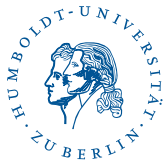


HUMBOLDT-UNIVERSITÄT ZU BERLIN



L^AT_EX for Linguists

L^AT_EX 1: Basics

Sebastian Nordhoff & Antonio Machicao y Priemer

www.linguistik.hu-berlin.de/staff/amp

LOT 2019, Amsterdam

December 31, 2018

Inhaltsverzeichnis

1 What is L^AT_EX?

- History
- WYSIWYG vs. WYGIWYN
- Examples
- How does L^AT_EX work?

2 Document structure 1

- Dokumentklasse

3 Befehle

4 Zeichen & Umbrüche

- Zeichen & Sonderzeichen
- Leerzeichen & Zeilenumbrüche

5 (Aus-)Kommentieren

6 Hausaufgabe 0

Grundlage & empfohlene Lektüre

...basierend auf Freitag und Machicao y Priemer (2015) und auf Machicao y Priemer und Kerkhof (2016)

→ [LINK](#)

1 What is L^AT_EX?

2 Document structure 1

3 Befehle

4 Zeichen & Umbrüche

5 (Aus-)Kommentieren

6 Hausaufgabe 0

History

- $\tau\epsilon\chi$ (T_EX) was developed between 1977 and 1986 by Donald E. Knuth.
- ℒ_AT_EX is an interface with helpful macros for the T_EX system. It was written by Leslie Lamport (= **L**amport **T**_E**X**).
- Pronunciation: [ˈlaɪ.tɛç], [ˈleɪ.tɛç], [ˈleɪ.tɛkh]
- ℒ_AT_EX works with markup tagging conventions – similar to HTML – to
 - define the structure of the document (e.g. chapters and sections),
 - for typographic marking (e.g. bold and italics),
 - for cross-references (e.g. citations)

WYSIWYG vs. WYGIWYN

- MS Word or Libre Office: **WYSIWYG** (*what-you-see-is-what-you-get*)

This is a headline

This word is **bold** and this one is in *italics*.

- L^AT_EX: **WYGIWYN** or **WYGIWYM** (*what-you-get-is-what-you-need/mean*)

▼ `\section{This is a headline}`

This word is `\textbf{bold}` and this one is in `\textit{italics}`.

Examples

What can you do with L^AT_EX?

Books & Articles

Discourse-level implicature: A case for QUD

Katja Jasinskaja Fabienne Salfner Constantin Freitag

August 12, 2014

Abstract

This paper argues that multi-sentence discourses give rise to Gricean quantity implicatures that go beyond the mere sum of the implicatures of the sentences they consist of. We formulate two theories of discourse-level implicature: the null theory, which only has a mechanism for sentence-level implicature and not even any notion of discourse structure; and a theory that assumes that discourse is hierarchically structured by questions under discussion (QUD) and that questions under discussion can guide the derivation of quantity implicatures at all levels of discourse structure. In two experiments using the inference task paradigm and focusing on sequences of sentences with contrastive topic, the QUD-based theory is shown to make more accurate predictions than the null theory. This finding provides additional motivation for the QUD-based approach to discourse structure.

1 Introduction

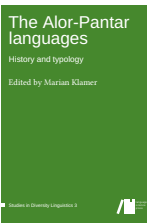
The goal of this paper is to show how the notion of Question under Discussion (QUD) can be employed to describe the phenomenon of discourse-level implicatures. In the classical picture of pragmatics going back to Paul Grice (1975), implicatures are defeasible inferences that result from reasoning about what the speaker has said, has not said, could have said, and what he or she must have meant, on the assumption that he or she behaves cooperatively and observes the Maxims of Conversation (Quality, Quantity, Manner). The input to this reasoning process is the *speech act*, and the usual task assumption is that a speech act has roughly the size of one sentence. As a result, the bulk of work on implicatures talks about implicatures of individual sentences.

