

SelfLinux-0.13.0



## LaTeX-Schnelleinführung



Autor: Andreas Kalbitz ([felix@pingos.org](mailto:felix@pingos.org))  
Formatierung: Matthias Hagedorn ([matthias.hagedorn@selflinux.org](mailto:matthias.hagedorn@selflinux.org))  
Lizenz: GPL

## Inhaltsverzeichnis

### 1 Anfängliche Fragestellungen

- 1.1 Was ist LaTeX?
- 1.2 Was ist anders beim Erstellen von LaTeX-basierten Texten?
- 1.3 Welcher Editor ist am besten für LaTeX geeignet?
- 1.4 Gibt es grafische Benutzeroberflächen für LaTeX?
- 1.5 Wie erhalte ich meinen Text in druckbarer Fassung?

### 2 Die Arbeit mit den LaTeX-Standards

- 2.1 Der Grundaufbau eines LaTeX-Dokuments
- 2.2 Die Standard-Klassen
- 2.3 Bilder im Text
- 2.4 Farbe im Text
- 2.5 Tabellen im Text

### 3 Mathematische Formeln in LaTeX

### 4 Zusammenfassung

# 1 Anfängliche Fragestellungen

## 1.1 Was ist LaTeX?

LaTeX ist ein Drucksatzsystem, dessen Ergebnisse weit über das hinausgehen, was man mit einer Schreibmaschine schreiben könnte. Die Qualität der gesetzten Zeichen unterscheidet sich auch von den Druckergebnissen von herkömmlichen Textverarbeitungsprogrammen, da LaTeX Ligaturen verwendet. Das Ausmaß der Fähigkeiten wird aber erst richtig deutlich, wenn man mit Notensatz, Kreuzworträtseln, mathematischen Formeln oder chemischen Symboldarstellungen zu tun hat. Selbst Diagramme lassen sich bei Kenntnis der entsprechenden LaTeX-Befehle programmieren.

## 1.2 Was ist anders beim Erstellen von LaTeX-basierten Texten?

LaTeX ist ein System, das wie eine Programmiersprache Befehle bereitstellt, um ein gewünschtes Ergebnis zu erzielen. Man muss also einen Überblick über die LaTeX-Befehle erhalten, um damit arbeiten zu können. Dies ist auch der Grund, warum häufig gestresste Linux-Anwender eher zum Textprogramm greifen, als LaTeX zu nutzen. Wer die ersten Grundbefehle allerdings gelernt hat, greift gern auf LaTeX zurück, wenn neben dem schnellen Text auch das Layout eine Rolle spielt.

LaTeX benötigt ein installiertes LaTeX-System und einen Editor - mehr nicht. Die Ergebnisse der ersten Verarbeitungsstufe eines Textes liegen im **DeVice-Independend Format** (DVI) vor. Diese Dateien lassen sich dann beispielsweise ins Postscript- oder in PDF-Format verwandeln. Mit dem Programm `pdflatex` lassen sich auch direkt PDF-Dateien erzeugen.

## 1.3 Welcher Editor ist am besten für LaTeX geeignet?

Wichtig für den guten Überblick ist das Syntax-Highlighting, wie es auch in der Programmierung gern genutzt wird. Editoren wie *Vim*, *EMACS*, *Nedit* oder *Kate* bringen die entsprechenden Voraussetzungen mit. So behält man den Überblick über öffnende und schließende Klammern. Ansonsten gibt es keine Besonderheiten zu beachten. Am besten geeignet ist der Lieblingseditor, weil man mit ihm am besten umgehen kann.

## 1.4 Gibt es grafische Benutzeroberflächen für LaTeX?

Ja, in der Tat, die gibt es. Für KDE2 wurde *Ktexmaker2* entwickelt. Sein Nachfolger für KDE3 ist *Kile*. Doch schon vor KDE2 gab es nützliche Oberflächen, wie die *TeX-Shell* (ts). Sie alle bieten die Möglichkeit die Sourcen zu editieren und anschließend eines der LaTeX-Programme aufzurufen, um die DVI-Dateien zu erzeugen. *LyX* geht dabei noch etwas weiter. **WYSIWYM** (What you see, is what you mean) nennt sich die Ansicht, die mit diesem Editor erzeugt wird. *LyX* stellt dabei die Sourcen zusammen. Aus dem Mac-User-Bereich stammt das Programm *TeXMacs*.

## 1.5 Wie erhalte ich meinen Text in druckbarer Fassung?

Zunächst muss der Text, wie im [Abschnitt 2](#) folgt, erstellt werden. Abgespeichert wird er mit der Erweiterung `.tex`. Diese Quelldatei wird dann mit den Befehlen

```
user@linux / $ latex quelldatei.tex
user@linux / $ dvips quelldatei.dvi
user@linux / $ ps2pdf quelldatei.ps
```

übersetzt. Wer keine PDF-Datei als Ergebnis benötigt, kann die letzte Zeile weglassen.

Die einzelnen Dateien lassen sich mit den Programmen `xdvi quelldatei.dvi`, `kghostview quelldatei.ps`, oder `xpdf quelldatei.pdf` bzw. `acroread quelldatei.pdf` betrachten.

