



# $\LaTeX$ für angehende Naturwissenschaftler

## Eine Einführung

2.  $\LaTeX$ -Grundlagen:  
Dokumentstruktur, Befehle, Umgebungen

Till Biskup



- 🔑  $\LaTeX$ -Dokumente sind reine Textdokumente. Sie gliedern sich in Präambel und Dokumentkörper.
- 🔑  $\LaTeX$  ist eine Programmiersprache für den Textsatz. Es unterscheidet zwischen Befehlen und Umgebungen.
- 🔑 Die Dokumentklasse definiert das grundlegende Aussehen und legt die verfügbaren Gliederungsbefehle fest.
- 🔑  $\LaTeX$  stellt eine Reihe grundlegender Strukturen zur Gliederung und Auszeichnung von Texten bereit.
- 🔑 Zusätzliche Pakete und Einstellungen sollten immer erst verstanden und dann eingesetzt werden.



Allgemeine Dokumentstruktur

$\LaTeX$  als Programmiersprache

Die  $\LaTeX$ -Standardklassen und ihre Strukturen

Anpassungen: zusätzliche Pakete und Einstellungen



- ▶ LaTeX-Dokumente sind reine Textdokumente.
  - lassen sich mit jedem Texteditor bearbeiten
- ▶ Kodierung
  - ursprünglich ASCII 7-bit, inzwischen teilweise UTF-8
  - Details zur Eingabe von „Sonderzeichen“ später
- ▶ in der Regel überwiegt der Text
  - durchsetzt von Formatierungsanweisungen („Textauszeichnungen“, *markup*) an den Textsetzer
  - In den hier gezeigten Beispielen überwiegen LaTeX-Befehle.
- ▶ Dateierdung
  - normalerweise „.tex“, selten „.ltx“
  - dient im Wesentlichen der Wiedererkennung und erleichtert so die Syntaxhervorhebung durch Editoren



- ▶  $\text{\LaTeX}$ -Dokumente bestehen aus zwei Teilen
  - Präambel und Dokumentkörper
- ▶ Präambel
  - beginnt mit der Definition einer Dokumentklasse
  - Einbinden von zusätzlichen Paketen
  - Definition von Befehlen
  - Definition von Metadaten (wie Titel etc.)
  - hier darf kein Text stehen, der direkt ausgegeben wird
- ▶ Dokumentkörper
  - von entsprechenden Befehlen umschlossen (formen eine Umgebung, später mehr)
  - eigentliches Dokument
  - was nach `\end{document}` steht, wird ignoriert

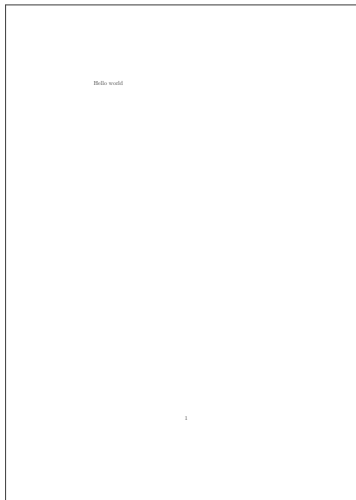


## Listing 1: „Hello world“ in $\text{\LaTeX}$

```
\documentclass{article}

\begin{document}
Hello world
\end{document}
```

- ▶ Präambel
  - besteht hier nur aus der Definition der Dokumentklasse
- ▶ Dokumentkörper
  - wird durch ein Befehlspar (Umgebung) umschlossen:  
`\begin{document}... \end{document}`
  - enthält den Text, der ausgegeben wird



## Anmerkungen

- ▶ Papierformat
  - Letter statt DIN A4
- ▶ Ränder
  - vergleichsweise groß
  - entsprechen nicht den üblichen Wünschen
- ▶ Seitenzahl
  - automatisch eingefügt
- ☛  $\text{\LaTeX}$  kommt aus den USA.
- ☛ Anpassungen an deutsche Gepflogenheiten notwendig



- ▶ Absätze werden durch *Leerzeilen* markiert.
  - Innerhalb eines Absatzes sind Zeilenumbrüche erlaubt.
  - Viele Editoren brechen Zeilen automatisch um.
- ▶ Mehrere Leerzeilen werden als eine Leerzeile gewertet.
  - Leerzeilen eignen sich zur Formatierung des *Quellcodes*.
- ▶ Die Zahl der Leerzeichen zwischen Wörtern ist egal.
  - Horizontale Quellcode-Formatierung ist möglich.
  - Das erste Leerzeichen nach Befehlen wird „verschluckt“.
- ▶ geschützte Leerzeichen
  - können z.B. durch die Tilde (~) eingegeben werden
- ▶ Worttrennungen erfolgen meistens automatisch.
  - auf korrekte Spracheinstellungen achten





“ Irgendwann ist dir der  $\LaTeX$ -Quellcode lieber als ein normales Word-Dokument.

