

L^AT_EX-Beamer-Kurs

Basics Technisches

Martin Glatz
ig-mathe



1. Was ist \LaTeX
2. Technik
3. Dieser Kurs ...
4. Wie arbeitet man mit \LaTeX
5. Wo finde ich Hilfe



Leseanleitung

Programmcode:

```
abgesetzter_Programmcode  
(Leerzeichen_notfalls_entfernen)  
%Kommentare_im_Code
```

Normaler Text sowie Programmcode im Text normaler Text

Achtung! Gefahr?

Besonders Wichtiges oder Gefährliches ist so gekennzeichnet! Also eventuelle Unstimmigkeiten, Kompatibilitätsprobleme, ...

Solche Hinterlegungen dienen nur der optischen/inhaltlichen Strukturierung und haben ansonsten keine besondere Bedeutung.

Starke Gliederung

Falls notwendig und sinnvoll, gibt es auch Gliederungen mit Überschriften ...

1. Was ist \LaTeX
2. Technik
3. Dieser Kurs ...
4. Wie arbeitet man mit \LaTeX
5. Wo finde ich Hilfe



L^AT_EX-Kurs
Basics
Technisches

Inhalt

Was ist L^AT_EX

Features

Nachteile

Technik

Dieser Kurs ...

Wie arbeitet man
mit L^AT_EX

Wo finde ich Hilfe

L^AT_EX // Grundlegendes

- ⊙ ein sogenanntes Textsatzprogramm
- ⊙ pdf-Dokumente als Ergebnis (m.E.)
- ⊙ kostenlos, legal frei verfügbar und plattformübergreifend (Windows, Linux, Mac)
- ⊙ weit verbreitet in allen (Natur-)Wissenschaften
- ⊙ DAS Standardprogramm für mathematische Texte



L^AT_EX-Kurs
Basics
Technisches

Inhalt

Was ist L^AT_EX

Features
Nachteile

Technik

Dieser Kurs ...

Wie arbeitet man
mit L^AT_EX

Wo finde ich Hilfe

Anwendungen

- ⊙ kurze Dokumente (Übungszettel, Laborprotokolle, ...)
- ⊙ längere Berichte mit umfassenden Verzeichnissen (Seminar-, Bachelor-, Master- und Diplomarbeiten)
- ⊙ ganze Bücher mit klassischer Typographie (Schmutztitel etc)
- ⊙ programmierte, exakte Grafiken eingebunden erzeugbar
- ⊙ (Serien-)Briefe
- ⊙ pdf-Präsentationen (Beweisstück A: diese Präsentation hier!)



L^AT_EX // Mathematik und NAWI

- ermöglicht logischen, hochqualitativen Formelsatz mit perfekter Formatierung (Abständen)
- Formeln können ohne lästiges Dauergeklicke eingegeben werden – beliebig verschachtelt/kompliziert
- eigene, professionelle Mathematik-Schriften (fett, Blackbord, Symbole) (passend zur Textschrift) vorhanden
- Sätze, Definitionen, Beweise etc mit automatischer Nummerierung/Formatierung – bereits vorgefertigt
- automatisierte Nummerierung von Gleichungen oder Gleichungssystemen mit einfachster Verweis-Möglichkeit
- intuitive Formatierung von chemischen Summenformeln und Reaktionsgleichungen (ohne lästiges Dauertiefstellen etc)
- einfachste, automatisierte Einbindung von Programmcode direkt aus den Code-Dateien
- ...



L^AT_EX // Grafiken und Tabellen

- hochqualitative Tabellen nach typographischen Standards setzen und bearbeiten
- keine Ruckelprobleme auch bei vielen Tabellen und Grafiken
- Tabellenlayout mit wenig Aufwand durchgehend änderbar und reproduzierbar (kein Dauergeklicke für jede Zelle, falls doch eine dünnere Linie gewünscht ist)
- kann externe, hochqualitative Vektorgrafiken und pdf-Dokumente/Teile ohne Qualitätsverlust einbinden (→ ∞-scharf, unverblickelt)
- Grafiken können nachträglich noch mit der Textschrift pixelgenau beschriftet bzw. bedruckt werden
- automatische Nummerierung der Objekte inkl. Tabellen- & Abbildungsverzeichnissen mit geringstem Aufwand machbar
- eigene Grafiken sind pixelgenau programmierbar
- ...