## 2 Die Arbeit mit den LaTeX-Standards

### 2.1 Der Grundaufbau eines LaTeX-Dokuments

Jedes LaTeX-Hauptdokument wird in Präambel und Body aufgeteilt. In der Präambel werden LaTeX-Klasse und LaTeX-Styles sowie einige zusätzliche Angaben zu bestimmten Dokumentarten angegeben. In den Body wird der Inhalt mit Befehlen strukturiert eingegeben.

Quelltext 1: Grobgerüst eines deutschsprachigen Artikels

```
\documentclass[a4paper,11pt]{article}
\usepackage{ngerman}
\usepackage[latin1]{inputenc}
\begin{document}
Hier wird der Text des Dokumentes eingefügt.
\end{document}
```

Der Befehl `\documentclass[]{}{}` definiert die Art des Dokumentes. Mehr dazu steht im nächsten Abschnitt. Die Dokumentklasse wird in den geschweiften Klammern angegeben. Die eckigen Klammern nehmen Optionen entgegen. In unserem Beispiel sind das die Voreinstellungen für den Druck auf DIN-A4-Papier mit einer 11pt-Standardschriftgröße. Jedes Hauptdokument darf nur eine Dokumentklasse aufrufen. Die `\usepackage[]{}{}`-Befehle binden alle gewünschten Zusatzpakete ein.

### 2.2 Die Standard-Klassen

LaTeX stellt standardmäßig die Klassen `article`, `book`, `report`, `proc` und `plain` zur Verfügung. Jede hat ihre Besonderheiten. Sie unterscheiden sich in der Art, wie die Gliederung verarbeitet wird, ja sogar die Gliederungsbefehle sind nicht in allen Klassen gleich.

So macht es beim Satz eines Buches Sinn, den Befehl `\part{Titel}` zu benutzen, wenn es in mehrere Teile aufgeliedert werden soll, die dann wiederum Kapitel, Sektionen usw. beinhalten. Dieser Befehl steht darüberhinaus in der Klasse `report` zur Verfügung. Alle anderen Klassen können darauf verzichten. Eine weitere Besonderheit ist die Voreinstellung der `book`-Klasse, Kapitelüberschriften auf eine einzelne Seite zu setzen. Beschränken wir uns hier auf diese Angaben zum `part`-Befehl. Alle anderen Gliederungsebenen sind gleich verwendbar.

## Quelltext 2: Gliederungsbefehle innerhalb eines 1. Kapitels

```
\begin{document}
\chapter{Kapiteltitel}
\section{Thementitel}
\subsection{Titel eines Unterthemas}
\subsubsection{Titel eines Unterthemas der nächst tieferen
Gliederungsstufe}
\paragraph[optinaler hervorhebbarer Absatzanfang] Absatztext der
nicht gesondert hervorgehoben wird.
Hier wird der Text des Dokumentes eingefügt.
\section{Thementitel}
\subsection{Titel eines Unterthemas}
\subsubsection{Titel eines Unterthemas der nächst tieferen
Gliederungsstufe}
\paragraph*{} Absatz der nicht extra in die Gliederung
aufgenommen wird.
\end{Document}
```

So ausgerüstet bin ich in der Lage einen reinen Text zu schreiben, dessen Layout ich eigentlich nicht beeinflusst habe. Latex übernimmt einige Standardeinstellungen die der Fortgeschrittene Benutzer zu modifizieren weiß.

## 2.3 Bilder im Text

Bilder können auf unterschiedliche Weise erzeugt werden. So gibt es zum einen die bereits erwähnte Möglichkeit direkt über entsprechende Befehle Grafiken zu erzeugen. Das gehört aber in keine Kurzbeschreibung. Um schnell ein Bild in einen Text einzubinden, muss man die Präambel erweitern.

## Quelltext 3: Änderungen in der Präambel und im Textkörper zum

```
Einfügen einer eps-Grafik.
....
\usepackage[dvips]{graphicx}
\usepackage{epsfig}
....
\begin{document}
....
\includegraphics[width=13cm]{Bilddatei}
....
\end{document}
```

Dabei wird das Bild in den Absatz eingefügt. Und ein Zeilenumbruch angehängt. So erscheint der Text ober und unterhalb des Bildes. Einen Textumlauf erreichen wir damit noch nicht. In diesem Beispiel haben wir eine **eps**-Grafik verwendet. (Encapsulated Postscript). Das Paket **graphicx** sorgt für die Konvertierung und Einbindung in die DVI-Datei.

## 2.4 Farbe im Text

Um Farbe in den Text zu bringen, brauchen wir das Zusatzpaket **color** in der Präambel. Quelltext 4 demonstriert, wie man einen Kasten erzeugt, der 3 verschiedene Farbwerte für den Rand, das Innere und den Text eines Kastens akzeptiert.