There is ongoing vivid debate on whether implicatures are generated only at the level of whole sentences, or also at a substantial level, also known as the local-globalist debate (Chierchia, 2004; Geurts and Poussencolour, 2009; Geurts, 2010; Sauerland, 2010; Chien and Specter, 2011, and others). The question that by comparison has received much less attention is whether implicatures also exist above the sentence level—for larger discourse units and whole discourses. It is a common view in discourse semantics that a discourse is itself a complex speech act that consists of simple speech acts (see e.g. Asher and Lascarides, 2003). On this view, one should expect Gricean reasoning to apply to single-sentence speech acts and multi-sentence speech acts alike, so discourse-level implicatures should be generated in much the same way as sentence-level implicatures.

Bar Geurts has recently advocated this position (Geurts, 2007, 2010). He gives examples the next way as sentence-level implicatures.

- (1) Tony: Which places did you see on your trip to Italy?

1



Language Science Press

Book

Language Science Press
P.O. Box 100
1000 Leiden, The Netherlands

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Ordering information

Trees

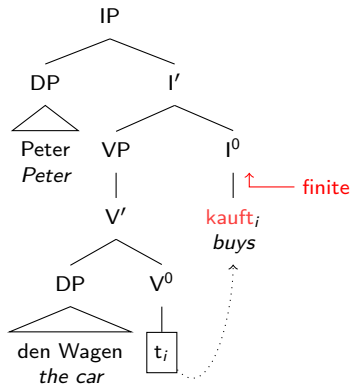


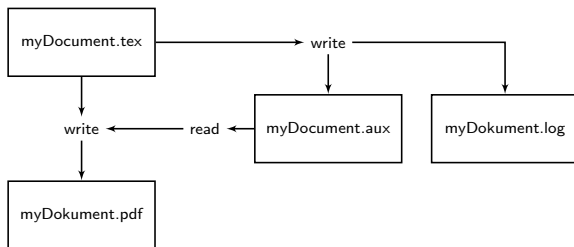
Fig. 1: Head movement

Glossing & IPA

- (1)
 - a. Der Mann schläf -t.
 the.NOM man.NOM sleep -s
 ‘The man is sleeping.’
 - b. Der Mann hat dem Jungen ein Buch über
 the.NOM man.NOM has the.DAT boy.DAT a.ACC book.ACC about
 Linguistik gegeben.
 linguistics give.PTCP.PRF
 ‘The man gave the boy a book about linguistics.’
- (2)
 - a. ⟨phonetics⟩
 - b. /fəˈnɛ.tɪks/
 - c. [fəˈnɛtɪks]

How does L^AT_EX work?

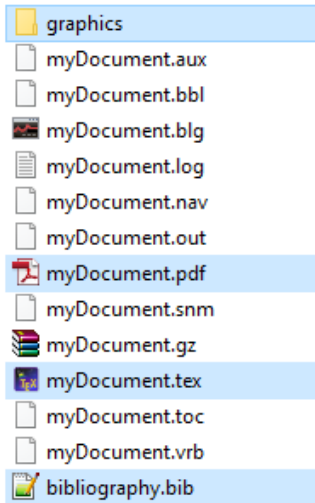
By compiling your document, L^AT_EX creates further **auxiliary files** to improve the next compilations.



- your document: `.tex`
- your product: `.pdf`

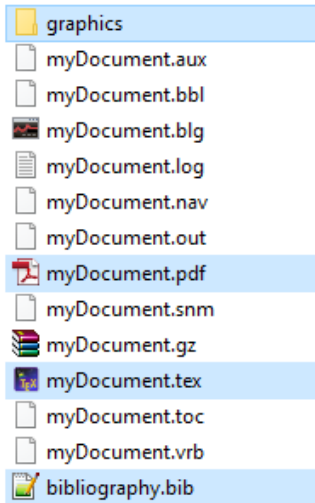
The auxiliary files can be **deleted** after your work is done. They will be created again when you compile.

- .log → information about the compiling process
- .bbl → information for the bibliography
- .nav → information for the navigation through slides
- .toc → information for the table of contents
- ...



The following files are important and **should not be deleted**. They are not created in the compiling process:

- `.tex` → this is the document you are working on.
- `.pdf` → you can delete your PDF, but this is what you normally want as your result
- `.bib` → this file contains your bibliography data base (if you have one)
- folder `graphics` → here could be your graphics (if you need some)



1 What is L^AT_EX?

2 Document structure 1

3 Befehle

4 Zeichen & Umbrüche

5 (Aus-)Kommentieren

6 Hausaufgabe 0

Document structure 1

A L^AT_EX document consists of (at least) two parts: **preamble** and **body**.

