

# Schnelleinführung in das Textsatzsystem „ $\text{\LaTeX}$ “

PD Dr. Karin Halupczok  
Email: Karin.Halupczok@math.uni-freiburg.de

22. und 23. Januar 2010

Zweitätiges MeMPhys-Seminar  
Fakultät für Mathematik und Physik

Teil 1: Allgemeine Einführung und Grundlagen,  
Benutzung unter Linux/emacs

<http://home.mathematik.uni-freiburg.de/halupczok/diverses.html>

Allgemeine Einführung: Was ist  $\text{\LaTeX}$ ?

Allgemeine Grundlagen: Aufbau eines  $\text{\LaTeX}$ -Dokuments

Benutzung von  $\text{\LaTeX}$  unter Linux, der Editor emacs

# Was ist $\text{\LaTeX}$ ? – Zur Geschichte von $\text{\LaTeX}$

$\text{\LaTeX}$  (gesprochen „Laa-Tech“) ist ein Computer -  
Satzprogramm mit sehr umfangreichen Möglichkeiten.

# Was ist $\text{\LaTeX}$ ? – Zur Geschichte von $\text{\LaTeX}$

$\text{\LaTeX}$  (gesprochen „Laa-Tech“) ist ein Computer -  
Satzprogramm mit sehr umfangreichen Möglichkeiten.  
1977: Donald E. Knuth entwickelte die Programmiersprache  $\text{\TeX}$   
(tau epsilon chi – sprich „Tech“) und Metafont zum professionellen  
Textsatz.

# Was ist $\text{\LaTeX}$ ? – Zur Geschichte von $\text{\LaTeX}$

$\text{\LaTeX}$  (gesprochen „Laa-Tech“) ist ein Computer -  
Satzprogramm mit sehr umfangreichen Möglichkeiten.  
1977: Donald E. Knuth entwickelte die Programmiersprache  $\text{\TeX}$   
(tau epsilon chi – sprich „Tech“) und Metafont zum professionellen  
Textsatz.

In den 1980er Jahren wurde  $\text{\LaTeX}$  (sprich „Laa-Tech“) von Leslie  
Lamport zunächst als ein Makropaket für  $\text{\TeX}$  entwickelt, d. h. als  
eine Sammlung von Befehlen und Funktionen in  $\text{\TeX}$ .

# Was ist $\text{\LaTeX}$ ? – Zur Geschichte von $\text{\LaTeX}$

$\text{\LaTeX}$  (gesprochen „Laa-Tech“) ist ein Computer -  
Schriftsatzprogramm mit sehr umfangreichen Möglichkeiten.  
1977: Donald E. Knuth entwickelte die Programmiersprache  $\text{\TeX}$   
(tau epsilon chi – sprich „Tech“) und Metafont zum professionellen  
Textsatz.

In den 1980er Jahren wurde  $\text{\LaTeX}$  (sprich „Laa-Tech“) von Leslie  
Lamport zunächst als ein Makropaket für  $\text{\TeX}$  entwickelt, d. h. als  
eine Sammlung von Befehlen und Funktionen in  $\text{\TeX}$ .

Es gibt  $\text{\LaTeX}$ -Versionen für alle gängigen Betriebssysteme.

# Für welche Betriebssysteme gibt es $\text{\LaTeX}$ ?

$\text{\LaTeX}$  ist frei nutzbar und betriebssystemunabhängig.

# Für welche Betriebssysteme gibt es $\text{\LaTeX}$ ?

$\text{\LaTeX}$  ist frei nutzbar und betriebssystemunabhängig.

Die älteste Implementierung: Unter Unix/Linux.

Besonderheiten von Unix-Systemen: Rechnernetze für mehrere Benutzer, Dateien mit Benutzerrechten, „multitasking“, Stabilität, etc.

# Für welche Betriebssysteme gibt es L<sup>A</sup>T<sub>E</sub>X?

L<sup>A</sup>T<sub>E</sub>X ist frei nutzbar und betriebssystemunabhängig.

Die älteste Implementierung: Unter Unix/Linux.

Besonderheiten von Unix-Systemen: Rechnernetze für mehrere Benutzer, Dateien mit Benutzerrechten, „multitasking“, Stabilität, etc.

Für Linux bei DANTE e. V. verfügbar unter

<http://www.dante.de/software/ctan/>

# Für welche Betriebssysteme gibt es $\text{\LaTeX}$ ?

$\text{\LaTeX}$  ist frei nutzbar und betriebssystemunabhängig.

Die älteste Implementierung: Unter Unix/Linux.

Besonderheiten von Unix-Systemen: Rechnernetze für mehrere Benutzer, Dateien mit Benutzerrechten, „multitasking“, Stabilität, etc.

Für Linux bei DANTE e. V. verfügbar unter

<http://www.dante.de/software/ctan/>

Bekannteste Implementierung für Macintosh: Oz $\text{\TeX}$  unter

<http://www.trevorrow.com/oztex/>

# Für welche Betriebssysteme gibt es $\text{\LaTeX}$ ?

$\text{\LaTeX}$  ist frei nutzbar und betriebssystemunabhängig.

Die älteste Implementierung: Unter Unix/Linux.

Besonderheiten von Unix-Systemen: Rechnernetze für mehrere Benutzer, Dateien mit Benutzerrechten, „multitasking“, Stabilität, etc.

Für Linux bei DANTE e. V. verfügbar unter

<http://www.dante.de/software/ctan/>

Bekannteste Implementierung für Macintosh: Oz $\text{\TeX}$  unter

<http://www.trevorrow.com/oztex/>

Bekannteste Implementierung für MS-Windows: MiK $\text{\TeX}$  unter

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

# Grundlegendes Prinzip von L<sup>A</sup>T<sub>E</sub>X

- ▶ Texte werden strukturiert: Erst kommt der Inhalt, dann das Layout, welches zentral veränderbar ist. Damit ist es leicht, ein konsistentes Aussehen zu erzeugen.

# Grundlegendes Prinzip von $\text{\LaTeX}$

- ▶ Texte werden strukturiert: Erst kommt der Inhalt, dann das Layout, welches zentral veränderbar ist. Damit ist es leicht, ein konsistentes Aussehen zu erzeugen.
- ▶ Anweisungen an den „Setzer“ (also an  $\text{\LaTeX}$ ) des Dokuments, wie z. B. „hier aktuelles Datum einsetzen“, stehen in der  $\text{\LaTeX}$ -Quelldatei.  
Beispiel: `\today` setzt das aktuelle Datum in den Text ein.

