>\arcsin</code>	<code>\cosh</code>	<code>\deg</code>	<code>\gcd</code>	<code>\lg</code>	<code>\ln</code>	<code>\Pr</code>	<code>\sup</code>
<code>\arctan</code>	<code>\cot</code>	<code>\det</code>	<code>\hom</code>	<code>\lim</code>	<code>\log</code>	<code>\sec</code>	<code>\tan</code>
<code>\arg</code>	<code>\coth</code>	<code>\dim</code>	<code>\inf</code>	<code>\liminf</code>	<code>\max</code>	<code>\sin</code>	<code>\tanh</code>
$\left\{$	<code>\rmoustache</code>	\int	<code>\lmoustache</code>	$\right)$	<code>\rgroup</code>	$\left($	<code>\lgroup</code>
$ $	<code>\arrowvert</code>	$\ $	<code>\Arrowvert</code>	$ $	<code>\bracevert</code>		

Table 10: Large Delimiters

\widetilde{abc}	<code>\widetilde{abc}</code>	\widehat{abc}	<code>\widehat{abc}</code>
\overleftarrow{abc}	<code>\overleftarrow{abc}</code>	\overrightarrow{abc}	<code>\overrightarrow{abc}</code>
\overline{abc}	<code>\overline{abc}</code>	\underline{abc}	<code>\underline{abc}</code>
\overbrace{abc}	<code>\overbrace{abc}</code>	\underbrace{abc}	<code>\underbrace{abc}</code>
\sqrt{abc}	<code>\sqrt{abc}</code>	$\sqrt[n]{abc}$	<code>\sqrt[n]{abc}</code>
f'	<code>f'</code>	$\frac{abc}{xyz}$	<code>\frac{abc}{xyz}</code>

6.3.5 Funktionsnamen

Eine der abgedruckten Tabellen enthielt eine Sammlung mathematischer Funktionen und feststehender Ausdrücke. Dies hat einen einfachen Sinn. Man betrachte die folgende Ungleichung:

$$\sin \alpha \neq \mathit{sin} \alpha.$$

Der inhaltliche Unterschied ist durchaus bemerkenswert, es sollte schon ein Unterschied sein, ob man den Sinus von Alpha oder das Produkt s mal i mal n mal Alpha bildet. Natürlich kann man auch kurzerhand selbst Funktionsnamen definieren: „`\$a \operatorname{eats} b`“ wird zu „*a eats b*“. Dies benötigt das Paket „`amsmath`“.

Weitere Informationen zum Mathesatz finden sich in [KOP96] sowie [NID].

6.4 Vielgenutzte Umgebungen

Hier folgt nur ein kleiner Ausschnitt – es gibt eine Unmenge an Umgebungen, jede für ihren Spezialfall. Da dies dem Titel nach allerdings nur eine „Einführung“ werden soll, beschränke ich mich mal auf wenige, die dafür im täglichen Gebrauch von größerer Bedeutung sind.

6.4.1 itemize oder: „Listen, unnummeriert“

Für eine Liste von Items ist die „`itemize`“-Umgebung die einfachste Variante. Zu Beachten ist: Jeder einzelne Eintrag beginnt mit `\item` — und hat als Enddefinition nur den Beginn des nächsten Items oder das Ende der Listenumgebung. Als Parameter kann optional ein anderes

Symbol für den einzelnen Punkt gesetzt werden. LaTeX kommt mit Schachtelungen ohne Probleme zurecht, hierbei werden die Symbole gewechselt, die dem einzelnen Punkt der Aufzählung vorangestellt sind.

6.4.2 description oder: „Liste, eigene Überschrift“

Mithilfe der Umgebung „description“ ist es möglich, daß die einzelnen Punkte einer Liste selbstdefinierte Überschriften tragen. Die Überschrift wird als optionaler Parameter in eckigen Klammern übergeben.

6.4.3 enumerate oder: „Listen, numeriert“

Wenn man nun aber eine durchnummerierte Liste wünscht? Auch nicht viel komplizierter: Die Umgebung heißt „enumerate.“. Es werden automatisch arabische Ziffern benutzt.

