

Einführung in L^AT_EX

Luis Imsande

Fachschaftsrat Mathematik/Informatik
Martin-Luther-Universität Halle-Wittenberg

Impressum

Herausgeber	Fachschaftsrat Mathematik/Informatik der Studierendenschaft der Martin-Luther-Universität Halle-Wittenberg K.d.ö.R.
Autor	Luis Imsande
Kontakt	limsande@yahoo.com

Gesetzt mit \LaTeX

Inhaltsverzeichnis

1. Über dieses Dokument	2
2. Was ist LaTeX?	2
2.1. WYSIWYG vs. WYSIWYM	3
2.2. Zur Arbeit mit LaTeX	3
2.2.1. Befehle	3
2.2.2. Kommentare	4
3. Ein minimales LaTeX-Dokument	5
4. Schriftformatierung & Dokumentstruktur	6
4.1. Absätze und Umbrüche	6
4.2. Schriftgröße etc.	6
4.3. Titel & Dokumentkopf	7
4.4. Gliederung (Zwischenüberschriften)	8
5. Funktion erweitern mit Paketen	8
6. Mathematik	8
6.1. Mathematische Formeln setzen	8
6.2. Den richtigen Befehl zu einem Symbol finden	11
7. Abbildungen und Tabellen (Gleitumgebungen)	11
7.1. Abbildungen	11
7.2. Tabellen	12
7.3. Gleitumgebungen	13
8. Dokumentklassen	15
Anhang	16
A. Die wichtigsten Pakete	16
B. Wo gibt's Hilfe?	16

1. Über dieses Dokument

Dieses Dokument, inklusive sein LaTeX-Quellcode, ist frei verfügbar auf GitHub.com¹. Es ist gedacht als ausführliche Form des Crashkurses zum Textsatzsystem LaTeX der Fachschaft Mathematik/Informatik an der Martin-Luther-Universität Halle-Wittenberg. Es ist definitiv keine vollständige Beschreibung von LaTeX und all seinen Funktionen, sondern soll nur das vermitteln, was wir für den Einstieg wichtig finden. Nach dem Lesen sollten zusammen mit den Inhalten des Kurses das Erstellen von Lösungen der Übungsseries der ersten Semester gut möglich sein.

Um den Einstieg noch einfacher zu machen, hat der FSR eine Vorlage für die Übungsseries erstellt, die praktisch alle Einstellungen automatisch vornimmt. Es muss also nur noch der Text eingefügt werden. Diese Vorlage wurde im Kurs vorgestellt und steht auf unserer Website² zur Verfügung. Einiges, was hier erklärt wird, nimmt euch die Vorlage bereits ab.

Im Kurs wurde als Arbeitsumgebung der Online-LaTeX-Editor Overleaf³ vorgestellt. Eine Online-Lösung halten wir für besonders einfach, da so die ganze Administration einer eigenen LaTeX-Installation entfällt und nichts auf dem eigenen Rechner installiert werden muss. Deshalb wird hier auch keine Installationshilfe gegeben, lediglich die bekanntesten Distributionen erwähnt. Trotzdem stehen wir natürlich immer für Fragen, nicht nur zum Thema LaTeX, zur Verfügung, z. B. über unsere Webseite⁴. Außerdem bietet das Institut für Informatik den Informatik-Treff⁵ an, wo euch Studenten höherer Semester bei Fragen zur Seite stehen. In Abschnitt B am Ende des Dokuments ist eine Sammlung von Links zu LaTeX-Hilfen.

2. Was ist LaTeX?

LaTeX (sprich: Latech) ist ein Textsatzsystem. Seine Stärken liegen ganz besonders bei mathematischen und technischen Darstellungen. LaTeX ist tatsächlich eine Erweiterung des Textsatzsystems TeX aus dem Jahr 1978, welches entwickelt wurde aus dem damaligen Mangel an Möglichkeiten, mathematische und generell wissenschaftliche Texte in hoher Qualität zu setzen. Heute ist LaTeX de-facto Standard für wissenschaftliche Arbeiten. Es ist *open source*, d. h. sein Programmcode ist für jeden frei einsehbar, und für alle gängigen Betriebssysteme verfügbar. Dieses Dokument wurde übrigens auch mit LaTeX erstellt.

¹<https://github.com/Limsande/latex-intro.git>

²<https://fachschaft.mathinf.uni-halle.de>

³www.overleaf.com

⁴<https://fachschaft.mathinf.uni-halle.de/kontakt/>

⁵<https://www.informatik.uni-halle.de/studium/informatik-treff/>

2.1. WYSIWYG vs. WYSIWYM

Systeme zum Erstellen von Texten lassen sich in die Kategorien „WYSIWYG“ („*What you see is what you get*“) und „WYSIWYM“ („*What you see is what you mean*“) einteilen. Zur ersten Kategorie gehören z. B. Office-Programme wie Microsoft Word und LibreOffice. Im Grunde bedeutet „WYSIWYG“: Was ich in meinem Editor während des Schreibens sehe, ist bereits das fertige Dokument (Schreiben des Textes und Erstellen des Dokumentes ist also ein einziger Schritt).