# Grundlegendes Prinzip von L<sup>A</sup>T<sub>E</sub>X

- ▶ Texte werden strukturiert: Erst kommt der Inhalt, dann das Layout, welches zentral veränderbar ist. Damit ist es leicht, ein konsistentes Aussehen zu erzeugen.
- ▶ Anweisungen an den „Setzer“ (also an L<sup>A</sup>T<sub>E</sub>X) des Dokuments, wie z. B. „hier aktuelles Datum einsetzen“, stehen in der L<sup>A</sup>T<sub>E</sub>X-Quelldatei.  
Beispiel: `\today` setzt das aktuelle Datum in den Text ein.
- ▶ Man erstellt also eine Datei *name.tex* mit dem Quellcode. In dieser wird das gewünschte Ergebnis programmiert.

## ... Grundlegendes Prinzip von L<sup>A</sup>T<sub>E</sub>X

- ▶ Die Quelldatei *name.tex* wird mit dem Programm *latex* bzw. *pdflatex* kompiliert. Dieses erzeugt eine Ausgabedatei *name.dvi* bzw. *name.pdf*

## ... Grundlegendes Prinzip von L<sup>A</sup>T<sub>E</sub>X

- ▶ Die Quelldatei *name.tex* wird mit dem Programm *latex* bzw. *pdflatex* kompiliert. Dieses erzeugt eine Ausgabedatei *name.dvi* bzw. *name.pdf*
- ▶ *dvi* ist das ursprüngliche Ausgabeformat von L<sup>A</sup>T<sub>E</sub>X; es kann mit dem Programm *dvips* in PostScript (Druckersprache; Dateierendung *ps*) verwandelt werden. Mit *ps2pdf* kann eine *ps*-Datei in eine *pdf*-Datei umgewandelt werden.

## ... Grundlegendes Prinzip von L<sup>A</sup>T<sub>E</sub>X

- ▶ Die Quelldatei *name.tex* wird mit dem Programm *latex* bzw. *pdflatex* kompiliert. Dieses erzeugt eine Ausgabedatei *name.dvi* bzw. *name.pdf*
- ▶ *dvi* ist das ursprüngliche Ausgabeformat von L<sup>A</sup>T<sub>E</sub>X; es kann mit dem Programm *dvips* in PostScript (Druckersprache; Dateiendung *ps*) verwandelt werden. Mit *ps2pdf* kann eine *ps*-Datei in eine *pdf*-Datei umgewandelt werden.
- ▶ Daneben gibt es Hilfsdateien:  
stets *name.log* (Log-Datei), *name.aux* (allgemeine Hilfsdatei);  
bei Bedarf andere, z. B. *name.toc* (für Inhaltsverzeichnis)

# Nachteile von L<sup>A</sup>T<sub>E</sub>X

- ▶ man sieht nicht sofort das Ergebnis

# Nachteile von L<sup>A</sup>T<sub>E</sub>X

- ▶ man sieht nicht sofort das Ergebnis
- ▶ man muss die „Programmiersprache“ lernen  
→ lange Einarbeitungszeit

# Nachteile von L<sup>A</sup>T<sub>E</sub>X

- ▶ man sieht nicht sofort das Ergebnis
- ▶ man muss die „Programmiersprache“ lernen  
→ lange Einarbeitungszeit
- ▶ etwas umständlich für schnelle, kleine Texte

# Nachteile von L<sup>A</sup>T<sub>E</sub>X

- ▶ man sieht nicht sofort das Ergebnis
- ▶ man muss die „Programmiersprache“ lernen  
→ lange Einarbeitungszeit
- ▶ etwas umständlich für schnelle, kleine Texte
- ▶ der Quellcode ist sensibel gegenüber kleinen Fehlern

# Nachteile von L<sup>A</sup>T<sub>E</sub>X

- ▶ man sieht nicht sofort das Ergebnis
- ▶ man muss die „Programmiersprache“ lernen  
→ lange Einarbeitungszeit
- ▶ etwas umständlich für schnelle, kleine Texte
- ▶ der Quellcode ist sensibel gegenüber kleinen Fehlern
- ▶ manchmal gibt es Unverträglichkeiten zwischen verschiedenen Erweiterungspaketen

# Vorteile von L<sup>A</sup>T<sub>E</sub>X

- ▶ präzise Programmierung des Ergebnisses möglich

## Vorteile von L<sup>A</sup>T<sub>E</sub>X

- ▶ präzise Programmierung des Ergebnisses möglich
- ▶ keine versteckten Steuerzeichen

# Vorteile von L<sup>A</sup>T<sub>E</sub>X

- ▶ präzise Programmierung des Ergebnisses möglich
- ▶ keine versteckten Steuerzeichen
- ▶ „ästhetischer Textsatz“

# Vorteile von L<sup>A</sup>T<sub>E</sub>X

- ▶ präzise Programmierung des Ergebnisses möglich
- ▶ keine versteckten Steuerzeichen
- ▶ „ästhetischer Textsatz“
- ▶ bestes Programm für mathematischen Formelsatz

# Vorteile von L<sup>A</sup>T<sub>E</sub>X

- ▶ präzise Programmierung des Ergebnisses möglich
- ▶ keine versteckten Steuerzeichen
- ▶ „ästhetischer Textsatz“
- ▶ bestes Programm für mathematischen Formelsatz
- ▶ automatisches Inhaltsverzeichnis, bequeme Querverweise, u.ä.

# Vorteile von L<sup>A</sup>T<sub>E</sub>X

- ▶ präzise Programmierung des Ergebnisses möglich
- ▶ keine versteckten Steuerzeichen
- ▶ „ästhetischer Textsatz“
- ▶ bestes Programm für mathematischen Formelsatz
- ▶ automatisches Inhaltsverzeichnis, bequeme Querverweise, u.ä.
- ▶ Möglichkeit zum Programmieren eigener Befehle

# Vorteile von L<sup>A</sup>T<sub>E</sub>X

- ▶ präzise Programmierung des Ergebnisses möglich
- ▶ keine versteckten Steuerzeichen
- ▶ „ästhetischer Textsatz“
- ▶ bestes Programm für mathematischen Formelsatz
- ▶ automatisches Inhaltsverzeichnis, bequeme Querverweise, u.ä.
- ▶ Möglichkeit zum Programmieren eigener Befehle
- ▶ Quelldateien sind Textdateien, also klein, lesbar und systemunabhängig (aber Zeichenkodierung beachten!)