L^AT_EX preamble

part of the document where **global characteristics** of the document are specified.

- The preamble **begins** (**obligatorily**) with the `\documentclass{}` command.
- In the preamble you will install **packages** for further L^AT_EX functions.
- **Optional** (either in the preamble or in the body – preferably in the preamble)
 - your **own commands** and
 - **metadata**
- The preamble **ends** with the command `\begin{document}`.

L^AT_EX body

part of the document where **local characteristics** of the document are specified and where you write your document.

- The body **begins** with the `\begin{document}` command (end of preamble).
- The body **ends** with `\end{document}`.
- Everything following the command `\end{document}` will not be interpreted by L^AT_EX.

Exercise

Insert the following lines in your `.tex` file and compile.

```
%%%%%%%%%%
% Compile: XeLaTeX BibTeX XeLaTeX XeLaTeX
%%%%%%%%%
\documentclass{scrartcl}

%%%%%%%%%PACKAGES%%%%%%%%%
%%%%%%%%%COMMANDS%%%%%%%%%
%%%%%%%%%META DATA%%%%%%%%%
%%%%%%%%%END PREAMBLE%%%%%%%%%
%%%%%%%%%BEGIN DOCUMENT%%%%%%%%%

\begin{document}

This is my first \LaTeX file.

\end{document}

%%%%%%%%%END DOCUMENT%%%%%%%%%
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compile: PDFLaTeX BibTeX PDFLaTeX PDFLaTeX
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\documentclass[10pt,paper=a4,abstracton]{scrartcl}

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\usepackage[utf8]{inputenc}
\usepackage[english,ngerman]{babel}

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\newcommand{\obj}[1]{\emph{#1}}

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\author{Antonio Machicao y Priemer}
\title{\LaTeX\ für Linguisten}
\subtitle{Testdatei}
\date{23. Februar 2016}

```

```

\begin{document}

```

Diese Datei ist dazu gemacht, dass alle Workshopteilnehmer einige Vorzüge von \LaTeX\ austesten können.

```

\end{document}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
END DOCUMENT%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Diese Datei ist dazu gemacht, dass alle Workshopteilnehmer einige Vorzüge von L^AT_EX austesten können.

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 zur Verfügung gestellt werden (die wir verwenden werden):

- `scrbook` für Bücher
- `scrreprt` für längere Schriften mit etlichen Kapiteln, z. B. eine Dissertation
- `scrartcl` für Artikel, ohne Kapitel nur mit Abschnitten
- `scrlttr2` für Briefe

Für Details über das KOMA-Script siehe Kohm und Morawski (2014) und <https://www.komascript.de/>

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compile: PDFLaTeX BibTeX PDFLaTeX PDFLaTeX
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\documentclass[10pt,paper=a4,abstracton]{scrartcl}

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\usepackage[utf8]{inputenc}
\usepackage[english,ngerman]{babel}

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\newcommand{\obj}[1]{\emph{#1}}

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\author{Antonio Machicao y Priemer}
\title{\LaTeX\ für Linguisten}
\subtitle{Testdatei}
\date{23. Februar 2016}

```

```

\begin{document}

```

Diese Datei ist dazu gemacht, dass alle Workshopteilnehmer einige Vorzüge von \LaTeX\ austesten können.

```

\end{document}

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
END DOCUMENT%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Zudem kann man die Optionen dieser Dokumentklassen in dem `documentclass`-Befehl festlegen.

Häufigste Optionen und 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.

Dokumentklasse

```
\documentclass[10pt,paper=a4]{scrartcl}  
\begin{document}  
Text Text Text  
\end{document}
```

Hinweis: TeXstudio bietet einen Assistenten, der die Präambel für Sie schreibt. Schauen Sie in der Toolbar unter Assistenten/Assistent für ein neues Dokument nach. Dort können Sie alles weitere einstellen.

1 What is LaTeX?

2 Document structure 1

3 Befehle

4 Zeichen & Umbrüche

5 (Aus-)Kommentieren

6 Hausaufgabe 0

Befehle (commands)

- L^AT_EX arbeitet mit einer **logischen Textauszeichnung**
- Wenn Sie etwas kursiv setzen wollen, müssen Sie nicht irgendwo „klicken“, sondern **einen Befehl eingeben**

Syntax von Befehlen:

- Alle L^AT_EX-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}
```