L^AT_EX // Text und Literatur

- perfekte Strukturierung durch Kapitel etc inkl. automatischer Einträgen ins Inhaltsverzeichnis
- beim Schreiben kann man sich auf den Inhalt und die Struktur konzentrieren
- viele Sonderzeichen etc leicht verwendbar
- simpels System für Master- bzw. Hauptdokument und Tochterdokumente
- einfaches System für Fußnoten (inkl. Fußmarken, Formatierung)
- Verwaltung und Zitate auf verwendete Literatur
- beliebige Verweise innerhalb des Dokuments (auch auf Formeln, Bilder, Tabellen etc) mit automatischer Aktualisierung
- verschiedene Verzeichnisse sowie Index/Register ohne Gefummel machbar
- minimale Dateigrößen bei den Codes
- automatische lebende Kolumnentitel
- ...



IATeX-Kurs
Basics
Technisches

Inhalt

Was ist IATeX

Features

Nachteile

Technik

Dieser Kurs ...

Wie arbeitet man
mit IATeX

Wo finde ich Hilfe

Typographie und Layout

- hervorragende Schrifteinbindung: Ligaturen, Kapitälchen, Unterschneidungen, Sperrungen ...
- wunderschöner Blocksatz mit automatischer Worttrennung (optischer Randausgleich usw)
- professionell abgestimmte Schriftgrößen und Abstände
- automatische Auszeichnungen von Überschriften auf verschiedenen Ebenen (inkl. Nummerierung, falls gewünscht)
- spezielle Einstellungen für ein- bzw. zweiseitigen Satz möglich
- Einstellungen der Seitenränder nach klassischen typographischen Regeln sehr einfach möglich
- millimetergenaue, automatische Formatierungen bei (mehrfach) verschachtelten Aufzählungen
- ...



Automatisierung und Programmierbarkeit

- ⊙ einfachstes Wiederverwenden von Vorlagen bzw. Einstellungen
- ⊙ hohe Flexibilität bei nachträglichen Änderungswünschen
- ⊙ Variablen und Zähler können selbst gemacht werden
- ⊙ eigene Befehle und Umgebungen können erzeugt werden
- ⊙ oft reicht ein Befehl (z. B. für Änderung der Aufzählungssymbole aller Listen)
- ⊙ ...



Nachteile von L^AT_EX

- ⊙ spezielle (Layout-)Wünsche z. T. nur aufwendig/ mit Erfahrung zu erreichen (L^AT_EX ist kein Layout-Programm wie indesign)
- ⊙ bei der (Farb-)Tabellen-Gestaltung nicht alles (einfach) machbar
- ⊙ nicht alle Schriftarten frei verfügbar
- ⊙ manche Funktionen benötigen eigene (frei verfügbare) Pakete
- ⊙ es gibt oft nicht die »einzig richtige« Möglichkeit
- ⊙ Einarbeitungszeit & Gewöhnungsphase

1. Was ist \LaTeX

2. Technik

Kernel-Grundsystem

Editoren

3. Dieser Kurs ...

4. Wie arbeitet man mit \LaTeX

5. Wo finde ich Hilfe



T_EXnischer Hintergrund // L^AT_EX ...

- ⊙ ist grundsätzlich kein WYSIWYG-Programm (wie Word oder OpenOffice)
- ⊙ arbeitet intern mit einer »Programmiersprache«
- ⊙ benötigt Befehle, um Formatierungen etc vorzunehmen
- ⊙ bietet Pakete und Vorlagen für verschiedenste Anwendungen und Themengebiete (z. B. Chemische Summenformeln, Mathematische Texte, Präsentationen, ...)
- ⊙ benötigt einen Editor (Entwicklungsumgebung) zur Bearbeitung der Quellcode-Datei
- ⊙ erzeugt grundsätzlich (pdf-)Dokumente aus einer Quellcode-Datei (.tex) (Kompillieren/Übersetzen)