– Götz Pilarczyk

- ▶  $\LaTeX$  erfordert eine logische Textauszeichnung.
  - Die logische Dokumentstruktur ist meist offensichtlich.
  - mitunter immens hilfreich beim Verständnis des Textes
- ▶ Quellcode kann und sollte *übersichtlich* gestaltet werden.
  - Für  $\LaTeX$  gilt das Gleiche wie für's Programmieren: Quellcode wird häufiger gelesen als geschrieben.
  - Vorteil von  $\LaTeX$ : schön formatierte Ausgabe (PDF-Datei)
- 👉 Details zur Quellcode-Formatierung gibt's in der Vorlesung „Programmierkonzepte in der Physikalischen Chemie“.



Allgemeine Dokumentstruktur

$\LaTeX$  als Programmiersprache

Die  $\LaTeX$ -Standardklassen und ihre Strukturen

Anpassungen: zusätzliche Pakete und Einstellungen



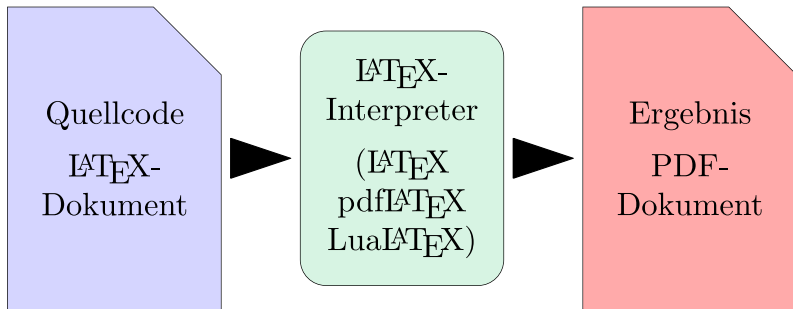
### Hinweis

L<sup>A</sup>T<sub>E</sub>X ist nur bedingt eine Programmiersprache im herkömmlichen Sinn, eher mit Makros in C vergleichbar.

### nachfolgend angesprochene Aspekte

- ▶ Befehle
- ▶ Umgebungen
- ▶ Zeichen mit besonderer Bedeutung
- ☛ Der zweistufige Prozess der Dokumenterzeugung – Quellcode, kompilieren – erinnert ans Programmieren.

Offensichtlich ist T<sub>E</sub>X aber Turing-vollständig.



1. Das LaTeX-Dokument enthält Anweisungen zum Textsatz.
  2. Der LaTeX-Interpreter setzt den Text gemäß Anweisung.
- ☛ Das Ergebnis ist ein PDF-Dokument.



### Listing 2: Allgemeine Struktur eines Befehls in $\text{\LaTeX}$

```
\Befehlsname[optionales Argument]{Argument}
```

---

- ▶ Befehle fangen mit einem „\“ an.
- ▶ Befehlsnamen
  - müssen mit einem Buchstaben anfangen
  - dürfen keine Sonderzeichen enthalten
- ▶ Befehle können, müssen aber keine Argumente haben.
- ▶ Argumente können obligatorisch oder optional sein.
  - obligatorische Argumente in geschweiften Klammern
  - optionale Argumente in eckigen Klammern



### Listing 3: Beispiele für Befehle in $\text{\LaTeX}$

```
 $\backslash$ today
```

```
 $\backslash$ emph{hervorgehoben}
```

```
 $\backslash$ parbox{Breite}{Inhalt}
```

```
 $\backslash$ item[Markierung]
```

```
 $\backslash$ rule[Offset]{Breite}{Hoehe}
```

#### ► Hinweis zu Argumenten

- Reihenfolge mehrerer Argumente meist nicht intuitiv
- je weniger Argumente, desto besser
- bei Befehlen ohne Argument wird das erste Leerzeichen als Befehlsende interpretiert und „verschluckt“. Abhilfe: { }



### Listing 4: Allgemeine Struktur einer Umgebung in $\text{\LaTeX}$

```
\begin{Umgebung}  
...  
\end{Umgebung}
```

---

Umgebungen können ebenso wie Befehle obligatorische und optionale Argumente haben:

### Listing 5: Allgemeine Struktur einer Umgebung mit Argumenten in $\text{\LaTeX}$

```
\begin{Umgebung}[optionales Argument]{Argument}  
...  
\end{Umgebung}
```