# Vorteile von L<sup>A</sup>T<sub>E</sub>X

- ▶ präzise Programmierung des Ergebnisses möglich
- ▶ keine versteckten Steuerzeichen
- ▶ „ästhetischer Textsatz“
- ▶ bestes Programm für mathematischen Formelsatz
- ▶ automatisches Inhaltsverzeichnis, bequeme Querverweise, u.ä.
- ▶ Möglichkeit zum Programmieren eigener Befehle
- ▶ Quelldateien sind Textdateien, also klein, lesbar und systemunabhängig (aber Zeichenkodierung beachten!)
- ▶ Open–Source–Programm:  
kostenlos und kann von jedem weiterentwickelt werden  
(daher reichhaltiges Angebot an Erweiterungen)

# Vorteile von L<sup>A</sup>T<sub>E</sub>X

- ▶ präzise Programmierung des Ergebnisses möglich
- ▶ keine versteckten Steuerzeichen
- ▶ „ästhetischer Textsatz“
- ▶ bestes Programm für mathematischen Formelsatz
- ▶ automatisches Inhaltsverzeichnis, bequeme Querverweise, u.ä.
- ▶ Möglichkeit zum Programmieren eigener Befehle
- ▶ Quelldateien sind Textdateien, also klein, lesbar und systemunabhängig (aber Zeichenkodierung beachten!)
- ▶ Open–Source–Programm:  
kostenlos und kann von jedem weiterentwickelt werden  
(daher reichhaltiges Angebot an Erweiterungen)
- ▶ verschiedene Versionen sind untereinander verträglich

# Welche Möglichkeiten hat L<sup>A</sup>T<sub>E</sub>X?

Je nach Anwendung erlaubt L<sup>A</sup>T<sub>E</sub>X eine Fülle an Erweiterungen mittels dazuladbaren Paketen.

Diese können sein:

- ▶ Sprachen (griechisch, russisch, arabisch, japanisch, hindi, ägyptische Hieroglyphen. . .)

# Welche Möglichkeiten hat $\text{\LaTeX}$ ?

Je nach Anwendung erlaubt  $\text{\LaTeX}$  eine Fülle an Erweiterungen mittels dazuladbaren Paketen.

Diese können sein:

- ▶ Sprachen (griechisch, russisch, arabisch, japanisch, hindi, ägyptische Hieroglyphen. . .)
- ▶ Diagramme, Grafik, Schaltkreise, Feynman-Diagramme

# Welche Möglichkeiten hat $\text{\LaTeX}$ ?

Je nach Anwendung erlaubt  $\text{\LaTeX}$  eine Fülle an Erweiterungen mittels dazuladbaren Paketen.

Diese können sein:

- ▶ Sprachen (griechisch, russisch, arabisch, japanisch, hindi, ägyptische Hieroglyphen. . .)
- ▶ Diagramme, Grafik, Schaltkreise, Feynman-Diagramme
- ▶ Chemische Formeln

# Welche Möglichkeiten hat L<sup>A</sup>T<sub>E</sub>X?

Je nach Anwendung erlaubt L<sup>A</sup>T<sub>E</sub>X eine Fülle an Erweiterungen mittels dazuladbaren Paketen.

Diese können sein:

- ▶ Sprachen (griechisch, russisch, arabisch, japanisch, hindi, ägyptische Hieroglyphen. . .)
- ▶ Diagramme, Grafik, Schaltkreise, Feynman-Diagramme
- ▶ Chemische Formeln
- ▶ Musik: Notensetzen

# Welche Möglichkeiten hat L<sup>A</sup>T<sub>E</sub>X?

Je nach Anwendung erlaubt L<sup>A</sup>T<sub>E</sub>X eine Fülle an Erweiterungen mittels dazuladbaren Paketen.

Diese können sein:

- ▶ Sprachen (griechisch, russisch, arabisch, japanisch, hindi, ägyptische Hieroglyphen. . .)
- ▶ Diagramme, Grafik, Schaltkreise, Feynman-Diagramme
- ▶ Chemische Formeln
- ▶ Musik: Notensetzen
- ▶ Schachpartien

# Welche Möglichkeiten hat L<sup>A</sup>T<sub>E</sub>X?

Je nach Anwendung erlaubt L<sup>A</sup>T<sub>E</sub>X eine Fülle an Erweiterungen mittels dazuladbaren Paketen.

Diese können sein:

- ▶ Sprachen (griechisch, russisch, arabisch, japanisch, hindi, ägyptische Hieroglyphen. . .)
- ▶ Diagramme, Grafik, Schaltkreise, Feynman-Diagramme
- ▶ Chemische Formeln
- ▶ Musik: Notensetzen
- ▶ Schachpartien
- ▶ Mathematik

# Welche Möglichkeiten hat $\text{\LaTeX}$ ?

Je nach Anwendung erlaubt  $\text{\LaTeX}$  eine Fülle an Erweiterungen mittels dazuladbaren Paketen.

Diese können sein:

- ▶ Sprachen (griechisch, russisch, arabisch, japanisch, hindi, ägyptische Hieroglyphen. . .)
- ▶ Diagramme, Grafik, Schaltkreise, Feynman-Diagramme
- ▶ Chemische Formeln
- ▶ Musik: Notensetzen
- ▶ Schachpartien
- ▶ Mathematik
- ▶ beamer-Paket statt Powerpoint!

# Welche Möglichkeiten hat $\text{\LaTeX}$ ?

Je nach Anwendung erlaubt  $\text{\LaTeX}$  eine Fülle an Erweiterungen mittels dazuladbaren Paketen.

Diese können sein:

- ▶ Sprachen (griechisch, russisch, arabisch, japanisch, hindi, ägyptische Hieroglyphen. . .)
- ▶ Diagramme, Grafik, Schaltkreise, Feynman-Diagramme
- ▶ Chemische Formeln
- ▶ Musik: Notensetzen
- ▶ Schachpartien
- ▶ Mathematik
- ▶ beamer-Paket statt Powerpoint!
- ▶ und viel mehr. . .