Kernel-Grundsystem

Die Übersetzer-Software

TeX Live

für Windows und Linux

- ⊙ <http://www.tug.org/texlive/>
- ⊙ Installer-Download unter <http://mirror.ctan.org/systems/texlive/tlnet/install-tl.zip>
- ⊙ Installationsanleitungen auf der Webseite verfügbar
- ⊙ Ubuntu: Download über SoftwareCenter

MacTeX

für Mac OS X

- ⊙ die TeXLive-Distribution speziell für Mac
- ⊙ <http://www.tug.org/mactex/>
- ⊙ MacTeX.mpkg.zip als Grund-Software
- ⊙ MacTeXtras.zip für weitere Editoren etc



IATeX-Kurs
Basics
Technisches

Inhalt

Was ist IATeX

Technik

Kernel-Grundsystem

Editoren

Dieser Kurs ...

Wie arbeitet man
mit IATeX

Wo finde ich Hilfe

(Spezielle) Editoren I

texmaker

<http://www.xmlmath.net/texmaker/index.html>

- ⊙ plattformübergreifend (Windows, Mac, Linux)
- ⊙ unser Standard-Editor

Kile

<http://kile.sourceforge.net/>

- ⊙ Linux (und mittlerweile auch für Windows)
- ⊙ Autovervollständigung und viele Shortcuts
- ⊙ ideal für größere Projekte (Dokumente mit mehreren Quelldateien)
- ⊙ Download auch über das Ubuntu-Softwarecenter



(Spezielle) Editoren II

LyX

<http://www.lyx.org/WebDe.Home>

- ⊙ Windows, Mac, Linux
- ⊙ bietet (meist) grafische-Oberfläche
- ⊙ L^AT_EX-Befehle werden meist simultan übersetzt
- ⊙ vergleichsweise geringe Umstellung von Word etc
- ⊙ hat »eigenes« Quell-Dateiformat
- ⊙ viele L^AT_EX-Funktionen nur mühsam nutzbar

TeXnicCenter

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

- ⊙ nur für Windows

1. Was ist \LaTeX

2. Technik

3. Dieser Kurs ...

Ziele

Ja

Nein

4. Wie arbeitet man mit \LaTeX

5. Wo finde ich Hilfe



IATeX-Kurs
Basics
Technisches

Inhalt

Was ist IATeX

Technik

Dieser Kurs ...

Ziele

Ja

Nein

Wie arbeitet man
mit IATeX

Wo finde ich Hilfe

Ziele dieses Kurses

- ⊙ ein grundlegendes Verständnis der Thematik/Problematik rund um IATeX, Typographie, Textsatz und Naturwissenschaften erzeugen
- ⊙ den Nutzen von IATeX klarmachen
- ⊙ selbstständiges Arbeiten ermöglichen
- ⊙ zum selbstständigen Recherchieren befähigen
- ⊙ Vorlagen und Code-Schnippsel haben
- ⊙ viele Themengebiete mit passenden Pakete kennenlernen



IATeX-Kurs
Basics
Technisches

Inhalt

Was ist IATeX

Technik

Dieser Kurs ...

Ziele

Ja

Nein

Wie arbeitet man
mit IATeX

Wo finde ich Hilfe

Inhalt des restlichen Kurses

- 1 Einführung (jetzt)
- 2 Arbeiten mit IATeX
- 3 Standard-Dokumente erzeugen (inkl Basic-Befehle)
- 4 Grafiken einbinden und Tabellen erzeugen
- 5 Mathematik und Naturwissenschaften
- 6 Gliederung, Verzeichnisse, Verweise in längeren Texten
- 7 Design: Layout, Typographie, Umgestaltungen
- 8 pdf-Folien für Präsentationen erstellen und gestalten



Was geboten wird

Alles, was man für Handouts, Laborprotokolle, Seminararbeiten (auch mathematisch-physikalisch) grundsätzlich braucht!