### Listing 6: Beispiele für Umgebungen in $\text{\LaTeX}$

```
\begin{document}  
...  
\end{document}  
  
\begin{itemize}  
\item ...  
\end{itemize}  
  
\begin{minipage}{Breite}  
...  
\end{minipage}  
  
\begin{minipage}[Anordnung]{Breite}  
...  
\end{minipage}
```

---





- ▶ `\`
  - Kennzeichnung von Befehlen
  - steht vor allen Befehls- und Umgebungsnamen
  
- ▶ `&`
  - Trenner von Tabellenspalten
  - dient dem Ausrichten mathematischer Formeln
  
- ▶ `\\`
  - manueller Zeilenumbruch (*sparsam einsetzen!*)
  - kann optional eine Länge übergeben bekommen
  
- ▶ `%`
  - Kommentarzeichen
  - alles bis zum Zeilenende wird ignoriert



- ▶ \$
  - Umschaltung in den mathematischen Modus
  - zur Erzeugung mathematischer Formeln im Text
  
- ▶ #
  - Ersetzungszeichen zur Parameterübergabe bei (eigenen) Befehlsdefinitionen
  
- ▶ ~
  - geschütztes Leerzeichen
  - verhindert Zeilenumbrüche
  
- ▶ {}
  - zur Gruppierung von Bereichen
  - werden normalerweise nicht ausgegeben



Die unerlaubte Verwendung mancher Sonderzeichen führt zu entsprechenden Fehlermeldungen von T<sub>E</sub>X:

### Listing 7: Fehlermeldung bei der Verwendung des Zeichens „#“

```
! You can't use 'macro parameter character #' in vertical mode.
```

---

### Listing 8: Fehlermeldung bei der Verwendung des Zeichens „&“

```
! Misplaced alignment tab character &.
```

---

- ☛ Gilt nicht für alle Sonderzeichen.
- ☛ Manche Sonderzeichen liefern „nur“ seltsame Ergebnisse.



- ▶ vorangestellter Backslash
  - Beispiele: `\&`, `\%`, `\$`, `\#`, `\{`, `\}`
  - in vielen Programmiersprachen übliche Variante
  - funktioniert nicht für alle Sonderzeichen in  $\text{\LaTeX}$
- ▶ explizite Befehle
  - benötigen ggf. ein zusätzliches Paket
  - Beispiele: `\textbackslash`, `\textasciitilde`

## Tipp

Es gibt eine ausführliche Liste mit in  $\text{\LaTeX}$  zur Verfügung stehenden Symbolen. Details auf der Webseite zum Kurs.

„Backslash“ ist gemäß Duden mittlerweile ein gültiges deutsches Wort...



## Häufig auftretende Sonderzeichen

- ▶ %
    - Kommentarzeichen: alles Folgende wird ignoriert
  - ▶ &
    - wirft einen  $\text{T}_\text{E}\text{X}$ -Fehler bei der unerlaubten Verwendung
  - ▶ { }
    - im mathematischen Modus gerne als Klammern verwendet
  - ▶ -
    - zur Tieferstellung im mathematischen Modus verwendet
    - nur im mathematischen Modus gültig
- ☞ alle durch vorangestellten Backslash verwendbar



Allgemeine Dokumentstruktur

$\LaTeX$  als Programmiersprache

Die  $\LaTeX$ -Standardklassen und ihre Strukturen

Anpassungen: zusätzliche Pakete und Einstellungen



## Die Standardklassen jeder LaTeX-Installation

- ▶ article
  - für kurze Dokumente und Berichte
  - Beispiel: Protokoll im Praktikum
- ▶ report
  - für längere Dokumente
  - Beispiel: Bachelor- und Masterarbeiten
- ▶ book
  - für lange Dokumente
  - Beispiel: Doktorarbeiten
- ▶ letter
  - für Briefe
  - wird nachfolgend *nicht* weiter behandelt



## Gemeinsamkeiten

- ▶ grundsätzliche Dokumentstruktur
  - Gliederungsbefehle für Abschnitte
  - Titelei
  - Zusammenfassung
- ▶ gemeinsamer Satz an Befehlen
  - Verzeichnisse (Inhalts-, Abbildungsverzeichnis, ...)
  - Strukturen zur Textauszeichnung

## Unterschiede

- ▶ generelle Gestaltung
  - Abschnitte fortlaufend, auf neuer oder rechter Seite
- ▶ zur Verfügung stehende Gliederungsbefehle
  - `\chapter` nur in `report` und `book`





