

L^AT_EX-Einführung für Linguisten

Constantin Freitag Antonio Machicao y Priemer
Universität Konstanz Freie Universität Berlin

23. Februar 2015

6. linguistischer Methodenworkshop Humboldt-Universität zu Berlin

Inhaltsverzeichnis

1	Was ist es und wie geht das?	6
1.1	Geschichte	6
1.2	Funktion – WYSIWYG vs. WYGIWYN	6
1.3	Warum sollte ich es benutzen?	7
2	Grundlagen	9
2.1	Wie funktioniert L ^A T _E X?	9
2.2	Die Software	10
2.2.1	Texmaker-Konfiguration	10
2.3	Befehle	11
2.4	Zeichen und Sonderzeichen	12
2.5	Leerzeichen und Zeilenumbrüche	13
2.6	(Aus-)Kommentieren	14
3	Dokumentstruktur	15
3.1	Präambel	15
3.1.1	Dokumentklasse	15
3.1.2	Pakete einbinden	16
3.2	Body	17
3.2.1	Metadaten	17
3.2.2	Textauszeichnung	18
3.2.3	Gliederung	19
3.2.4	Inhaltsverzeichnis	20
3.2.5	Fußnoten	20
3.3	Übungen	21

4	Textumgebungen	21
4.1	Zitate	21
4.2	Listenumgebungen	22
4.2.1	Itemize	23
4.2.2	Enumerate	23
4.2.3	Description	23
4.2.4	Kombinierte Listen	24
4.2.5	Individualisierte Anpassung	25
4.2.6	Allgemeine Listenumgebung	25
4.2.7	Pakete für Aufzählungen	25
4.3	Theoreme	26
4.3.1	Pakete für Theoreme	27
4.4	Wörtliche Wiedergabe	27
4.4.1	Probleme mit <code>verbatim</code>	28
4.4.2	Pakete für wörtliche Wiedergaben	28
4.5	Abstract	28
4.6	Vers	29
4.7	Flattersatz	29
4.7.1	Linksbündiger Text	30
4.7.2	Rechtsbündiger Text	31
4.8	Übungen	32
5	Nicht-textbezogene Elemente	32
5.1	Grafiken	32
5.1.1	Größe, Darstellung und Formate	33
5.1.2	Grafikpfad	34
5.1.3	Paket zur Erstellung von Grafiken	35
5.2	Tabellen	35
5.2.1	Linien	36
5.2.2	Erweitertes Layout	37
5.2.3	Pakete für Tabellen	39
5.3	Gleitumgebungen	40
5.4	Abbildungs- und Tabellenverzeichnis	43
6	Querverweise	43
6.1	Einfache Querverweise	43
6.2	Präfixe	44
6.3	Pakete für Querverweise	45
7	Bibliographie	45
7.1	BibTeX-Einträge	48
7.2	Das <code>natbib</code> -Paket	49
7.3	Literaturverwaltungssysteme	50
7.4	Weitere Pakete für Bibliographien	50

8	Pakete für Linguisten	50
8.1	Nummerierte Beispiele und Glossen	50
8.1.1	Beispiele	50
8.1.2	Grammatikalitätsurteile	51
8.1.3	Glossen	52
8.2	IPA-Notation	54
8.3	Bäume	55
8.3.1	Das <code>forest</code> -Paket	55
8.3.2	Das <code>qtree</code> -Paket	57
8.3.3	Das <code>tikz-qtree</code> -Paket	58
8.4	Attribute-Value Matrix (AVM)	60
9	Mathematik-Umgebung	60
9.1	Weitere Pakete für Mathematik-Umgebungen	62
9.2	Übungen	62
10	Typographie	63
10.1	Leerzeichen	63
10.2	Abkürzungen	64
10.3	Ellipsen – Auslassungspunkte	65
10.4	Anführungszeichen	66
10.5	Der Strich	66
11	Längere Arbeiten – Multifile Parsing	67
12	Finaler Satz	68
12.1	Titelseite	68
12.2	Satzspiegel	69
12.2.1	Generelles Seitenlayout	69
12.2.2	Ränder	69
12.2.3	Bindekorrektur	70
12.2.4	Schriftgröße	70
12.2.5	Zeilenabstand	70
12.2.6	Kopf- und Fußzeile	71
12.2.7	Seitenzahlen	71
12.3	PDF Optionen	71
12.4	Silbentrennung	72
12.4.1	Silbentrennung anpassen	72
12.4.2	Silbentrennung unterdrücken	73
12.5	Seitenkorrektur	73
13	Programmieren – Definiere deine eigenen Makros	73
13.1	Eigene Makros definieren	73
13.1.1	Befehle ohne Argumente	74

13.1.2 Befehle mit Argumenten	75
13.1.3 Weitere Beispiele für nützliche Befehle	77
13.2 Eigene Umgebungen	77
14 Pakete und Handbücher	78
Literatur	81
Index	82

Vorwort

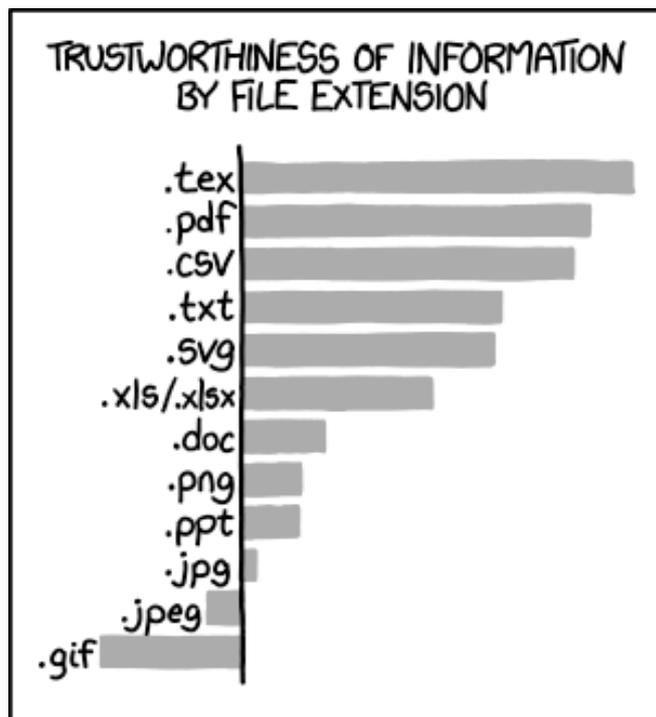


Abbildung 1: Nichts hinzuzufügen! (xkcd.com)

Der folgende Reader wurde gemacht für eine \LaTeX -Einführung von Constantin an der Universität Konstanz und für einen \LaTeX -Workshop von Constantin und Antonio beim **6. linguistischen Methodenworkshop** an der Humboldt-Universität zu Berlin erweitert. Dieser Reader kann aber auch für das \LaTeX -Selbststudium verwendet werden. Als \LaTeX -Distribution hierfür wurde $\text{\textsc{MiKTeX}}$ und als Editor $\text{\textsc{Texmaker}}$ benutzt.

Dieser Reader beinhaltet einen Index von Begriffen und Befehlen mit den entsprechenden Seitenzahlen am Ende des Dokuments, darüber hinaus findet man eine Liste von allen hier besprochenen Paketen und Software mit den entsprechenden URLs im Abschnitt 14 und am Ende einiger Kapitel **Short-cuts** oder **Hinweise**, die sich i. d. R. auf $\text{\textsc{Texmaker}}$ beziehen, wie der folgende Hinweis:

Hinweis: Achte darauf, dass $\text{\textsc{Texmaker}}$ Quellcode und PDF-Ansicht nebeneinander anzeigt. Gehe dafür zu `Optionen / Texmaker konfigurieren / Befehle /` und aktiviere dort `embed`.

1 Was ist es und wie geht das?

1.1 Geschichte

- $\tau\epsilon\chi$ (T_EX) wurde zwischen 1977 und 1986 von Donald E. Knuth entwickelt, da er mit den damaligen Textverarbeitungsprogrammen unzufrieden war.
- L^AT_EX ist ein Interface, das eine Sammlung an nützlichen *Makros* zur Verfügung stellt, um einfacher das T_EX-System verwenden zu können. Es wurde von Leslie Lamport geschrieben. Der Name ist eine Abkürzung für **L**amport **T**e**X**.

1.2 Funktion – WYSIWYG vs. WYGIWYN

Im Vergleich zu Textverarbeitungsprogrammen wie *MS Word* oder *Libre Office*, die nach dem WYSIWYG-Prinzip (*what-you-see-is-what-you-get*) funktionieren, arbeitet L^AT_EX nach einem WYGIWYN- (*what-you-get-is-what-you-need*) oder WYGIWYM-Prinzip (*what-you-get-is-what-you-mean*). L^AT_EX arbeitet mit einer *logischen Textauszeichnung* – ähnlich wie HTML – um Textelemente zu definieren.

Warum ist es besser?

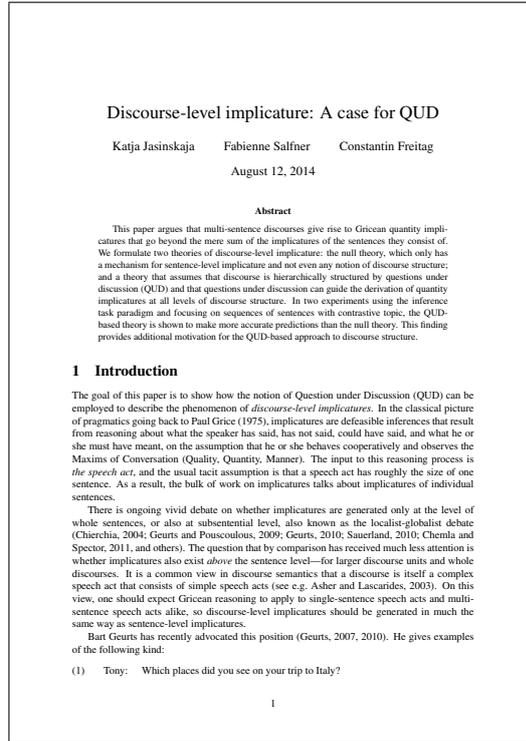
- Es funktioniert! In einem 100-seitigen Dokument wird sich nicht plötzlich die Nummerierung von Kapiteln und Beispielen ändern, oder die Schriftart an einigen Stellen auf Comic Sans umstellen!
- T_EX stellt mithilfe eines Algorithmus sicher, dass die Leerzeichen zwischen Wörtern nicht zu groß werden und die besten Stellen für eine korrekte Silbentrennung gefunden werden, sodass Absätze in Blocksatz optimal aussehen.
- Es ist für alle Plattformen kostenlos. Die Dokumente sind somit plattformunabhängig.
- Es erlaubt eine stabile Verwendung von Automatismen wie: Referenzen, Bibliographien, Indizes, ...
- Es benötigt nur wenig Rechenleistung für große und komplexe Dokumente.
- Es kann mathematische Formeln korrekt darstellen.
- Es wird immer weiter von der *Community* entwickelt, sodass immer wieder neue (kostenlose) Pakete heruntergeladen werden können.
- Es trennt den Inhalt von der Form.

1.3 Warum sollte ich es benutzen?

Weil es dich zu einem besseren Linguisten macht. Man hört auf, Sachen zu tun, die der Computer für einen erledigen kann. Daher hat man mehr Zeit, sich mit dem Inhalt zu befassen.

Einige Beispiele, was man mit L^AT_EX tun kann:

Bücher & Artikel



Poster & Briefe

Filler complexity in long-distance wh-extractions in German
 Constantin Freitag & Sophie Repp
 Humboldt-Universität zu Berlin

I. Introduction

Häfner and Sag (2010) and Hofmeister (2008) found that the processing of filler-gap dependencies with an extracted wh-phrase in English is sensitive to the complexity of the wh-phrase (Sag and Chung, 2002).

- Reading times of the word after the extracted wh-phrase were longer after a which-phrase than after a bare wh-word.
- All the gap sites, in contrast, RTs were shorter for 'which' gaps than for bare 'wh' gaps.

Suggested reasons: greater informativity of which-phrases, therefore –

- reduced likelihood of misanalysis (early gap filling) as no continued structure
- easier retrieval due to increased activation and resistance to interference

Our findings from German call the generality of these findings and their interpretation into question.

II. Methods

Exp. 1: Self-Paced Reading

- Non-cumulative word-by-word presentation in a stationary window
- Task: reading + answer task
- 64 Participants saw 30 critical items and 120 unrelated fillers each

Exp. 2: Acceptability Judgments

- Thermoneter Judgment paradigm (Friedrichsen, 2009) – Participants judged sentence pairs relative to two fixed anchor sentence pairs.
- 30 Participants saw 30 critical items and 64 unrelated fillers each

2 x 3 Designs

- SENTENCE TYPE: BASELINE – no island violation; CNPC – violation of the Complex Noun Phrase Constraint
- FILLER TYPE: BARE – bare wh-phrase; WHICH – which-NP phrase; ASJ – a which-NP phrase with adjective and adverb

III. Materials

Context: Nils will betray (the fact) that Jan has backed the beggar who was on the run and is sufficiently well-known.

Question:

BASELINE: Wie/ welchen Betrüger/ Welchen Anrechtlos Bekannten Betrüger wird Nils verraten, dass Jan t_i gedeckt hat?

CNPC: Wie/ welchen Betrüger/ Welchen Anrechtlos Bekannten Betrüger wird Nils die Tatsache verraten, dass Jan t_i gedeckt hat?

ASJ: Wie/ welchen Betrüger/ Welchen Anrechtlos Bekannten Betrüger wird Nils die Tatsache verraten, dass Jan t_i gedeckt hat?

Answers (only ASJ):

CORRECT: den richtigen Betrüger RELATED: den madden Betrüger DISTRACTIO: den großen Dieb

'the right beggar' 'the mad beggar' 'the tall thief'

IV. Results

Exp. 1: Residual Reading Times
(2-way repeated measures ANOVA)

- No main effects of SENTENCE TYPE.
- Main effects of FILLER TYPE: Matrix clause (17 words after filler) longer RTs for ASJ than for BARE/WHICH.

Exp. 2: Answer Times
(2-way repeated measures ANOVA)

- Main effect of SENTENCE TYPE: BASELINE faster than CNPC.
- Main effect of FILLER TYPE: ASJ faster than WHICH/BARE.
- High overall answer accuracy (93%-95% per condition).

Exp. 3: Acceptability Judgments
(2-Stevens and 2-way repeated measures ANOVA)

- Main effect of SENTENCE TYPE: BASELINE more acceptable than CNPC.
- Interaction of SENTENCE TYPE and FILLER TYPE: BASELINE-BARE-WHICH-ASJ < CNPC-BARE-ASJ.

V. Discussion

Summary

- Exp. 1: Retrieving a more complex wh-phrase is more costly than retrieving a less complex wh-phrase – evidenced by the reading times for ASJ.
- The disadvantage of a very complex wh-filler at the retrieval site might be due to the activation of more semantic features, which is indicated by the advantage of ASJ in the answer times (exclusion of answer alternatives).
- Exp. 2: In grammatical island violations,

increasing informativity of the wh-filler improves overall acceptability.

Interpretation

Probably a trade-off: more complex structural information is more costly to process but more semantic information also helps to identify referents in a discourse.

Possible reasons for differences with other studies:

- Highly greater syntactic complexity of the most informative condition (ASJ)
- Verb final sentence structure of German: retrieval region + sentence wrap-up region

- German wh-phrases are more informative than English ones (case)
- Maybe not only reading time effects but also answer task preparation
- Implications:
 - Higher processing effort pays off in terms of better success at interpretation.
 - Processing in non-island and island structures does not differ.
 - More research on languages with differing morpho-syntax is needed.

Prof. Dr. Marc H. Scholl
 FB Human- & Information Sciences
 Universitätsstraße 10
 78454 Konstanz
 +49 (0) 75 31 68 - 44 52
 Fax: +49 (0) 75 31 68 - 35 77
 Marc.Scholl@uni-konstanz.de
 dhs.uni-konstanz.de

Universität Konstanz: Fach 188 - 78457 Konstanz
 Frau
 Julia Wandt
 Universität Konstanz
 – Kommunikation und Marketing –
 Universitätsstraße 10
 78464 Konstanz

16. 10. 2014

Die KOMA-Script Letter Klasse aktualisiert und eine Briefvorlage für **BT&X**-Nutzer
 Seite 1 / 1

Sehr geehrte Frau Wandt,

Hier der Standardbrief mit Bankzelle im Fuß.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

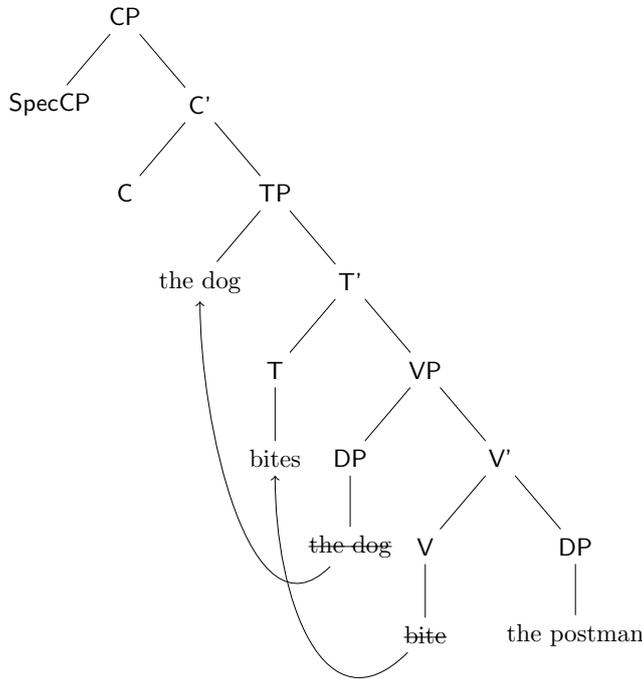
Lorem ipsum dolor sit Mathe: $1 + 2 = 5\sqrt{10}$ amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Mit freundlichen Grüßen

Marc H. Scholl

BW-Bank Konstanz IBAN: DE32 6005 0101 7485 012 74 BIC: SOLA DE 33

Bäume



Glossen & IPA

(1) DEUTSCH

- a. Der Mann hat dem Jungen ein Buch über Linguistik
the man.NOM has the boy.DAT a book.ACC about linguistics
gegeben.
give.PTCP.PRF/gave
'The man gave the boy a book about linguistics.'

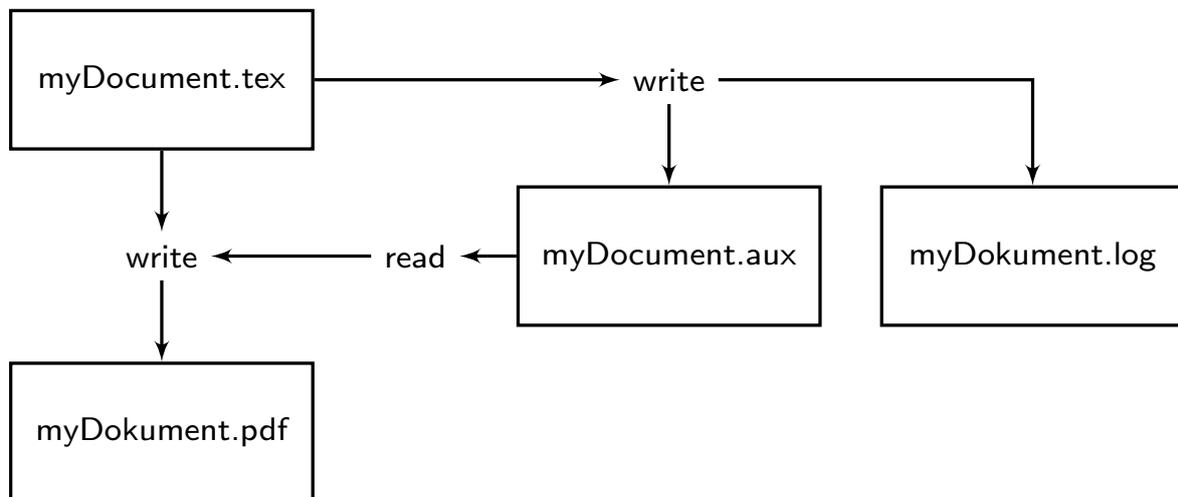
(2) <Phonetik>
/fə'.nɛ.tɪks/
[fə'netɪks]

2 Grundlagen

Zunächst das Wichtigste:

TeXing macht einfach Spaß!

2.1 Wie funktioniert L^AT_EX?



- Der Text wird in einer `.tex`-Datei¹ (z. B. `myDocument.tex`) in einem Editor eurer Wahl (z. B. `Texmaker`) geschrieben.
- Der Editor kompiliert die Datei (z. B. mit `PDFLaTeX`).

¹.`tex` meint die Endung des Dokuments, so wie `.doc` die Endung von Worddokumenten, `.txt` die Endung von allgemeinen Textdokumenten, oder `.jpg` die Endung von Bildern sind. Wenn in diesem Reader von der `.tex`-Datei die Rede ist, meinen wir die L^AT_EX -Datei, an der gearbeitet wird.

- Der Editor schreibt dabei verschiedene Hilfsdateien und darunter auch das PDF-Ergebnis `myDocument.pdf`.
- Die `.log`-Datei beinhaltet das Protokoll der letzten Kompilierung mit \LaTeX .
- Die `.aux`-Datei transportiert Information von einem Kompilierungsvorgang zum nächsten, z. B. über das Inhaltsverzeichnis, Querverweise, ...

2.2 Die Software

In diesem Kurs werden wir mit dem Editor **Texmaker** arbeiten (s. 50 im Abschnitt 14), welcher folgende Vorteile anbietet:

- kostenlos,
- kompatibel mit Linux-, Windows- und Mac OS-Rechnern,
- Unicode-Unterstützung, Rechtschreibkontrolle, Autovervollständigung von Befehlen, PDF-Viewer,
- Einfach zu verwenden und zu konfigurieren, ...

Außer dem Editor benötigt man eine \LaTeX -Distribution. In diesem Fall werden wir **MiKTeX** (für Windows-User) verwenden. Für Linux-User ist **TeXLive** eine sehr bekannte Alternative und für Mac OS-User **MacTeX** (s. 39, 49 und 36 im Abschnitt 14). *Zuerst* soll die Distribution (z. B. **MiKTeX**) und erst dann den Editor (z. B. **Texmaker**) installiert werden!

Hinweis: Die hier in den Kapiteln angegebenen Short-cuts beziehen sich auf **Texmaker**, sollten sie sich auf die Distribution beziehen, wird es explizit gesagt.

2.2.1 Texmaker-Konfiguration

Texmaker bietet eine sehr ausführliche und benutzerfreundliche Dokumentation, außerdem gibt es ein Benutzerhandbuch von **Texmaker** auf Deutsch. Letzteres ist zwar von der Version 3.1 der Software, für die meisten Fragen ist sie jedoch aktuell genug (s. 50 und 51 im Abschnitt 14).