# Literatur und Hilfe

- ▶ „T<sub>E</sub>X users group“: [www.tug.org](http://www.tug.org)

# Literatur und Hilfe

- ▶ „T<sub>E</sub>X users group“: [www.tug.org](http://www.tug.org)
- ▶ „Dante e. V.“: [www.dante.de](http://www.dante.de)

# Literatur und Hilfe

- ▶ „T<sub>E</sub>X users group“: [www.tug.org](http://www.tug.org)
- ▶ „Dante e. V.“: [www.dante.de](http://www.dante.de)
- ▶ verschiedene Einführungen im Netz, z.B.  
Oetiker et.al. „The not so short introduction to L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>“  
[www.dante.de/CTAN/info/lshort/english/lshort.pdf](http://www.dante.de/CTAN/info/lshort/english/lshort.pdf)

# Literatur und Hilfe

- ▶ „T<sub>E</sub>X users group“: [www.tug.org](http://www.tug.org)
- ▶ „Dante e. V.“: [www.dante.de](http://www.dante.de)
- ▶ verschiedene Einführungen im Netz, z.B.  
Oetiker et.al. „The not so short introduction to L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>“  
[www.dante.de/CTAN/info/lshort/english/lshort.pdf](http://www.dante.de/CTAN/info/lshort/english/lshort.pdf)
- ▶ Thomas F. Sturm: „L<sup>A</sup>T<sub>E</sub>X – Einführung in das Textsatzsystem“, RRNZ-Handbuch (günstiges Skript, gibt es im RZ, vgl. <http://portal.uni-freiburg.de/rz/dienste/software/docu>)

# Literatur und Hilfe

- ▶ „T<sub>E</sub>X users group“: [www.tug.org](http://www.tug.org)
- ▶ „Dante e. V.“: [www.dante.de](http://www.dante.de)
- ▶ verschiedene Einführungen im Netz, z.B.  
Oetiker et.al. „The not so short introduction to L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>“  
[www.dante.de/CTAN/info/lshort/english/lshort.pdf](http://www.dante.de/CTAN/info/lshort/english/lshort.pdf)
- ▶ Thomas F. Sturm: „L<sup>A</sup>T<sub>E</sub>X – Einführung in das Textsatzsystem“, RRNZ-Handbuch (günstiges Skript, gibt es im RZ, vgl. <http://portal.uni-freiburg.de/rz/dienste/software/docu>)
- ▶ Helmut Kopka „L<sup>A</sup>T<sub>E</sub>X“, Addison Wesley)  
(früher *das* Buch für Einsteiger, jetzt in drei Bänden)

# Literatur und Hilfe

- ▶ „T<sub>E</sub>X users group“: [www.tug.org](http://www.tug.org)
- ▶ „Dante e. V.“: [www.dante.de](http://www.dante.de)
- ▶ verschiedene Einführungen im Netz, z.B.  
Oetiker et.al. „The not so short introduction to L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>“  
[www.dante.de/CTAN/info/lshort/english/lshort.pdf](http://www.dante.de/CTAN/info/lshort/english/lshort.pdf)
- ▶ Thomas F. Sturm: „L<sup>A</sup>T<sub>E</sub>X – Einführung in das Textsatzsystem“, RRNZ-Handbuch (günstiges Skript, gibt es im RZ, vgl. <http://portal.uni-freiburg.de/rz/dienste/software/docu>)
- ▶ Helmut Kopka „L<sup>A</sup>T<sub>E</sub>X“, Addison Wesley)  
(früher *das* Buch für Einsteiger, jetzt in drei Bänden)
- ▶ Mittelbach, Goossens „Der L<sup>A</sup>T<sub>E</sub>X-Begleiter“, Addison Wesley  
„Das Bernhardiner-Buch“, einer der Klassiker. Beschreibt vollständig den neuen Standard L<sup>A</sup>T<sub>E</sub>X2e und zahlreiche Pakete unter anderem aus den Bereichen Grafik, Tabellen und Mathematik. Geeignet für Einsteiger und Fortgeschrittene

## ...mehr Literatur und Hilfe

- ▶ Walter Schmidt u. a.:  $\text{\LaTeX}$ 2e Kurzbeschreibung, sehr gute Einführung in  $\text{\LaTeX}$ , ist für den Anfang völlig ausreichend, unter <ftp://ftp.dante.de/tex-archive/info/lshort/german/>

## ...mehr Literatur und Hilfe

- ▶ Walter Schmidt u. a.:  $\text{\LaTeX}2\text{e}$  Kurzbeschreibung, sehr gute Einführung in  $\text{\LaTeX}$ , ist für den Anfang völlig ausreichend, unter <ftp://ftp.dante.de/tex-archive/info/lshort/german/>
- ▶ Liste der  $\text{\LaTeX}$ -Symbole im Netz:  
[www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf](http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf)

## ...mehr Literatur und Hilfe

- ▶ Walter Schmidt u. a.:  $\text{\LaTeX}2\text{e}$  Kurzbeschreibung, sehr gute Einführung in  $\text{\LaTeX}$ , ist für den Anfang völlig ausreichend, unter <ftp://ftp.dante.de/tex-archive/info/lshort/german/>
- ▶ Liste der  $\text{\LaTeX}$ -Symbole im Netz:  
[www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf](http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf)
- ▶ Manuela Jürgens:  $\text{\LaTeX}$ - eine Einführung und ein bißchen mehr... Einführung in  $\text{\LaTeX}$ , Teil 1: <ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a0260003.pdf>

## ...mehr Literatur und Hilfe

- ▶ Walter Schmidt u. a.:  $\text{\LaTeX}2\text{e}$  Kurzbeschreibung, sehr gute Einführung in  $\text{\LaTeX}$ , ist für den Anfang völlig ausreichend, unter <ftp://ftp.dante.de/tex-archive/info/lshort/german/>
- ▶ Liste der  $\text{\LaTeX}$ -Symbole im Netz:  
[www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf](http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf)
- ▶ Manuela Jürgens:  $\text{\LaTeX}$ - eine Einführung und ein bißchen mehr... Einführung in  $\text{\LaTeX}$ , Teil 1: <ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a0260003.pdf>
- ▶ Manuela Jürgens: :  $\text{\LaTeX}$ - Fortgeschrittene Anwendungen Einführung in  $\text{\LaTeX}$ , Teil 2: <ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a0279510.pdf>