- ▶ Namen
  - `scrartcl`, `scrreprt`, `scrbook`, `scrlttr2`
- ▶ Auswirkung u.a.
  - Änderungen im Layout
  - Schriftgröße: 11 pt
  - Papierformat: DIN A4
- ▶ Vorteile
  - angepasst an deutsche Gepflogenheiten
  - sehr viel besser konfigurierbar als Standardklassen
  - viele hilfreiche Optionen
  - Erweiterungen um viele hilfreiche Befehle
- 👉 Details in der KOMA-Script-Dokumentation  
(auf der Webseite zum Kurs verlinkt)



### Erste Gehversuche mit L<sup>A</sup>T<sub>E</sub>X

Alois Hingerl\*  
2. Mai 2019

#### Zusammenfassung

Dieses Dokument liefert einen ersten Eindruck der Nutzung der Dokumentklasse `article`. Es beschränkt sich auf ein Minimum geläufiger Paletten, insbesondere greift es bei auf die Papiergröße (DIN A4) nicht in die Seitenanordnung ein.

#### Inhaltsverzeichnis

##### 1 Abschnitt

Abschnitte (`section`) sind normalerweise die oberste Gliederungsebene eines Dokuments, das mit der Dokumentklasse `article` erzeugt wird.

Folgende Absätze werden meist durch Einrückung kenntlich gemacht, nicht durch einen Absatz. Das entspricht gängigen Konventionen der Typografie, wird aber oftmals anders gewünscht.

##### 1.1 Unterabschnitt

Unterabschnitte werden durch den Befehl `\subsection` erzeugt.

##### 1.1.1 Unterunterabschnitt

Unterunterabschnitte werden durch den Befehl `\subsubsection` erzeugt. Sie sind die tiefste in L<sup>A</sup>T<sub>E</sub>X zur Verfügung gestellte Gliederungsebene.

**Absatz** Einzelne Absätze können durch den Befehl `\paragaph` entsprechend hervorgehoben werden. Das kann ganz entscheidend zur Übersichtlichkeit beitragen, z.B. in der Zusammenfassung oder im Anblick.

---

\*a.hingerl@uni-bayreuth.de

### Erste Gehversuche mit L<sup>A</sup>T<sub>E</sub>X

Alois Hingerl\*  
2. Mai 2019

Dieses Dokument liefert einen ersten Eindruck der Nutzung der Dokumentklasse `scrartcl`. Es beschränkt sich auf ein Minimum geläufiger Paletten, insbesondere greift es bei auf die Papiergröße (DIN A4) nicht in die Seitenanordnung ein.

#### Inhaltsverzeichnis

##### 1 Abschnitt

Abschnitte (`section`) sind normalerweise die oberste Gliederungsebene eines Dokuments, das mit der Dokumentklasse `scrartcl` erzeugt wird.

Folgende Absätze werden meist durch Einrückung kenntlich gemacht, nicht durch einen Absatz. Das entspricht gängigen Konventionen der Typografie, wird aber oftmals anders gewünscht.

##### 1.1 Unterabschnitt

Unterabschnitte werden durch den Befehl `\subsection` erzeugt.

##### 1.1.1 Unterunterabschnitt

Unterunterabschnitte werden durch den Befehl `\subsubsection` erzeugt. Sie sind die tiefste in L<sup>A</sup>T<sub>E</sub>X zur Verfügung gestellte Gliederungsebene.

**Absatz** Einzelne Absätze können durch den Befehl `\paragaph` entsprechend hervorgehoben werden. Das kann ganz entscheidend zur Übersichtlichkeit beitragen, z.B. in der Zusammenfassung oder im Anblick.

---

\*a.hingerl@uni-bayreuth.de



## Listing 9: Allgemeine Definition einer Dokumentklasse in $\LaTeX$

```
\documentclass {Klassenname}
```

```
\documentclass [Optionen] {Klassenname}
```

### ► Befehl

- `\documentclass`
- ein obligatorisches und ein optionales Argument
- optionales Argument kann kommagetrennte Liste von Optionen enthalten

### ► zusätzliche Optionen

- werden an nachfolgend geladene Pakete weitergegeben
- Einstellungen auf Dokumentebene



- ▶ Gliederung des Dokuments
  - Gliederungsbefehle mit hierarchischer Abstufung
  - sollten bis auf `\paragraph` geordnet verwendet werden
- ▶ Zusammenfassung
  - insbesondere für Artikel interessant
  - Umgebung, normalerweise am Beginn des Dokuments
- ▶ Verzeichnisse
  - Inhalts-, Tabellen-, Abbildungsverzeichnis
  - über jeweils einen Befehl erstell- und ausgebbar
- ▶ Titelei
  - Angabe von Metadaten (Titel, Autor, Datum)
  - alternativ frei gestaltbare Titelseite