Zunächst soll die Zeichenkodierung eingestellt werden. Dafür gehe zur Toolbar unter **Optionen/Texmaker konfigurieren** und klicke dann links auf den **Editor**-Button. Dort können u. a.

- die Schriftfamilie,
- Schriftgröße und
- Zeichenkodierung

des Editors eingestellt werden. D. h. wenn die Schrift beim Schreiben zu klein ist, oder wenn man sich eine andere Schriftart beim Schreiben wünscht, kann sie ebenso dort eingestellt werden. Bitte beachte, dass diese Einstellungen *nicht* das Outputdokument, sondern die Eingabe betreffen! Um die Schriftgröße oder -art des Outputs zu ändern s. Abschnitt 3.

Es wird empfohlen hier die Zeichenkodierung auf `utf-8` zu setzen, da wir später mit der Option `utf-8` vom Paket `inputenc` arbeiten werden. Es ist wichtig, dass die Zeichenkodierung des Editors mit der Zeichenkodierung in der Präambel des Dokuments übereinstimmt (s. . Abschnitt 3.1.2).

An der gleichen Stelle kann die Sprache der automatischen Rechtschreibüberprüfung eingestellt werden. Nochmal, es handelt sich um die Rechtschreibüberprüfung des Editors! `Texmaker`, ähnlich wie Word oder andere Texteditoren, überprüft während der Eingabe die Rechtschreibung. Falsch geschriebene (oder unbekannte) Wörter werden rot unterstrichen. Dafür muss jedoch die Sprache der Eingabe eingestellt werden. Klicke dafür auf den Ordner-Icon unter `Rechtschreibwörterbuch`, und wähle die passende Sprache:

- `de_DE.dic` für Deutsch,
- `en_GB.dic` für britisches Englisch,
- `en_US.dic` für amerikanisches Englisch, ...

Weitere Hilfestellungen zur Konfiguration werden in späteren Kapitel dieses Readers gegeben.

Hinweis: Um sich die Korrekturvorschläge eines falsch geschriebenen Worts anzeigen zu lassen, klicke auf das rot unterstrichene Wort mit der rechten Maustaste, anschließend wähle aus dem Menü den gewünschten Korrekturvorschlag und klicke darauf mit der linken Maustaste.

2.3 Befehle

`LATEX` kennt im Prinzip drei Arten von Befehlen:

- **Einfache Befehle:** Alle `LATEX` -Befehle beginnen mit einem Backslash „`\`“, gefolgt vom Namen des Befehls, von optionalen Argumenten in eckigen Klammern und von obligatorischen Argumenten in geschweiften Klammern.

```
\Befehlsname[optional]{obligatorisch}
\Befehlsname[opt1, opt2=Wert]{obl1}{obl2}
```

- **Umgebungen:** Umgebungen bestehen aus einem `begin`- und einem `end`-Befehl.

```
\begin{Umgebung}[optional]
...
\end{Umgebung}
```

- **Deklarationen:** Deklarationen verändern Parameter. Der Skopus von Deklarationen kann so definiert sein, dass er an bestimmten Grenzen – wie an einem Absatzschluss – endet, oder dass er nur auf einen von geschweiften Klammern bestimmten Skopus beschränkt ist.

```
\Deklaration ... [Skopusende]
{\Deklaration ...} außerhalb des Skopus
```

Hinweis: Sobald die ersten Buchstaben des Befehls eingegeben wurden, bietet **Texmaker** (per *Autovervollständigung*) an, den Befehl automatisch zu vervollständigen. Mit den Pfeiltasten kann der passende Befehl ausgewählt werden, anschließend wird die Enter-Taste gedrückt. Bei Umgebungen werden so **begin-** und **end-**Befehl eingegeben.

2.4 Zeichen und Sonderzeichen

Die folgenden Zeichen können problemlos verwendet werden:

```
a...z A...Z 0...9
. , : ; ? ! ` ' " ( ) [ ] + - * =
```

Achtet darauf, welche Art von Anführungszeichen durch ` ' " generiert werden. Mehr zur typographisch korrekten Verwendung von Anführungszeichen seht ihr im Abschnitt 10.

Die folgenden Zeichen haben in \LaTeX eine besondere Bedeutung und können nicht einfach im Fließtext verwendet werden.

```
# $ & ~ _ ^ { } < > | \ %
```

Um diese Zeichen verwenden zu können, musst man den in \LaTeX vordefinierten Funktionen dieser Zeichen *entkommen*. Bei einigen Zeichen kann man den vordefinierten Funktionen durch Voranstellen eines Backslashes „\“ entkommen.

```
\# \$ \% \_ \^ \{ \} \< \>
```

Dem *Backslash*, der *größer als*- und *kleiner als*-Zeichen, der Tilde, dem Zirkumflex und dem senkrechten Strich kann man nicht mit dem Backslash entkommen.

Da die Folge \\ für Zeilenumbrüche reserviert ist, kann man dem einfachen Backslash „\“ nicht mit Verwendung eines vorangestellten Backslashes entkommen. Dafür sollte der folgende Befehl benutzt werden:

```
\textbackslash
```

Die *größer als*- „>“ und *kleiner als*-Symbole „<“ können im Text durch die folgenden Befehle oder durch die Verwendung des Mathematikmodus', d. h. durch die Klammerung in $\$$ -Zeichen erzeugt werden (mehr zum Mathematikmodus im Abschnitt 9).

```
\textgreater $>$
\textless $<$
```

Ähnliches gilt für den senkrechten Strich (,pipe‘). Um ihn darzustellen, kann man entweder den Befehl `vert` oder den Strich in der Mathematikmodusklammerung eingeben oder den Befehl `textbar` außerhalb des Mathematikmodus’.

```
\vert$
|$
\textbar
```

Die Tilde „~“ hat in \LaTeX die Funktion eines geschützten Leerzeichens (s. Abschnitt 10). Um dieser Funktion zu entkommen kann man nicht den Backslash verwenden (`\~`), denn dadurch erscheint der folgende Buchstabe mit einer Tilde. So bei der Eingabe „`\~nicht`“, erscheint „ñicht“. Will man auch dieser Funktion entkommen, muss der folgende Befehl (ähnlich wie bei dem Backslash) benutzt werden:

```
\textasciitilde
```

Das gleiche Problem taucht beim Zirkumflex „^“ auf, welcher als Akzent z. B. im Französischen gebraucht wird. Daher erscheint bei der Eingabe „`s^ur`“ der folgende Output: „sûr“. Aus diesem Grund benötigt man den folgenden Befehl um den Zirkumflex als Output zu haben:

```
\textasciicircum
```

2.5 Leerzeichen und Zeilenumbrüche

- \LaTeX hat eine gesonderte Behandlung von Leerzeichen, die viele typographische Fehler automatisch korrigiert.
- Es macht keinen Unterschied zwischen einem Leerzeichen (,blank‘) oder einem Tabulator (,tab‘).
- Es zählt keine aufeinanderfolgenden Leerzeichen, d. h. mehrere konsekutive Leerzeichen werden nur als eins behandelt.
- Ein Leerzeichen zu Beginn einer Zeile wird einfach ignoriert.
- Ein Zeilenumbruch im Code wird als einzelnes Leerzeichen interpretiert.
- Eine *Leerzeile* (d. h. zwei Zeilenumbrüche hintereinander) legen das Ende eines Absatzes fest.
- Mehr Leerzeilen (oder Zeilenumbrüche) werden als *eine* einzelne Leerzeile interpretiert.

Im Abschnitt 10.1 werden verschiedene Arten von Leerzeichen behandelt. Hier ein Beispiel:

```
Hier ist der      Beispieltext mit viel
zu vielen Leerzeichen ,
die in Word immer zu sehen sind.
Und hier auch noch
ein Zeilenumbruch.
```

```
Erst hier, durch zwei Zeilenumbrüche, ein neuer Absatz.
```

```
Und hier auch, auch wenn hier mehr Umbrüche benutzt wurden.
```

```
Hier ist der Beispieltext mit viel zu vielen Leerzeichen , die in Word immer zu sehen
sind. Und hier auch noch ein Zeilenumbruch.
Erst hier, durch zwei Zeilenumbrüche, ein neuer Absatz.
Und hier auch, auch wenn hier mehr Umbrüche benutzt wurden.
```

2.6 (Aus-)Kommentieren

Das % ist das Zeichen um L^AT_EX-Code auszukommentieren, d. h. L^AT_EX wird den gesamten folgenden Text bis zum Zeilenumbruch *ignorieren*. Der Text nach dem Prozentzeichen wird weder interpretiert noch im Output wiedergegeben.

Kommentare sind sehr hilfreich beim Programmieren.

- Man kann auf einfacher Art und Weise den Code/Text verstecken, ohne ihn zu löschen.
- Fehler können durch Auskommentieren und Kompilieren leichter in Zeilen oder größeren Regionen gefunden werden.
- Man kann Leerzeichen oder Leerzeilen in langen Eingabezeilen unterbinden.
- Kommentare können im Code geschrieben werden, ohne dass sie als Text gedruckt werden.

```
Hier ist etwas Code, das angezeigt werden soll.
%und hier sind meine geheimen und wichtigen Notizen
Etwas Text und Kommentare, die sogar ein Wort teilen:
Rindfleischetikettierungs% wichtiges Zeugs
überwachungsaufgaben% mehr wichtiges Zeugs
übertragungsgesetz.
```

```
Hier ist etwas Code, das angezeigt werden soll. Etwas Text und Kommentare, die sogar
ein Wort teilen: Rindfleischetikettierungsüberwachungsaufgabenübertragungsgesetz.
```

Short-cuts: STRG+T: Markierten Text auskommentieren, STRG+U: Kommentare ausschalten

Hinweis: `Texmaker` markiert auskommentierten Text in grau, so dass es deutlich wird, ob Text aktiv ist oder nicht.

3 Dokumentstruktur

Ein `LaTeX`-Dokument besteht (zumindest) aus zwei Teilen

- **Die Präambel:**

In der Präambel werden alle globalen Eigenschaften des Dokuments definiert. Der notwendige Teil einer Präambel ist der `\documentclass{}`-Befehl.

Die Präambel endet mit dem `\begin{document}`-Befehl.

- **Der Body:**

Der Body beinhaltet den eigentlichen Text des Dokuments sowie lokale Definitionen.

Er beginnt mit dem `\documentclass{}`-Befehl und endet mit `\end{document}`. Alles, was diesem `end`-Befehl folgt, wird von `LaTeX` nicht interpretiert.

3.1 Präambel

3.1.1 Dokumentklasse

Der `\documentclass{}`-Befehl legt die Parameter des allgemeinen Dokument-Layouts fest. Die wichtigsten Klassen sind:

- `book` für Bücher
- `report` für längere Schriften mit etlichen Kapiteln, z. B. eine Dissertation
- `article` für Artikel, ohne Kapitel nur mit Abschnitten
- `letter` für Briefe

Da diese Klassen häufig für amerikanische Formate spezifiziert sind, gibt es Varianten dieser Klassen, die von `KOMA-Script` (s. 28 in Abschnitt 14) zur Verfügung gestellt werden: `scrbook`, `scrreprt`, `scrartcl` und `sclttr2`.

Zudem kann man die Optionen dieser Dokumentklassen in dem `documentclass`-Befehl festlegen, hier sind die häufigsten Optionen und das dazugehörige Minimalbeispiel:

- **Schriftgröße** für die (Default-)Schriftgröße: `10pt`, `11pt`, `12pt`
Default → `10pt`

- **Papierformat:** letterpaper, a4paper

Default → letterpaper

In den KOMA-Script-Klassen sollte man paper=a4 oder paper=letter statt a4paper bzw. letterpaper verwenden.

```
\documentclass[10pt,a4paper]{scrartcl}
\begin{document}
Text Text Text
\end{document}
```

Hinweis: Texmaker bietet einen Assistenten, der die Präambel für euch schreibt. Schaut in der Toolbar unter **Assistent/Assistent für ein neues Dokument** nach. Dort könnt ihr alles weitere einstellen.

3.1.2 Pakete einbinden

Die Breite an Funktionen, zu denen man mit L^AT_EX Zugang hat, ist beschränkt. Um das erwünschte Layout mit den Extra-Features passend zu den eigenen Bedürfnissen zu verwenden, müssen zusätzliche Pakete geladen werden.

Die Pakete müsse in der Präambel mit dem folgenden Befehl geladen werden :

```
\usepackage[parameter1, parameter2]{packagename}
```

Normalerweise sind bereits viele der benötigten Pakete in der L^AT_EX-Distribution (z. B. MiKTeX) vorinstalliert. (Fast) Alle anderen Pakete mit den entsprechenden Benutzerhandbüchern können kostenfrei aus der Webseite von CTAN – The Comprehensive T_EX Archive Network (www.ctan.org) heruntergeladen werden.

Die folgenden Pakete werden am häufigsten benötigt:

- **babel** stellt sprachspezifische Funktionen (Überschriften der Gliederung, Silbentrennung, Zeichen-Support, Datum, usw.) für *andere* Sprachen als Englisch oder für *zusätzliche* Sprachen bereit. Die letzte Sprache bestimmt die Haupteigenschaften des Dokuments.

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

- **inputenc** definiert die Kodierung der eingegebenen Charaktere, die sich von `ascii` unterscheiden, wie z. B. `macce` oder `applemac` (Mac), `latin1` (Linux), `ansinew` (Win)

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

- **fontenc** spezifiziert die Schriftart, die im Output-Dokument benutzt wird (Lateinisch, Kyrillisch, usw.). Wird nicht nicht richtige Schriftart spezifiziert, könnte L^AT_EX Probleme bei der Silbentrennung in Wörtern mit diakritischen Zeichen aufweisen.

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

- `lmodern` aktiviert die Schriftart *Latin Modern*. Sie ist eine Abwandlung der T_EX-Standard-Schriftart *Computer Modern*, die für den Gebrauch in PDF-Dokumenten optimiert wurde.

```
\usepackage{lmodern}
```

Hinweis: Wenn ein Paket nicht in der MiKTeX-Distribution vorinstalliert ist, erhält man eine automatische Anfrage, ob das Paket heruntergeladen und installiert werden soll. Sollte dies nicht funktionieren reicht häufig, die `.sty`-Datei² des entsprechenden Pakets aus dem Internet herunterzuladen und *lokal* (d. h. im selben Ordner wie die aktuelle `.tex`-Datei) zu speichern.

3.2 Body

3.2.1 Metadaten

Zu Beginn des Dokuments können bestimmte Metadaten spezifiziert werden, so zum Beispiel:

- `\author{Vorname1 Nachname1 \and Vorname2 Nachname2}`
- `\title{Dokumenttitel}`
- `\subtitle{Untertitel}`
- `\date{23. Februar 2015}` oder `\date{\today}` oder `\date{}`
Default → `\date{\today}`

Mit dem Befehl `\maketitle` im Beginn des Dokuments werden diese Informationen im Dokument wiedergegeben.

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

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

\author{Constantin Freitag \and Antonio Machicao y Priemer}
\title{\LaTeX\ für Linguisten}
\subtitle{Eine Einführung}
\date{23. Februar 2015}

\begin{document}
```

²Die `.sty`-Datei enthält den Code des entsprechenden Pakets, sie ist die *Style*-Datei.

```
\maketitle

Text Text Text

\end{document}
```

Hinweis: Die Metadaten können auch in der Präambel eingegeben werden. Der Befehl `\maketitle` muss jedoch *immer* zu Beginn des Bodys eingegeben werden. `Texmaker` bietet zudem die Autovervollständigung der Befehle an.

3.2.2 Textauszeichnung

\LaTeX bietet verschiedene Befehle zur Textauszeichnung:

```
\textbf{bold}
\textit{italics}
\textsl{slanted}
\emph{emphasized}
\underline{underline}
\texttt{typewriter}
\textsc{small caps}
ex\textsuperscript{up}
```

bold
<i>italics</i>
<i>slanted</i>
<i>emphasized</i>
<u>underline</u>
typewriter
SMALL CAPS
ex ^{up}

Das Paket `ulem` bietet einen Befehl `\sout{strike out}` ~~strike out~~ an, aber achtet darauf, das Paket mit der Option `normalem` zu laden (`\usepackage[normalem]{ulem}`), andernfalls bewirkt der Befehl `\emph{emphasized text}` das Folgende: emphasized text.

Short-cuts: Um die Short-cuts für Textauszeichnung in `Texmaker` zu sehen, geht bitte in der Toolbar zu `LaTeX/Schriftart`. Außerdem findet ihr eine seitliche Toolbar mit einigen dieser Befehle.

\LaTeX bietet auch die Möglichkeit an, die Schriftgröße³ zu ändern. Dies ist jedoch in wissenschaftlichen Arbeiten nicht empfehlenswert.

```
{\tiny tiny}
{\scriptsize scsize}
{\footnotesize fsize}
{\small small}
{\normalsize normal}
{\large large}
{\Large Large}
{\LARGE LARGE}
```

```
{\huge huge}
{\Huge Huge}
```

³Die Schriftgrößenbefehle können entweder als *Deklarationen* wie auch als *Umgebungen* angegeben werden, s. Abschnitt 2.3.

tiny
scriptsize
footnotesize
small
normalsize
large
Large
LARGE
huge
Huge

Hinweis: In der seitlichen Toolbar von **Texmaker** können die Schriftgrößen angeklickt werden. Diese erscheinen in Form von Umgebungen.

3.2.3 Gliederung

Es wurde bereits gezeigt, wie man neue Absätze generiert (zwei Zeilenumbrüche hintereinander). Weitere Befehle für die lokale Strukturierung des Textes sind:

- `\par` beendet einen Absatz.
- `\newline` und `\\` bewirken einen Zeilenumbruch im Outputtext ohne den Absatz zu beenden.
- `\noindent` verhindert die Einrückung nach einem Zeilenumbruch, der jedoch nicht als neuer Absatz markiert werden soll.

Zudem bietet \LaTeX die folgenden Gliederungsbefehle:

- `\part [Kurztitel] {Titel}`
- `\chapter [Kurztitel] {Titel}` (nur in `book` und `report`, bzw. in `scrbook` und `scrreprt`)
- `\section [Kurztitel] {Titel}`
- `\subsection [Kurztitel] {Titel}`
- `\subsubsection [Kurztitel] {Titel}`
- `\paragraph [Kurztitel] {Titel}`
- `\subparagraph [Kurztitel] {Titel}`

Der in der Option `Kurztitel` eingegebene Text wird im Inhaltsverzeichnis und in den Kopfzeilen erscheinen.

Hinweis: Hat man eine Stelle im PDF-Dokument, die man bearbeiten will, oder will man sehen, wie eine Stelle des `.tex`-Dokuments im PDF-Ergebnis aussieht, geht das mit `Texmaker` problemlos. Man muss nur mit der rechten Maustaste auf die Stelle im PDF-Viewer bzw. im `.tex`-Quelltext klicken, und dann `Zur Zeile im Quelltext springen` bzw. `Zum PDF wechseln` auswählen.

3.2.4 Inhaltsverzeichnis

Ein Inhaltsverzeichnis wird durch den folgenden Befehl generiert:

```
\tableofcontents
```

Dieser Befehl muss an der Stelle im Dokument angegeben werden, an der das Inhaltsverzeichnis erscheinen soll. Die Ebenen, die im Inhaltsverzeichnis erscheinen, und die Seitenzahlen werden automatisch von `LATEX` aus der Gliederung des Dokuments (s. Abschnitt 3.2.3) entnommen (z. B. `\section[Kurztitel]{Titel}`).

Um die Tiefe im Inhaltsverzeichnis zu bestimmen, kann der folgende Befehl angegeben werden:

```
\setcounter{tocdepth}{Tiefe als Zahlwert}
```

Damit können weitere Ebenen (z. B. `subsection`) in das Inhaltsverzeichnis hinzugefügt zu werden.

Hinweis: Die Aktualisierung des Inhaltsverzeichnisses benötigt häufig, dass das Dokument zweimal kompiliert wird. `Texmaker` bietet einen Button namens `Schnelles Übersetzen` an, mit dem der Kompilierungsvorgang eingestellt werden kann. Geht zur Toolbar, geht dort zum Bereich `Optionen/Texmaker konfigurieren/`. Klickt dann auf `Schnelles Übersetzen`, wählt dort unter `Befehl für schnelles Übersetzen` die Option `PDFLaTeX + Bib(la)Tex + PDFLaTeX (x2) + PDF anzeigen` an und klickt auf `OK`. Damit habt ihr den Kompilierungsvorgang für `Schnelles Übersetzen` eingerichtet.

3.2.5 Fußnoten

Um eine Fußnote zu generieren, muss nur der folgende Befehl an der Stelle, an der der Fußnotenindex erscheinen soll, angegeben werden:

```
\footnote{Inhalt der Fußnote}
```

Hier ein Beispiel zur Verwendung von diesem Befehl:

```
Hier kommt etwas Text und hier eine Fußnote\footnote{Das  
ist keine Literaturangabe, sondern ein weiterer  
kluger \emph{Einfall}.} in einer tollen Hausarbeit.
```

Hier kommt etwas Text und hier eine Fußnote⁴ in einer tollen Hausarbeit.

Das Paket `footnote` stellt weitere interessante Optionen für Fußnoten bereit.

⁴Das ist keine Literaturangabe, sondern ein weiterer kluger *Einfall*.

3.3 Übungen

1. Generiere ein \LaTeX -Dokument mit **Texmaker** und verwende die bisher erklärten Pakete und Befehle.
2. Suche nach einer wissenschaftlichen Arbeit, die du geschrieben hast. Speichere diese Arbeit als PDF und während des Kurses, beginne diese Datei in \LaTeX zu übersetzen. Vergleiche am Ende die beiden Ergebnisse und genieße es!

4 Textumgebungen

\LaTeX bietet mehrere Textumgebungen an. Die bekanntesten sind Zitate, Listen, „wörtliche Wiedergaben“ und Abstracts.

4.1 Zitate

Wörtliche Zitate, die länger als zwei Zeilen lang sind, sollen vom Fließtext getrennt werden. \LaTeX stellt dafür zwei Umgebungen zur Verfügung: `quote` und `quotation`. Beide Umgebungen sind rechts und links eingerückt. Der Unterschied zwischen den beiden betrifft die Absatzgrenzen. `quote` trennt die Absätze mit vertikalem Abstand, während `quotation` die erste Zeile des Absatzes einrückt.

Quote-Umgebung

Das ist der Text vor der `\texttt{quote}`-Umgebung.