- Beispiele:

```
\textbf{fettgedruckter Text}  
\citep[vgl.] [3--5]{Freitag&MyP15a}
```

(3) **fettgedruckter Text**

(4) (vgl. Freitag und Machicao y Priemer, 2015: 3–5)

L^AT_EX kennt im Prinzip **drei Arten von Befehlen**:

- **Einfache Befehle:**

Backslash „\“ + Befehlsnamen + optionale Argumenten in eckigen Klammern + obligatorische Argumenten in geschweiften Klammern

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

- Beispiele:

```
\textit{kursive Hervorhebung}  
\citet[vgl.] [3--5]{Freitag&MyP15a}
```

(5) *kursive Hervorhebung*

(6) Freitag und Machicao y Priemer (vgl. 2015: 3–5)

- **Umgebungen:**

Umgebungen bestehen aus einem `begin`- und einem `end`-Befehl. Der Befehl gilt im Bereich zwischen `begin` und `end`.

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

- Beispiele:

```
\begin{multicols}{2}[Beispiel]  
Das ist die erste Spalte  
  
Das ist die zweite Spalte  
\end{multicols}
```

(7) Beispiel

Das ist die erste Spalte

Das ist die zweite Spalte

- **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
```

- Beispiele:

```
{\small Hello world!} \Huge Hello world!
```

(8) Hello world! **Hello world!**

1 What is LaTeX?

2 Document structure 1

3 Befehle

4 Zeichen & Umbrüche

5 (Aus-)Kommentieren

6 Hausaufgabe 0

Zeichen & Sonderzeichen

- Die folgenden Zeichen können problemlos verwendet werden:

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

- Achten Sie darauf, welche Art von **Anführungszeichen** durch ‘ ’ “ generiert werden (vgl. Machicao y Priemer, 2018).
- Die **Umlaute** „ä, ö, Ä, Ö, ...“, **Akzente** „á, à, ...“ und das **Eszett** „ß“ können (bei PDF-L^AT_EX) mithilfe des folgenden Pakets `\usepackage[utf8]{inputenc}` direkt eingegeben werden. Andernfalls müssen sie mit Extrabefehlen geschrieben werden:

```
\"A \"O \"a \"o \'a \'o \ss{}
oder \"{A} {\\"O} {\ss}
```

(9) Ä Ö ä ö á ò ß oder Ä Ö ß

- Die folgenden Zeichen haben in L^AT_EX eine **besondere Bedeutung** und können nicht einfach im FlieSStext verwendet werden:

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

- Um diese Zeichen verwenden zu können, musst man den in L^AT_EX 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** (*pipe*) 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 später).

```
\textgreater $>$
```

```
\textless $<$
```

- Um den **senkrechten Strich** („pipe“) 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 L^AT_EX die Funktion eines geschützten Leerzeichens. 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
```

- Weiteres zu Sonderzeichen in L^AT_EX:
https://de.wikibooks.org/wiki/LaTeX/_Akzente_und_Sonderzeichen

Leerzeichen & Zeilenumbrüche

- L^AT_EX 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 konsequente 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.

Hier ein Beispiel:

Hier ist ein Beispieltext mit viel
zu vielen Leerzeichen .
In Word sind sie immer zu sehen.
Hier verwenden wir einen
Zeilenumbruch.

Zwei Zeilenumbrüche ergeben einen neuen Absatz.

Mehr als zwei Umbrüche ergeben nur einen neuen Absatz.

Hier ist ein Beispieltext mit viel zu vielen Leerzeichen . In Word sind sie immer zu sehen. Hier verwenden wir einen Zeilenumbruch.
Zwei Zeilenumbrüche ergeben einen neuen Absatz.
Mehr als zwei Umbrüche ergeben nur einen neuen Absatz.

1 What is LaTeX?

2 Document structure 1

3 Befehle

4 Zeichen & Umbrüche

5 (Aus-)Kommentieren

6 Hausaufgabe 0

(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.