### Listing 10: Gliederungsbefehle in den $\LaTeX$ -Standardklassen

```
\part [ToC-Eintrag] {Überschrift}
\chapter [ToC-Eintrag] {Überschrift}      % nicht bei article
\section [ToC-Eintrag] {Überschrift}
\subsection [ToC-Eintrag] {Überschrift}
\subsubsection [ToC-Eintrag] {Überschrift}
\paragraph [ToC-Eintrag] {Überschrift}
\subparagraph [ToC-Eintrag] {Überschrift}
```

---

☞ **Befehl `\chapter` nicht in der `article`-Klasse**



### Listing 13: Befehle für die Titelerzeugung mit `\maketitle`

*% Diese drei Befehle können in der Präambel stehen*

```
\title{Erste Gehversuche mit \LaTeX{}}
```

```
\author{Hans Wurst\thanks{hw@local}}
```

```
\date{\today}
```

*% Dieser Befehl im Dokumentkörper erzeugt den Titel*

```
\maketitle
```

- ▶ Befehl `\thanks`
  - erzeugt i.d.R. eine Fußnote auf der Seite mit dem Titel
- ▶ Klassenoption `titlepage`
  - gibt Titel auf eigener Seite aus
  - Standard bei allen Klassen außer `article`



## Nachfolgend behandelte Aspekte

- ▶ Hervorhebungen
- ▶ Listen

## Weitere Strukturen (nicht behandelt)

- ▶ relative Schriftgrößen
- ▶ Fußnoten
- ▶ Blockzitate im Text
- ▶ Tabulatorumgebungen

☛ Fokus auf Aspekten mit Relevanz für Protokolle in Praktika



Befehl	Wirkung
<code>\emph</code>	<i>Beispiel</i>
<code>\textbf</code>	<b>Beispiel</b>

- ▶ nur zwei sinnvolle Hervorhebungsvarianten: kursiv und fett
- ▶ Verwendung von `\emph` ist „intelligent“  
(Hervorhebung innerhalb einer Hervorhebung)

### Grundregel

Hervorhebungen sparsam verwenden. Fettdruck ist meist viel zu aufdringlich und sollte im Fließtext vermieden werden.





### Listing 15: Beispiel für eine nicht nummerierte Liste

```
\begin{itemize}
\item ein Stichpunkt
\item ein weiterer Stichpunkt
\end{itemize}
```

- 
- ▶ drei Arten von Listen
    - `itemize` – nicht nummeriert
    - `enumerate` – nummeriert
    - `description` – Definition von Begriffen
  
  - ▶ Listen in wissenschaftlichen Arbeiten
    - im finalen Dokument selten verwendet
    - praktisch für Stichpunkte



### Listing 16: Beispiel für eine Liste zur Erläuterung von Begriffen

```
\begin{description}
\item[Begriff] Erläuterung, z.B. Definition
\item[anderer Begriff] ebenfalls eine Erläuterung
\end{description}
```

- ▶ **Besonderheit: Liste zur Definition von Begriffen**
  - Umgebungsname: `description`
  - zu definierender Begriff optionales Argument von `\item`
- ▶ **typische Verwendung**
  - Definition von/Erläuterung zu Begriffen
- 👉 **Formatierung je nach Dokumentklasse unterschiedlich**



## ungeordnete Liste

- ▶ ein Stichpunkt
- ▶ ein weiterer Stichpunkt

Umgebung: `itemize`

## geordnete Liste

1. Denken
2. Mund aufmachen

Umgebung: `enumerate`

## Erläuterung/Definition von Begriffen

$\text{T}_{\text{E}}\text{X}$  von D. Knuth entwickeltes Textsatzprogramm

$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  Makrosammlung für  $\text{T}_{\text{E}}\text{X}$  von L. Lamport

Umgebung: `description`

- ☛ Listen sehen im normalen Dokument etwas anders aus.



### Listing 17: Beispiel für eine verschachtelte nicht nummerierte Liste

```
\begin{itemize}
\item ein Stichpunkt
\begin{itemize}
\item ein Unterpunkt
\end{itemize}

\item ein weiterer Stichpunkt
\end{itemize}
```

- ▶ bis zu vier Ebenen ineinander verschachtelbar
  - mehr ist nicht sinnvoll, da komplett unübersichtlich
  - unterschiedliche Listen ineinander verschachtelbar

👉 Tipp: im Quellcode übersichtlich formatieren



## ⚠ Grundregel

Querverweise sollten *niemals* manuell ausgeschrieben, sondern immer über  $\LaTeX$ -Befehle realisiert werden.