- ⊙ Einführung ins Dokumente-Erstellen von 0 weg
- ⊙ Umgang mit KOMA-Klassen sowie der beamer-Klasse
- ⊙ komprimierte Übersicht mit unverzichtbaren/sinnvollen Paketen zu den jeweiligen Themengebieten
- ⊙ viele Code-Beispiele (die 1:1 verwendet werden können)
- ⊙ viele Befehle und Optionen, die das Aussehen verändern (Seitenlayout, Kopf-/Fußzeile, Schriften, Überschriften, Listen, Präsentationsfolien)



Was nicht geboten wird/ werden kann

Nicht gewollt

- Der Kurs ist keine theoretische Schulung, wie welche Programme im Hintergrund arbeiten (für die durchschnittliche Anwendung völlig uninteressant)
- T_EX-Grundlagen und umfassende Programmier-Befehle werden nicht behandelt
- Postscript bzw. DVI werden nicht thematisiert
- andere Klassen als KOMA/beamer werden nur erwähnt

Nicht möglich

- Der Kurs ersetzt nicht das selbstständige/selbsttätige Arbeiten (Übung macht den Meister)
- Das Nachlesen/Suchen zu spezielleren Themen ersparen
- Nicht alle Layout-Sonderwünsche werden abgedeckt

1. Was ist \LaTeX

2. Technik

3. Dieser Kurs ...

4. Wie arbeitet man mit \LaTeX

Programmiersprache

Dokumentklasse

Pakete

Befehle

Umgebungen

Neu/Umdefinieren

Sonderzeichen

Fehler

Workflow

Fragestellungen

5. Wo finde ich Hilfe



Ein erster Beispielcode

```
%Praeambel_(Vorspann) :  
\documentclass{scrartcl}
```

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

```
%eigentliches_Dokument-Inhalt  
\begin{document}
```

```
\section{Einleitung}
```

```
Hier_kommt_alles,_\textbf{was}_gesetzt_werden  
soll_...
```

```
\begin{center}  
..._zentrierter_Text_...  
\end{center}
```

```
\end{document}
```



Dokumentklasse

Dokumentklasse laden

```
\documentclass[optionale_Einstell.]{Klassenname}
```

```
\documentclass[10pt,a4paper]{scrartcl}
```

- ⦿ muss die erste nicht kommentierte Textzeile sein
- ⦿ genau eine Dokumentklasse ist nötig
- ⦿ bestimmt das grobe Aussehen/Funktion des Endprodukts
- ⦿ Klassen für kurze Artikel, Bücher, pdf-Präsentationen etc
- ⦿ optionale Einstellungen: häufig für Grundschriftgröße, ein- oder zweiseitiges Layout, ...
- ⦿ liefern z. T. auch eigene/neue Befehle



Pakete

Pakete laden:

```
\usepackage[optionale_Einstellungen]{Paketname}
```

```
\usepackage[latin9]{inputenc}
```

- ⊙ müssen nach `\documentclass[...]{...}` kommen
- ⊙ müssen vor `\begin{document}` stehen
- ⊙ stellen grundlegende Funktionalitäten bereit: Grafiken einbinden, Farbe ermöglichen, ...
- ⊙ stellen Befehle bereit: neue Listen, neue mathematische Symbole, neue Befehle für Layout-Änderungen
- ⊙ haben häufig Optionen, mit denen die Funktionalitäten spezifiziert werden können
- ⊙ z. T. Kompatibilitätsprobleme, z. T. Reihenfolge wichtig
- ⊙ öfters gibt es mehrere Pakete für die selbe Funktion (Recherchieren, welches Standard ist bzw. welche Funktionen wirklich gebraucht werden)
- ⊙ z. T. sind Pakete überholt (obsolet)



Befehle (commands)

Grundsyntax:

```
\befehl  
\befehl{verpflichtender_Parameter}  
\befehl[opt._Parameter]{verpfl._Parameter}  
\befehl[opt._Parameter]{vParameter1}{vParameter2}{vParameter3}  
...
```

Beispiele

- ⊙ `\delta` liefert δ
- ⊙ `\textbf{fett}` für **fett**
- ⊙ `\sqrt{2}` bzw. `sqrt[3]{2}` liefert $\sqrt{2}$ bzw. $\sqrt[3]{2}$
- ⊙ werden immer mit Backslash eingeleitet
- ⊙ können optionale bzw. verpflichtende Parameter haben
- ⊙ Reihenfolge der Parameter beachten
- ⊙ ermöglichen Gliederungen, Formatierungen, Sonderzeichen, ...
- ⊙ können verschachtelt sein
- ⊙ eigene Befehle können definiert werden



Umgebungen (environments)

```
\begin{Umgebungsname}  
Inhalt_...  
\end{Umgebungsname}
```

Beispiel:

```
\begin{huge}  
Dieser_Text_wird_riesig_geschrieben_...  
\end{huge}
```

Dieser Text wird riesig geschrieben ...