```
\begin{quote}
```

```
Die grammatischen Phänomene in einer Sprache zerfallen  
in zwei Teilbereiche:\newline
```

```
kerngrammatische Phänomene und randgrammatische  
Phänomene (\emph{Ausnahmen}).\par
```

```
Kerngrammatische Phänomene werden von einer  
Kerngrammatik der Sprache mit relativ einfachen theoretischen  
Mitteln erfasst. \citep{Nolda&Co14a}
```

```
\end{quote}
```

Das ist der Text nach der `\texttt{quote}`-Umgebung.

Das ist der Text vor der `quote`-Umgebung.

Die grammatischen Phänomene in einer Sprache zerfallen in zwei Teilbereiche:

kerngrammatische Phänomene und randgrammatische Phänomene (*Ausnahmen*).

Kerngrammatische Phänomene werden von einer Kerngrammatik der Sprache mit relativ einfachen theoretischen Mitteln erfasst. (Nolda et al., 2014)

Das ist der Text nach der `quote`-Umgebung.

Quotation-Umgebung

Das ist der Text vor der `\texttt{quotation}`-Umgebung.

```
\begin{quotation}
```

```
Die grammatischen Phänomene in einer Sprache zerfallen  
in zwei Teilbereiche:\newline
```

```
kerngrammatische Phänomene und randgrammatische  
Phänomene (\emph{Ausnahmen}).\par
```

```
Kerngrammatische Phänomene werden von einer  
Kerngrammatik der Sprache mit relativ einfachen theoretischen  
Mitteln erfasst. \citep{Nolda&Co14a}
```

```
\end{quotation}
```

Das ist der Text nach der `\texttt{quotation}`-Umgebung.

Das ist der Text vor der `quotation`-Umgebung.

Die grammatischen Phänomene in einer Sprache zerfallen in zwei Teilbereiche:

kerngrammatische Phänomene und randgrammatische Phänomene (*Ausnahmen*).

Kerngrammatische Phänomene werden von einer Kerngrammatik der Sprache mit relativ einfachen theoretischen Mitteln erfasst. (Nolda et al., 2014)

Das ist der Text nach der `quotation`-Umgebung.

4.2 Listenumgebungen

L^AT_EX hat drei vordefinierte Listenumgebungen:

- `itemize`,
- `enumerate`,
- `description`,

und eine allgemeine Listenumgebung:

- `list`.

Jeder einzelne Eintrag in Listen beginnt mit `\item`.

4.2.1 Itemize

Die `itemize`-Umgebung wird für ungeordnete Listen verwendet.

```
\begin{itemize}
\item Erster Punkt
\item Zweiter Punkt
  \begin{itemize}
  \item Erster Unterpunkt
  \item Zweiter Unterpunkt
  \end{itemize}
\item Dritter Punkt
\end{itemize}
```

- Erster Punkt
- Zweiter Punkt
 - Erster Unterpunkt
 - Zweiter Unterpunkt
- Dritter Punkt

4.2.2 Enumerate

Nummerierte Listen werden mit der `enumerate`-Umgebung erzielt.

```
\begin{enumerate}
\item Erster Punkt
\item Zweiter Punkt
  \begin{enumerate}
  \item Erster Unterpunkt
  \item Zweiter Unterpunkt
  \end{enumerate}
\item Dritter Punkt
\end{enumerate}
```

1. Erster Punkt
2. Zweiter Punkt
 - a) Erster Unterpunkt
 - b) Zweiter Unterpunkt
3. Dritter Punkt

4.2.3 Description

Die `description`-Umgebung generiert Listen von Begriffen mit den entsprechenden Beschreibungen.

```
\begin{description}
\item[Begriff:] entsprechende Beschreibung
  \begin{description}
  \item[Unterbegriff:] entsprechend eingebettete Beschreibung
  \end{description}
\item[Begriff:] entsprechende sehr sehr sehr lange, nicht
enden wollende, fast unendliche Beschreibung.\
```

```
Weiterführende Beschreibung nach einem Absatz.  
\end{description}
```

Begriff: entsprechende Beschreibung

Unterbegriff: entsprechend eingebettete Beschreibung

Begriff: entsprechende sehr sehr sehr lange, nicht enden wollende, fast unendliche Beschreibung.

Weiterführende Beschreibung nach einem Absatz.

Bitte beachtet im Abschnitt 3.2.3 den Unterschied zwischen den Befehlen `\` und `\par`.

4.2.4 Kombinierte Listen

Listen können auch rekursiv in einander eingebettet werden.

Die entsprechenden Aufzählungszeichen werden relativ zur aktuellen Umgebung automatisch ausgewählt. Das heißt, die erste Ebene der `enumerate`-Umgebung erscheint mit arabischen Ziffern, auch wenn sie in der `itemize`-Umgebung eingebettet ist, die wiederum in der `description`-Umgebung eingebettet ist.

```
\begin{description}  
\item[Linguistik:] eine wissenschaftliche Disziplin  
  \begin{itemize}  
    \item Sie hat etwas mit Sprache zu tun.  
    \item Sie interagiert mit anderen Disziplinen:  
      \begin{enumerate}  
        \item Philosophie  
        \item Psychologie  
        \item Soziologie  
      \end{enumerate}  
    \end{itemize}  
  \end{description}
```

Linguistik: eine wissenschaftliche Disziplin

- Sie hat etwas mit Sprache zu tun.
- Sie interagiert mit anderen Disziplinen:
 1. Philosophie
 2. Psychologie
 3. Soziologie

4.2.5 Individualisierte Anpassung

Einzelne Aufzählungszeichen können mittels eines eingegebenen optionalen Parameters durch gewünschte Zeichen ersetzt werden. Die Syntax dafür ist ähnlich wie bei der `description`-Umgebung.

```
\begin{itemize}
\item Standardzeichen
\item[+] Individualisiert
\end{itemize}
```

- Standardzeichen
- + Individualisiert

```
\begin{enumerate}
\item Standardzeichen
\item[a)] Individualisiert
\item Standardzeichen
\end{enumerate}
```

1. Standardzeichen
- a) Individualisiert
2. Standardzeichen

Auch wenn die Möglichkeit besteht, Aufzählungszeichen für einzelne Listenelemente manuell zu ändern, sollte dies eher die Ausnahme sein. Normalerweise werden solche Änderungen global (d. h. für das gesamte Dokument) vorgenommen.

4.2.6 Allgemeine Listenumgebung

\LaTeX stellt ebenso eine allgemeine Listenumgebung zur Verfügung, die nach den Bedürfnissen des Benutzers gestaltet werden kann. Die Syntax für diese selbstdefinierte Liste lautet:

```
\begin{list}{Aufzählungszeichen}{Listenname}
\item Inhalt
\end{list}
```

Außer in den sehr seltenen Fällen, in denen ein Listentyp ein einziges Mal in einem Dokument verwendet werden soll, wird die `list`-Umgebung normalerweise als „Variable“ für einen neuen Listentyp im gesamten Dokument definiert.

4.2.7 Pakete für Aufzählungen

Mit der Verwendung von weiteren Paketen können die Funktionen in Listenumgebungen erweitert werden.

- Das Paket `enumerate` eröffnet die Möglichkeit den Stil des Zählers in der `enumerate`-Umgebung zu verändern.

- Das Paket `mdwlist` bietet eine Variante der Listenbefehle mit Asterisk (z. B. `itemize*`), welche einen reduzierten vertikalen Abstand zwischen Listenitems ausweist. Außerdem hat dieses Paket die Möglichkeit die Nummerierung in der `enumerate`-Umgebung auszusetzen, und wieder aufzunehmen.
- Das Paket `paralist` erlaubt die Aufzählungszeichen für die ganze Liste als optionales Argument auszuwählen. Es bietet zudem die Umgebung `compactitem` mit einem reduzierten vertikalen Abstand zwischen Listenitems und zwischen der Liste und dem Text davor und danach. Darüber hinaus gibt es damit die Möglichkeit die Befehle `inpara`, `inparaitem`, `inparaenum` und `inparadesc` zu verwenden, welche Listenpunkte in den Fließtext einbauen lässt.
- Das Paket `enumitem` erlaubt es – in einer vergleichsweise einfachen Form – Listen umzudefinieren und daher alle vorher angesprochenen Eigenschaften selbst einzubauen.

Short-cut: Mit STRG+UMSCHALT+I erscheint in `Texmaker` automatisch ein neuer `\item`-Befehl.

4.3 Theoreme

`LATEX` hat eine installierte Umgebung für nummerierte Theoreme. Zunächst muss die Umgebung definiert werden, die Syntax sieht wie folgt aus:

```
\newtheorem{id}{Name}
```

Hier kommt das konkrete Beispiel:

```
\newtheorem{bsp}{Beispiel}
```

Anschließend kann die neu definierte `bsp`-Umgebung verwendet werden. Es ist empfehlenswert solche Definitionen, wenn sie *global* verwendet werden sollen, in der Präambel zu definieren (s. Abschnitt 3).

```
\begin{bsp}
Colorless green ideas sleep furiously.
\end{bsp}
```

Beispiel 1. *Colorless green ideas sleep furiously.*

Außerdem kann eine weitere Beschreibung als optionales Argument hinzugefügt werden.

```
\begin{bsp}[Revidiert]
Colorless \emph{red} ideas sleep furiously.
\end{bsp}
```

Beispiel 2 (Revidiert). *Colorless red ideas sleep furiously.*

Der Zähler kann ebenso manipuliert werden, so dass er die aktuelle `section` (oder `subsection`) als Präfix verwendet. Dafür muss bloß in der Definition die Option `section` hinzugefügt werden.

```
\newtheorem{defin}{Definition}[section]
```

```
\begin{defin}
```

```
Ein Morphem ist die kleinste bedeutungstragende Einheit.
```

```
\end{defin}
```

Definition 4.1. *Ein Morphem ist die kleinste bedeutungstragende Einheit.*

Damit Theoreme den gleichen Zähler benutzen, sollte die `id` des zu teilenden Zählers als erster optionaler Parameter hinzugefügt werden.

```
\newtheorem{ax}[defin]{Axiom}
```

```
\begin{ax}
```

```
Erstes Axiom, welches den Zähler mit \texttt{defin} teilt,  
und daher die \texttt{section} im Zähler enthält.
```

```
\end{ax}
```

Axiom 4.2. *Erstes Axiom, welches den Zähler mit `defin` teilt, und daher die `section` im Zähler enthält.*

4.3.1 Pakete für Theoreme

Das `theorem`-Paket bietet weitere Möglichkeiten um Theoremumgebungen auf die eigenen Bedürfnisse anzupassen.

4.4 Wörtliche Wiedergabe

Die `verbatim`-Umgebung wird verwendet, um den Input-Text (inklusive Leerzeichen) als Output auszugeben. Sie wird häufig verwendet um Programmiercode wiederzugeben. Als Default-Einstellung gibt die `verbatim`-Umgebung den Input-Text in Schreibmaschinenschrift als Output wieder.

Die Silbentrennung und der Zeilenumbruch im Output-Text sind in der `verbatim`-Umgebung nicht aktiv!

```
\begin{verbatim}
```

```
Das ist Code mit vielen, unnötigen Leerzeichen auf
```

```
einer viel zu langen Zeile, die nicht mehr auf  
dem Blatt passt.  
\end{verbatim}
```

Das ist Code mit vielen, unnötigen Leerzeichen auf einer viel zu langen Zeile, die

Es gibt auch eine *inline*-Variante der `verbatim`-Umgebung: `\verb`. Der Input-Text wird durch zwei Zeichen umklammert. Es ist nicht wichtig, *welche* Zeichen dafür ausgewählt werden. Wichtig ist nur, dass sie nicht in dem einzugebenden Code vorkommen.

```
Etwas Text mit Code \verb|\code*+##| darin
```

Etwas Text mit Code `\code*+##` darin

4.4.1 Probleme mit `verbatim`

- `verbatim` ist eine sehr anfällige Umgebung und kann zu Problemen in einigen Umgebungen wie Fußnoten, sections, usw. führen.
- Da Input-Text in `verbatim` von $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ nicht interpretiert wird, gibt es keine Unterstützung für `utf-8`-Zeichen.

4.4.2 Pakete für wörtliche Wiedergaben

Das `listings`-Paket bietet eine mächtige Alternative zur `verbatim`-Umgebung. Sie kann sowohl in der Umgebungsvariante als auch in der *inline*-Variante benutzt werden.

4.5 Abstract

Die `abstract`-Umgebung generiert ein Abstract in Dokumentklassen wie `scrartcl`. Der Text wird in einer einzigen Spalte erscheinen, auch wenn der Artikeltext in zwei Spalten geteilt wird. Der Titel „Abstract“ erscheint, wenn die Option `abstracton` zum `documentclass`-Befehl hinzugefügt wurde (`\documentclass[abstracton]{scrartcl}`).

```
\begin{abstract}  
Ein Abstract ist eine kurze Zusammenfassung über den Inhalt  
der Arbeit. Das Abstract wird immer am Anfang des Dokuments  
positioniert.\par  
Es ist auch möglich das Abstract in mehrere Absätze  
zu teilen.  
\end{abstract}
```

Zusammenfassung

Ein Abstract ist eine kurze Zusammenfassung über den Inhalt der Arbeit. Das Abstract wird immer am Anfang des Dokuments positioniert.
Es ist auch möglich das Abstract in mehrere Absätze zu teilen.

4.6 Vers

L^AT_EX stellt ebenso eine *verse*-Umgebung für die Dichter unter uns.

```
\begin{verse}
Eine kurze Zeile\\
und eine weitere, die jedoch leider lang, viel zu lang geraten
sogar so lang, dass sie gebrochen werden muss.

Noch ein Vers\\
ein Zweizeilenvers.
\end{verse}
```

Eine kurze Zeile
und eine weitere, die jedoch leider lang, viel zu lang geraten sogar so lang,
dass sie gebrochen werden muss.

Noch ein Vers
ein Zweizeilenvers.

4.7 Flattersatz

Um Text zentriert, rechts- oder linksbündig anstatt im Blocksatz anzuzeigen, können Umgebungen verwendet werden. Das folgende Beispiel zeigt die Umgebung zum zentrieren:

```
\begin{center}
Zentrierter Text
\end{center}
```

Zentrierter Text

Innerhalb einer Umgebung kann ebenso der folgende Befehl benutzt werden **\centering**

```
Das ist Text im Blocksatz oberhalb des Zitats.  
\begin{quote}  
\centering Zentrierter Text in der \texttt{quote}-Umgebung  
\end{quote}  
Das ist Text im Blocksatz unterhalb des Zitats.
```

Das ist Text im Blocksatz oberhalb des Zitats.

Zentrierter Text in der quote-Umgebung

Das ist Text im Blocksatz unterhalb des Zitats.

Um nur eine Zeile zu zentrieren, kann der folgende Befehl verwendet werden: `\centerline{}`. Im Vergleich zur `center`-Umgebung wird dabei kein weiterer vertikaler Abstand vor oder nach der zentrierten Zeile generiert.

```
Das ist Text im Blocksatz oberhalb der zentrierten Zeile.  
  
\centerline{Zentrierte Zeile ohne vertikalen Abstand}  
  
Das ist Text im Blocksatz unterhalb der zentrierten Zeile.
```

Das ist Text im Blocksatz oberhalb der zentrierten Zeile.

Zentrierte Zeile ohne vertikalen Abstand

Das ist Text im Blocksatz unterhalb der zentrierten Zeile.

4.7.1 Linksbündiger Text

Analog zum Zentrieren können die folgenden Befehle bzw. Umgebungen für linksbündigen Text benutzt werden:

```
Das ist Text im Blocksatz oberhalb des linksbündigen Textes.  
\begin{flushleft}  
Das ist linksbündiger Text, bei dem es keine Silbentrennung  
gibt, aber einen Zeilenumbruch.  
\end{flushleft}  
Das ist Text im Blocksatz unterhalb des linksbündigen Textes.
```

Das ist Text im Blocksatz oberhalb des linksbündigen Textes.

Das ist linksbündiger Text, bei dem es keine Silbentrennung gibt, aber einen Zeilenumbruch.

Das ist Text im Blocksatz unterhalb des linksbündigen Textes.

```
Das ist Text im Blocksatz über der \texttt{quote}-Umgebung.
\begin{quote}
\raggedright
Das ist linksbündiger Text in einer Umgebung, bei dem es
keine Silbentrennung gibt, aber einen Zeilenumbruch.
\end{quote}
Das ist Text im Blocksatz unter der \texttt{quote}-Umgebung.
```

Das ist Text im Blocksatz über der `quote`-Umgebung.

Das ist linksbündiger Text in einer Umgebung, bei dem es keine Silbentrennung gibt, aber einen Zeilenumbruch.

Das ist Text im Blocksatz unter der `quote`-Umgebung.

```
Das ist Text im Blocksatz oberhalb der linksbündigen Zeile.

\leftline{Linksbündige Zeile ohne vertikalen Abstand}

Das ist Text im Blocksatz unterhalb der linksbündigen Zeile.
```

Das ist Text im Blocksatz oberhalb der linksbündigen Zeile.

Linksbündige Zeile ohne vertikalen Abstand

Das ist Text im Blocksatz unterhalb der linksbündigen Zeile.

4.7.2 Rechtsbündiger Text

Analog zum linksbündigen Text können die folgenden Befehle bzw. Umgebungen für rechtsbündigen Text benutzt werden:

```
Das ist Text im Blocksatz oberhalb des rechtssbündigen Textes.
\begin{flushright}
Das ist rechtsbündiger Text, bei dem es keine Silbentrennung
gibt, aber einen Zeilenumbruch.
\end{flushright}
Das ist Text im Blocksatz unterhalb des rechtsbündigen Textes.
```

Das ist Text im Blocksatz oberhalb des rechtssbündigen Textes.

Das ist rechtsbündiger Text, bei dem es keine Silbentrennung gibt, aber einen
Zeilenumbruch.

Das ist Text im Blocksatz unterhalb des rechtsbündigen Textes.

```

Das ist Text im Blocksatz über der \texttt{quote}-Umgebung.
\begin{quote}
\raggedleft
Das ist rechtsbündiger Text in einer Umgebung, bei dem es
keine Silbentrennung gibt, aber einen Zeilenumbruch.
\end{quote}
Das ist Text im Blocksatz unter der \texttt{quote}-Umgebung.

```

Das ist Text im Blocksatz über der `quote`-Umgebung.

Das ist rechtsbündiger Text in einer Umgebung, bei dem es keine
Silbentrennung gibt, aber einen Zeilenumbruch.

Das ist Text im Blocksatz unter der `quote`-Umgebung.

```

Das ist Text im Blocksatz oberhalb der rechtsbündigen Zeile.

\rightline{Rechtsbündige Zeile ohne vertikalen Abstand}

Das ist Text im Blocksatz unterhalb der rechtsbündigen Zeile.

```

Das ist Text im Blocksatz oberhalb der rechtsbündigen Zeile.

Rechtsbündige Zeile ohne vertikalen Abstand

Das ist Text im Blocksatz unterhalb der rechtsbündigen Zeile.

Hinweis: Wenn man sich mit dem Cursor am Beginn oder Ende einer Umgebung befindet, markiert `Texmaker` die gesamte Umgebung. Dies ist sehr hilfreich um herauszufinden, ob man einen `begin`- oder `end`-Befehl vergessen hat.

4.8 Übungen

1. Öffne die Datei `LaTeX_text_environments.tex` und bearbeite den Inhalt in den entsprechenden Bereichen mit den Umgebungen und Befehlen, die in diesem Abschnitt gezeigt wurden.
2. Nimm dein Übungsdokument und bearbeite alle Stellen, die mit den hier gezeigten Befehlen und Umgebungen modifiziert werden können.

5 Nicht-textbezogene Elemente

5.1 Grafiken

\LaTeX erlaubt sowohl das Einfügen von externen Grafiken, als auch das Generieren eigener Grafiken. In diesem Kurs werden wir uns nur mit dem Einfügen externer Grafiken

befassen.

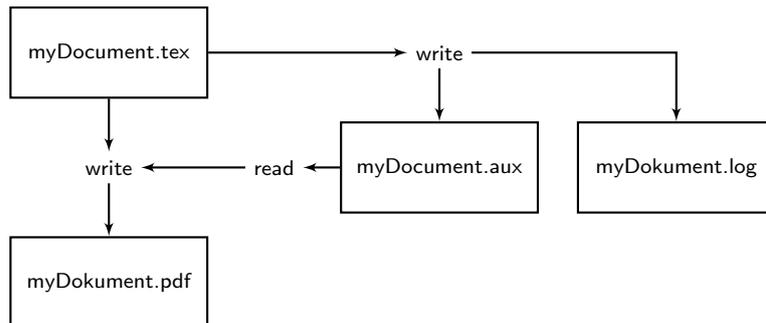
Um Grafiken einzufügen, muss das Paket `graphicx` in der Präambel mit dem folgenden Befehl geladen werden: `\usepackage{graphicx}`

Anschließend können mit dem Befehl `\includegraphics` und der folgenden Syntax Grafiken in das Dokument eingefügt werden:

```
\includegraphics [Größe] {Pfad/Dateiname}
```

Ein konkretes Beispiel:

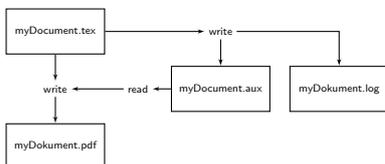
```
\includegraphics{LaTeX_flowchart_1.pdf}
```



5.1.1 Größe, Darstellung und Formate

Die Größe der Grafik im Dokument kann *relativ* zur Originalgröße der Grafik spezifiziert werden, wie in dem folgenden Beispiel:

```
\includegraphics [scale=0.5] {LaTeX_flowchart_1.pdf}
```



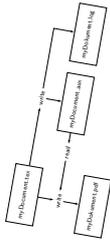
Die Größenangabe `scale=0.5` meint, dass die Größe der Grafik im Dokument 50 % von der Originalgröße betragen soll.

Die Grafiken können auch mit *absoluten* Größenangaben (s. `width` und `height`) geladen werden:

```
\includegraphics [width=10cm] {LaTeX_flowchart_1.pdf}
\includegraphics [width=\linewidth] {LaTeX_flowchart_1.pdf}
\includegraphics [height=10cm] {LaTeX_flowchart_1.pdf}
```

Grafiken können auch mit Hilfe des `angle`-Parameters gedreht werden, wobei die Zahl den Drehungsgrad angibt.

```
\includegraphics [angle=80, scale=0.3] {
  graphics/LaTeX_flowchart_1.pdf}
```



Das Paket `graphicx` stellt ebenso verschiedene Arten der Bilddarstellung mit Befehlen wie z. B. `bb` (Zeichen-Box), `viewport` (Darstellungsfeld), `trim` (Bildbeschneidung) und `clip` (Befestigen einer Grafik an eine Zeichen-Box) zur Verfügung. Außerdem kann auch *Text* mit den Befehlen `scalebox` bzw. `rotatebox` skaliert bzw. gedreht werden. Um mehr dazu zu erfahren schaut im `graphicx`-Handbuch (s. Punkt 24 im Abschnitt 14) nach. Um Grafiken übereinander zu legen kann das Paket `overpic` verwendet werden.

Die folgenden Formate werden bei der Kompilierung unterstützt:⁵

- `.pdf`, Format für Vektorgrafiken
- `.png`, Format für Rastergrafiken
- `.jpg`, Format für Rastergrafiken
- `.eps`, Format für Vektorgrafiken (nur mit dem `epstopdf`-Paket benutzbar)

5.1.2 Grafikpfad

Der Befehl `includegraphics` nimmt den Dateinamen mit dem Pfad zum aktuellen Ort der Grafikdatei. Wenn sich die Datei im gleichen Ordner wie die `.tex`-Datei befindet, muss *nur* der Dateiname, andernfalls muss zusätzlich der Pfad angegeben werden. Der Pfad zur Grafik muss *relativ* zur `.tex`-Datei angegeben werden.

Wenn alle Grafiken in einem Ordner gesammelt werden (z. B. `graphics`), dann muss der Pfad zu diesem Ordner präzisiert werden.