6.4.4 verbatim oder: „Nur-Text“

Immer mal wieder kommt es vor, daß man (z.B. weil man ein LaTeX-Buch schreiben will) auch Textsequenzen so ausgeben möchte, wie sie eingegeben wurden. Dazu müsste man gewissermaßen den Interpreter auf Zeit abstellen können. Genau dazu dient die Verbatim-Umgebung. Entweder als echte Umgebung oder man definiert sich per `\makeshortverb{\#}` ein eigenes Zeichen, mit dem im Text ein Verbatim-Abschnitt beginnt (welches man dafür aber nicht mehr normal benutzen kann) und wieder beendet wird. Hierzu muß das Paket „verbatim“ eingebunden werden.

6.4.5 tabular oder: „Tabelle, simpel“

Die „tabular“-Umgebung erlaubt simple Tabellen. Diese Tabellenform ist allerdings noch nicht in der Lage, mit Seitenumbrüchen umzugehen. Ein Beispiel:

Überschrift		
a	b	c
d	e	f

```
\begin{tabular}{|c|l r|}\hline
\multicolumn{3}{|c|}{Überschrift}\\
\hline
a & b & c\\
d & e & f\\
\hline
\end{tabular}
```

...und die Quelle. Nicht schwer zu erkennen: Mit den „|“ wird die Vergabe der vertikalen Trennlinien gesteuert; c,r und l steuern, ob der Zelleninhalt zentriert, links- oder rechtsbündig ausgegeben werden soll. Eine multicolumn-Zeile beginnt einen Eintrag der sich über mehrere Spalten erstreckt und die „&“-Zeichen grenzen die einzelnen Einträge gegeneinander ab. Der Befehl „\hline“ schließlich erzuegt eine horizontale Linie.

6.4.6 Istlisting oder: Programmlistings einbinden

Mithilfe der Umgebung „Istlisting“ ist es ohne Probleme möglich, Quelltext aus verschiedenen Programmiersprachen formatiert auszugeben. Der Quelltext kann natürlich auch aus einer externen Datei eingebunden werden und muß nicht wie hier gezeigt im LaTeX-Dokument stehen. Das Ergebnis kann dann wie folgt aussehen:

```
1 #include <stdio.h>
2 int main (int argc , char * argv [])
3 {
4     printf("Ich habe %i Argumente bekommen!\n\n" , argc );
5     return 0;
6 }
```

Der Quelltext dafür sieht wie folgt aus:

```
\lstset{language=c}
\begin{lstlisting}[numbers=left,frame=single]
#include <stdio.h>
int main (int argc, char * argv[])
{
    printf("Ich habe %i Argumente bekommen!\n\n",argc);
    return 0;
}
\end{lstlisting}
```

6.4.7 minipage oder: Seite in der Seite

Die „minipage“-Umgebung erlaubt es, innerhalb einer Seite Bereiche zu definieren, die sich selbst wieder wie Seiten verhalten. Ihnen wird zwangsläufig mitgegeben, wie breit sie sind.

	Ich bin auch eine Seite.
Ich bin eine Seite! Ich bin exakt sechs Zentimeter breit!	Aber ich bin nur vier Zentimeter breit. Dafür passe ich neben die andere Seite.

Der Quelltext sieht wie folgt aus:

```
\begin{minipage}{6cm}
Ich bin eine Seite! Ich bin exakt sechs Zentimeter breit!
\end{minipage}
\begin{minipage}{4cm}
Ich bin auch eine Seite. Aber ich bin nur vier Zentimeter
breit. Dafür passe ich neben die andere Seite.
\end{minipage}
```

6.4.8 Literaturverzeichnis

Jede größere naturwissenschaftliche Arbeit stützt sich zu nicht unerheblichen Anteilen auf Zitate aus Drittliteratur (Primär- und Sekundärquellen), die entsprechend gekennzeichnet werden müssen. Dazu ist in den meisten Distributionen von LaTeX das Programm BibTeX enthalten. Für größere Projekte sollte unbedingt eine separate BibTeX-Datenbank benutzt werden, für kleinere Projekte reicht die hier vorgestellte Technik durchaus auch.