- ⦿ es muss ein `\begin{...}` und ein `\end{...}` geben (sonst Fehlermeldung)
- ⦿ zuständig für Formatierungen, Grafiken, Tabellen, Matrizen ...
- ⦿ können verschachtelt sein
- ⦿ stellen innerhalb z. T. neue Befehle bereit



Umgebungen (environments) (Fortsetzung)

Der Code (die Einrückungen dienen nur der Übersichtlichkeit:

```
\begin{itemize}
  \item erster Punkt
  \begin{itemize}
    \item erster Unterpunkt
    \item zweiter Unterpunkt
  \end{itemize}
  \item zweiter Punkt
  \item dritter Punkt
\end{itemize}
```

... und das Ergebnis:

- erster Punkt
 - erster Unterpunkt
 - zweiter Unterpunkt
- zweiter Punkt
- dritter Punkt



Eigene Befehle definieren

```
\newcommand{\Name}{Definition}  
\newcommand{\Name}[Anz]{Definition}  
\newcommand{\Name}[Anz][default]{Definition}
```

- ⊙ \Name ist der Name des Befehls
- ⊙ Definition ist Text /. Code und sagt L^AT_EX, was es machen soll
- ⊙ Anz ist die Anzahl der Argumente, die der Befehl haben soll
- ⊙ die Argumente können mit #1 bis maximal #9 in der Definition aufgerufen werden
- ⊙ ist ein optionales Argument gewünscht:
 - default (Text, Befehl, etc) ist der Standardwert
 - #1 wird zum optionalen Argument

Beispiel 1:

```
%Definition:  
\newcommand{\igm}{InteressengemeinschaftuMathematik}  
  
%Verwendung:  
...uu\igm{}u...u\igm\u...uu\igmu...
```



Eigene Befehle definieren (Fortsetzung)

Beispiel 2

```
%Definition:  
\newcommand{\fettust}[1]{\textbf{\underline{#1}}}  
  
... \fettust{fett_und_untersrichten} ...
```

Beispiel 3

```
%Definition:  
\newcommand{\tabzeile}[3]{#1_&#2_&#3_\\_}  
  
%Verwendung  
\tabzeile{x}{y}{z}
```

Beispiel 4

```
%Definition:  
\newcommand{\abstand}[1][2cm]{\hspace*{#1}}
```

x	\abstand	y	liefert	x	y
x	\abstand[1cm]	y	liefert	x	y

Inhalt

Was ist L^AT_EX

Technik

Dieser Kurs ...

Wie arbeitet man
mit L^AT_EX

Programmiersprache

Dokumentklasse

Pakete

Befehle

Umgebungen

Neu/Umdefinieren

Sonderzeichen

Fehler

Workflow

Fragestellungen

Wo finde ich Hilfe



Eigene Umgebungen definieren

```
\newenvironment{Name} [Anz] {Beginn}{Ende}  
\newenvironment{Name} [Anz] [default] {Beginn}{Ende}  
%Aufruf:  
\begin{Name}_\dots_\end{Name}
```

- ⦿ Name ist der Name der Umgebung (ohne Backslash)
- ⦿ Beginn sind die Befehle/Text, die am Umgebungsbeginn ausgeführt werden
- ⦿ Ende sind die Befehle/Text, die am Umgebungsende ausgeführt werden
- ⦿ Anz ist die Anzahl der Argumente
- ⦿ default siehe `newcommand` ...