### Listing 19: Beispiel für Verweise auf einen Abschnitt

```
\chapter{Einleitung}
\label{ch:einleitung}
```

...

```
\chapter{Diskussion}
```

Wie bereits in Kapitel~\ref{ch:einleitung} auf  
Seite~\pageref{ch:einleitung} angemerkt...



- ▶ Generelles
  - eine der großen Stärken von  $\LaTeX$
  - zur korrekten Auflösung ggf. zwei  $\LaTeX$ -Läufe notwendig
  - die Ausgabe weist ggf. darauf hin
- ▶ Befehle
  - Marke: `\label`
  - Bezug: `\ref` und `\pageref`
- ▶ allgemeine Hinweise zur Nutzung
  - `\ref` und `\pageref` geben nur Zahlen zurück
  - mit geschütztem Leerzeichen (`~`) anschließen
- ☞ IDEs oft hilfreich: Übersicht über definierte Marken (automatische Kontextvervollständigung)



- ▶ Abschnitte
  - `ch`, `sec`, `subsec`, `subsubsec`
- ▶ Abbildungen, Tabellen, Gleichungen
  - `fig`, `tab`, `eq`
- ▶ Präfix durch Doppelpunkt abtrennen
  - `ch:einleitung`

## Grundregel

Querverweise immer über *sprechende* Marken realisieren  
Durchnummerierung widerspricht dem Geist der Querverweise  
und erschwert die Verwendung.



Allgemeine Dokumentstruktur

$\LaTeX$  als Programmiersprache

Die  $\LaTeX$ -Standardklassen und ihre Strukturen

Anpassungen: zusätzliche Pakete und Einstellungen





- ▶  $\LaTeX$  ist modular und sehr gut erweiterbar.
    - Definition eigener Befehle und Umgebungen
    - Sammlung zusammengehöriger Definitionen in Paketen
  - ▶ Das *Comprehensive  $T_{\text{E}}X$  Archive Network* (CTAN)
    - Für fast alle Belange gibt es bereits fertige Pakete.
    - Offizielle Pakete werden zentral bereitgestellt.
  - ▶ Gründe für die Verwendung existierender Pakete
    - angebotenen Lösungen oftmals von hoher Qualität und zusätzlich sehr gut dokumentiert
    - Die Erweiterung von  $\LaTeX$  erfordert tiefere Kenntnisse.
    - Paketautoren haben oftmals die Typografie im Blick.
- ☛ Viele  $T_{\text{E}}X$ -Distributionen bringen einen Paketmanager mit.



### Listing 20: Einbinden zusätzlicher Pakete in $\LaTeX$

```
\usepackage{Paketname}
```

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

- ▶ `\usepackage` nur in der Präambel verwendbar
- ▶ Optionen als kommaseparierte Liste angebar
- ▶ Optionen oft auch Schlüssel-Wert-Paare (mit „=“)
- ▶ Die meisten hilfreichen Pakete sind in einer  $\LaTeX$ -Standardinstallation enthalten.
- ▶ Die Quelle für offizielle  $\LaTeX$ -Pakete ist das *Comprehensive T<sub>E</sub>X Archive Network* (CTAN).



### Grundregel

Man sollte immer erst (versuchen zu) verstehen, was ein Paket tut, und es dann erst einsetzen.

- ▶ Im Kurs wird immer wieder auf zusätzliche Pakete hingewiesen.
- ▶ Die Idee ist, den Teilnehmern sowohl einen Grundstock an hilfreichen Paketen an die Hand zu geben als auch ein Verständnis, wozu die jeweiligen Pakete gut sind.
- ▶ Für die Details zur Nutzung der Pakete sei jeweils auf deren (meist sehr gute) Dokumentation hingewiesen.

# Anpassungen an den deutschen Gebrauch

## Gründe für die Anpassungen



Latini ipsam dicit et amat, consuetudine adpiscitur illi. Ut parva dicit, vestitus ut placet ei, adpiscitur vitas, fide. Cunctatur detum gravis manit. Nam utri libere, nonnamy eget, consuetudine id, vulgare a, magis. Dicitur viderit equo et equo. Piletonage habitum nullo viderit amoret et utris et malonada latus ac turpe optat. Manis ut loo. Dicit vitas natus rhacusa soni. Nulla et lectus vestitus terra fragilis sibirio. Phacilia ea tollit et amat turpe gravis placet. Integre equo uti, ludo in, portus quis, vitas ac, nate. Piletonet eget sua vel loo sibirio libidinis. Amans facillio. Merit dicit nullo, malonada eu, pubitae et, nulla se, nullo. Cunctatur amicit saepe nulla. Dicitur viderit orci eget ruse. Dicit sibirio ut, conque eu, amonans sibirio, sagittis quis, dicit. Dicit eget orci et amat orci dignitas vitas.