Am Ende des Dokumentes (noch vor `\end{document}`) werden die Literaturangaben folgendermaßen eingegeben:

```
\begin{thebibliography}{99}
\bibitem[Q]{W} ERZ: Mein Feind.
\end{thebibliography}
```

Erläuterung: Der Wert 99 ist willkürlich. Dort soll ein „Musterwert“ stehen, der die Länge des längsten Eintrags hat, der darunter als für den Leser sichtbar angezeigt werden wird. Jeder Eintrag des Literaturverzeichnisses beginnt mit dem Kommando „`\bibitem`“. Darauf folgt („Q“) als optionaler Parameter. Dieser Text wird auch im ausgegebenen Text als Markierung für das entsprechende Werk verwendet. Dem folgt das erste obligatorische Argument (hier „W“). Dieser Bezeichner wird intern als Referenz benutzt und kann entsprechend unsinnige Werte besitzen. Schließlich folgt die Beschreibung des Buches, wie sie am Ende ausgegeben werden wird. Hier können auch Formatierungsanweisungen gesetzt werden.

Diese Einträge können mit dem Befehl `\cite{W}` referenziert werden. In diesem Falle wird der Eintrag „W“ angegeben. Dabei wird vor der Ausgabe als Referenzsymbol „Q“ eingesetzt.

Achtung: Möglicherweise sind mehrere LaTeX-Läufe notwendig, da auch hier wieder undefinierte Symbole erst nach ihrer Benutzung aufgelöst werden. Auf die Ausgabe beim LaTeX-Lauf achten!

Ich habe hier nur die simple Methode einer Bibliographie innerhalb eines LaTeX-Dokumentes angegeben; es geht mittels einer externen BibTeX-Datenbank eleganter wie auch einfacher.

6.5 Graphiken im eps-Format einbinden

Zum Einbinden der vorangehenden Graphiken wurden der Befehl

```
\includegraphics[scale=0.5]{schoenesBild.eps}
```

benutzt. Dabei wurden die Graphiken durch den Skalierungsfaktor auf die Hälfte zusammgezogen. Wenn die LaTeX-Datei mit latex in eine dvi-Datei umgewandelt werden muß, so wird hier eine Datei im Encapsulated-Postscript-Format eingebunden. Wenn mit pdflatex gearbeitet wird, muss eine einzubindende Graphik als PNG oder PDF vorliegen. Es ist möglich, daß LaTeX während des Laufes Konverter startet, aber das möge man sich aus geeigneter Fachliteratur aneignen.

6.6 Fußnoten

Bevor ich es vergesse: Fußnoten.

Nichts ist einfacher als Fußnoten¹¹: `\footnote{Fußnotentext}`.

6.7 Textformatierung

Natürlich es es auch möglich, direkten Einfluß auf die Formatierung zu nehmen. Aber dies ist nicht Sinn der Benutzung von LaTeX. Das physikalische Auszeichnen von Textbestandteilen ist Teil des LaTeX–Laufes und wird durch die Dokumentenklasse sowie die eingebundenen Pakete bestimmt.

6.7.1 Textteile hervorheben

Um wichtigen Text hervorzuheben gibt es zunächst die einfache Möglichkeit, den hervorzuhebenden Text mittels *emph* für „emphasize“ zu behandeln. Der Aufruf lautet „`\emph{text}`“.

6.7.2 Schriften

Hier soll es nicht um konkrete Schriften wie „Times New Roman“ oder „Arial“ gehen, sondern um Schrifttypen. Es gibt die drei Schrifttypen „Roman“ (Standard), „Typewriter“ und „serifenlos“. Diese werden mittels wie Schaltern funktionierender Tags eingeschaltet. Das bedeutet: Sobald ein entsprechendes Tag gefunden wird, wird der Schrifttyp solange geändert und beibehalten, bis das nächste kommt. Die Tags lauten folgendermassen: `\ttfamily` für einen Typewriter, `\rmfamily` für die Standardschrift und schließlich `\sffamily` für eine serifenlose Schrift. Die „Grundschriftart“ läßt sich durch das Einbinden von Paketen steuern, so kann beispielsweise „times“ oder „palatino“ eingebunden werden.

¹¹...füßelig...