Eigene Umgebungen definieren (Fortsetzung)

Beispiel

```
\newenvironment{auszeichnung}_\%Name  
{\bfseries\itshape}_\%Beginn  
{\mdseries\upshape}_\%Ende  
  
%Aufruf:  
\begin{auszeichnung}  
..._automatisch_fetter,_kursiver_Text_...  
\end{auszeichnung}
```

Beispiel

```
\newenvironment{liste}_\%Name  
{\begin{itemize}}_\%Beginn  
{\end{itemize}}_\%Ende  
  
%Aufruf:  
\begin{liste}  
\item_xxx  
\item_yyy  
\end{liste}
```



Savebox-Verwendung

Inhalte »speichern« und »drucken«

```
\newsavebox{\speicherbox}
```

```
\newenvironment{Name}
```

```
{  
\begin{lrbox}{\speicherbox}  
\begin{minipage}{\textwidth}  
}
```

```
{  
\end{minipage}  
\end{lrbox}  
%  
\hspace*{-\fboxsep}  
\colorbox{farbe}{\usebox{\speicherbox}}  
}
```

- ⦿ alles zwischen `\begin{lrbox}{\speicherbox}` und `\end{lrbox}` wird in `\speicherbox` gespeichert
- ⦿ mit `\usebox{\speicherbox}` wird der Inhalt ausgegeben



Eigene Pakete erstellen

Für jedes selbst erstellte Paket muss ein Tex-File mit dem Code als sty-File (style-File) gespeichert werden

Inhalt des selbst erstellten Pakets

- ⊙ Laden von vorhandenen Paketen (+ individuellen Einstellungen)
- ⊙ Definieren von eigenen Befehlen
- ⊙ Definieren von eigenen Umgebungen
- ⊙ kein Text, kein `\begin{document} ... \end{document}`
- ⊙ es können auch Optionen programmiert werden

```
%Beginn_des_Files_paketname.sty
```

```
\ProvidesPackage{paketname}
```

```
\usepackage{...}
```

```
...
```

```
\newcommand{\eigenerbefehl}[1]{...}
```

```
...
```

```
\newenvironment{eigeneumgebung}{...}{...}
```



Speicherort von eigenen Paketen I

Für generelle Verwendung des Pakets

- im LaTeX-Strukturbaum (wo alle Pakete gespeichert sind):
 - Windows XP `C:\Documents and Settings\\texmf\tex\latex\local\` oder so ähnlich
 - Windows Vista/7
`C:\Users\\texmf\tex\latex\local\` oder so ähnlich
 - Linux `~/texmf/tex/latex/local/`
 - Mac OS X
`/Users/<user name>/Library/texmf/tex/latex/local/`
oder `usr/library/tex/local/latex/`
- sinnvoll: pro Paket einen Unterordner machen (Auch Grafiken etc. können darin gespeichert werden.)
- Man muss evtl. den Verzeichnisbaum aktualisieren:
 - Windows: Konsole: `texhash`
 - Mac: Terminal (siehe Dienstprogramme): Befehle: `sudo texhash`



Speicherort von eigenen Paketen II

Für einmalige Verwendung des Pakets

- ⦿ in den Ordner kopieren, in dem die .tex-Datei ist, in die man das eigene Paket einbindet
- ⦿ Vorteil: keine »Installation« nötig
- ⦿ Nachteil: das .sty-File muss in den Ordner jeder tex-Datei kopiert werden, in der man darauf zugreifen möchte



Sinnvoller Einsatz von eigenen Paketen

vorteilhaft, wenn

- man sich viele eigene Befehle oder Umgebungen gemacht hat, die man immer benutzen möchte
- sich eine gutes Layout (z. B. mit Kopf- und Fußzeilen) gemacht hat, das man regelmäßig verwenden möchte
- man sowieso immer die gleichen Pakete lädt, da man das Laden gleich in ein eigenes .sty-File auslagern kann
- man sich schon etwas mit IATeX auskennt, da man dadurch die Vorteile der Programmierbarkeit (Variablen etc) ausnützen kann

nicht günstig, wenn

- noch ständig etwas an seinem Layout ändert
- man eigene Befehlen immer wieder ändert
- noch ganz wenig Erfahrung hat, welche Pakete und welche Einstellungen nützlich sind
- man alte Dokumente wieder verwenden möchte, obwohl man ständig die Befehle im .sty-File verändert