## ...mehr Literatur und Hilfe

- ▶ Walter Schmidt u. a.:  $\text{\LaTeX}2\text{e}$  Kurzbeschreibung, sehr gute Einführung in  $\text{\LaTeX}$ , ist für den Anfang völlig ausreichend, unter <ftp://ftp.dante.de/tex-archive/info/lshort/german/>
- ▶ Liste der  $\text{\LaTeX}$ -Symbole im Netz:  
[www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf](http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf)
- ▶ Manuela Jürgens:  $\text{\LaTeX}$ - eine Einführung und ein bißchen mehr... Einführung in  $\text{\LaTeX}$ , Teil 1: <ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a0260003.pdf>
- ▶ Manuela Jürgens: :  $\text{\LaTeX}$ - Fortgeschrittene Anwendungen Einführung in  $\text{\LaTeX}$ , Teil 2: <ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a0279510.pdf>
- ▶ Beschreibung der  $\text{\TeX}$ -Pakete:  
<http://texcatalogue.sarovar.org/brief.html>

## ...mehr Literatur und Hilfe

- ▶ Walter Schmidt u. a.:  $\text{\LaTeX}2\text{e}$  Kurzbeschreibung, sehr gute Einführung in  $\text{\LaTeX}$ , ist für den Anfang völlig ausreichend, unter <ftp://ftp.dante.de/tex-archive/info/lshort/german/>
- ▶ Liste der  $\text{\LaTeX}$ -Symbole im Netz:  
[www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf](http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf)
- ▶ Manuela Jürgens:  $\text{\LaTeX}$ - eine Einführung und ein bißchen mehr... Einführung in  $\text{\LaTeX}$ , Teil 1: <ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a0260003.pdf>
- ▶ Manuela Jürgens: :  $\text{\LaTeX}$ - Fortgeschrittene Anwendungen Einführung in  $\text{\LaTeX}$ , Teil 2: <ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/a0279510.pdf>
- ▶ Beschreibung der  $\text{\TeX}$ -Pakete:  
<http://texcatalogue.sarovar.org/brief.html>
- ▶ Viel Material unter:  
<http://cips02.physik.uni-bonn.de/pool/soft/tex/#a5b1>

Allgemeine Einführung: Was ist  $\text{\LaTeX}$ ?

Allgemeine Grundlagen: Aufbau eines  $\text{\LaTeX}$ -Dokuments

Benutzung von  $\text{\LaTeX}$  unter Linux, der Editor emacs

## Wie sieht eine .tex-Datei aus?

Erst kommt der sogenannte Vorspann, auch Präambel genannt, in dem allgemeine Parameters des Dokuments beschrieben werden, dann in der `document`-Umgebung der eigentliche Text.

## Wie sieht eine .tex-Datei aus?

Erst kommt der sogenannte Vorspann, auch Präambel genannt, in dem allgemeine Parameters des Dokuments beschrieben werden, dann in der document-Umgebung der eigentliche Text. Z. B.:

```
\documentclass[12pt]{article}
\usepackage{ngerman}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
```

## Wie sieht eine .tex-Datei aus?

Erst kommt der sogenannte Vorspann, auch Präambel genannt, in dem allgemeine Parameters des Dokuments beschrieben werden, dann in der document-Umgebung der eigentliche Text. Z. B.:

```
\documentclass[12pt]{article}
\usepackage{ngerman}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}

\begin{document}
```

Hier steht nun der `\textit{eigentliche}` Text des Dokuments.

```
\end{document}
```

## Wie sieht eine .tex-Datei aus?

Erst kommt der sogenannte Vorspann, auch Präambel genannt, in dem allgemeine Parameters des Dokuments beschrieben werden, dann in der document-Umgebung der eigentliche Text. Z. B.:

```
\documentclass[12pt]{article}
\usepackage{ngerman}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}

\begin{document}
```

Hier steht nun der `\textit{eigentliche}` Text des Dokuments.

```
\end{document}
```

Bemerkung: Manchmal ist latin1 statt utf8 nötig

## Bemerkung zum praktischen Loslegen

Mit den vorigen Befehlen kann man bereits ein fertiges  $\text{\LaTeX}$ -Dokument erstellen. Z. B. kann man in den Textteil einfach mit der Maus kopierte Absätze aus dem Internet (etwa aus wikipedia) einfügen.

Dabei ist oft Folgendes zu beachten:

Gänsefüßchen, sofern man als Sprache „german“ bzw. „ngerman“ (alte bzw. neue Rechtschreibung) eingestellt hat, sind umzuwandeln in die Befehle:

## Bemerkung zum praktischen Loslegen

Mit den vorigen Befehlen kann man bereits ein fertiges  $\text{\LaTeX}$ -Dokument erstellen. Z. B. kann man in den Textteil einfach mit der Maus kopierte Absätze aus dem Internet (etwa aus wikipedia) einfügen.

Dabei ist oft Folgendes zu beachten:

Gänsefüßchen, sofern man als Sprache „german“ bzw. „ngerman“ (alte bzw. neue Rechtschreibung) eingestellt hat, sind umzuwandeln in die Befehle:

“ ‘ (backtick) oder `\g1qq` für die deutschen Gänsefüßchen unten

## Bemerkung zum praktischen Loslegen

Mit den vorigen Befehlen kann man bereits ein fertiges  $\text{\LaTeX}$ -Dokument erstellen. Z. B. kann man in den Textteil einfach mit der Maus kopierte Absätze aus dem Internet (etwa aus wikipedia) einfügen.

Dabei ist oft Folgendes zu beachten:

Gänsefüßchen, sofern man als Sprache „german“ bzw. „ngerman“ (alte bzw. neue Rechtschreibung) eingestellt hat, sind umzuwandeln in die Befehle:

“ ‘ (backtick) oder `\g1qq` für die deutschen Gänsefüßchen unten

“ ’ (tick, der gerade Strich) oder `\grqq` für die deutschen Gänsefüßchen oben

## Aufbau der Befehle in $\LaTeX$

Eine Anweisung in  $\LaTeX$  beginnt immer mit einem Schrägstrich nach links, dem sogenannten *Backslash*, dann folgt der Befehlsname, dem in geschweiften Klammern *Argumente* bzw. *Parameter* übergeben werden können. Vor den Argumenten können in eckigen Klammern sogenannte *Optionen* übergeben werden, damit werden Einstellungen für den Befehl vorgenommen.