6.7.3 Schriftanschnitt und -größe

Befehl	Ergebnis
Schnitt	
<code>\textit{txt}</code>	<i>Kursiv</i>
<code>\textsl{txt}</code>	<i>geneigt</i>
<code>\textsc{txt}</code>	KAPITÄLCHEN
<code>\textup{txt}</code>	Aufrecht
<code>\textbf{txt}</code>	Fett
<code>\textmd{text}</code>	normaler Text
Größe	
<code>\tiny{txt}</code>	winzig
<code>\scriptsize{txt}</code>	sehr klein
<code>\footnotesize{txt}</code>	klein
<code>\small{txt}</code>	klein
<code>\normalsize{txt}</code>	normal
<code>\large{txt}</code>	groß
<code>\Large{txt}</code>	größer
<code>\LARGE{txt}</code>	noch größer
<code>\huge{txt}</code>	riesig
<code>\Huge{txt}</code>	größtmögl.

6.8 Zeilen- und Seitenumbruch

Um eine neue Seite zu erzwingen, wird der Befehl „`\newpage`“ genutzt, für eine neue Zeile wird „`\\`“ benutzt, wobei optional noch der Abstand bis zur nächsten Zeile mit angegeben werden kann: „`\\ [2cm]`“

7 DVI, Postscript und PDF

Es gibt zwei grundlegende Herangehensweisen an LaTeX: Die eine zielt auf ein DVI-Dokument als Ausgabemedium und ist die einleitend Beschriebene. Die Andere ist wesentlich praxisnäher und erzeugt gleich eine Ausgabedatei im Portable Document Format.

Die DVI-Datei kann mit `xdvi` und Konsorten betrachtet werden. Man kann sie aber auch mit dem Befehl „`dvips <dateiname.dvi>`“ in eine entsprechende Postscriptdatei umwandeln. Um hieraus nun ein pdf-Dokument zu machenn, ist der Befehl „`ps2pdf <dateiname.ps>`“ notwendig. Jetzt kann es zur großen Ernüchterung kommen, wenn sich das Ergebnis als PDF ansieht. Es wirkt unscharf und ausgesprochen unschön. Wenn man dagegen von dieser Datei druckt hat man ein ansehnliches Ergebnis. Dies ist ein Problem bei der Einbettung von Schriften. Bindet man dagegen ein Schriftarten-Paket wie „`times`“ oder „`palatino`“ mit ein, sieht es wieder recht ansehnlich aus.

Geht man hingegen mit `pdflatex` an das Dokument heran, so erhält man ein PDF ohne diesen Umweg und hat dazu noch den Vorteil, daß mithilfe des Paketes „`hyperref`“ erzeugte Hyperlinks

tatsächlich anklickbar sind. Allerdings hat man auch ein Problem: Graphiken im EPS-Format können nicht eingebunden werden. Graphiken, die in einem PDF auftauchen sollen, müssen entweder als png oder selbst schon als pdf vorliegen.

7.1 Was ist PostScript?

Bei PostScript handelt es sich um eine richtige Programmiersprache¹². Es lassen sich mit wenigen Befehlen komplexe Dokumente schaffen. Mit ganz simplem Mitteln lässt sich beispielsweise Millimeterpapier erzeugen. Zu beachten ist die polnische Notation, die bei Postscript intensiv genutzt wird.

7.2 Was ist PDF?

PDF, so wurde mir einst erklärt, sei ein „extended Subset“ von PostScript. Ich will hier nicht kleinlich sein und mich fragen, ob der Ausdruck nicht eher ein Oxymoron ist, aber ich werde den Ausdruck verteidigen. PDF basiert auf grossen Teilen von Postscript, bietet aber lange nicht alles, was PostScript kann. Dafür hat es Fähigkeiten (wie erwähnte Hyperlinks), die Postscript selbst fehlen. Hier ist der Übergang zwischen einer Programmier- und einer Seitenbeschreibungssprache zu beobachten.