Verwendung von Variablen

Vorgehensweise

- 1 erstellen (z. B. in eigenem Paket)
- 2 definieren/Werte zuweisen (z. B. im .tex-Dokument)
- 3 aufrufen/Werte abrufen (z. B. im .tex-Dokument oder eigenem Paket)

- 1 erstellen im .sty-File:

```
\newcommand*{\lvname}[1]{\def\@lvname{#1}}
```

- 2 Wert zuweisen im .tex-File:

```
\lvname{Analysis_1_V0}
```

- 3 Wert abrufen im .tex-File:

```
\title{Skriptum_zur_\@lvname}
```

Wert abrufen im .sty-File:

```
\fancyhead[L]{Mitschrift_zur_LV_\@lvname}
```

IATeX-Kurs
Basics
Technisches

Inhalt

Was ist IATeX

Technik

Dieser Kurs ...

Wie arbeitet man
mit IATeX

Programmiersprache

Dokumentklasse

Pakete

Befehle

Umgebungen

Neu/Umddefinieren

Sonderzeichen

Fehler

Workflow

Fragestellungen

Wo finde ich Hilfe



Zeichen mit besonderer Bedeutung

- ⊙ geschweifte Klammern:
 - { } als Programmierstrukturierung bzw. bei Befehlen nötig
 - für { } im Text: \{ bzw. \}
- ⊙ Prozentzeichen:
 - % zum Kommentieren: nachfolgender Text in der Zeile wird nicht als Befehl/Text aufgefasst. Nützlich, um komplizierteren Code nachvollziehbar zu halten.
 - % im Text: \%
- ⊙ Kaufmännisches Und:
 - & Als Tabulatorzeichen: Trennt z. B. Einträge in einer Tabelle oder einer Matrix.
 - & im Text: \&
- ⊙ eckige Klammern:
 - für optionale Parameter bei Befehlen/Umgebungen.
 - Im Text einfach so verwendbar
- ⊙ (doppelter) Backslash:
 - \\ erzeugt Zeilenumbruch.
 - Im Text: \verb+\\+ (Schreibmaschinenschrift!) oder \textbackslash für \



Typische Fehlermeldungen

Wenn das Kompilieren nicht funktioniert ...

- ⊙ `Too many }s ... etc`
Irgendwo fehlt eine Klammer oder ist eine zuviel
- ⊙ `missing \end{itemize} etc`
Umgebungen müssen immer ein `\begin{...}` und ein `\end{...}` haben
- ⊙ `unkown command ...`
wahrscheinlich ein Tippfehler bei einem Befehl
- ⊙ `not allowed in textmode etc.`
spezielle Befehle (z. B. `^` oder `_`) dürfen nur im Mathematik-Modus benutzt werden



Workflow // Arbeitsvorgehen I

Technische Ebene:

- 1 Ordner (+ Unterordner) erstellen
- 2 Dateien erstellen (.tex Files)
- 3 Dokumentklasse wählen, (Standard)pakete laden

logische Ebene

- 1 Dokumentstruktur überlegen
- 2 grobe logische Gliederung machen
- 3 notwendige Konzepte überlegen
- 4 Platzhalter-Befehle definieren (z. B. liste)
- 5 weites Vorausdenken macht sich bezahlt



Workflow // Arbeitsvorgehen II

Textebene

- 1 Gliederungsbefehle verwenden
- 2 Text/Formeln schreiben
- 3 Befehle verwenden
- 4 regelmäßig übersetzen (→ Fehlersuche)

Layout-Ebene und Feinschliff

- 1 Grundlayout (Kopf-Fußzeile etc) gestalten
- 2 Platzhalterbefehle anpassen, falls nötig
- 3 Sonderformatierung machen (wenn notwendig)
- 4 Zeilen-Seitenumbrüche anpassen
- 5 regelmäßig übersetzen (→ Fehlersuche)



Interessante Fragestellungen

gute Fragen:

- ⊙ Gibt es Pakete, die meine Layout-Wünsche oder thematischen Ansprüche erfüllen?
- ⊙ Wo finde ich die Dokumentationen zu diesen Paketen?
- ⊙ Kann ich mir selbst eigene Befehle programmieren, die mir das Arbeiten erleichtern?

unerwünschte Fragen:

- ⊙ Wie kombiniere ich verschiedene Schriftarten wie Arial und Verdana?
- ⊙ Wie erzeuge ich Wordarts?
- ⊙ Wie verändere ich nur die Schriftgröße **einer** Gliederungsebene?
- ⊙ Wie verändere ich den Abstand zwischen zwei speziellen Absätzen?