## Aufbau der Befehle in L<sup>A</sup>T<sub>E</sub>X

Eine Anweisung in L<sup>A</sup>T<sub>E</sub>X beginnt immer mit einem Schrägstrich nach links, dem sogenannten *Backslash*, dann folgt der Befehlsname, dem in geschweiften Klammern *Argumente* bzw. *Parameter* übergeben werden können. Vor den Argumenten können in eckigen Klammern sogenannte *Optionen* übergeben werden, damit werden Einstellungen für den Befehl vorgenommen.

`\documentclass` ist immer der erste Befehl in L<sup>A</sup>T<sub>E</sub>X. Einige der wichtigsten Optionen für `\documentclass` lauten

`a4paper` Din A4-Format, falls Anpassung nötig

`12pt` größere Schrift

`twocolumn` zweispaltige Ausgabe

## Aufbau der Befehle in L<sup>A</sup>T<sub>E</sub>X

Eine Anweisung in L<sup>A</sup>T<sub>E</sub>X beginnt immer mit einem Schrägstrich nach links, dem sogenannten *Backslash*, dann folgt der Befehlsname, dem in geschweiften Klammern *Argumente* bzw. *Parameter* übergeben werden können. Vor den Argumenten können in eckigen Klammern sogenannte *Optionen* übergeben werden, damit werden Einstellungen für den Befehl vorgenommen.

`\documentclass` ist immer der erste Befehl in L<sup>A</sup>T<sub>E</sub>X. Einige der wichtigsten Optionen für `\documentclass` lauten

`a4paper` Din A4-Format, falls Anpassung nötig

`12pt` größere Schrift

`twocolumn` zweispaltige Ausgabe

Mehrere Optionen können, mit Kommata getrennt, gleichzeitig genutzt werden, also etwa

```
\documentclass[12pt,twocolumn]{article}
```

Texte werden direkt in den Hauptteil innerhalb der `document`-Umgebung eingegeben,  $\LaTeX$  kümmert sich dann automatisch um die Verteilung der Wörter pro Zeile, den Blocksatz, die Ausrichtung von Abständen, Trennung, Seitentrennung usw.

Texte werden direkt in den Hauptteil innerhalb der `document`-Umgebung eingegeben,  $\text{\LaTeX}$  kümmert sich dann automatisch um die Verteilung der Wörter pro Zeile, den Blocksatz, die Ausrichtung von Abständen, Trennung, Seitentrennung usw.

Eine Leerzeile in der `.tex`-Datei bedeutet, daß  $\text{\LaTeX}$  an dieser Stelle einen neuen Absatz beginnen soll. Abstände zwischen Absätzen, Leerzeilen usw. muß man mit Befehlen eingeben.

Texte werden direkt in den Hauptteil innerhalb der document-Umgebung eingegeben,  $\LaTeX$  kümmert sich dann automatisch um die Verteilung der Wörter pro Zeile, den Blocksatz, die Ausrichtung von Abständen, Trennung, Seitentrennung usw.

Eine Leerzeile in der .tex-Datei bedeutet, daß  $\LaTeX$  an dieser Stelle einen neuen Absatz beginnen soll. Abstände zwischen Absätzen, Leerzeilen usw. muß man mit Befehlen eingeben.

Leerzeichen und Zeilenenden werden in der .tex-Datei wie Worttrenner behandelt. Leerzeichen können aber mit „\ “ oder ~ erzwungen werden, falls nötig. Beispiel: die Eingabe von  $\backslash\text{LaTeX kann's}$  ergibt  $\LaTeX\text{kann's}$ . Gibt man  $\backslash\text{LaTeX}\backslash\text{kann's}$  ein, ergibt dies  $\LaTeX\text{ kann's}$ .

## Befehlszeichen

Die folgenden zehn Zeichen haben in  $\text{\LaTeX}$  eine besondere Bedeutung und können nicht als Zeichen im Text verwendet werden. Mit dem *backslash* kann man Sie aber bequem durch die darunterstehenden  $\text{\LaTeX}$ -Befehle erhalten:

Zeichen: { } \$ & # % \_ ^ ~

in  $\text{\LaTeX}$ : \{ \} \\$ \& \# \% \\_ \^{} \~{}

\ geht mit `\textbackslash` (mit Paket *textcomp*).

## Befehlszeichen

Die folgenden zehn Zeichen haben in  $\text{\LaTeX}$  eine besondere Bedeutung und können nicht als Zeichen im Text verwendet werden. Mit dem *backslash* kann man Sie aber bequem durch die darunterstehenden  $\text{\LaTeX}$ -Befehle erhalten:

Zeichen: { } \$ & # % \_ ^ ~

in  $\text{\LaTeX}$ : \{ \} \\$ \& \# \% \\_ \^{} \~{}  
\ geht mit `\textbackslash` (mit Paket *textcomp*).

Umlaute können problemlos in den Text eingefügt werden, sofern die Präambel die folgenden Befehle enthält:

```
\usepackage{ngerman}  
\usepackage[utf8]{inputenc}  
\usepackage[T1]{fontenc}
```

Man stellt so sicher, daß  $\text{\LaTeX}$  nach deutschen Trennregeln trennt, Umlaute richtig umsetzt usw.

# Befehle ohne Parameter

Befehle ohne Parameter haben die Form `\Befehl`

# Befehle ohne Parameter

Befehle ohne Parameter haben die Form `\Befehl`

Z.B. `\newline`

# Befehle ohne Parameter

Befehle ohne Parameter haben die Form `\Befehl`

Z. B. `\newline`

Die Eingabe

```
Text und \newline noch viel mehr Text
```

# Befehle ohne Parameter

Befehle ohne Parameter haben die Form `\Befehl`

Z.B. `\newline`

Die Eingabe

```
Text und \newline noch viel mehr Text
```

ergibt die Ausgabe

```
Text und  
noch viel mehr Text
```

# Befehle ohne Parameter

Befehle ohne Parameter haben die Form `\Befehl`

Z.B. `\newline`

Die Eingabe

```
Text und \newline noch viel mehr Text
```

ergibt die Ausgabe

```
Text und  
noch viel mehr Text
```

`\\` ist eine bequeme Abkürzung für `\newline`

## Befehle mit Parametern

Im allgemeinen haben Befehle ein oder mehrere *Parameter*; optionale Parameter stehen in eckigen Klammern, notwendige Parameter in geschweiften Klammern. Bei einem optionalen und zwei notwendigen Parameters sieht ein allgemeiner Befehl so aus:

```
\Befehl [optionalerParameter] {Parameter1}{Parameter2}
```