Quelltext 4: Änderungen in der Präambel und im Textkörper für die Benutzung farbiger Rahmen und farbiger Texte

```
....  
\usepackage{color}  
....  
\begin{document}  
....  
\fcolorbox{Rahmenfarbe}{Füllfarbe}{\textcolor{Textfarbe}{Text im  
Kasten.}}  
....  
\end{document}
```

Leider stoßen wir hier auf ein Problem. Der Befehl `\fcolorbox` färbt nur die Elemente einer Zeile. Ist der Text zu lang, bricht die Box aus dem normalen Text aus und wird über den rechten Rand verlängert. Es gibt also keinen Zeilenumbruch mehr. Abhilfe schafft hier zum Beispiel die `parbox`-Umgebung.

Quelltext 5: Parbox innerhalb einer farbigen Box

```
....  
\fcolorbox{red}{blue}{\parbox{\textwidth}{\textcolor{white}{  
Textinhalt der weit über die Textbreite hinausragt.  
belegt werden können.}}}  
....
```

Mit der Angabe `\textwidth` benutzen wir eine LaTeX-Variable die den Wert der Textbreite enthält. Hier kann auch eine Breitenangabe wie 10cm stehen. Die `Parbox` nimmt den Text auf und sorgt für den Zeilenumbruch. Ob der Text darin nun gefärbt wird oder nicht, hat auf das Verhalten der `Parbox` keine Auswirkung. `Parbox` funktioniert übrigens auch ohne den `fcolorbox`-Befehl.

## 2.5 Tabellen im Text

Tabellen sind Gestaltungselemente die für Übersicht sorgen sollen. Wer einen Tabellenquellcode liest, der viele Spalten und Zeilen enthält kann schnell mal den Überblick verlieren. Dennoch sieht in der Ausgabedatei später alles so aus, wie in einer richtigen Tabelle.

Es gibt in LaTeX verschiedene Möglichkeiten, Tabellen zu erzeugen. Die einfachste für kleine tabellarische Übersichten ist die `Tabular`-Umgebung.

Quelltext 6: Beispieltabelle mit der `Tabular`-Umgebung

```
....  
\begin{tabular}{|l|c|r|}  
\hline  
linksbündig & zentriert & rechtsbündig \\  
\hline  
1 & 2 & 3,141\\  
\hline  
\end{tabular}  
....
```

Die Angaben in den geschweiften Klammern nach dem `\begin{tabular}`-Befehl dienen der Formatierung

der Tabelle. Die senkrechten Striche stehen für die senkrechten Spaltenbegrenzungen der Tabelle. Lässt man sie weg, werden die Spalten ohne Linien formatiert. Die Buchstaben `l`, `c` und `r` stehen für die Ausrichtung der Zelleninhalte. Der Befehl `\hline` fügt horizontale Linien in die Tabelle ein. Jede Tabellenzeile darf nur so viele Feldinhalte haben, wie formatierte Felder in der Format-Klammer stehen. Ein `{rrrr}` ergibt also eine Tabelle mit vier rechtsbündig ausgerichteten Spalten. Eine Zelle wird von der anderen durch das `&`-Zeichen getrennt. Jede Tabellenzeile schließt mit einem `\\` (=Zeilenumbruch) ab.

Will man die Tabelle für die Formatierung längerer Textpassagen benutzen, kann in der Format-Klammer anstelle von `l`, `c` oder `r` auch ein Absatz mit fester Länge eingegeben werden.

Quelltext 6: Beispieltabelle Absatz-Zellen

```
...
\begin{tabular}{|p{3cm}|p{7cm}|}
\hline
Zeit & Ereignis \\
\hline
2001 & Linuxtag in Stuttgart\\
2002 & Linuxtag in Karlsruhe\\
2003 & Linuxtag in Karlsruhe\\
\hline
\end{tabular}
...
```

Die Absatzbreiten sollten hier auf jeden Fall so gewählt werden, dass sie in der Summe die Textbreite nicht überschreiten.

### 3 Mathematische Formeln in LaTeX

Wer nur mal zwischendurch in LaTeX eine Formel eingeben will, kann das ganz schnell durch Umschalten in den Math-Modus durch Einklammerung der Formel in zwei  $\$$ -Zeichen erreichen. Hier ein paar Beispiele für Gleichungen mit Potenz-, Wurzel- und Bruchausdrücken.

Quelltext 6: Formelbeispiele

```
...  
$ f(x)=3x^2+2x+2 $\\  
$ f(x)=\frac{x+3}{x^2-9} $\\  
$ f(x)=\frac{1}{\sqrt{x^2}} $\\  
$ y=\sqrt[3]{8} $\\  
...
```

Die Beispiele enthalten Befehle, die sich ineinander verschachteln lassen. Wer ausführlichere Formelkonstrukte erstellen möchte, sollte aber unbedingt auf die [AMS-LaTeX-Pakete](#) zurückgreifen, die verschiedene Umgebungen für alle nur erdenklichen mathematischen Formeln und Symbole bereitstellen.

### 4 Zusammenfassung

Eine Kurzbeschreibung kann nur in ein Thema einführen. Mit dieser Beschreibung sollte es möglich sein, erste Gehversuche mit LaTeX zu machen. Hier kann nicht der Anspruch auf Vollständigkeit erhoben werden. Eine umfangreichere Referenz ist im Entstehen und wird in den nächsten SelfLinux-Versionen stetig ausgebaut.