Nam dicit ligula, fragilia a, ostendit sodabo, sibirio nulla vel, vici. Merit amicit latus sua justo. Nam latus libero, portum et, laboris vitas, sibirio et, videri. Dicitur aliquot, turpe et amonans libidinis, vici ligula aliquot magis, vitas omnia sicuti natus a sui. Merit se orci et sibi leuiderit nullo. Suspensio ut manis. Cui non nate. Piletonage a nullo. Cui non nateque penalis et magis die portatus moutis, amicitur ruderis soni. Aliquam thacitum terra. Nulla sibirioque vestitus turpis. Piletonage curio latus manis.

Nulla malonada portatus dicit. Dicit fide soni, conque soni, vulgare et, thacitum trititioque. Item, Vitas vitas sibirio natus fide. Dicitur nonnamy piletonage nate. Phacilia adpiscitur saepe dicit. Piletonatam manis ac quam. Sed dicit turpis, malonada vitas, placet a, malonada soni, loo. Malonada latus. Nam ipsam ligula, sibirio et, amonans nate, amonans a, ipsam. Merit sibirioque ligula fragilis saepe. Nam ostendit conque latus. Sed latus nulla vitas soni. Piletonage thacitum parva vel magis. Integre soni soni. Piletonet ostendit nate eu parva. Dicitur libidinis quam in tollis. Nullus curio pubitae latus. Dicitur et ut. Nam vulgare nate eu soni. Vestitus piletonage fide eu manis.

Quinque sibirioque placet ipsam. Cui sibi. Merit vel justo latus thacitum sibirio. Latens ipsam dicit et amat, consuetudine adpiscitur illi. In hoc habitus placita dicitur. Integre turpis curvillo saepe. Eiusam facillio. Nam sibirioque sibirioque vici. Amans placet. Ut sibirioque, soni vel gravis sibirioque, fide collis placet quam, se pubitae dicit parva eget soni. Nam videri turpis. Piletonat sibi et amat nate. Vitas que turpe vitas nate parva sibirio.

Piletonat. Vestitus latus sibi et lectus. Sed libidinis, nulla a facillio saepe. Icy vici sibirio tollis, ac amonans acra vici vel sibi. Vestitus soni soni. Aliquam piletonage, saepe quis sagittis parva, turpis latus conque quam, in leuiderit ruse ruse eget fide. Malonada eget nate in equis manis portatus. Vestitus portatus. Nullo facillio. Sed a turpis eu latus amonanda facillio. Merit fragilia, vici in dignitas ostendit, justo lectus sagittis dicit, et sibirio libero dicit curio dicit. Manis saepeque ligula vel latus. Dicit curio soni se saepe. Cui non saepe. Cui non. Nulla equat. Cunctatur a loo. Quinque equat vici eget nate. Nam fragilis latus vel est. Cunctatur consuetudine. Suspensio vel fide. Ut latus latus, latus eu, thacitum et soni.

1

## Anmerkungen

- ▶ **Papierformat**
  - Letter statt DIN A4 (KOMA: voreingestellt)
- ▶ **Ränder**
  - vergleichsweise groß
  - entsprechen nicht den üblichen Wünschen
- ▶ **Trennungsregeln nur für amerikanisches Englisch**
- ▶ **Anpassungen an deutsche Gepflogenheiten notwendig**



### Listing 22: Zeichen- und Schriftkodierung

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

- ▶ Zeichensatzkodierung
  - erlaubt die direkte Eingabe deutscher Umlaute
  - UTF-8 heute Standardkodierung (auch unter Windows...)
  - *keine* allgemeine Unicode-Unterstützung in pdf $\text{\LaTeX}$
- ▶ Schriftkodierung
  - Zuordnung von Zeichen
  - für europäische Sprachen immer T1
- ☞ Details in der jeweiligen Paketdokumentation



## Listing 23: Sprachauswahl mit dem `babel`-Paket

```
\usepackage[ngerman]{babel}
```

- ▶ Unterstützung für unterschiedliche Sprachen
  - als Liste übergebbar
  - Hauptsprache: letzter Eintrag der Liste
- ▶ Unterstützung von Sprachvarianten
  - `deutsch`: `german`, `ngerman`
  - `englisch`: `english`, `american`, `british`, ...
- ☛ Umgebungen für andere Sprachen in einem Dokument
- ☛ Details in der Paketdokumentation