## Befehle mit Parametern

Im allgemeinen haben Befehle ein oder mehrere *Parameter*; optionale Parameter stehen in eckigen Klammern, notwendige Parameter in geschweiften Klammern. Bei einem optionalen und zwei notwendigen Parameters sieht ein allgemeiner Befehl so aus:

```
\Befehl [optionalerParameter] {Parameter1} {Parameter2}
```

Zum Beispiel hat der Befehl `\textit` einen notwendigen Parameter, den er in kursiv setzt. Die Eingabe

```
Text mit einem \textit{kursiven} Wort
```

## Befehle mit Parametern

Im allgemeinen haben Befehle ein oder mehrere *Parameter*; optionale Parameter stehen in eckigen Klammern, notwendige Parameter in geschweiften Klammern. Bei einem optionalen und zwei notwendigen Parameters sieht ein allgemeiner Befehl so aus:

```
\Befehl [optionalerParameter] {Parameter1} {Parameter2}
```

Zum Beispiel hat der Befehl `\textit` einen notwendigen Parameter, den er in kursiv setzt. Die Eingabe

```
Text mit einem \textit{kursiven} Wort
```

ergibt die Ausgabe

```
Text mit einem kursiven Wort
```

# Umgebungen

Umgebungen haben die Form

```
\begin{Umgebungsname}
```

```
\end{Umgebungsname}
```

# Umgebungen

Umgebungen haben die Form

```
\begin{Umgebungsname}
```

```
\end{Umgebungsname}
```

Z. B. setzt die `itshape`-Umgebung allen dazwischenstehenden Text kursiv. Die Eingabe

```
Text  
\begin{itshape}  
  und noch mehr  
\end{itshape}  
Text
```

# Umgebungen

Umgebungen haben die Form

```
\begin{Umgebungsname}
```

```
\end{Umgebungsname}
```

Z. B. setzt die `itshape`-Umgebung allen dazwischenstehenden Text kursiv. Die Eingabe

```
Text  
\begin{itshape}  
  und noch mehr  
\end{itshape}  
Text
```

ergibt die Ausgabe

```
Text und noch mehr Text
```

# Klassen und Pakete

Die verschiedenen Klassen, mit der ein Dokument mittels `\documentclass` spezifiziert werden kann, lauten

`article` für mittelgroße und kleine Dokumente

`report` für umfangreiche Dokumente

`book` für komplette Bücher

# Klassen und Pakete

Die verschiedenen Klassen, mit der ein Dokument mittels `\documentclass` spezifiziert werden kann, lauten

`article` für mittelgroße und kleine Dokumente

`report` für umfangreiche Dokumente

`book` für komplette Bücher

Für verschiedene Anwendungen steht eine große Auswahl an Paketen zur Verfügung, die mit `\usepackage` und eventuellen Optionen in die Präambel eingefügt werden können.

Diese stellen dann weitere Kommandos zur Verfügung. Beispiele:

# Klassen und Pakete

Die verschiedenen Klassen, mit der ein Dokument mittels `\documentclass` spezifiziert werden kann, lauten

`article` für mittelgroße und kleine Dokumente

`report` für umfangreiche Dokumente

`book` für komplette Bücher

Für verschiedene Anwendungen steht eine große Auswahl an Paketen zur Verfügung, die mit `\usepackage` und eventuellen Optionen in die Präambel eingefügt werden können.

Diese stellen dann weitere Kommandos zur Verfügung. Beispiele:

`(n)german` für deutsche Texte

`graphicx` für Bildereinbindung

`mutlicol` für mehrspaltigen Text

`longtable` für mehrseitige Tabellen

`amsmath`, `amsthm`, `amssymb` umfangreiche Mathematik-Pakete

## Klassen und Pakete

Die verschiedenen Klassen, mit der ein Dokument mittels `\documentclass` spezifiziert werden kann, lauten

`article` für mittelgroße und kleine Dokumente

`report` für umfangreiche Dokumente

`book` für komplette Bücher

Für verschiedene Anwendungen steht eine große Auswahl an Paketen zur Verfügung, die mit `\usepackage` und eventuellen Optionen in die Präambel eingefügt werden können.

Diese stellen dann weitere Kommandos zur Verfügung. Beispiele:

`(n)german` für deutsche Texte

`graphicx` für Bildereinbindung

`mutlicol` für mehrspaltigen Text

`longtable` für mehrseitige Tabellen

`amsmath`, `amsthm`, `amssymb` umfangreiche Mathematik-Pakete

Mit `\usepackage` können auch mehrere Pakete gleichzeitig eingeladen werden, z. B. `\usepackage{ngerman, amsmath}`

Allgemeine Einführung: Was ist  $\text{\LaTeX}$ ?

Allgemeine Grundlagen: Aufbau eines  $\text{\LaTeX}$ -Dokuments

Benutzung von  $\text{\LaTeX}$  unter Linux, der Editor emacs

## emacs

Emacs ist ein sehr umfangreiches Editor-Programm unter Linux, das zum Tippen von  $\text{\LaTeX}$ -Dateien benutzt werden kann und zahlreiche Möglichkeiten bietet. (Eventuell den X-Symbol-Mode ausschalten!)

## emacs

Emacs ist ein sehr umfangreiches Editor-Programm unter Linux, das zum Tippen von  $\text{\LaTeX}$ -Dateien benutzt werden kann und zahlreiche Möglichkeiten bietet. (Eventuell den X-Symbol-Mode ausschalten!)

Schön: Syntax-Highlighting, einfaches Kompilieren von  $\text{\LaTeX}$ -Dateien, etc. Und wenn man sich versehentlich in einem unbekanntem Modus verirrt hat: **C-g** drücken!

## emacs

Emacs ist ein sehr umfangreiches Editor-Programm unter Linux, das zum Tippen von  $\text{\LaTeX}$ -Dateien benutzt werden kann und zahlreiche Möglichkeiten bietet. (Eventuell den X-Symbol-Mode ausschalten!)

Schön: Syntax-Highlighting, einfaches Kompilieren von  $\text{\LaTeX}$ -Dateien, etc. Und wenn man sich versehentlich in einem unbekanntem Modus verirrt hat: **C-g** drücken!

Die wichtigsten Emacs-Befehle über die Tastatur: (C steht für die Control-, und M für die Meta-Taste, das ist die Alt-Taste. Blättern geht mit den Bild-hoch- und runter-Pfeiltasten.)