```
\includegraphics[scale=0.5]{graphics/LaTeX_flowchart_1.pdf}
```

Ist die Grafik außerhalb des Ordners, in dem sich die `.tex`-Datei befindet, dann kann man eine Ebene höher in der Ordnerstruktur mit dem Präfix `../` gelangen.

```
\includegraphics[scale=0.5]{../LaTeX_flowchart_1.pdf}
```

Das `graphicx`-Paket bietet mit dem Befehl `\graphicspath{Ordnerliste}` die Möglichkeit an, den Pfad für alle Grafiken global (in der Präambel) festzulegen, wie in dem folgenden Beispiel:

```
\graphicspath{graphics/}
\graphicspath{{graphics/}{pictures/}}
```

⁵Wir nehmen an, dass ihr mit PDFLaTeX kompiliert. Diese ist die Einstellung, die wir beim Kompilieren in Texmaker empfohlen haben (s. Hinweis im Abschnitt 3.2.4). Wenn ihr einen CVI-Kompilierer wie LaTeX verwendet, solltet ihr nur PostScript-Formate verwenden (`.ps`, `.eps`).

Wenn der Pfad bereits global festgelegt wurde, kann die Grafik einfach mit dem Befehl `includegraphics` und dem Namen der Grafik eingefügt werden.

5.1.3 Paket zur Erstellung von Grafiken

Um Grafiken in \LaTeX selber zu erstellen, ist das Paket `pgf` die mächtigste Alternative. Es kommt mit einer benutzerfreundlichen `TikZ`-Syntax ausgestattet ist. Schaut im `pgf`-Handbuch nach für weitere Informationen (s. 43 im Abschnitt 14).

5.2 Tabellen

Im Grunde ist die Erstellung von Tabellen in \LaTeX sehr einfach, wenn auch zunächst etwas gewöhnungsbedürftig. Die Umgebung für Tabellen heißt `tabular` und nimmt ein optionales und ein obligatorisches Argument.

```
\begin{tabular}[Position]{Layout}
...
\end{tabular}
```

Die Option *Position* kann die Werte `t` (top), `c` (center), oder `b` (bottom) annehmen. Diese Positionswerte geben die vertikale Positionierung der gesamten Tabelle in Bezug zur aktuellen Zeile (zur zuletzt geschriebenen Zeile), die Default-Einstellung ist in diesem Fall *center*.

Das obligatorische Argument *Layout* definiert die Ausrichtung in den Spalten der Tabelle. Für jede einzelne Spalte muss ein Wert angegeben werden:

- `l`: linksbündig
- `c`: Zentriert
- `r`: rechtsbündig
- `p{length}`: feste Breite
- `|` (pipe): vertikale Linien zwischen Spalten werden eingefügt

Tabellen werden Zeile für Zeile geschrieben. Das Et-Zeichen `&` trennt zwei Zellen von einander und der doppelte Backslash `\\` markiert das Ende einer Zeile.

```
Aktuelle Zeile
\begin{tabular}[c]{lc|rp{1.7cm}|}
l-bündig & zentriert & r-bündig & feste Breite \\
\hline
langer Inhalt & langer Inhalt & langer Inhalt & 
langer Inhalt \\
kurz & & kurz \\
\end{tabular}
```

	l-bündig	zentriert	r-bündig	feste Breite
Aktuelle Zeile	langer Inhalt	langer Inhalt	langer Inhalt	langer Inhalt
	kurz		kurz	

Das vorige Beispiel zeigt einige der Eigenschaften von Tabellen:

- Zellen können auch leer bleiben.
- Zellen am Ende einer Zeile können auch weggelassen werden.
- Das Weglassen von Zellen am Zeilenende beeinflusst das Verhalten von vertikalen Linien.
- Inhalt in Zellen mit fester Breite wird automatisch in Silben getrennt.

In der Mathematik-Umgebung (s. Abschnitt 9) kann man auch die `array`-Umgebung verwenden, welche die gleiche Syntax wie `tabular`-Umgebung aufweist. Eine dritte mögliche Umgebung für Tabellen ist die `tabular*`-Umgebung, welche ein weiteres obligatorisches Argument für eine feste Tabellenbreite hat. Für Tabellen mit fester Breite wird jedoch eher das Paket `tabularx` empfohlen (s. Abschnitt 5.2.3).

Wenn das Tabellenlayout symmetrisch ist, kann die folgende Abkürzung verwendet werden: `*{Anzahl}{Layout}`. Dieser Befehl wiederholt das *Layout* so häufig wie es durch *Anzahl* spezifiziert wurde.

```
\begin{tabular}{c*{4}{l}}
one & two & three & four & five \\
thirteen & fourteen & fifteen & sixteen & seventeen \\
\end{tabular}
```

one	two	three	four	five
thirteen	fourteen	fifteen	sixteen	seventeen

```
\begin{tabular}{c*{2}{r|l}}
one & two & three & four & five \\
thirteen & fourteen & fifteen & sixteen & seventeen \\
\end{tabular}
```

one	two	three	four	five
thirteen	fourteen	fifteen	sixteen	seventeen

5.2.1 Linien

Wie bereits gezeigt, werden vertikale Linien bereits im Layoutargument mittels `|` spezifiziert. Die weiteren Linien werden jedoch in der Tabelle direkt angegeben:

- `\hline`: fügt eine horizontale Linie in der ganzen Tabellenbreite ein. Sie kann nicht in einer Zelle, sondern nur vor oder nach der Zelle eingefügt werden.
- `\cline{n-m}`: fügt eine horizontale Linie von der linken Grenze der Spalte n bis zur rechten Grenze der Spalte m ein.
- `\vline`: fügt eine vertikale Linie mit der Höhe einer Zelle ein.

```
\begin{tabular}{c*{2}{r|l}}
\hline
one & two & three & four & five \\
\cline{1-2} \cline{4-5}
thirteen \vline & fourteen & fifteen & sixteen & seventeen \\
\end{tabular}
```

one	two	three	four	five
thirteen	fourteen	fifteen	sixteen	seventeen

5.2.2 Erweitertes Layout

Damit eine Zelle mehrere Spalten umfasst, kann der folgende Befehl verwendet werden: `\multicolumn{Anzahl}{Layout}{Inhalt}`.

Um den Abstand zwischen zwei Spalten zu manipulieren, oder etwas Text zwischen zwei Spalten hinzuzufügen, kann man den folgenden Befehl benutzen: `@{Inhalt}`.

```
\begin{tabular}{c @{:} r l r @{\hspace{3em}} l}
one & \multicolumn{2}{l}{two-plus-three} & four & five \\
\cline{1-2} \cline{4-5}
thirteen & fourteen & fifteen & sixteen & seventeen \\
\end{tabular}
```

one	: two-plus-three	four	five
thirteen	fourteen	fifteen	sixteen
		sixteen	seventeen

Wenn Zellen mehrere Zeilen umfassen, kann man das Paket `multirow` verwenden, mit dem folgenden Befehl: `\multirow{Anzahl}{Breite}[Korrektur]{Inhalt}`.

Das Paket `booktabs` bietet weitere horizontale Linien mit Extra-Abständen, um besondere Tabellen in \LaTeX zu generieren:

- `\toprule[Breite]`
- `\midrule[Breite]`
- `\bottomrule[Breite]`
- `\cmidrule[Breite](<Schnitt>){a-b}`: analog zu `cline` außer wegen des optionalen `Schnitt`-Arguments

Hier ein Paar Beispiele:

Item		
article	unit	price
proofreading	per words	0.02
layout	per page	0.80
printing	per page	0.99
typesetting	per article	40.33

Item		
article	unit	price
proofreading	per words	0.02
layout	per page	0.80
printing	per page	0.99
typesetting	per article	40.33

Item		
article	unit	price
proofreading	per words	0.02
layout	per page	0.80
printing	per page	0.99
typesetting	per article	40.33

Item		
article	unit	price
proofreading	per words	0.02
layout	per page	0.80
printing	per page	0.99
typesetting	per article	40.33

Und hier der dazu gehörige Code:

```
\begin{tabular}[t]{t}{llr}
\multicolumn{2}{c}{Item} & \\
article & unit & price \\
proofreading & per words & 0.02 \\
layout & per page & 0.80 \\
printing & per page & 0.99 \\
typesetting & per article & 40.33 \\
\end{tabular}
```

```
\begin{tabular}[t]{l|l|r}
\hline
\multicolumn{2}{c}{Item} & \\
\hline
article & unit & price \\
\hline
proofreading & per words & 0.02 \\
\hline
layout & per page & 0.80 \\
\hline
printing & per page & 0.99 \\
\hline
typesetting & per article & 40.33 \\
\hline
\end{tabular}
```

```

\begin{tabular}[t]{l}
\hline
\multicolumn{2}{c}{Item} & \\
\cline{1-2}
article & unit & price \\
\hline
proofreading & per words & 0.02 \\
layout & per page & 0.80 \\
printing & per page & 0.99 \\
typesetting & per article & 40.33 \\
\hline
\end{tabular}

```

```

\begin{tabular}[t]{l}
\toprule
\multicolumn{2}{c}{Item} & \\
\cmidrule{1-2}
article & unit & price \\
\midrule
proofreading & per words & 0.02 \\
layout & per page & 0.80 \\
printing & per page & 0.99 \\
typesetting & per article & 40.33 \\
\bottomrule
\end{tabular}

```

5.2.3 Pakete für Tabellen

- `multirow`: um Zellen zu bilden, die über mehrere Zeilen gehen
- `dcolumn`: um Inhalt an einer Trennstelle auszurichten, bspw. an einem Dezimalzeichen
- `array`: um Präfixe und Suffixe für Spalteninhalt zu verwenden, sodass eigene Spaltentypen definiert werden können, bspw. für Schriftartenoptionen
- `tabularx`: für Tabellen mit einer festen Breite
- `longtable`: für Tabellen, die größer als eine Seite sind
- `ltxtable`: für lange Tabellen mit einer festen Breite
- `xcolor` oder `colortbl`: für farbige Elemente bzw. für farbige Tabellen. Das Laden vom `xcolor`-Paket mit der Option `table` (`\usepackage[table]{xcolor}`) bewirkt das automatische Laden des `colortbl`-Pakets.

- `rotating`: stellt `sidewaystable`-Umgebung zur Verfügung, um Tabellen zu drehen.

Hinweis: Texmaker bietet einen Tabellen-Assistenten an. Schaut in der Toolbar unter `Assistent/Tabellen-Assistent` für ein neues Dokument nach. Dort könnt ihr Anzahl an Spalten und Zellen, Zellenausrichtung, Text in den Zellen und alles weitere einstellen.

5.3 Gleitumgebungen

Manchmal erscheinen Grafiken und Tabellen nicht auf der Seite, wo sie per Befehl eingesetzt wurden. Ist auf der Seite mit dem Befehl nicht genügend Platz, erscheint die Grafik/Tabelle auf der nächsten Seite mit genügend Platz. \LaTeX sucht dabei mittels eines Algorithmus, wo die Elemente am besten eingesetzt werden können. Manchmal will man aber selber entscheiden, wo die Grafiken auftreten sollen. Dafür bietet \LaTeX zwei besondere Umgebungen: `figure` und `table`.

Diese Umgebungen schaffen den Platz mit Textbreite, in den die Grafik/Tabelle platziert werden kann.

```
\begin{table}[Position]
  \centering
  \caption[Kurztitel]{Langer Titel}

  \begin{tabular}[t]{t}{1|1}
    \hline
    Eins & Zwei \\
    Drei & Vier \\
    \hline
  \end{tabular}

  \caption[Kurztitel]{Langer Titel}
  \label{Präfix:ID}
\end{table}
```

Der *Position*sparameter kann die folgenden Werte annehmen:

- h** *here*: Das Gleitobjekt wird genau an die Stelle positioniert, wo der Inputbefehl eingegeben wurde (wenn möglich).
- t** *top*: Das Gleitobjekt wird an die oberste Stelle der aktuellen Seite oder Spalte positioniert, *wenn* der vorige Text unter dem Gleitobjekt auf der gleichen Seite Platz findet. Wenn das nicht möglich ist, wird das Gleitobjekt an die oberste Stelle der nächsten Seite/Spalte positioniert.
- b** *bottom*: Das Gleitobjekt wird an die unterste Stelle der aktuellen Seite oder Spalte positioniert (wenn möglich). Wenn das nicht möglich ist, erscheint das Gleitobjekt an der untersten Stelle der folgenden Seite/Spalte.

p *page of floats*: Die Gleitobjekte im Dokument werden gesammelt. Alle Objekte werden gemeinsam in einem Bereich nur für Gleitobjekte (Ende eines Kapitels oder Ende des Dokuments) wiedergegeben.

! Dieses Zeichen kann mit allen anderen kombiniert werden, um den Befehl zu erzwingen.

Die *Position*parameter können kombiniert werden. Die lineare Reihenfolge von links nach rechts gibt die Präferenz an, d. h. der am weitesten links stehende Parameter hat die meiste Präferenz. Es ist empfehlenswert die Einstellung [htbp] zu verwenden. L^AT_EX setzt als Default-Einstellung [tbp].

```
\begin{table}[htbp]
```

Innerhalb der Gleitumgebung können die folgenden Deklarationen für die Ausrichtung der Objekte verwendet werden:

```
\centering  
\raggedright  
\raggedleft
```

Im Prinzip können auch die bekannten Ausrichtungs*umgebungen* benutzt werden (s. Abschnitt 4.7). Dies ist jedoch nicht empfehlenswert, da diese Umgebungen vertikalen Abstand vor und nach dem Befehl zusätzlich zu dem von der Gleitumgebung hinzugefügten Abstand dazurechnet:

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

Mit dem `caption`-Befehl kann eine Grafik-/Tabellenbeschriftung für das Gleitobjekt eingefügt werden. Die Beschriftung kann sowohl *über* als auch *unter* die Grafik/Tabelle angebracht werden. Ähnlich dem `section`-Befehl erlaubt der `caption`-Befehl einen Kurztitel, der im Abbildungs- bzw. Tabellenverzeichnis angezeigt wird.

Jede `caption` wird automatisch präfigiert („Tabelle n :“), und nummeriert („ n “). Die Sprache in der die Präfigierung erfolgt, hängt von der eingestellten Sprache im Sprachunterstützungspaket (z. B. `babel`) ab. Die Art der Nummerierung hängt von der *Dokumentklasse* ab. In der *Artikelklasse* ist die Nummerierung einfach (d. h. n), während in der *Buchklasse* die Nummerierung die Kapitelnummer und die Objektzahl (d. h. $k.n$) enthält.

Wenn die Beschriftung oberhalb der Tabelle erscheint, kann man die Option `tablecaptionabove` zur `documentclass` im KOMA-Script (s. Abschnitt 3.1.1) für den richtigen vertikalen Abstand hinzufügen.

```
\caption[short title]{long title}
```

Der `label`-Befehl wird für Querverweise benutzt. Da der `caption`-Befehl die Nummerierung erstellt, ist es wichtig, dass `label` erst *nach* `caption` kommt, damit der richtige

Querverweis erfolgt. Genauer zur Verwendung vom `label`-Befehl wird im Abschnitt 6 erklärt. Es sei hier nur vorweggenommen, dass das Voranstellen der Präfixe `tab` für Tabellen und `fig` für Grafiken (engl. „figures“) wichtig für einige Bearbeitungsfunktionen in Editoren ist.

```
\label{Präfix:ID}
```

Hier das Beispiel dazu:

```
\begin{table}[htbp]
  \centering
  % \caption{Beschriftung oben}{Lange Beschriftung oben
  % (auskommentiert)}

  \begin{tabular}[t]{l}{ll}
    \hline
    Eins & Zwei \\
    Drei & Vier \\
    \hline
  \end{tabular}

  \caption{Beschriftung unten}{Lange Beschriftung unten}
  \label{tab:beispiel-tabelle1}
\end{table}
```

Eins	Zwei
Drei	Vier

Tabelle 1: Lange Beschriftung unten

Das gleiche gilt auch für Grafiken, wie das folgende Beispiel zeigt.

```
\begin{figure}[htbp]
  \centering
  \includegraphics[scale=0.5]{LaTeX_flowchart_1.pdf}
  \caption{Durchlaufplan in \LaTeX\ }
  \label{fig:latex-flowchart}
\end{figure}
```

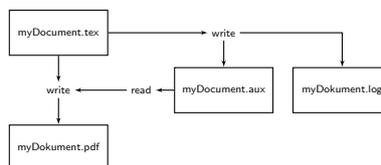


Abbildung 2: Durchlaufplan in \LaTeX

Will man eine neue Seite beginnen, kann man den Befehl `\newpage` verwenden. Dieser Befehl beendet die aktuelle Seite (bzw. die aktuelle Spalte) sofort und füllt den übrigen Platz mit Weißraum auf. Wenn aber Gleitobjekte darauf warten positioniert zu werden, ist eher der Befehl `\clearpage` (oder `\cleardoublepage` für doppelseitige Dokumente) empfehlenswert. Dieser Befehl positioniert *zuerst* die Gleitobjekte und beendet erst *dann* die aktuelle Seite.

Hinweis: Die Verwendung von Präfixen im `label`-Befehl ist ebenso in `Texmaker` hilfreich, da die Autovervollständigung von `Texmaker` bei der Verwendung von Querverweisen die IDs nach Präfixen geordnet anbietet. Dies macht die Suche nach dem passenden Querverweis einfacher!

5.4 Abbildungs- und Tabellenverzeichnis

Abbildungs- und Tabellenverzeichnisse werden einfach mit den folgenden Befehlen automatisch erstellt:

- `\listoffigures`
- `\listoftables`

Dieser Befehl soll an der Position im Dokument stehen, an der das entsprechende Verzeichnis im Output erscheinen soll (i. d. R. nach dem Inhaltsverzeichnis). `LATEX` wird automatisch die Informationen aus den `caption`-Informationen sammeln.

6 Querverweise

6.1 Einfache Querverweise

Mit `LATEX` ist es sehr einfach mit Querverweisen zu arbeiten. Es sind nur zwei Sachen dafür notwendig:

Anker: Dafür wird der Befehl `\label{ID}` verwendet. Die ID muss natürlich einzigartig im Dokument sein.

Verweis: Dafür wird der Befehl `\ref{ID}` benutzt, damit wird auf die (Beispiel-, Abbildungs- oder Tabellen-) Nummer verwiesen. Mit dem Befehl `\pageref{ID}` wird dagegen auf die Seitenzahl verwiesen, in der sich das Element befindet.

Das `label` muss immer dem logischen Textauszeichnungsbefehl folgen, auf das es sich bezieht (z. B. `section`, `item`, `caption`, ...). Um Probleme zu vermeiden, empfiehlt es sich das `label` immer *unmittelbar* nach dem Textauszeichnungsbefehl zu positionieren.

Wenn `LATEX` die ID des Eintrags nicht findet, weil man sich vielleicht verschrieben hat, wird statt des Verweises ein doppeltes Fragezeichen ?? stehen.

6.2 Präfixe

Wie bereits gesehen (s. Abschnitt 5.3), ist es empfehlenswert das `label` mit einem Präfix zu versehen. Die Präfixe stehen vor der ID und werden von ihr durch einen Doppelpunkt (`:`) getrennt. Editoren und Pakete können häufig die ID weiter verwenden. Die häufigsten Präfixe sind:

sec für alle Überschriften

cha/chap nur für Kapitel (**sec** kann auch benutzt werden)

part nur für Bücher, die auch in Teile gegliedert sind (**sec** kann auch benutzt werden)

fig für Abbildungen

tab für Tabellen

item für Listenpunkte

eqn für Gleichungen

fn für Fußnoten

Hier folgt das entsprechende Beispiel. Darin ist zu sehen, dass die Kategorie des Elements (z. B. Tabelle, Abbildung, usw.), auf das verwiesen wird, per Hand eingegeben werden muss und durch ein geschütztes Leerzeichen (`~`) mit dem `ref`-Befehl verbunden wird. So wird vermieden, dass die Nummer allein auf der nächsten Zeile erscheint. Einige Pakete geben automatisch die Kategorie des referierten Elements wieder (s. Abschnitt 6.3).

Hier ist ein Querverweis auf die Tabelle~\ref{tab:beispiel-tabelle2}, die nach diesem Text kommt. Außerdem zeigen wir einen Verweis auf die Tabelle~\ref{tab:beispiel-tabelle1} auf Seite~\pageref{tab:beispiel-tabelle1} im Abschnitt~\ref{sec:float-environments}.