Durchs Auskommentieren kann man:

- **Code/Text verstecken**, ohne ihn zu löschen;
- leichter in Zeilen oder gröSSeren Regionen **Fehler** finden;
- **Leerzeichen oder Leerzeilen** in langen Eingabezeilen **unterbinden**;
- **Kommentare in den Code schreiben**, ohne dass sie als Text gedruckt werden

(Aus-)Kommentieren

```
Hier ist etwas Code, der angezeigt werden soll.  
%hier sind wichtige Notizen
```

```
Kommentare können sogar ein Wort teilen:  
Rindfleischetikettierungs% Notiz: Fugen-s  
überwachungsaufgaben% Notiz: Fugen-n  
übertragungsgesetz.
```

Hier ist etwas Code, der angezeigt werden soll. Kommentare können sogar ein Wort teilen: Rindfleischetikettierungsüberwachungsaufgabenübertragungsgesetz.	Rindfleischetiket-
---	--------------------

1 What is L^AT_EX?

2 Document structure 1

3 Befehle

4 Zeichen & Umbrüche

5 (Aus-)Kommentieren

6 Hausaufgabe 0

Hausaufgabe 0: L^AT_EX-Vorbereitung

- Laden Sie `MiKTeX` und `TeXstudio` – wie in der Anleitung in Moodle angegeben – herunter.
- Installieren Sie beide Programme.
- Folgen Sie dabei der Anleitung in Moodle.
- Falls Sie **Probleme bei der Installation** haben, melden Sie sich bitte bei Pia Linscheid *vor* der nächsten Sitzung! Andernfalls werden Sie die kommenden Hausaufgaben nicht abgeben können.
- **Alternative:** Anstatt die Programme zu installieren, können Sie versuchen Ihre Hausaufgaben mit Overleaf zu lösen (Siehe Anleitung in Moodle).

Hausaufgabe 0: Lektüre

- Lesen Sie den **LingStudi-Guide** (s. Moodle/Allgemeines)

Hausaufgabe 0: Mitgestalten

- Schreiben Sie bei Moodle im Bereich „Was will ich in diesem Kurs lernen?“ **einen bis zwei Stichpunkte** auf. Erklären Sie kurz – wenn nötig – was Sie damit meinen.

Quellen I

- Grafik: File Extensions – xkcd, A webcomic of romance, sarcasm, math, and language
<https://xkcd.com/1301/>
[Zugriff: 10.04.2017]
- Link: Akzente und Sonderzeichen in L^AT_EX.
https://de.wikibooks.org/wiki/LaTeX/_Akzente_und_Sonderzeichen
[Zugriff: 10.10.2017]
- Software: MiKTeX
<https://miktex.org/>
[Zugriff: 10.04.2017]
- Software: TeXstudio
<https://www.texstudio.org/>
[Zugriff: 10.04.2017]

Literatur I

Freitag, Constantin und Antonio Machicao y Priemer (2015). L^AT_EX-Einführung für Linguisten. Manuskript. URL

<https://www.linguistik.hu-berlin.de/de/staff/amp/latex-einfuehrung>, Zugriff: 12.04.2015.

Knuth, Donald E. (1986). *The T_EXbook*. Boston, MA: Addison-Wesley.

Kohm, Markus und Jens-Uwe Morawski (2014). Die Anleitung: KOMA-Script.

Online-Handbuch. URL <http://www.ctan.org/pkg/koma-script>, Zugriff: 28.01.2014.

Kopka, Helmut (1994). *L^AT_EX: Einführung. Band 1*. Bonn: Addison-Wesley.

Machicao y Priemer, Antonio (2018). Hinweise für Seminararbeiten. Manuskript. URL

https://www.linguistik.hu-berlin.de/de/staff/amp/downloads/myp2018-04-06-hinweise_seminararbeit.pdf, Zugriff: 16.10.2018.

Machicao y Priemer, Antonio und Robyn Kerkhof (2016). L^AT_EX-Einführung für Linguisten – Slides. Präsentation beim 7. linguistischen Methodenworkshop an der Humboldt-Universität zu Berlin – 22.–24. Februar 2016. URL

<https://www.linguistik.hu-berlin.de/de/staff/amp/latex-einfuehrung>, Zugriff: 23.02.2016.