**C-x C-c** Emacs verlassen

**C-x C-f** *Datei Return* Datei laden

**C-x C-s** Datei speichern

**C-x C-i** Datei einfügen

## Schreiben einer $\text{\LaTeX}$ -Datei

Emacs starten: Im xterm eingeben: `emacs beispiel.tex &`

# Schreiben einer L<sup>A</sup>T<sub>E</sub>X-Datei

Emacs starten: Im xterm eingeben: `emacs beispiel.tex &`

oder: `emacs &`, dann im Emacs `C-x C-f beispiel.tex` *Return*  
eingeben

# Schreiben einer L<sup>A</sup>T<sub>E</sub>X-Datei

Emacs starten: Im xterm eingeben: `emacs beispiel.tex &`

oder: `emacs &`, dann im Emacs `C-x C-f beispiel.tex` *Return* eingeben

Dann wird die `.tex`-Datei im Emacs getippt und abgespeichert mit `C-x C-s`

# Schreiben einer $\text{\LaTeX}$ -Datei

Emacs starten: Im xterm eingeben: `emacs beispiel.tex &`

oder: `emacs &`, dann im Emacs `C-x C-f beispiel.tex` *Return* eingeben

Dann wird die `.tex`-Datei im Emacs getippt und abgespeichert mit `C-x C-s`

Die `.tex`-Datei ist fertig zum Kompilieren, d. h. fertig, um  $\text{\LaTeX}$  anzuwenden. Dafür gibt es folgende Methoden:

# Kompilieren einer $\text{\LaTeX}$ -Datei

1. im xterm eingeben: `latex beispiel.tex`, dann Ansicht mit `xdvi beispiel.dvi &`

## Kompilieren einer $\text{\LaTeX}$ -Datei

1. im xterm eingeben: `latex beispiel.tex`, dann Ansicht mit `xdvi beispiel.dvi &`
2. im xterm eingeben: `pdflatex beispiel.tex`, dann Ansicht mit `xpdf beispiel.pdf &`

# Kompilieren einer $\text{\LaTeX}$ -Datei

1. im xterm eingeben: `latex beispiel.tex`, dann Ansicht mit `xdvi beispiel.dvi &`
2. im xterm eingeben: `pdflatex beispiel.tex`, dann Ansicht mit `xpdf beispiel.pdf &`
3. im Emacs eingeben: *C-c C-c Return*, dann Ansicht ebenso mit *C-c C-c Return*

## Kompilieren einer $\text{\LaTeX}$ -Datei

1. im xterm eingeben: `latex beispiel.tex`, dann Ansicht mit `xdvi beispiel.dvi &`
2. im xterm eingeben: `pdflatex beispiel.tex`, dann Ansicht mit `xpdf beispiel.pdf &`
3. im Emacs eingeben: *C-c C-c Return*, dann Ansicht ebenso mit *C-c C-c Return*

$\text{\LaTeX}$ -Fehlermeldungen im xterm können durch die Eingabe der Taste `x` ignoriert werden.

# Kompilieren einer $\text{\LaTeX}$ -Datei

1. im xterm eingeben: `latex beispiel.tex`, dann Ansicht mit `xdvi beispiel.dvi &`
2. im xterm eingeben: `pdflatex beispiel.tex`, dann Ansicht mit `xpdf beispiel.pdf &`
3. im Emacs eingeben: *C-c C-c Return*, dann Ansicht ebenso mit *C-c C-c Return*

$\text{\LaTeX}$ -Fehlermeldungen im xterm können durch die Eingabe der Taste `x` ignoriert werden.

Fehlermeldungen können im Emacs durch die Eingabe von `C-c `` angezeigt werden. Das Fehlerfenster kann geschlossen werden, wenn man mit der Maus ins Hauptfenster geht und dort `C-x 1` eingibt.

## Bemerkungen zum Kompilieren

Bis die fertige  $\text{\LaTeX}$ -Datei korrekt angezeigt wird, können mehrere Kompilier-Durchgänge mit  $\text{\LaTeX}$  nötig sein.

## Bemerkungen zum Kompilieren

Bis die fertige  $\text{\LaTeX}$ -Datei korrekt angezeigt wird, können mehrere Kompilier-Durchgänge mit  $\text{\LaTeX}$  nötig sein.

Nicht immer kann das Programm `xdvi` das Ergebnis richtig anzeigen. Dann muß man die `.dvi`-Datei in ein anderes Format umwandeln.

# Bemerkungen zum Kompilieren

Bis die fertige  $\text{\LaTeX}$ -Datei korrekt angezeigt wird, können mehrere Kompilier-Durchgänge mit  $\text{\LaTeX}$  nötig sein.

Nicht immer kann das Programm `xdvi` das Ergebnis richtig anzeigen. Dann muß man die `.dvi`-Datei in ein anderes Format umwandeln.

Umwandlung vom Format `.dvi` ins Format `.ps`:  
`dvips beispiel.dvi`

## Bemerkungen zum Kompilieren

Bis die fertige  $\text{\LaTeX}$ -Datei korrekt angezeigt wird, können mehrere Kompilier-Durchgänge mit  $\text{\LaTeX}$  nötig sein.

Nicht immer kann das Programm `xdvi` das Ergebnis richtig anzeigen. Dann muß man die `.dvi`-Datei in ein anderes Format umwandeln.

Umwandlung vom Format `.dvi` ins Format `.ps`:  
`dvips beispiel.dvi`

Umwandlung vom Format `.ps` ins Format `.pdf`:  
`ps2pdf beispiel.ps`, umgekehrt mit `pdf2ps beispiel.pdf`

# Ansicht und Drucken

Ansicht von .ps-Dateien mit `gv beispiel.ps &`, von dort aus ist auch Drucken möglich.

# Ansicht und Drucken

Ansicht von .ps-Dateien mit `gv beispiel.ps &`, von dort aus ist auch Drucken möglich.

Dies geht im xterm auch mit `lpr beispiel.ps` oder `lpr beispiel.pdf`, falls eine .ps- oder .pdf-Datei erzeugt wurde.

# Ansicht und Drucken

Ansicht von .ps-Dateien mit `gv beispiel.ps &`, von dort aus ist auch Drucken möglich.

Dies geht im xterm auch mit `lpr beispiel.ps` oder `lpr beispiel.pdf`, falls eine .ps- oder .pdf-Datei erzeugt wurde.

Mit dem Programm `xpdf` können .pdf-Dateien angesehen und ausgedruckt werden.