```
\begin{table}[htbp]
  \centering
  \begin{tabular}{lll}
    Eins & Zwei & Drei \\
    Vier & Fünf & Sechs
  \end{tabular}
  \caption{Beispieltabelle für Querverweise}
  \label{tab:beispiel-tabelle2}
\end{table}
```

Eins Zwei Drei
Vier Fünf Sechs

Tabelle 2: Beispieltabelle für Querverweise

Hier ist ein Querverweis auf die Tabelle 2, die nach diesem Text kommt. Außerdem zeigen wir einen Verweis auf die Tabelle 1 auf Seite 42 im Abschnitt 5.3.

Querverweise funktionieren auch in Listen, es ist aber empfehlenswert Querverweise nur in `enumerate`-Umgebungen zu verwenden, da nur diese eine *eindeutige* Nummerierung aufweisen:

```
\begin{enumerate}
\item Erster Punkt
\item \label{item:punkt2} Es klappt!
\item Dritter Punkt
\end{enumerate}
```

Wie in Punkt~\ref{item:punkt2} deutlich wird,
es klappt!

1. Erster Punkt
2. Es klappt!
3. Dritter Punkt

Wie in Punkt 2 deutlich wird, es klappt!

6.3 Pakete für Querverweise

Das `varioref`-Paket bietet hilfreiche Befehle, um auf Nummerierungen und Seitenzahlen zu verweisen, als auch um automatisierte Referenzphrasen wie „auf der nächsten Seite“, usw. hinzuzufügen.

Das Paket `fancyref` fügt u. a. automatisch (in der von `babel` eingestellten Sprache) die Kategorie des referierten Elements (d. h. Tabelle, Abbildung, Abschnitt, usw.) hinzu.

7 Bibliographie

L^AT_EX bietet das Bib_TE_X-Tool, um in Dokumenten Quellen und Bibliographien einfach und vor allem *einheitlich* handzuhaben (für mehr Information siehe 7 im Abschnitt 14).

Bib_TE_X verwendet dafür die folgenden Komponenten:

- eine Bibliographiedatenbank, die aus einem einfachen Textdokument (mit der En-

dung `.bib`) besteht⁶,

- in den Text eingebaute Quellen, deren Angabe ähnlich wie bei Querverweisen funktioniert,
- einen Bibliographiestil.

Ein Beispiel eines Eintrags in einer Bib_TE_X-Datei sieht wie folgt aus:

```
@book{Knuth1986,  
  address = {Boston, MA},  
  publisher = {Addison-Wesley},  
  author = {Knuth, Donald E.},  
  title = {The TeXbook},  
  year = {1986}  
}
```

Jeder Eintrag in der Bibliographiedatenbank besteht aus den folgenden Teilen:

- `@book` meint den Eintragstyp, in diesem Fall handelt es sich um ein `book`. Diese Information ist wichtig wegen der Notation in der Bibliographie!
- `Knuth1986` ist die ID des Eintrags, d. h. der Anker, auf den verwiesen wird (vgl. Abschnitt 6).
- Die anderen Felder tragen die Information des Eintrags zusammen und erklären sich von allein.
- Beachte, dass jeder einzelne `.bib`-Eintrag von der ID bis zum Schluss durch geschweifte Klammern eingeschlossen werden muss, und dass die einzelnen Informationspunkte durch Kommata getrennt sind.

Eine Literaturangabe funktioniert im Prinzip genau so wie der Befehl `ref` bei Querverweisen (vgl. Abschnitt 6) nur mit dem Befehl `cite`:

```
\cite{ID}
```

Am Ende des Dokuments (oder an der Position, an der die Literaturliste erscheinen soll) wird der Bibliographiestil festgelegt und die Verlinkung zur eigenen Bibliographiedatenbank erstellt, mit den folgenden Befehlen:

```
\bibliographystyle{Stilname}  
\bibliography{.bib-Datei-Name}
```

Hier ein Beispiel wie Bib_TE_X im Fließtext verwendet wird⁷:

⁶Die Endung des `.txt`-Dokuments muss in `.bib` geändert werden!

⁷Um zu sehen, wie das Literaturverzeichnis aussieht, siehe die Bibliographie des vorliegenden Readers.

Wenn Sie mehr über die Grundmechanismen vom `\TeX`-System erfahren wollen, lesen Sie `\cite{Knuth1986}`.

```
\bibliographystyle{alpha}
\bibliography{meinebibliographie}
```

Wenn Sie mehr über die Grundmechanismen vom `\TeX`-System erfahren wollen, lesen Sie Knuth (1986).

Der Befehl `cite` greift beim Kompilieren automatisch auf den Eintrag mit der entsprechenden ID in der `.bib`-Datei zu und fügt die passende Literaturangabe zum Fließtext hinzu. Außerdem wird dann eine Bibliographie erstellt, die einheitlich nach dem ausgesuchten Stil formatiert wird.

Die Notation der Literaturangabe im Text und der Bibliographie werden von dem ausgesuchten Bibliographiestil durch den Befehl `bibliographystyle` bestimmt. Die folgenden Stile stehen standardmäßig zur Verfügung⁸: `alpha`, `plain`, `unsrt`, und `abbrv`

Es ist zudem wichtig, dass der Name der Bibliographie, der in dem Befehl `bibliography` eingefügt wird, ohne Endung geschrieben wird, d. h. `meinebibliographie` und nicht `meinebibliographie.bib`.

Außerdem sollte sich die `.bib`-Datei in dem selben Ordner wie die `.tex`-Datei befinden, andernfalls, sollte der gesamte Pfad zur Datei angegeben werden (vgl. Abschnitt 5.1.2).

Wenn `\LaTeX` die `.bib`-Datei oder die ID des Eintrags nicht findet, wird statt der Literaturangabe ein doppeltes Fragezeichen `??` stehen.

Damit ein Literaturverzeichnis erstellt wird, ist es notwendig das Dokument mehrmals zu kompilieren (vgl. Abschnitt 3.2.4):

1. kompilieren mit `PDFLaTeX`, um die Literaturangaben zu finden (und in die `.aux`-Datei zu speichern),
2. kompilieren mit `BibTeX`, um die Literaturangaben aus der `.aux`-Datei mit denen aus der `.bib`-Datei zu vergleichen (es wird eine `.bbl`-Datei generiert),
3. kompilieren mit `PDFLaTeX`, um Literaturangaben einzusetzen und die Bibliographie (aus der `.bbl`-Datei) zu erstellen,
4. kompilieren mit `PDFLaTeX`, falls die Literaturangaben oder die Bibliographie die Seitenzahlen des Dokuments geändert haben.

Hinweis: Um den Kompilierungsprozess zu automatisieren bietet `Texmaker` den Button `Schnelles Übersetzen` an. Wie dieser Button eingerichtet wird, wurde im Hinweis im Abschnitt 3.2.4 erläutert.

⁸Für weitere in der Linguistik häufiger gebrauchte Stile s. Abschnitt 7.2.

7.1 BibTeX-Einträge

Die wichtigsten Eintragstypen sind:

article für Zeitschriftenartikel

book für veröffentlichte Bücher

incollection für Artikel in Sammelbänden

inproceedings für Artikel in Proceedings von Konferenzen

misc ein Joker, falls alles andere nicht passt

phdthesis für Dissertationen

unpublished für unveröffentlichte Manuskripte

All diese Eintragstypen haben sowohl obligatorische als auch optionale und ignorierte Felder. Für mehr Information siehe 7 im Abschnitt 14.

Hier sind einige wenige Tipps um die Einträge der `.bib`-Datei richtig einzugeben. Die Ergebnisse können im Literaturverzeichnis dieses Readers unter Wiese und Piñango (2014), Butler et al. (2014) und Krifka (2014):

author: in der Notation `Nachname, Vorname`, auch wenn andere Notationen möglich sind. Mehrere Autoren werden durch `and` getrennt. Das gleiche gilt für **editor**.

```
author = {Wiese, Heike and Piñango, Maria},
```

title: mit der konventionellen Großschreibung bei Inhaltswörtern (im Englischen). BibTeX konvertiert automatisch die Großschreibung in Kleinschreibung in Artikeltiteln und belässt die Großschreibung bei Buchtiteln (abhängig vom Stil!).

Wenn ein Wort *nicht* von Groß- zu Kleinschreibung konvertiert werden soll (besonders wichtig für deutsche Nomina!), sollte der Buchstabe *geschützt* werden. Dies erreicht man, indem man den Buchstaben in geschweiften Klammern schreibt.

```
title = {... plural marking in {Y}ucatec {M}aya},
```

Um Emphase in Titeln zu markieren, sollte die *emphasize-Deklaration* verwendet werden. Sie schützt jedoch nicht vor der Konversion von Groß- auf Kleinschreibung!

```
title = {\em Ja}, {\em nein}, {\em doch} als ...},
```

7.2 Das natbib-Paket

Das natbib-Paket bietet eine große Breite an Funktionen. Um den in der Linguistik häufig benutzte `author(year)`-Stil⁹ zu verwenden, sollte das Paket mit dieser Option geladen werden:

```
\usepackage[authoryear]{natbib}
```

Dafür sollte dementsprechend ein `bibliographystyle` ausgewählt werden, welcher mit der `author(year)` Notation arbeitet, wie `chicago` oder `apalike`.

Hier einige Beispiele der neuen Befehle, die das natbib-Paket anbietet für Literaturverweise mit Klammern:

<code>\citet{Knuth1986}</code>	Knuth (1986)
<code>\citet[36]{Knuth1986}</code>	Knuth (1986, 36)
<code>\citep{Knuth1986}</code>	(Knuth, 1986)
<code>\citep[36]{Knuth1986}</code>	(Knuth, 1986, 36)
<code>\citep[vgl.] [36]{Knuth1986}</code>	(vgl. Knuth, 1986, 36)
<code>\citep[vgl.] [] {Knuth1986}</code>	(vgl. Knuth, 1986)

Hier einige Beispiele für Literaturverweise ohne Klammern:

<code>\citealt{Knuth1986}</code>	Knuth 1986
<code>\citealp{Knuth1986}</code>	Knuth, 1986

Hier einige Beispiele um nur Teile der Information zu erhalten:

<code>\citeauthor{Knuth1986}</code>	Knuth
<code>\citeyear{Knuth1986}</code>	1986
<code>\citeyearpar{Knuth1986}</code>	(1986)

Diese Befehle können bspw. verwendet werden, um Verweise in Genitiv zu setzen:

```
... wie in  
\citeauthor{Knuth1986}s \citeyearpar{Knuth1986}  
Buch bereits gesehen ...
```

```
... wie in Knuths (1986) Buch bereits gesehen ...
```

Um mehr als einen Literaturverweis zu zitieren, gibt man sie einfach getrennt durch Kommata an:

```
Hier ist ein Verweis mit drei Namen  
\citep[vgl.] [] {Knuth1986,Simon1974,Ross1967}.
```

⁹Andere Stile sind u. a. der `author-page-` und der Fußnotenstil.

Hier ist ein Verweis mit drei Namen (vgl. Knuth, 1986; Simon, 1974; Ross, 1967).

Bib_TE_X kürzt automatisch die Literaturverweise mit „et al.“, wenn dort mehr als zwei Namen vorhanden sind.

Dieser Verweis (Finkbeiner und Meibauer, 2014) enthält nur zwei Namen, während dieser (Nolda et al., 2014) drei Namen hat.

7.3 Literaturverwaltungssysteme

Es gibt viele Literaturverwaltungssysteme wie *JabRef*, *EndNote* oder *Citavi*¹⁰, um die eigene Bibliothek einfach zu verwalten und zu aktualisieren. Einige davon sind kostenfrei (bspw. *JabRef*), andere nicht (*EndNote* oder *Citavi*), allerdings bieten Universitäten häufig kostenlose Lizenzen dieser Programme für Studenten und Mitarbeiter!

Literaturverwaltungssysteme generieren (unter den vielen weiteren Funktionen) auch Bib_TE_X-Datenbanken, die einfach mit der `.tex`-Datei verbunden werden können.

7.4 Weitere Pakete für Bibliographien

Für die Sprachunterstützung (in anderen Sprachen als Englisch) sollte das Paket `babelbib` benutzt werden, sofern `babel` als Sprachunterstützungspaket benutzt wird (s. Abschnitt 3.1.2).

8 Pakete für Linguisten

8.1 Nummerierte Beispiele und Glossen

8.1.1 Beispiele

Für nummerierte Beispiele können die folgenden Programme verwendet werden:

`gb4e` ist etwas mächtiger, aber auch etwas umständlicher in der Syntax.

`linguex` ist einfacher und deutlicher von der Syntax her als `gb4e`, allerdings hat es auch weniger Optionen ...für den „faulen“ Linguisten

Beide Pakete laden automatisch `cgloss4e`, um Beispiele mit Glossen zu verwenden.

In diesem Handbuch wird nur das Paket `linguex` erläutert. Das Paket `gb4e` verfügt über eine gute Dokumentation, kann aber Probleme bei der Interaktion mit anderen

¹⁰*EndNote* ist sowohl für Windows als auch für Mac erhältlich, *Citavi* nur für Windows.

Paketen aufweisen¹¹.

Hier ein Beispiel mit dem `linguex`-Paket:

```
\ex. Hier ist ein neues Beispiel.  
Das neue Beispiel endet mit einer Leerzeile.
```

```
\ex. Hier ist ein anderes Beispiel.  
\a. Hier der Beginn eines neuen Levels  
\b. Ein neuer Punkt im neuen Level  
\b. und noch ein Punkt  
\a. Hier ein neues Level  
\z. %Damit wird das Level geschlossen.  
\b. Ein Punkt in dem Level davor.
```

- (3) Hier ist ein neues Beispiel. Das neue Beispiel endet mit einer Leerzeile.
- (4) Hier ist ein anderes Beispiel.
- a. Hier der Beginn eines neuen Levels
 - b. Ein neuer Punkt im neuen Level
 - c. und noch ein Punkt
 - (i) Hier ein neues Level
 - d. Ein Punkt in dem Level davor.

Es ist auch möglich, dass die Buchstabennummerierung direkt in der arabischen Nummerierung eingebettet wird.

```
\ex. %Leer lassen!  
\a. Die arabische Nummerierung bettet die  
Buchstabennummerierung ein  
\b. Dasselbe gilt hier.
```

- (5)
 - a. Die arabische Nummerierung bettet die Buchstabennummerierung ein
 - b. Dasselbe gilt hier.

8.1.2 Grammatikalitätsurteile

Grammatikalitätsurteile werden bekanntlich durch die folgenden Zeichen angegeben:

```
* ? # %
```

¹¹So ist z. B. die Reihenfolge, in der die Pakete geladen werden, für `gb4e` von Bedeutung. Beispielsweise muss `gb4e` immer *nach* `forest` geladen werden, da `gb4e` bestimmte Befehle, die für `forest` nötig sind undefiniert. Für mehr Informationen schaut im Punkt 22 im Abschnitt 14 nach. Bitte beachte auch, dass die Verwendung von `gb4e` in der `beamer`-Klasse Probleme bei der Nummerierung erzeugen kann.

Um die Raute und das Prozentzeichen verwenden zu können, muss man deren \LaTeX -Funktionen entkommen (vgl. Abschnitt 2.4)!

Grammatikalitätsurteile werden immer am linken Rand des Beispiels angegeben, direkt nach der Nummerierung, getrennt durch ein Leerzeichen vom Nummerierungsbefehl und durch ein weiteres Leerzeichen vom Beispielsatz. Die Beispielsätze sollen aber alle links auf der gleichen Höhe beginnen. Das Paket `linguex` regelt diese Ausrichtung automatisch. Die maximale Anzahl an Zeichen ist vom Platz zwischen Nummerierung und Beispielsatz abhängig. Wenn zu viele Zeichen dort vorhanden sind, schlägt die Ausrichtung fehl.

```
\ex.
\a. * Das ungrammatischer ist Beispiel.
\b. ?? Das ist stark markiertes Beispiel.
\b. ? Das ist markiert nur.
\b. \# Das ist semantisch markiert.
\b. Dieses Beispiel ist wohlgeformt.
\b. \#\#\# Hier sind zu viele Zeichen!
```

- | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>(6) a. *Das ungrammatischer ist Beispiel.
 b. ??Das ist stark markiertes Beispiel.
 c. ?Das ist markiert nur.
 d. #Das ist semantisch markiert.
 e. Dieses Beispiel ist wohlgeformt.
 ####Hier sind zu viele Zeichen!</p> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

8.1.3 Glossen

Sowohl für `linguex` als auch für `gb4e` wird die Glossierung von Beispielen durch das Paket `cgloss4e` erledigt, welches sie automatisch mitinstallieren. Wie bereits erwähnt, ist `linguex` weniger mächtig als `gb4e` und daher stehen nicht *alle* Funktionen zur Verfügung, aber i. d. R. genügende.

Um Beispiele zu glossieren, muss nur ein `g` zum Beispielbefehl vor dem Punkt hinzugefügt werden.

```
\exg. Every farmer who owns a donkey beats it. \\
jeder Bauer wer besitzt einen Esel schlägt es \\
`Jeder Bauer, der einen Esel besitzt, schlägt ihn.'
```

- | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>(7) Every farmer who owns a donkey beats it.
 jeder Bauer wer besitzt einen Esel schlägt es
 ‘Jeder Bauer, der einen Esel besitzt, schlägt ihn.’</p> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Nach dem Beispielbefehl `exg.` folgt ein Leerzeichen, anschließend folgt der Beispielsatz. Die ersten zwei Zeilen (d. h. Beispielsatz und Glossierung) sind obligatorisch und enden mit `\\`. Die Übersetzung ist optional. Die Wörter werden automatisch vertikal ausgerichtet.

Beispielsätze, die mehrere Zeilen lang sind, werden automatisch korrekt angeordnet, d. h. eine Zeile Beispielsatz, nächste Zeile Glosse, usw.

```
\exg. Nils wird die Identität verraten, die der Betrüger
gedeckt hat, der auf der Flucht war und hinreichend
bekannt ist.\\
Nils will the identity betray which the beguiler backed
has who on the escape was and acceptably known is\\
`Nils will betray the identity which has backed the
beguiler who was on the run and is acceptably
known.'
```

- (8) Nils wird die Identität verraten, die der Betrüger gedeckt hat, der auf der Flucht war und hinreichend bekannt ist.
 Nils will the identity betray which the beguiler backed has who on the escape was and acceptably known is
 ‘Nils will betray the identity which has backed the beguiler who was on the run and is acceptably known.’

Um Wörter zu gruppieren oder Platzhalter zu verwenden, kann man die geschwungenen Klammern benutzen¹².

```
\ex.
\ag. Auch Mehrwortelemente können glossiert werden.\\
also {more.word.elements} can.\textsc{3.pl} glossed be\\
```

- (9) a. Auch Mehrwortelemente können glossiert werden.
 also more.word.elements can.3.PL glossed be

```
\ex.
\ag. The vegetablesi that ti looked fantastic\\
das Gemüse das {} aussah toll\\
```

- (10) a. The vegetables_i that t_i looked fantastic
 das Gemüse das aussah toll

¹²Die Regeln für eine korrekte Glossierung werden in Comrie et al. (2008) erklärt.

Mit dem `linguex`-Paket kann man ebenso Klammern mit Labels ausstatten, z. B. für syntaktische Strukturen. Siehe dazu 30 im Abschnitt 14.

8.2 IPA-Notation

Für die IPA-Notation wird das Paket `tipa` verwendet. Es ist empfehlenswert, das Paket mit der Option `safe` zu laden. Für mehr Information siehe das Handbuch unter 54 im Abschnitt 14.

```
\usepackage[safe]{tipa}
```

Die Kombination aus den Paketen `fontenc`, `linguex` und `tipa` kann zu Problemen führen. Will man diese drei Pakete gemeinsam benutzen, z. B. um IPA-Symbole in Beispielen zu verwenden, dann sollten die Pakete mit den folgenden Optionen geladen werden (das gleiche gilt auch, wenn nur `fontenc` und `tipa` benutzt werden!). Für mehr Information siehe das Handbuch unter 54 im Abschnitt 14.

```
\usepackage[T3,T1]{fontenc}
\usepackage{linguex}
\usepackage[noenc,safe]{tipa}
```

Das `tipa`-Paket bietet 3 Möglichkeiten IPA-Symbole zu verwenden:

- Einzelne Makros:

```
[\textglotstop{} n . \textesh{} \textinvscr{}
\texttoptiebar{a\textsci{}} . \textschwa{} n]
```

[ʔan.ɸʁ̥äi.ən]

- Makrogruppierung:

```
\texttipa{[Pan.SK\texttoptiebar{aI}.\@n]}
```

[ʔan.ɸʁ̥äi.ən]

- `tipa`-Umgebung:

```
\begin{IPA}
[Pan.SK\texttoptiebar{aI}.\@n]
\end{IPA}
```

[ʔan.ɸʁ̥äi.ən]

Für eine gute Übersicht der verschiedenen IPA-Symbole schaut im Punkt 29 im Abschnitt 14 nach.

8.3 Bäume

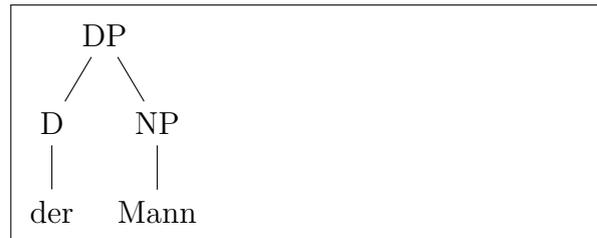
8.3.1 Das forest-Paket

Das `forest`-Paket bietet eine der neusten und mächtigsten Formen, um Bäume in \LaTeX zu zeichnen.

```
\usepackage{forest}
```

Innerhalb einer `forest`-Umgebung werden die Bäume mithilfe der bekannten syntaktischen Klammernotation angegeben. Die Syntax kann linear oder hierarchisch angegeben werden.

```
\begin{forest}
[DP
 [D
 [der]]
 [NP
 [Mann]]
 ]
\end{forest}
```

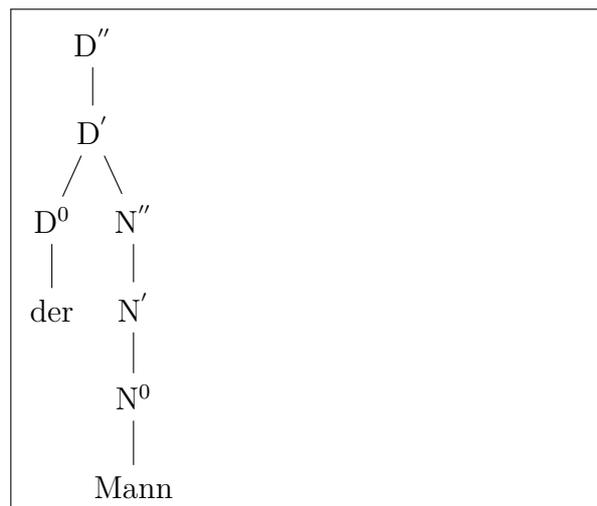


Die Notation für die Zwischenprojektionen werden mit den folgenden Befehlen angezeigt:

```
$^{' '$}
${'}$   Dabei bedeuten diese Zeichenketten folgendes: Das erste $-Zeichen aktiviert
${0}$
```

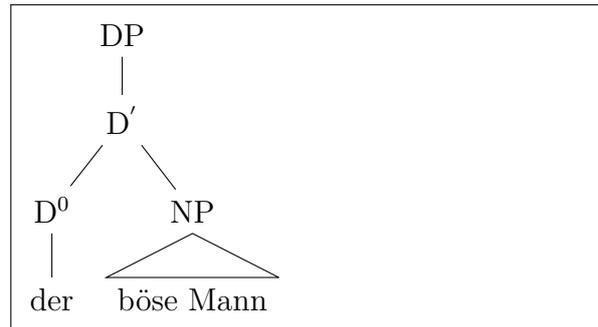
den Mathematikmodus (s. Abschnitt 9). Der Zirkumflex $\hat{}$ stellt im Mathematikmodus die folgenden Zeichen hoch. Die geschwungenen Klammern $\{$ und $\}$ geben den Skopus vom Zirkumflex an. Anschließend deaktiviert das $\$$ -Zeichen den Mathematikmodus.