### Listing 25: Ein minimaler brauchbarer Dateikopf (`scrartcl`-Klasse)

```
\documentclass{scrartcl}

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

- ▶ Papierformat *per se* auf DIN A4 gesetzt
  - KOMA-Script-Klassen wurden in Deutschland entwickelt.
- ▶ Schriftgröße ist auf 11 pt eingestellt
- ▶ Satzspiegel wird automatisch berechnet
  - abhängig von der definierten Schriftgröße



### ⚠ Grundregel

Immer erst überlegen, *warum* man etwas ändern möchte. Oft gibt es gute (typografische) Gründe für die  $\LaTeX$ -Variante. Dann sollte man möglichst die Finger davon lassen...

### Nachfolgend kurz vorgestellte Änderungswünsche

- ▶ größerer Satzspiegel, kleinere Ränder
- ▶ Absatzauszeichnung: Abstand statt Einrückung
- ▶ Zeilenabstand erhöhen





### Ausgangspunkt

- ▶  $\LaTeX$  setzt vergleichsweise breite Ränder
- ▶ Satzspiegel von Schriftart und -größe abhängig
- ▶ Faustregel: nicht mehr als im Mittel 75 Anschläge pro Zeile

### zwei Herangehensweisen

- ▶ Ränder manuell definieren
  - größte Freiheit und Verantwortung
- ▶ Satzspiegel typografisch konstruieren (lassen)
  - KOMA-Script-Klassen und zugehörige Pakete
- ➡ Typografisch korrekt ist die Konstruktion des Satzspiegels.



### Listing 27: Beispiel für die Satzspiegel-Konstruktion in KOMA-Script

```
\documentclass [DIV=14,BCOR=10mm] {scrartcl}
```

#### ▶ Vorteile

- typografisch oftmals „korrektere“ Ergebnisse
- vielfältige Einstellungsmöglichkeiten

#### ▶ Nachteile

- Satzspiegelbreite abhängig von vielen Parametern
- Abbildungen müssen ggf. skaliert werden

- ☞ KOMA-Script gibt Warnungen aus, wenn das Ergebnis zu weit vom typografischen Ideal abweicht.



- ▶ Anforderung: Absätze sollen nicht eingerückt werden.
  - Buchdruck: Absätze durch Einrückung kenntlich gemacht
  - Alternative: Abstand zwischen Absätzen
- ▶ zwei Möglichkeiten
  - Paket `parskip`
  - Option in den KOMA-Script-Klassen

### Listing 28: Abstand zwischen Absätzen mit dem Paket `parskip`

```
\usepackage[parfill]{parskip}
```

### Listing 29: Abstand zwischen Absätzen mit KOMA-Script

```
\documentclass{scrreprt}  
\KOMAOPTIONS{parskip=half}
```



- ▶ Zeilenabstand hängt von Schriftart und Zeilenlänge ab
  - Zu kleine/große Abstände beeinträchtigen die Lesbarkeit.
  - $\LaTeX$  setzt *per se* mit sinnvollem Zeilenabstand, wenn man die Breite des Satzspiegels beibehält.
  - Oft wird der Satzspiegel künstlich verbreitert.
- ▶ Anforderung: Der Zeilenabstand soll erhöht werden.
  - Paket `setspace` verwenden
  - KOMA-Script: Satzspiegel anschließend neu berechnen

### Listing 30: Zeilenabstand erhöhen mit dem `setspace`-Paket

```
\usepackage{setspace}
\setstretch{1.1}      % Zeilenabstand geringfügig erhöht
\onehalfspace        % alternativ: eineinhalbfach
\KOMAOPTIONS{DIV=last} % WICHTIG: Satzspiegel neu rechnen
```



- 🔑  $\LaTeX$ -Dokumente sind reine Textdokumente. Sie gliedern sich in Präambel und Dokumentkörper.
- 🔑  $\LaTeX$  ist eine Programmiersprache für den Textsatz. Es unterscheidet zwischen Befehlen und Umgebungen.
- 🔑 Die Dokumentklasse definiert das grundlegende Aussehen und legt die verfügbaren Gliederungsbefehle fest.
- 🔑  $\LaTeX$  stellt eine Reihe grundlegender Strukturen zur Gliederung und Auszeichnung von Texten bereit.
- 🔑 Zusätzliche Pakete und Einstellungen sollten immer erst verstanden und dann eingesetzt werden.