1. Was ist \LaTeX

2. Technik

3. Dieser Kurs ...

4. Wie arbeitet man mit \LaTeX

5. Wo finde ich Hilfe



Webseiten für L^AT_EX

Webseiten:

- ⊙ <http://de.wikipedia.org/wiki/LaTeX> L^AT_EX auf Wikipedia
- ⊙ <http://latex.tugraz.at/> L^AT_EX-Seite der TU Graz
- ⊙ <http://en.wikibooks.org/wiki/LaTeX> L^AT_EX-Wikibook
- ⊙ <http://www.weinelt.de/latex/> kleine (alte) Befehlsübersicht
- ⊙ <http://www.siart.de/typografie/latexipps.xhtml>
Übersichtsseite mit Links

Foren:

- ⊙ <http://www.mrunix.de/forums/forumdisplay.php?f=38>
- ⊙ <http://golatex.de/>
- ⊙ <http://www.latex-community.org/>

FAQs (frequently asked Questions):

- ⊙ <http://projekte.dante.de/DanteFAQ>
- ⊙ <http://www.tex.ac.uk/cgi-bin/texfaq2html?introduction=yes>

Oder einfach suchen!



Dokumentationen

<http://www.latex-project.org/guides/>
(offizielle) L^AT_EX-Dokumentationen:

- ⊙ <http://www.techscribe.co.uk/ta/latex-introduction.pdf>
einseitige Einführung
- ⊙ <http://www.tex.ac.uk/ctan/info/lshort/german/l2kurz.pdf>
Offizielle Einführung in L^AT_EX
- ⊙ <ftp://ftp.dante.de/tex-archive/info/german/l2tabu/l2tabu.pdf>
Do's und Dont's im Umgang mit L^AT_EX
- ⊙ <http://upload.wikimedia.org/wikipedia/commons/2/2d/LaTeX.pdf>
Wikibook als pdf
- ⊙ <http://www.tex.ac.uk/tex-archive/info/visualFAQ/visualFAQ.pdf>
- ⊙ <ftp://ctan.tug.org/tex-archive/info/epslatex.pdf>
Grafiken in L^AT_EX

Dokumentationen für Dokumentenklassen:

- ⊙ <ftp://ftp.dante.de/pub/tex/macros/latex/contrib/koma-script/scrguide.pdf>
Dokumentation für die KOMA-Klassen
- ⊙ <http://www.tex.ac.uk/CTAN/macros/latex/contrib/beamer/doc/beameruserguide.pdf>
Dokumentation für die beamer-Klasse



Typographie usw

Inhalt

Was ist IATeX

Technik

Dieser Kurs ...

Wie arbeitet man
mit IATeX

Wo finde ich Hilfe

- ⊙ <http://zvisionwelt.wordpress.com/downloads/> bzw.
<http://zvisionwelt.files.wordpress.com/2012/01/typokurz.pdf> bzw.
<http://zvisionwelt.files.wordpress.com/2012/01/praeambel.pdf>
- ⊙ [http://astro.uni-tuebingen.de/~nagel/pub/Kolleg/Latex/
VorlesungSchriften.pdf](http://astro.uni-tuebingen.de/~nagel/pub/Kolleg/Latex/VorlesungSchriften.pdf)
- ⊙ [http:
//homepage.ruhr-uni-bochum.de/georg.verweyen/latexfuerword.html](http://homepage.ruhr-uni-bochum.de/georg.verweyen/latexfuerword.html)
- ⊙ <http://www2.informatik.hu-berlin.de/sv/lehre/typographie.pdf>
- ⊙ <http://www.tug.dk/FontCatalogue/> Schriften in IATeX