```
\begin{forest}
[D$^{' '$}
 [D$^{' '$}
 [D$^{0}$ [der]]
 [N$^{' '$}
 [N$^{0}$ [Mann]]
 ]
 ]
 ]
\end{forest}
```



Die Abkürzung von Phrasen kann erreicht werden mit der Option `triangle` innerhalb der Klammer, die abgekürzt wird.

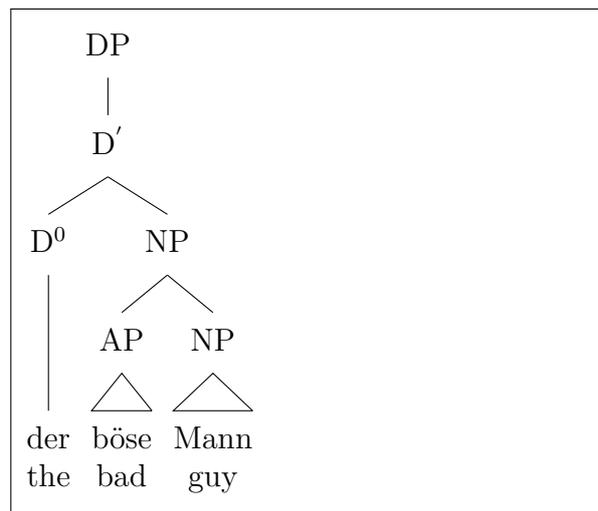
```
\begin{forest}
[DP
 [D$^{'}$
 [D$^{0}$ [der]]
 [NP [böse Mann,triangle]
 ]
 ]
 ]
\end{forest}
```



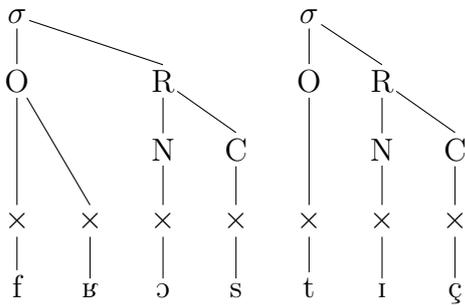
Das Paket `forest` erlaubt mit dem Befehl `forestset` zudem, dass globale Einstellungen für die Bäume in der Präambel geschrieben werden. Diese Einstellungen können jedoch auch lokal vorgenommen werden. So kann die folgende Definition übernommen werden um Glossen in den Bäumen mithilfe des doppelten Backslashes `\\` zu verwenden; mit der in diesem `forestset` definierten Option `sn edges` vor Beginn der Syntax des Baumes werden die Kanten bei jeder Mutter geschlossen und alle lexikalischen Einheiten erscheinen auf einer Linie.

```
\forestset{
sn edges/.style={for tree={parent anchor=south,
child anchor=north,align=center,base=bottom,
where n children=0{tier=word,inner xsep=0pt,
outer sep=0pt}{}}},
background tree/.style={for tree={text opacity=0.2,
draw opacity=0.2,edge={draw opacity=0.2}}}}
}
```

```
\begin{forest}
sn edges,
[DP
 [D$^{'}$
 [D$^{0}$ [der\\
 the]]
 [NP
 [AP [böse\\
 bad,triangle]]
 [NP [Mann\\
 guy,triangle]]
 ]
 ]
 ]
\end{forest}
```



Auch Silbenstrukturen lassen sich mit dem `forest`-Paket zeichnen. Dafür benötigt man den passenden Code im Befehl `forestset`. Um mehr zu diesem Paket zu erfahren, (u. a. woher man die passenden Codes für die Stile erhält) schau mal im Handbuch nach (s. 21 im Abschnitt 14).



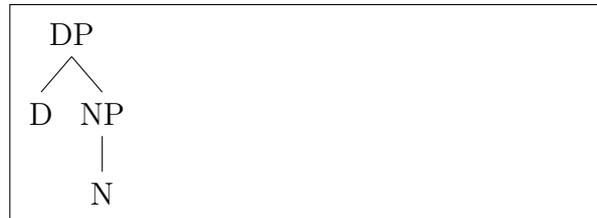
8.3.2 Das `qtree`-Paket

Das `qtree`-Paket erlaubt mit einer einfachen Syntax Bäume zu zeichnen. Es ist zwar nicht so mächtig wie `forest` aber für viele Anwendungen genügend.

Das `qtree`-Paket verwendet ebenso die Klammernotation. Das Zeichnen beginnt hier mit dem Makro `Tree`, das Ende des Baumes wird durch eine Leerzeile angegeben, dies bedeutet, dass keine Leerzeilen innerhalb des Baumcodes enthalten sein dürfen! Knoten im Baum werden durch einen vorangestellten Punkt angegeben.

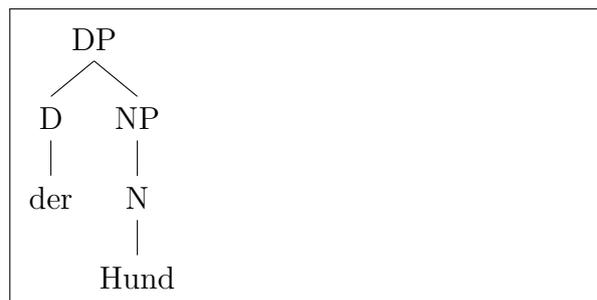
Hier ein Beispiel für einen syntaktischen Baum mit Kategorien:

```
\Tree
[.DP D
  [.NP N ]
]
```



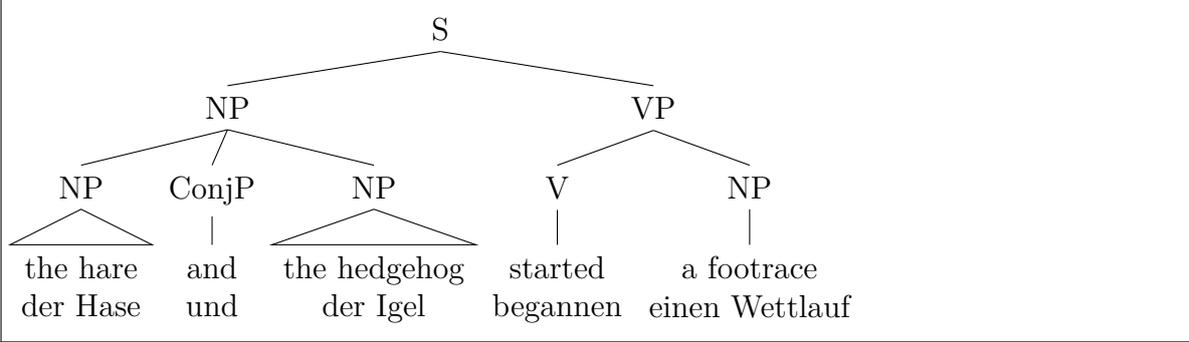
Hier der gleiche Baum mit lexikalischen Einträgen:

```
\Tree
[.DP
  [.D der ]
  [.NP
    [.N Hund ]
  ]
]
```



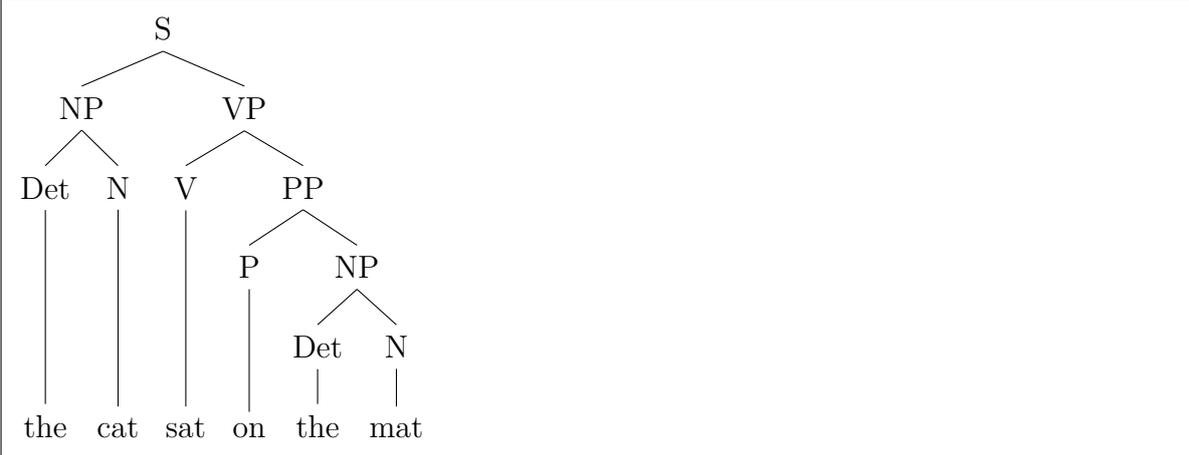
Und hier noch ein komplexeres Beispiel mit Abkürzungen (`\qroof`) und Glossen (`\\\`):

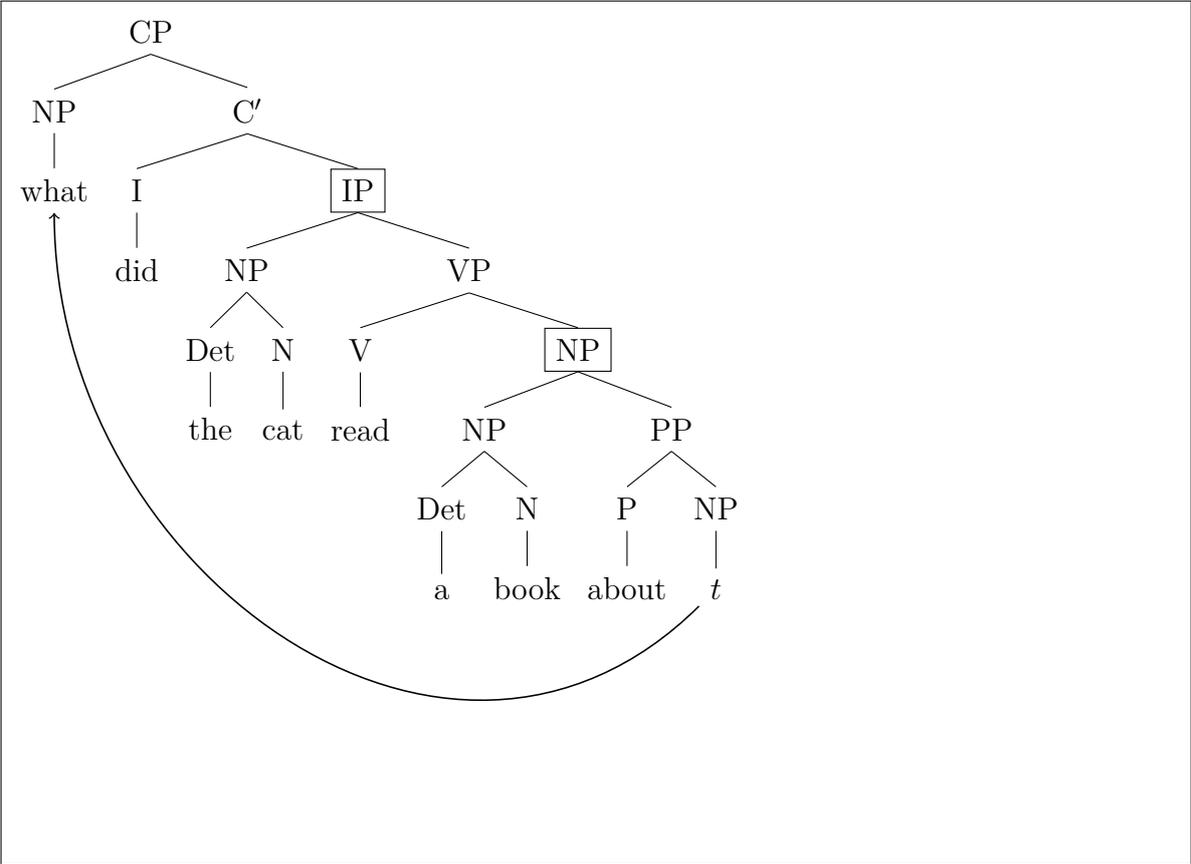
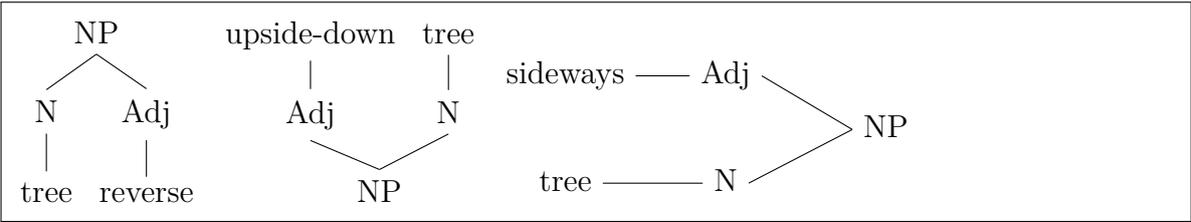
```
\Tree
[.S
  [.NP
    [\qroof{the hare\\der Hase}.NP ]
    [.ConjP and\\und ]
    [\qroof{the hedgehog\\der Igel}.NP ]
  ]
  [.VP
    [.V started\\begannen ]
    [.NP {a footrace\\einen Wettlauf} ]
  ]
]
```



8.3.3 Das tikz-qtrees-Paket

Das `tikz-qtrees`-Paket ist in gewisser Form eine Mischung aus `forest` und `qtrees`. Für mehr Informationen siehe das Handbuch unter Punkt 53 im Abschnitt 14. Hier werden nur einige Beispiele gezeigt, was man damit (oder mit `forest` mit einer etwas anderen Syntax) erschaffen kann.





8.4 Attribute-Value Matrix (AVM)

Mit den Paketen `avm` oder der Weiterentwicklung `avm+` können sehr komplexe AVMs – wie sie in HPSG und LFG verwendet werden – mit einer einfachen Syntax geschrieben werden.

cat subcat	$\langle \text{NP}_{it}, \text{NP}_{[2]}, \text{S}[\text{comp}]:[3] \rangle$								
content	<table border="1" style="border-collapse: collapse; width: 100%; text-align: left;"> <tr> <td style="padding: 2px 5px;">relation</td> <td style="padding: 2px 5px;">bother</td> </tr> <tr> <td style="padding: 2px 5px;">bothered</td> <td style="padding: 2px 5px;">[2]</td> </tr> <tr> <td style="padding: 2px 5px;">soa-arg</td> <td style="padding: 2px 5px;">[3]</td> </tr> </table>	relation	bother	bothered	[2]	soa-arg	[3]		
relation	bother								
bothered	[2]								
soa-arg	[3]								
subj	<table border="1" style="border-collapse: collapse; width: 100%; text-align: left;"> <tr> <td style="padding: 2px 5px;">pers</td> <td style="padding: 2px 5px;">3</td> </tr> <tr> <td style="padding: 2px 5px;">num</td> <td style="padding: 2px 5px;">sg</td> </tr> <tr> <td style="padding: 2px 5px;">gend</td> <td style="padding: 2px 5px;">masc</td> </tr> <tr> <td style="padding: 2px 5px;">pred</td> <td style="padding: 2px 5px;">pro</td> </tr> </table>	pers	3	num	sg	gend	masc	pred	pro
pers	3								
num	sg								
gend	masc								
pred	pro								
pred	eat⟨SUBJ, OBJ⟩								
obj	<table border="1" style="border-collapse: collapse; width: 100%; text-align: left;"> <tr> <td style="padding: 2px 5px;">pers</td> <td style="padding: 2px 5px;">3</td> </tr> <tr> <td style="padding: 2px 5px;">num</td> <td style="padding: 2px 5px;">pl</td> </tr> </table>	pers	3	num	pl				
pers	3								
num	pl								

9 Mathematik-Umgebung

Mathematische Ausdrücke können in zwei Varianten geschrieben werden:

- in der *inline*-Variante, umklammert durch Dollar-Zeichen (\$):

Wenn $2^2 + \sqrt{2} = c^4$, wie viel beträgt c ?

Wenn $2^2 + \sqrt{2} = c^4$, wie viel beträgt c ?

- im *Display*-Stil (auch als Mathematik-Umgebung im engeren Sinne bekannt) eingeschlossen in einer Kombination aus Backslash und eckigen Klammern $\left[\dots \right]$

Wenn $\left[2^2 + \sqrt{2} = c^4 \right]$, wie viel beträgt c ?

Wenn
$$2^2 + \sqrt{2} = c^4$$
, wie viel beträgt c ?

Für nummerierte Gleichungen kann die `equation`-Umgebung gebraucht werden. Mit Hilfe des `eqref`-Befehls aus dem Paket `amsmath` können Querverweise auf Gleichungen gemacht werden, ähnlich wie mit dem `ref`-Befehl¹³.

¹³Bitte beachte den Unterschied zwischen `eqref` und `ref`.

\dots siehe \eqref{eq:1}.

```
\begin{equation}
\label{eq:1}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{equation}
```

...siehe (1).

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6} \quad (1)$$

Formeln können auch im Fließtext angegeben werden, diese werden jedoch kleiner dargestellt.

Das ist eine Formel im Fließtext:

```
 $\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}$ Sie erscheint etwas klein.
Daher empfiehlt es sich, sie in einer Umgebung
zu verwenden:
\begin{equation}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{equation}
```

Das ist eine Formel im Fließtext: $\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$ Sie erscheint etwas klein. Daher empfiehlt es sich, sie in einer Umgebung zu verwenden:

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6} \quad (2)$$

Im Mathematikmodus werden alle Leerzeichen und Zeilenumbrüche ignoriert, solange sie nicht explizit durch den Befehl `\texttrm` angegeben werden. Außerdem erscheint der Text im Mathematikmodus kursiv gesetzt. Mit dem Befehl `\texttrm` kann Text in den Mathematikmodus eingebettet werden.

```
 $Text im Mathematikmodus. \texttrm{ texttrm gibt
Leerzeichen wieder.}$ Ein \texttrm{-Zeichen beendet den
Mathematikmodus und fügt wieder Leerzeichen und
Zeilenumbrüche ein. $Andernfalls geht der Text in einer
Zeile weiter und weiter und weiter und weiter
```

und weiter und weiter\$

TextimMathematikmodus. textrm gibt Leerzeichen wieder. Ein \$-Zeichen beendet den Mathematikmodus und fügt wieder Leerzeichen und Zeilenumbrüche ein. AndernfallsgehtderTextineinerZeileweiterundweiterundweiterundweiterundweiterundweiter

Viele Zeichen – bspw. die griechischen Buchstaben **alpha** (α), **beta** (β), **delta** (δ), usw. – können nur im Mathematikmodus. Viele davon wurden bereits in diesem Dokument besprochen (s. u. a. Abschnitt 2.4). Die Verwendung dieser Zeichen außerhalb des Mathematikmodus verhindert die Kompilierung des Dokuments!

9.1 Weitere Pakete für Mathematik-Umgebungen

Für mathematische Formel bietet das Paket **amsmath** besondere Hilfestellungen. Es hat z. B. eine gesternte Variante der **equation**-Umgebung (**equation***), welche nicht nummeriert wird und somit äquivalent ist zur bereits erwähnten Umgebung: [...]

```
Wenn
\begin{equation*}
  2^2+\sqrt{2}=c^4
\end{equation*}
, wie viel beträgt $c$?
```

Wenn
$$2^2 + \sqrt{2} = c^4$$
, wie viel beträgt c ?

Außerdem stellt **amsmath** den Befehl **text** zur Verfügung, welcher die Funktion von **textrm** übernimmt.

Hinweis: **Texmaker** markiert den Text in der Mathematikumgebung in grün, so dass nicht vergessen wird, die Umgebung wieder zu schließen! Dies ist ein häufiges Problem bei der Kompilierung. Schau immer gut nach, ob alle geöffneten Mathematikumgebungen geschlossen wurden. Das gleiche gilt bei allen anderen Umgebungen.

9.2 Übungen

1. Wie bereits erwähnt, können viele Befehle und Zeichen *nur* im Mathematikmodus verwendet werden. Durch diesen ganzen **L^AT_EX**-Reader wurde gezeigt, wie einige Befehle Argumente nehmen, während andere ohne Argumente benutzt werden. Anhand des folgenden Codes für die Gleichung (3) und für die Gleichung (4),
 - a) versuch herauszufinden, welche Befehle nur im Mathematikmodus benutzt werden können, und
 - b) versuch die Befehle zu verstehen und herauszufinden was sie bewirken.

```

\begin{equation}
\label{eq:ueb1}
2^2+\sqrt{2}=c^4
\end{equation}

\begin{equation}
\label{eq:ueb2}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{equation}

```

$$2^2 + \sqrt{2} = c^4 \quad (3)$$

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6} \quad (4)$$

10 Typographie – Die Bedeutung der kleinen Dinge

L^AT_EX erlaubt die totale Kontrolle über die Typographie im Dokument. Dies bedeutet jedoch auch, dass man wissen muss, welche typographische Konventionen verwendet werden!

10.1 Leerzeichen

L^AT_EX hat verschiedene Befehle für (leere) Zwischenräume (auch *Spatien* genannt) außer der normalen Eingabe durch die Leertaste:

implizites Spatium: Das meint den weißen Abstand zwischen zwei Charakteren im Textmodus. Beispielsweise, sind die impliziten Spatien (zwischen Buchstaben) größer im Sperrsatz. Dafür kann das Paket `microtype` verwendet werden.)

Leerzeichen: Das meint i. d. R. das „normale“ Leerzeichen zwischen Wörtern, welches mit der Leertaste eingegeben wird (aber nicht im Mathematikmodus, s. Abschnitt 9).

Beispiel Beispiel

Beispiel Beispiel

explizites Spatium: Das meint i. d. R. das „normale“ Leerzeichen zwischen Wörtern, welches *explizit* angegeben wird. Das explizite Spatium ist breiter (\approx Viertelgeviert) als das implizite. *Explizit* wird es bspw. nach Befehlen durch den `\` eingegeben.

<code>\LaTeX ist geil</code>	LaTeX ist geil
<code>\LaTeX\ ist geil</code>	LaTeX ist geil
<code>\LaTeX\ist geil</code>	ERROR!

geschütztes Spatium: Eine Tilde `~` fügt ein Leerzeichen ein, welches nicht getrennt wird, z. B. zwischen einer Abkürzung und dem Folgewort, damit die Abkürzung nicht alleine in einer Zeile steht.

<code>s.~Abschnitt</code>	s. Abschnitt
---------------------------	--------------

halbes (geschütztes) Spatium: Der Befehl `\,` wird benutzt für den halben Abstand eines normales Leerzeichens zwischen Wörtern. Er wird z. B. bei Abkürzungen mit mehreren Buchstaben (im Deutschen!), oder bei Maßeinheiten nach der Zahl verwendet (s. Abschnitt 10.2).

<code>z.\,B. 1\,m</code>	z. B. 1 m
--------------------------	-----------

kontextsensitives Spatium: Der Befehl `xspace` aus dem Paket `xspace` setzt ein ganzes Leerzeichen zwischen Wörtern, aber kein Leerzeichen, wenn ein Interpunktionszeichen folgt. Dieses Spatium wird z. B. in selbst gebastelten Definitionen benutzt (s. Abschnitt 13).

<code>Beispiel\xspace ? ein Wort</code>	Beispiel? ein Wort
<code>Beispiel\xspace , ein Wort</code>	Beispiel, ein Wort
<code>Beispiel\xspace ein Wort</code>	Beispiel ein Wort

breites Spatium: Der Befehl `quad` setzt ein Leerzeichen in der Breite eines Buchstabens M in der aktuellen Schriftart. Der Befehl `qqquad` für ein Leerzeichen der Breite zweier M in der aktuellen Schriftart.

<code>Beispiel Wort</code>	Beispiel Wort
<code>Beispiel\quad Wort</code>	Beispiel Wort
<code>Beispiel\qqquad Wort</code>	Beispiel Wort

10.2 Abkürzungen

In der Regel sollten aus Gründen der Lesbarkeit Abkürzungen vermieden werden. Sollten jedoch welche aus Platzgründen oder wegen des Stils verwendet werden müssen, sollten gewisse Konventionen beachtet werden.

Abkürzungen aus mehreren Wörtern wie „z. B. , d. h. oder i. d. R.“ werden im Deutschen durch ein halbes Spatium (`\,`) getrennt, das trifft jedoch nicht für die englischen Pendants wie „e.g., i.e., ...“ zu.