8 Fehlersuche

Ich möchte hier noch ein Wort zum Thema Fehlersuche verlieren. Gerade am Anfang hat man viel Ärger mit dem LaTeX-Interpreter. Ständig befindet man sich vor dem fragenden Prompt mit dem Fragezeichen. Gibt man dann ein H ein (in der Hoffnung auf Hilfe) kommt ein meist wenig sinnreicher Vorschlag¹³. In der Tat ist es fast immer sinnvoll, den LaTeX-Lauf abubrechen und sich die Quelldatei an der problematischen Stelle anzusehen. In den allermeisten Fällen sind es einfache Tippfehler, die den Lauf unterbrechen, knapp gefolgt vom Vergessen benötigter Pakete, in denen benutzte Befehle definiert sind. Schwer zu finden sind Folgefehler, die darauf aufbauen, daß eine der Hilfsdateien korrumpiert wurde (Umlaut eingeschlichen, usw). Hier hilft meist nur das Löschen sämtlicher Hilfs- und Ausgabedateien.

Auf noch eine Sache soll hingewiesen werden: Korrekturlesen. Nicht nur im Ursprungsdokument, auch und gerade in der Ausgabedatei. Denn ansonsten kann die Rechtschreibung wunderbar gelungen sein, aber nichtsdestoweniger kann einem dann noch die Zeilentrennung ins Wort fallen und ein wenig sinnentstellend arbeiten. Kommt nicht vor?

¹²Es existiert ein in PostScript geschriebener HTTP-Server, der pshttpd.

¹³Wie z.B. „I suspect you have forgotten a ‘}’, causing me to read past where you wanted me to stop. I’ll try to recover; but if the error is serious, you’d better type ‘E’ or ‘X’ now and fix your file.“

n“
a-
in
o-
ls
e-
e-
o-
er
on
n-

dererfamilie in einem Elendsviertel von Lyon zur Welt. Beide Eltern waren Analphabeten. Sein Aufstieg gilt als Beweis dafür, dass die französische Gesellschaft nicht aus sich reproduzierenden Kasten besteht. Und seine Kritik an Sarkozys Auftritten hat inzwischen Folgen: Als der Innenminister in der Nacht zum Donnerstag die Lage in einem Vorort inspizierte, geschah dies ohne das Fernsehen.

— Seite 29

Kommt doch vor. Voranstehend ein Beispiel aus dem „Tagesspiegel“ vom Freitag, den 04.11.2005, Seite 7.

Was man dagegen macht? Man gibt vor, wie getrennt werden darf: Hier wäre die Lösung ein $An\backslash-\alpha\backslash-\text{bet}$ gewesen, die nichtgewünschte Trennmöglichkeit werden dann vom Interpreter bewußt ignoriert.

9 Quellen

Zur Erstellung dieser Arbeit habe ich die ein- oder andere Information aus Literatur entnommen, allerdings den Großteil aus meinem Wissen um und meiner Erfahrung aus der praktischen Arbeit mit LaTeX. Somit ist meine Literaturliste eher als eine Liste von Buchempfehlungen für den Anfänger zu verstehen, aber nichtsdestoweniger bemühe ich mich hier einmal um Vollständigkeit.

Literatur

- [KOP96] Kopka, Helmut: „LaTeX - Band I“, Addison-Wesley, 1996 (Universitätsbibliothek Rostock ST 351 T28 K83(2))
- [NID] Elke u. Michael Niedermair: „LaTeX - das Praxisbuch“, Franzis, 2004 (Universitätsbibliothek Rostock ST 351 T28 N666)
- [SCH05] Petra Schlager und Manfred Thibud: „Wissenschaftlich mit LaTeX arbeiten“, Pearson Studium, 2005 (Universitätsbibliothek Rostock ST 351 T28 S338)

9.1 Internet

Desweiteren wurden die folgenden Internetseiten genutzt und sollen hiermit empfohlen werden:

- Herkunft der Symbolisten aus dem Matheteil → <http://www.feyrer.de/OS/latex.html>
- Deutsche Anwendervereinigung TeX e.V. → <http://www.dante.de>
- Das LaTeX-Kochbuch → <http://www.uni-giessen.de/hrz/tex/>

10 Changelog

- Erste Version: Veröffentlicht nach dem LaTeX-Tutorium am 08.12.2005.
- Zweite Version: Etwas mehr Information zu „inputenc“ und „fontenc“. Veröffentlicht am 22.12.2005.
- Dritte Version: Wesentliche Änderungen an Formulierungen. Nutzbarmachung als wissenschaftlicher Text. Veröffentlicht am 06.12.2006.