```
weder: z.B.  
noch: z.\ B.  
ja: z.\,B.
```

```
weder: z.B.  
noch: z. B.  
ja: z. B.
```

Zu Beginn von Sätzen sollten Abkürzungen vermieden werden.

```
nicht: Z.\,B. hat er \dots  
ja: Zum Beispiel hat er \dots
```

```
not: Z. B. hat er ...  
but: Zum Beispiel hat er ...
```

Abkürzungen sollten ebenso nach Artikeln und Zählwörtern vermieden werden.

```
siehe S.~24  
Die Seite~24 zeigt \dots
```

```
siehe S. 24  
Die Seite 24 zeigt ...
```

Wie oben gesehen, sollte das geschützte Spatium (~) benutzt werden, um einen Zeilenumbruch zwischen Abkürzung und Folgewort zu vermeiden. Das halbe Spatium (\,) ist ebenso geschützt, so dass Zeilenumbrüche zwischen

- Zahlen,

```
10\,000
```

```
10 000
```

- Abkürzungen,

```
m.\,E.
```

```
m. E.
```

- Zahl und Maßeinheit (12 cm),...

```
12\,cm
```

```
12 cm
```

vermieden werden.

10.3 Ellipsen – Auslassungspunkte

Auslassungen werden bei drei aufeinanderfolgenden Punkte. Der Platz zwischen den Punkten ist anders als bei drei „normalen“ aufeinanderfolgenden Punkten. \LaTeX stellt daher den Befehl `dots` für Auslassungen zur Verfügung.

```
nicht: Es war einmal ...  
ja: Es war einmal \dots
```

```
nicht: Es war einmal ...
ja: Es war einmal ...
```

Wenn die Punkte die Auslassung eines Wortteils repräsentieren, dann sollten sie an den Wort angehängt werden. Wenn sie die Auslassung eines einzelnen Wortes oder mehrerer Wörter repräsentieren, sollte ein Leerzeichen vor dem Befehl gesetzt werden (wie in dem obigen Beispiel).

```
nicht: Er hat Ars \dots\ gesagt!
ja: Er hat Ars\dots\ gesagt!
```

```
nicht: Er hat Ars ... gesagt!
ja: Er hat Ars... gesagt!
```

Wenn die Auslassungspunkte am Ende eines Satzes erscheinen, sollte kein zusätzlicher Punkt am Ende des Satzes gesetzt werden.

10.4 Anführungszeichen

Anführungszeichen werden in Sprachen verschieden dargestellt. Das Paket `babel` mit der Option `ngerman` (oder mit der Option `german`)¹⁴ liefert sowohl die Befehle `grqq` und `glqq` (,german right/left quotation quotation‘) für die doppelten als auch `grq` und `glq` (,german right/left quotation‘) für die einfachen Anführungszeichen im Deutschen¹⁵.

```
Er fragte: \glqq Wer hat \glq Arschloch\grq\ gesagt?\grqq
```

```
Er fragte: „Wer hat ‚Arschloch‘ gesagt?“
```

Englische Anführungszeichen können ohne weitere Pakete mit ``` ''`, bzw. mit `` '` geschrieben werden.

```
She asked: ``What does `erothetic' mean?''
```

```
She asked: "What does 'erothetic' mean?"
```

Das Paket `csquotes` bringt eine hilfreiche kontextsensitive Handhabung der Anführungszeichen in verschiedenen Sprachen.

10.5 Der Strich

Hier sind vier verschiedene Typen von Strichen, die in der Typographie verwendet werden:

Bindestrich (Viertelgeviert, `Divis`, ‚hyphen‘) Dieses Zeichen ist der kürzeste Strich und wird nur für die Trennung zwischen Wörtern oder Wortteilen. In \LaTeX wird er durch den einfachen Strich wiedergegeben -.

¹⁴Diese Befehle sind nicht bei \LaTeX von Hause aus integriert, daher benötigt man das Extrapaket `babel` für ihre Darstellung.

¹⁵Bitte beachte dabei die Setzung von Spatien bei den Befehlen.

Spieler-Trainer, kauf-

Spieler-Trainer, kauf-

Gedankenstrich (Halbgeviertstrich, Bis-Strich, ‚en-dash‘) Dieses Zeichen ist länger als der Bindestrich, es hat die Länge eines Buchstaben *n*. Dieses Zeichen wird eigentlich in allen anderen Fällen verwendet, in denen nicht der Bindestrich benutzt wird, d. h. für Einschübe im Text, als Bis-Zeichen, ... In \LaTeX wird er durch einen doppelten Bindestrich kodiert --.

Seite 3--4, Mo.--Fr.
In dieser Theorie -- und in
den späteren auch -- hat
er \dots

Seite 3–4, Mo.–Fr.
In dieser Theorie – und in den späteren
auch – hat er ...

Geviertstrich (‚em-dash‘) Dieses Zeichen ist der längste Strich. Es hat die Länge von einem Buchstaben *m*. Es wird meistens im Englischen als Gedankenstrich verwendet, und manchmal auch bei Geldbeträgen oder als Spiegelstrich in Listen. In \LaTeX wird er durch einen dreifachen Bindestrich kodiert ---.

45,--- Euro

45,— Euro

Minuszeichen Dieses mathematische Symbol hat die gleiche Länge wie ein Plus-Zeichen und ist etwas tiefer gelegen als der Gedankenstrich. In \LaTeX wird er durch einen Bindestrich in Mathematikmodus kodiert $\$-\$$

1 $\$+\$$ 2 $\$-\$$ 3 = 0

1 + 2 – 3 = 0

11 Längere Arbeiten – Multifile Parsing

Wenn du an größeren Arbeiten schreibst, kann es hilfreich sein den Inhalt auf mehrere Dateien aufzuteilen, zum Beispiel in Kapitel oder Abschnitte. Die einzelnen Inhalte werden dann alle in einer Master- oder Hauptdatei aufgerufen.

Nützlich ist das auch, wenn du mit Kollegen ein Dokument erstellt, bei dem jeder einen Teil bereitstellt oder eine Sammlung, zum Beispiel ein Booklet erstellt wird.

\LaTeX stellt dafür zwei Befehle zur Verfügung:

- $\backslash\text{input}\{\text{filename}\}$ bindet einfach die angegebene Datei ein. Dieser Befehl kann rekursiv verwendet werden.
- $\backslash\text{include}\{\text{filename}\}$ beginnt eine neue Seite, bevor der Inhalt der angegebenen Datei eingelesen wird. Dieser Befehl kann nicht rekursiv verwendet werden. Innerhalb der eingebundenen Datei können jedoch andere Dateien mit $\backslash\text{input}$ eingebunden werden.

Bedenke, dass alle Pfadangaben (für Textinhalt, Graphikdateien, usw.) immer relativ zur Masterdatei angegeben werden müssen. Das ist natürlich sehr wichtig, wenn du Dateien mehrfach ineinander verschachtelst. Weniger ist manchmal mehr! Erwähne dich an den `graphicspath`-Befehl aus Abschnitt 5.1.

Es gibt die Möglichkeit die Interpretation der Dateien zu beschränken, die du mit `\include{}` eingebunden hast. Das kann unter Umständen eine Menge Zeit beim Kompilieren sparen, wenn dein Dokument sehr groß ist. Einfach folgenden Befehl in die Präambel aufnehmen.

```
\includeonly{Dateiname1, Dateiname2,...}
```

Wenn du ein großes Dokument mit vielen Graphiken erzeugst, können die Kompilierungsvorgänge schon mal länger dauern. Dann füge die Option `draft` zur Dokumentenklasse hinzu. Graphiken werden dadurch zum Beispiel nicht geladen, stattdessen werden Platzhalterboxen in das Dokument eingefügt¹⁶.

12 Finaler Satz

Hier haben wir dir ein paar Vorschläge zusammengestellt, bezüglich der Schritte, die du durchführen kannst um dein \LaTeX -Dokument zu vollenden. Beachte, dass die Schritte eine logische Reihenfolge haben: je globaler der Effekt, desto eher sollte dieser Schritt ausgeführt werden.

12.1 Titelseite

Die Parameter für die Titelseite sind abhängig von der Dokumentenklasse. Im Folgenden findest du die Parameter für die Klassen im `KOMA-Script`. Die Informationen werden mit dem Befehl `\maketitle` gesetzt.

```
\titlehead{}
\subject{}
\title{}
\author{}
\date{}
\publishers{}
```

Einige Dokumentenklassen (`book`, `report`) setzen standardmäßig eine separate Titelseite. Bei den `KOMA`-Klassen kann dieses Verhalten mit der Dokumentenklassenoption `titlepage/notitlepage` gesteuert werden.

Für weitere Optionen – natürlich – ins Handbuch zum `KOMA-Script` schauen, klaro (s. 28 im Abschnitt 14).

¹⁶Obwohl `KOMA-Script` üblicherweise die `parameter=value` Syntax benutzt, also `draft=true`, benötigen einige Pakete, so wie `graphicx` die klassische Version, also lediglich `draft`.

12.2 Satzspiegel

12.2.1 Generelles Seitenlayout

Einige generelle Optionen können der Dokumentklasse als Parameter mitgegeben werden:

onecolumn Eine Spalte pro Seite

twocolumn Zwei Spalten pro Seite (journal style)

oneside Gerade und ungerade Seiten habe die gleichen Ränder und das gleiche Layout

twoside Gerade und ungerade Seiten haben unterschiedliche Ränder und Layout (Symmetrisch)

parskip Fügt eine Leerzeile zwischen Absätzen ein (Ohne Absatzeinzug!)

halfparskip Fügt eine halbe Zeile ein.

parindent Fügt bei der ersten Zeile eines Absatzes eine Einzug ein, ausgenommen die erste Zeile eines Abschnitts

Rate mal wo du weiter Optionen findest ...

12.2.2 Ränder

Die KOMA-Klassen berechnen die Ränder nach typographischen Gesichtspunkten basierend auf einem Raster (DIV).

```
\documentclass[a4paper, 12pt, DIV11]{scrartcl}
```

Der Standardwert ist 9. Du kannst den Faktor des Rasters mit einer Zahl >4 verändern. Je größer die Zahl, desto kleiner die Ränder und desto breiter der Textbereich (zu große Zahlen können zu merkwürdigen Ergebnissen führen).

Du kannst auch L^AT_EX den *optimalen* Wert für dich berechnen lassen.

```
\documentclass[a4paper, 12pt, DIVcalc]{scrartcl}
... %packages
\typearea[current]{calc}
```

Standardmäßig sind Kopf- und Fußzeilen vom Satzspiegel ausgeschlossen. Du kannst sie mit den Optionen `headinclude/footinclude` miteinbeziehen.

Wenn die Berechnung des Satzspiegels in Dokumenten vornehmen willst, die nicht zur KOMA-Klasse gehören, kann du das Paket `typearea` nutzen.

Für eine freie Wahl des Satzspiegels solltest du auf das Paket `geometry` zurückgreifen. Es erlaubt dir jeden Rand individuell zu bestimmen (...was meistens eine schlechte Idee ist).

12.2.3 Bindekorrektur

Die Bindekorrektur ist ein Extrarand an den Innenseiten eines zweiseitig gesetzten Dokuments, die für die Bindung benötigt wird. Du kannst die breite der Bindekorrektur explizit als Option der Dokumentenklasse angeben.

```
\documentclass[DIV11, BCOR10mm]{scrartcl}
```

12.2.4 Schriftgröße

Die Schriftgröße kann auch als Option der Dokumentenklasse angegeben werden.

```
\documentclass[a4paper, 12pt, DIV11]{scrartcl}
```

12.2.5 Zeilenabstand

Der Zeilenabstand oder Durchschuss sollte mit Hilfe des `setspace`-Pakets verändert werden, da es die Fußnoten von den Veränderungen ausnimmt. Es bietet die folgenden Paketoptionen:

```
\usepackage[singlespacing]{setspace}
```

```
\usepackage[onehalfspacing]{setspace}
```

```
\usepackage[doublespacing]{setspace}
```

Zusätzlich stellt es die folgenden Deklarationen bereit

```
\onehalfspacing
```

```
\doublespacing
```

```
\singlespacing
```

und die entsprechenden Umgebungen:

```
\begin{singlespace}  
...  
\end{singlespace}
```

```
\begin{onehalfspace}  
...  
\end{onehalfspace}
```

```
\begin{doublespace}  
...  
\end{doublespace}
```

Die zusätzliche `*singlespace`-Umgebung setzt einen schmaleren Leerraum zum vorhergehenden und nachfolgenden Text.

Mit der folgenden Umgebung

```
\begin{spacing}{number}  
text...  
\end{spacing}
```

kannst du einen benutzerdefinierten Zeilenabstand setzen. Wenn du den Zeilenabstand nur minimal verändern willst, beispielsweise für bestimmte Fonts, solltest du folgenden Befehl nutzen:

```
\linespread{1.05}
```

12.2.6 Kopf- und Fußzeile

Du kannst das Erscheinungsbild deiner Seiten global `\pagestyle{option}` (für alle Seiten die folgen) und lokal `\thispagestyle{option}` (nur für die aktuelle Seite) bestimmen.

empty leert die Kopf- und Fußzeile

plain nur Seitenzahlen in der Fußzeile

headings setzt den aktuellen Kapitel-/Abschnittstitel in die Kopfzeile

myheadings setzt benutzerdefinierte Informationen in Kopf- und Fußzeile. Nutze eher das `scrpage2`-Paket um deine eigenen Kopf- und Fußzeilen zu definieren und nicht das `fancyhdr`-Paket.

12.2.7 Seitenzahlen

Mit dem Befehl `\pagenumbering` kannst du den Typ der Seitennummerierung von der aktuellen Seite an definieren. Jeder Aufruf des Befehls setzt den Zähler zurück.

arabic arabische Ziffern: 1, 2, 3, ...

roman kleine römische Ziffern: i, ii, iii, ...

Roman große römische Ziffern: I, II, III, ...

alph Kleinbuchstaben: a, b, c, ...

Alph Großbuchstaben: A, B, C, ...

12.3 PDF Optionen

Du kannst eine Reihe von Parametern spezifizieren die später Teil der Metadaten deines .pdf-Dokuments werden, zum Beispiel `author`¹⁷ und `title`.

¹⁷Wenn du einen anonymen Abstract erstellst, solltest du sicher gehen, dass die Autoreninformation nicht Teil der pdf-Metadaten sind. Dazu brauchst du nur die `author`-Zeile löschen, auskommentieren oder mit einem leeren Inhalt bzw. explizit mit `anonymous` versehen.

Das `hyperref`-Paket bietet noch weitere Möglichkeiten die pdf-Metadaten zu spezifizieren, zum Beispiel `keywords`. Das Paket kreiert standardmäßig Lesezeichen (bookmarks) aus dem Inhaltsverzeichnis und Hyperlinks, zum Beispiel für Querverweise in deinem `.pdf`-Dokument.

```
\usepackage[bookmarksnumbered]{hyperref}
```

12.4 Silbentrennung

12.4.1 Silbentrennung anpassen

\LaTeX trennt Wörter an denen der Algorithmus es für ideal befindet. So manches Mal (für manche Sprachen deutlich häufiger) findet der Algorithmus eine falsche oder überhaupt keine Trennstelle.

Mit dem Befehl `hyphenation` kannst du eine Wortliste für dein Dokument anlegen und alle Trennstellen definieren. Klarerweise ist es äußerst sinnvoll diese Liste möglichst weit am Anfang des Dokumenten zu platzieren.

```
\hyphenation{All-tags-ge-gegen-stand Prag-ma-tism nobreak}
```

Beachte die folgende Syntax im `hyphenation`-Befehl:

- Trenne die Wörter nur durch Leerzeichen (kein Komma).
- Markiere **alle** Trennstellen mit Bindestrichen an. \LaTeX wird das jeweilige Wort nur an den angegebenen Stellen trennen.
- Wenn du keine Trennstelle angibst wird das Wort gar nicht getrennt, z. B. „nobreak“
- Das Trennmuster wird auf alle passenden Zeichenketten angewandt, unabhängig von Groß- und Kleinschreibung.

Das `babel`-Paket mit der Option `german` oder `ngerman` bietet noch ein paar weitere Befehle für die Silbentrennung an.

In der Trennliste sind keine Sonderzeichen erlaubt. Aber du kannst die Trennstellen auch lokal, d. h. am Wort selbst definieren.

Der Befehl `\-` spezifiziert die möglichen Trennstellen. Denke daran, alle möglichen Trennstellen eines Wortes zu definieren, denn wird nur eine Trennstelle angegeben, ist dies die einzige Stelle an der \LaTeX das entsprechende Wort trennen wird.

```
Hyphenating words like
elec\ -tro\ -en\ -ceph\ -a%
\ -lo\ -graph\ -i\ -cal\ -ly
```

```
Hyphenating words like electroencephalo-
graphically
```

Wenn du zwei Wörter mit einem Schrägstrich (slash) verbindest, wird die Zeichenkette als ein Wort behandelt und nicht am Schrägstrich getrennt. Wenn du die Trennung ermöglichen willst, nutze entweder den Befehl `\slash` oder definiere eine Trennstelle.

```
reading/writing\\
reading\slash writing\\
reading/\-writing
```

```
reading/writing
reading/writing
reading/writing
```

12.4.2 Silbentrennung unterdrücken

L^AT_EX bietet spezielle Befehle um Silbentrennung außer Kraft zu setzen.

- `~` setzt ein Leezeichen an dem nicht getrennt wird.
- `\mbox{Inhalt}` zeichnet eine unsichtbare Box um den Inhalt (nur eine Zeile). Der Inhalt wird nicht getrennt.

12.5 Seitenkorrektur

Um Schusterjungen und Hurenkinder zu vermeiden kannst du entweder Seitenumbrüche erzwingen oder Seiten (minimal) vergrößern.

Wir kennen bereits zwei Befehle für Seitenumbrüche:

- `\newpage` beginnt eine neue Seite
- `\clearpage` setzt erst alle noch zu setzenden Gleitumgebungen (Abbildungen, Tabellen) und beginnt dann eine neue Seite.

Du kannst eine Seite auch vergrößern, aber bitte sei maßvoll.

```
\enlargethispage{length}
\enlargethispage{2em}
```

13 Programmieren – Definiere deine eigenen Makros

13.1 Eigene Makros definieren

L^AT_EX bietet drei grundlegende Befehle, um eigene Makros (Befehle) zu definieren.

- `\newcommand{\Name}[Argumentenanzahl][Default]{Definition}` für die Definition neuer Befehle. Der Name des Befehls darf dabei nicht mit einem bereits bestehenden Befehl identisch sein, sonst knallt's.
- `\renewcommand{\Name}[Argumentenanzahl][Default]{Definition}` dient der Umdefinierung eines bereits bestehenden Befehls. Der Name des Befehls muss bereits existieren bzw. belegt sein, sonst knallt's.
- `\providecommand{\Name}[Argumentenanzahl][Default]{Definition}` definiert ebenfalls einen neuen Befehl. Sollte der Name bereits belegt sein, gilt weiterhin die alte Definition und die neue Definition wird ignoriert. Das ist eigentlich immer eine gute Idee, wenn man keine Ahnung hat, ob es den Namen schon gibt, nur darf man sich nicht wundern, wenn sich keine Wirkung zeigt.

Die einzelnen Parameter:

- `{\Name}` Der Name muss mit einem Backslash beginnen, gefolgt von großen und/oder kleinen Buchstaben A–Z a–z. Keine anderen Zeichen, z. B. Ziffern, sind erlaubt. Die einzigen Ausnahmen sind 2-Zeichenbefehle, welche uns oben schon begegnet sind, die aus einem Backslash gefolgt von einem nicht alphanumerischen Zeichen bestehen: `\`, und `\\`.
- `[Argumentenanzahl]` Die Anzahl der Argumente ist eine optionale Angabe. Falls sie spezifiziert wird muss sie eine Zahl zwischen 1 und 9 beinhalten.
- `[Default]` Hiermit kann ein Standardinhalt für das erste Argument voreingestellt werden. Diese Angabe ist ebenfalls optional.
- `{Definition}` Die eigentliche Befehlsdefinition ist natürlich obligatorisch. Hier wird definiert was \LaTeX im Hintergrund ausführen soll, wenn ihm der besagte Befehl begegnet.

13.1.1 Befehle ohne Argumente

Befehle ohne Argumente sind Vereinfachungen/Abkürzungen. Sie erlauben es dir lange, immer wiederkehrende Ausdrücke eventuell mit Auszeichnungsbefehlen einfach und schnell einzufügen.

```
\newcommand{\Name}{Definition}
```

Wenn du bspw. einen Syntaxaufsatz über Fragen schreibst, wirst du wohl öfter einen solchen Ausdruck verwenden:

```
\emph{wh}-Element
```

```
wh-Element
```

Wenn du keine Lust hast jedes mal diese Formatierung einzugeben, kannst du deine eigene Abkürzung definieren:

```
\newcommand{\whel}{\emph{wh}-Element}
```

```
Das erste \whel\ ist vorangestellt und  
das zweite \whel\ bleibt in-situ.
```

```
Das erste wh-Element ist vorangestellt und das zweite wh-Element bleibt in-situ.
```

Wie bei anderen Befehlen wird auch jeglicher Leerraum hinter den Befehlen *gefressen*. Wenn man also Zeichenketten (Ausdrücke) definiert, ist das `xspace`-Paket sehr hilfreich. Es stellt den gleichnamigen Befehl `\xspace` zur Verfügung. Dieser Befehl fügt automatisch ein Leerzeichen ein, wenn ein alphanumerischen Zeichen folgt. Wenn jedoch – z. B. am Satzende – ein Interpunktionszeichen folgt, wird auf das Leerzeichen verzichtet.

```
\newcommand{\whel}{\emph{wh}-element\xspace}
```

Das erste `\whel` ist vorangestellt, aber nicht das zweite `\whel`.

Das erste *wh*-element ist vorangestellt, aber nicht das zweite *wh*-element.

Solche Makros können auch äußerst hilfreich sein, wenn du zwar derzeit einen Arbeitstitel für einen Terminus benutzt, aber erst später entscheiden willst, wie der Begriff wirklich heißen soll.

Stell dir vor, du Berichtest über ein Experiment mit zwei Bedingungen, zwei Ausprägungen (levels) von lexikalischer Frequenz: `\lexfreqhigh` und `\lexfreqlow`. Du verweist zwar schon in deinem Dokument auf diese Bedingungen, aber dir fallen noch keine treffenden Bezeichnungen ein. Definiere dir einfach einen Befehl.

```
\newcommand{\lexfreqhigh}{\textsc{lex-freq-high}\xspace}
```

Kurze Reaktionszeiten für `\lexfreqhigh` implizieren, dass man auf Wörter in der `\lexfreqhigh`-Bedingung schneller Zugriff hat.

Kurze Reaktionszeiten für LEX-FREQ-HIGH implizieren, dass man auf Wörter in der LEX-FREQ-HIGH-Bedingung schneller Zugriff hat.

Wie auch immer du dich später entscheiden solltest, die Bedingungen bleiben die gleichen, lediglich der Name ändert sich. Um den Namen anzupassen, einfach den *Definition*steil deiner Befehlsdefinition entsprechend ändern und Voilà!

```
\newcommand{\lexfreqhigh}{LF-\textsc{high}\xspace}
```

Kurze Reaktionszeiten für `\lexfreqhigh` implizieren, dass man auf Wörter in der `\lexfreqhigh`-Bedingung schneller Zugriff hat.

Kurze Reaktionszeiten für LF-HIGH implizieren, dass man auf Wörter in der LF-HIGH-Bedingung schneller Zugriff hat.

13.1.2 Befehle mit Argumenten

Es geht natürlich auch noch ausgefuchster. Im Prinzip ist ein Befehl nur eine Variable, der du einen Inhalt zuweisen kannst, den \LaTeX verarbeitet, wenn es auf den Befehl trifft. Du könntest ja statt jedem Befehl auch den Inhalt der Befehlsdefinition in deinen Text kopieren.

Ein Argument eines Befehls ist nun eine Variable innerhalb dieser Befehlsdefinition. Also eine Variable innerhalb einer Variable, falls dass zur Verwirrung beiträgt.

Du kennst bereits viele solcher Befehle, die ihrerseits Argumente nehmen, z. B. einfache Textauszeichnungsbefehle wie `\textbf{Inhalt}`, der im Grunde besagt „Was auch immer innerhalb der geschweiften Klammern steht, setze es in Fettdruck“.

Also, lass es uns selbst machen. Wir geben den neuen Befehl `newbf` ein, mit einem Argument [1] und mit der Definition: das Argument #1 in `bfseries`, d. h. in Fettdruck gesetzt¹⁸.

```
\newcommand{\newbf}[1]{\bfseries #1}
```

```
Jetzt resultiert aus \newbf{Inhalt}
das gleiche wie aus \textbf{Inhalt}.
```

```
Jetzt resultiert aus Inhalt das gleiche wie aus Inhalt.
```

Wie du siehst, wird auf die Argumente in der Befehlsdefinition mit der Raute, gefolgt von der jeweiligen Nummer referiert.

Nun kannst du Schritt für Schritt die Zahl der Argumente erhöhen, je nachdem was du bewerkstelligen willst. Hier ein Beispiel, dass in der Grammatiktheorie Anwendung finden könnte:

```
\newcommand{\mergexy}[2]{merge (#1, #2)
$\rightarrow$ [#1[#1\ #2]]}
```

```
\mergexy{X}{Y}
```

```
merge (X, Y) → [X[X Y]]
```

```
\mergexy{$\alpha$}{$\beta$}
```

```
merge ( $\alpha$ ,  $\beta$ ) → [ $\alpha[\alpha \beta]$ ]
```

Nun, da uns das definieren schon leicht von der Hand geht, lass uns das optionale Standardargument ausprobieren. Wenn dieses Standardargument definiert wird, dann wird die Variable #1 mit dem jeweiligen Standardwert ersetzt. Wenn du den Befehl aber doch mit einem spezifischen Inhalt für das erste Argument aufrufst, dann wird der Standardwert überschrieben.

```
\newcommand{\headxy}[3][\$^0$]{[#2P [#3 #2#1]]}
```

```
\headxy{X}{Y}
```

```
[XP [Y X0]]
```

```
\headxy[-Kopf]{X}{Y}
```

```
[XP [Y X-Kopf]]
```

¹⁸Achte darauf, dass der Skopus der *Deklaration* `bfseries` durch die geschwungenen Klammern { und } markiert wird!

13.1.3 Weitere Beispiele für nützliche Befehle

Abkürzungen:

```
\newcommand{\zB}{z.\,B.\ }
```

wie \zB hier.

wie z. B. hier:

```
\newcommand{\s}{\mbox{s.~}}
```

dazu \s Kapitel \dots

dazu s. Kapitel ...

Einheiten:

```
\newcommand{\mm}[1]{#1\,mm\xspace}
```

mit \mm{3} Abstand

mit 3 mm Abstand

Auch hier mit optionalem Standardargument:

```
\newcommand{\xunit}[2][mm]{#2\,#1\xspace}
```

mit \xunit{25.4} oder \xunit[inch]{1} Abstand

mit 25.4 mm oder 1 inch Abstand

Zitierung mit Genitiv-*S* (mit dem natbib-Paket):

```
\newcommand{\citegen}[1]{\citeauthor{#1}'s  
(\citeyear{#1})\xspace}
```

So \citegen{Chomsky1995} contribution to \dots

So Chomsky's (1995) contribution to ...

13.2 Eigene Umgebungen

In gleicher Manier kannst du auch eigene Umgebungen definieren.

```
\newenvironment{Name}[Argumentenanzahl][Default]  
{BeginnDef}{EndeDef}  
\renewenvironment{Name}[Argumentenanzahl][Default]  
{BeginnDef}{EndeDef}
```

Ohne Argumente:

```
\newenvironment{itquote}{\begin{quote}\itshape}
{\end{quote}}
```

```
Etwas Text über deinem Zitat
\begin{itquote}
Hier ein Zitat nun kursiv\newline
mit einem Zeilenumbruch
\end{itquote}
Der Text nach dem Zitat
```

Etwas Text über deinem Zitat

*Hier ein Zitat nun kursiv
mit einem Zeilenumbruch*

Der Text nach dem Zitat

Mit Argumenten:

```
\newenvironment{headerquote}[1]{\centerline{\textbf{#1}}}
\begin{quote}}{\end{quote}}
```

```
Etwas Text über deinem Zitat\newline
\begin{headerquote}{Der Wahnsinn in Tüten}
Nach dem Titel für das Zitat, hier noch
das Zitat in der alten quote-Umgebung\newline
mit einem Zeilenumbruch
\end{headerquote}
Der Text nach dem Zitat
```

Etwas Text über deinem Zitat

Der Wahnsinn in Tüten

Nach dem Titel für das Zitat, hier noch das Zitat in der alten quote-
Umgebung
mit einem Zeilenumbruch

Der Text nach dem Zitat

14 Pakete und Handbücher

Hier ist eine Liste von URLs mit der besprochenen Software, den Paketen und den Handbüchern, die in diesem Dokument behandelt wurden. Eine schnelle Hilfe bietet ebenso Wikipedia, und vor allem das durchsuchbare *wikibook* „*L^AT_EX*-Kompendium“ (s. 60).

1. amsmath: <https://www.ctan.org/pkg/amsmath>
2. array: <https://www.ctan.org/pkg/array>

3. avm: <http://nlp.stanford.edu/manning/tex/>
4. avm+: <http://www.sfs.uni-tuebingen.de/~dm/02/winter/latex-tutorial/styles/avm/>
5. babel: <http://www.ctan.org/pkg/babel>
6. babelbib: <https://ctan.org/pkg/babelbib>
7. BibTeX: <https://ctan.org/pkg/bibtex> und <http://www.bibtex.org/>
8. booktabs: <http://www.ctan.org/pkg/booktabs>
9. Citavi: <https://www.citavi.com/de/>
10. colortable: <http://www.ctan.org/pkg/colortbl>
11. csquotes: <https://www.ctan.org/pkg/csquotes>
12. dcolumn: <http://www.ctan.org/pkg/dcolumn>
13. EndNote: <http://endnote.com/>
14. enumerate: <http://www.ctan.org/pkg/enumerate>
15. enumitem: <http://www.ctan.org/pkg/enumitem>
16. epstopdf: <http://www.ctan.org/pkg/epstopdf>
17. fancyref: <http://www.ctan.org/pkg/fancyref>
18. fancyhdr: <https://www.ctan.org/pkg/fancyhdr>
19. fontenc: <http://www.ctan.org/pkg/fontenc>
20. footnote: <http://www.ctan.org/pkg/footnote>
21. forest: <https://www.ctan.org/pkg/forest>
22. gb4e: <http://www.ctan.org/pkg/gb4e>
23. geometry: <https://www.ctan.org/pkg/geometry>
24. graphicx: <http://www.ctan.org/pkg/graphicx>
25. hyperref: <https://www.ctan.org/pkg/hyperref>
26. inputenc: <http://www.ctan.org/pkg/inputenc>
27. JabRef: <http://jabref.sourceforge.net/>
28. KOMA-Script: <http://www.ctan.org/pkg/koma-script>

29. LaTeXLinguisten: http://homepage.ruhr-uni-bochum.de/Alexander.Linke-2/linguistik/LaTeX/downloads/latex_fuer_linguisten.pdf
30. linguex: <http://www.ctan.org/pkg/linguex>
31. listings: <http://www.ctan.org/pkg/listings>
32. lmodern: <http://www.ctan.org/tex-archive/fonts/lm/>
33. natbib: <https://ctan.org/pkg/natbib>
34. longtable: <http://www.ctan.org/pkg/longtable>
35. ltxtable: <http://www.ctan.org/pkg/ltxtable>
36. MacTeX: <http://www.tug.org/mactex/>
37. mdwlist: <http://www.ctan.org/pkg/mdwlist>
38. microtype: <https://www.ctan.org/pkg/microtype>
39. MiKTeX: <http://miktex.org/>
40. multirow: <http://www.ctan.org/pkg/multirow>
41. overpic: <https://www.ctan.org/pkg/overpic>
42. paralist: <http://www.ctan.org/pkg/paralist>
43. pgf: <http://www.ctan.org/pkg/pgf>
44. qtree: <https://www.ctan.org/pkg/qtree>
45. rotating: <http://www.ctan.org/pkg/rotating>
46. scrpage2: <https://www.ctan.org/pkg/scrpage2>
47. setspace: <https://www.ctan.org/pkg/setspace>
48. tabularx: <http://www.ctan.org/pkg/tabularx>
49. TeXLive: <http://www.tug.org/texlive/>
50. Texmaker: <http://www.xmlmath.net/texmaker/>
51. TexmakerGerman: <http://www.xmlmath.net/texmaker/UserManualGerman.pdf>
52. theorem: <http://www.ctan.org/pkg/theorem>
53. tikz-qtree: <https://www.ctan.org/pkg/tikz-qtree>
54. tipa: <https://www.ctan.org/pkg/tipa>

- 55. typearea: <https://www.ctan.org/pkg/typearea>
- 56. ulem: <http://www.ctan.org/pkg/ulem>
- 57. varioref: <http://www.ctan.org/pkg/varioref>
- 58. xcolor: <http://www.ctan.org/pkg/xcolor>
- 59. xspace: <https://www.ctan.org/pkg/xspace>
- 60. wikibook: <http://de.wikibooks.org/wiki/LaTeX-Kompendium>

Literatur

- Butler, L. K., J. Bohnemeyer und T. F. Jaeger (2014). Syntactic constraints and production preferences for optional plural marking in Yucatec Maya. In A. Machicao y Priemer, A. Nolda, und A. Sioupi (Hrsgg.), *Zwischen Kern und Peripherie: Untersuchungen zu Randbereichen in Sprache und Grammatik*, S. 181–207. Berlin: De Gruyter.
- Chomsky, N. (1995). *The Minimalist Program*. Cambridge, MA: MIT Press.
- Comrie, B., M. Haspelmath und B. Bickel (2008, 05.02.2008). Leipzig Glossing Rules. (<http://www.eva.mpg.de/lingua/resources/glossing-rules.php> Zugriff: 24.06.2013).
- Finkbeiner, R. und J. Meibauer (2014). ‘Festschrift oder nicht Festschrift’: Syntax, Semantik und Pragmatik einer peripheren Konstruktion. In A. Machicao y Priemer, A. Nolda, und A. Sioupi (Hrsgg.), *Zwischen Kern und Peripherie: Untersuchungen zu Randbereichen in Sprache und Grammatik*, S. 69–88. Berlin: De Gruyter.
- Freitag, C. (2012). A Processing Account of Syntactic Islands and Grammaticalness: Towards a Reduction of Grammar. Masterarbeit, Humboldt-Universität zu Berlin.
- Geach, P. T. (1962). *Reference and Generality: An Examination of Some Medieval and Modern Theories*. Ithaca, NY: Cornell University Press.
- Knuth, D. E. (1986). *The TeXbook*. Boston, MA: Addison-Wesley.
- Krifka, M. (2014). *Ja, nein, doch* als sententiale Anaphern und deren pragmatische Optimierung. In A. Machicao y Priemer, A. Nolda, und A. Sioupi (Hrsgg.), *Zwischen Kern und Peripherie: Untersuchungen zu Randbereichen in Sprache und Grammatik*, S. 41–67. Berlin: De Gruyter.
- Nolda, A., A. Machicao y Priemer und A. Sioupi (2014). Die Kern/Peripherie-Unterscheidung: Probleme und Positionen. In A. Machicao y Priemer, A. Nolda, und A. Sioupi (Hrsgg.), *Zwischen Kern und Peripherie: Untersuchungen zu Randbereichen in Sprache und Grammatik*, S. 9–23. Berlin: De Gruyter.

- Ross, J. R. (1967). *Constraints on Variables in Syntax*. Ph. D. thesis, MIT, Cambridge, MA.
- Simon, H. A. (1974). How Big is a Chunk? *Science* 183, 482–488.
- Wiese, H. und M. Piñango (2014). Computations in the mental lexicon: Noun classes and the mass/count distinction. In A. Machicao y Priemer, A. Nolda, und A. Sioupi (Hrsgg.), *Zwischen Kern und Peripherie: Untersuchungen zu Randbereichen in Sprache und Grammatik*, S. 209–235. Berlin: De Gruyter.

Index

—, 67
 α , 62
 β , 62
 δ , 62
 ∞ , 60
 π , 60
 \rightarrow , 60
*, 51
*singlespace, 70
-, 66
—, 67
—, 67
../, 34
.aux, 10, 47
.bbl, 47
.bib, 46–48
.doc, 9
.eps, 34
.jpg, 9, 34
.log, 10
.pdf, 34, 71, 72
.png, 34
.ps, 34
.sty, 17
.tex, 9, 17, 20, 34, 47, 50
.txt, 9, 46
?, 51
??, 43, 47
[], 11, 60
#, 12, 51
\$, 12, 55, 60
%, 12, 14, 51
&, 12, 35
—, 12
 $\hat{\ } , 13, 55, 60$
 \sim , 13, 44, 64, 65, 73
 \backslash , 11, 12, 60, 63, 74
 $\backslash-$, 72
 $\backslash[$, 60, 62
 $\backslash]$, 60, 62
 $\backslash,$, 64, 65, 74
 $\backslash^$, 13
 \backsim , 13
 $\backslash\backslash$, 12, 19, 24, 35, 53, 56, 58, 74
|, 13, 36
>, 12
<, 12
{ } , 11, 12, 55, 76
' , 66
“ ”, 66
10pt, 15
a4paper, 16
abbrev, 47
abstract, 28
abstracton, 28
Alph, 71
alph, 71
alpha, 47, 62
amsmath, 60, 62, 78
angle, 33
ansinew, 16
apalike, 49
applemac, 16
arabic, 71
array, 36, 39, 78
article, 15, 48
ascii, 16
author, 17, 48, 68, 71
author(year), 49
author-page, 49
avm, 60, 79
avm+, 60, 79
b, 35, 40
babel, 16, 41, 45, 50, 66, 72, 79
babelbib, 50, 79
bb, 34
beamer, 51
begin, 11, 12, 32
beta, 62
bfseries, 76

bibliography, 46, 47
 bibliographystyle, 46, 47, 49
 BibTeX, 20, 45–48, 50, 79
 Bindekorrektur, 70
 book, 15, 19, 46, 48, 68
 bookmarksnumbered, 72
 booktabs, 37, 79
 bottomrule, 37

 c, 35
 calc, 69
 caption, 41, 43
 center, 29, 30
 centering, 29, 41
 centerline, 30
 cgloss4e, 50, 52
 cha, 44
 chap, 44
 chapter, 19
 chicago, 49
 Citavi, 50, 79
 cite, 46, 47
 citealp, 49
 citealt, 49
 citeauthor, 49
 citep, 49
 citet, 49
 citeyear, 49
 citeyearpar, 49
 cleardoublepage, 43
 clearpage, 43, 73
 cline, 37
 clip, 34
 cmidrule, 37
 colortable, 79
 colortbl, 39
 compactitem, 26
 csquotes, 66, 79
 CTAN, 16
 current, 69

 date, 17, 68
 dcolumn, 39, 79
 Deklaration, 18

 delta, 62
 description, 22–25
 DIV, 69
 DIVcalc, 69
 document, 15
 documentclass, 15, 28, 41
 dots, 65
 doublespacing, 70
 draft, 68

 editor, 48
 emph, 18
 emphasize, 48
 empty, 71
 enc, 54
 end, 11, 12, 15, 32
 EndNote, 50, 79
 enlargethispage, 73
 enumerate, 22–26, 45, 79
 enumitem, 26, 79
 epstopdf, 34, 79
 eqn, 44
 eqref, 60
 equation, 60, 62
 equation*, 62
 ex., 51
 exg., 52, 53

 fancyhdr, 71, 79
 fancyref, 45, 79
 fig, 42, 44
 figure, 40
 flushleft, 30
 flushright, 31
 fn, 44
 fontenc, 16, 54, 79
 footinclude, 69
 footnote, 20, 79
 footnotesize, 18
 forest, 51, 55–58, 79
 forestset, 56, 57
 frac, 60

 g, 52
 gb4e, 50–52, 79

geometry, 69, 79
german, 66
glq, 66
glqq, 66
graphicspath, 34, 68
graphicx, 33, 34, 68, 79
grq, 66
grqq, 66

h, 40
halfparskip, 69
headinclude, 69
headings, 71
height, 33
hline, 37
htbp, 41
HTML, 6
Huge, 18
huge, 18
hyperref, 72, 79
hyphenation, 72
Hyphentation, 72

ID, 43, 44, 46, 47
include, 67, 68
includegraphics, 33–35
includeonly, 68
incollection, 48
infty, 60
inpara, 26
inparadesc, 26
inparaenum, 26
inparaitem, 26
inproceedings, 48
input, 67
inputenc, 11, 16, 79
item, 23, 26, 43, 44
itemize, 22–24
itemize*, 26

JabRef, 50, 79

keywords, 72
KOMA-Script, 15, 16, 41, 68, 79

l, 35

label, 41–44
LARGE, 18
Large, 18
large, 18
LaTeX, 6, 9, 10
LaTeXLinguisten, 80
latin1, 16
leftline, 30
letter, 15
letterpaper, 16
lim, 60
linespread, 71
linguex, 50–52, 54, 80
list, 23, 25
listings, 28, 80
listoffigures, 43
listoftables, 43
lmodern, 17, 80
longtable, 39, 80
ltxtable, 39, 80

macce, 16
MacTeX, 10, 80
maketitle, 17, 18, 68
mbox, 73
mdwlist, 26, 80
microtype, 63, 80
midrule, 37
MiKTeX, 5, 10, 16, 17, 80
misc, 48
multicolumn, 37
multirow, 37, 39, 80
myheadings, 71

natbib, 49, 77, 80
newcommand, 73
newenvironment, 77
newline, 19
newpage, 43, 73
newtheorem, 26
ngerman, 66
noindent, 19
normalem, 18
normalsize, 18

notitlepage, 68
onecolumn, 69
onehalfspacing, 70
oneside, 69
overpic, 34, 80

p, 41
p{length}, 35
pagenumbering, 71
pageref, 43
pagestyle, 71
paper=a4, 16
paper=letter, 16
par, 19, 24
paragraph, 19
paralist, 26, 80
parindent, 69
parskip, 69
part, 19, 44
PDFLaTeX, 9, 20, 34, 47
pgf, 35, 80
phdthesis, 48
pi, 60
plain, 47, 71
providecommand, 73
publishers, 68

qquad, 64
qroof, 58
qtree, 57, 58, 80
quad, 64
quotation, 21
quote, 21

r, 35
raggedleft, 31, 41
raggedright, 30, 41
ref, 43, 44, 46, 60
renewcommand, 73
renewenvironment, 77
report, 15, 19, 68
rightline, 31
Roman, 71
roman, 71

rotatebox, 34
rotating, 40, 80

safe, 54
scale, 33
scalebox, 34
Schnelles Übersetzen, 20, 47
Schriftgröße, 70
scrartcl, 15, 28
scrbook, 15, 19
scriptsize, 18
scrlttr2, 15
scrpage2, 71, 80
scrreprt, 15, 19
sec, 44
section, 19, 27, 41, 43
setcounter, 20
setspace, 70, 80
sidewaystable, 40
singlespacing, 70
slash, 72
small, 18
sn edges, 56
sout, 18
sqrt, 60
subject, 68
subparagraph, 19
subsection, 19, 27
subsubsection, 19, 20
subtitle, 17
sum, 60

t, 35, 40
T1, 54
T3, 54
tab, 42, 44
table, 39, 40
tablecaptionabove, 41
tableofcontents, 20
tabular, 35, 36
tabular*, 36
tabularx, 36, 39, 80
tbp, 41
TeX, 6

TeXLive, 10, 80
Texmaker, 5, 9–12, 15, 16, 18–21, 26, 32,
43, 47, 62, 80
TexmakerGerman, 10, 80
text, 62
textasciicircum, 13
textasciitilde, 13
textbackslash, 12
textbar, 13
textbf, 18
textgreater, 12
textit, 18
textless, 12
textrm, 61, 62
textsc, 18
textsl, 18
textsuperscript, 18
texttt, 18
theorem, 27, 80
thispagestyle, 71
TikZ, 35
tikz-qtree, 58, 80
tiny, 18
tipa, 54, 80
title, 17, 48, 68
titlehead, 68
titlepage, 68
to, 60
tocdepth, 20
toprule, 37
Tree, 57
triangle, 56
trim, 34
true, 68
twocolumn, 69
twoside, 69
typearea, 69, 81

ulem, 18, 81
Umgebung, 18
underline, 18
unpublished, 48
unsrt, 47
usepackage, 16

utf-8, 11, 16, 28

varioref, 45, 81
verb, 28
verbatim, 27, 28
verse, 29
vert, 13
viewport, 34
vline, 37

width, 33
wikibook, 81
WYGIWYM, 6
WYGIWYN, 6
WYSIWYG, 6

xcolor, 39, 81
xspace, 64, 74, 81