L^AT_EX gehört zur zweiten Kategorie, bei der der Prozess aus zwei Schritten besteht: Der im Editor erstellte Text ist nur eine Rohfassung („Quelldatei“). Damit daraus ein fertiges Dokument wie bspw. ein PDF wird, muss er durch ein Programm verarbeitet („übersetzt“ oder „kompiliert“) werden. Erst dabei wird der Text mit vorher festgelegten Einstellungen formatiert (Schriftgröße, Schriftart, Hervorheben von Überschriften etc.), Abbildungen werden an vorher definierten Stellen eingefügt, Verweise generiert u.v.m.

Da die Formatierung komplett von L^AT_EX übernommen wird, kann es viele Dinge für das fertige Dokument optimieren. L^AT_EX ist z. B. bestrebt, alle Absätze, Abbildungen, Tabellen etc. so auf die einzelnen Seiten des Dokuments aufzuteilen, das jede einzelne Seite so gut aussieht wie möglich.

2.2. Zur Arbeit mit L^AT_EX

Der Quellcode ist im Grunde eine Textdatei (`.txt`) mit der Endung `.tex` und kann daher prinzipiell mit jedem Texteditor erstellt werden. Spezielle L^AT_EX-Editoren wie Texmaker⁶ erleichtern einem das Leben allerdings mit einigen Features wie intelligenter Vervollständigung und automatischem Syntaxcheck. Webbasierte Editoren wie Overleaf⁷ bieten außerdem die Möglichkeit, Dokumente gemeinsam in Echtzeit zu bearbeiten.

Möchte man L^AT_EX lieber lokal auf dem eigenen Rechner benutzen (was wir für den Einstieg nicht empfehlen), muss etwas Software installiert werden. Zwei gängige Distributionen des gesamten L^AT_EX-Systems, das zur Übersetzung gebraucht wird, sind TeX Live⁸ und MiKTeX⁹.

2.2.1. Befehle

Das Programm, das aus L^AT_EX-Quellcode ein PDF erstellt, heißt `pdflatex`. Beim Übersetzen durchsucht es die Quelldatei nach speziellen Wörtern („Befehlen“), die ihm sagen, wie

⁶<https://www.xmlmath.net/texmaker/>

⁷<https://www.overleaf.com/>

⁸<http://tug.org/texlive/>

⁹<https://miktex.org/>

es den Text verarbeiten soll. Ein L^AT_EX-Befehl beginnt mit einem Backslash (\). Beispiele sind `\tableofcontents`, `\textbf{}` und `\begin{equation}`. An diesen Beispielen sind drei verschiedene Typen von L^AT_EX-Befehlen zu sehen:

- Einfache, parameterlose Befehle stehen für sich und können verschiedenste Auswirkungen haben. `\tableofcontents` z. B. erzeugt ein Inhaltsverzeichnis des Dokuments.
- Befehle mit Parameter(n) benötigen zusätzliche Angaben, um zu funktionieren. Diese werden in geschweiften Klammern ({}) übergeben: `\textbf{Hallo}` führt dazu, dass „Hallo“ (der Parameter) fett gedruckt wird, `\section{Thema X}` erzeugt eine Überschrift mit dem Namen „Thema X“.

Befehle können außerdem *optionale* Parameter haben, d. h. zusätzliche Parameter, die nicht unbedingt angegeben werden müssen, der Befehl funktioniert auch ohne sie. Diese werden in eckigen Klammern ([]) geschrieben. Ein Beispiel ist `\includegraphics[width=10cm]{bild.png}`, wodurch das Bild „bild.png“ in das Dokument eingefügt und zusätzlich auf eine Breite von 10 cm skaliert wird. Ohne `[width=10cm]` würde das Bild in der Originalgröße eingefügt.

- Befehle zum Öffnen und Schließen von Umgebungen: `\begin{}` und `\end{}`. Eine Umgebung ist ein Bereich, in dem bestimmte Regeln für die Formatierung gelten. Die `equation`-Umgebung z. B. stellt ihren Inhalt als einzeilige mathematische Gleichung da, zentriert mit etwas Abstand darüber und darunter (wie man sie benutzt, wird in Abschnitt 6 erklärt). Einige Befehle sind nur in einer bestimmten Umgebung erlaubt.

Ganz allgemein markieren Befehle also Textstellen und bestimmen so, wie diese später im fertigen Dokument aussehen werden. Diese Form des Textsatzes nennt man auch *logisches Markup*.

Stößt `pdflatex` beim Übersetzen auf einen unbekannten Befehl (z. B. wegen eines Tippfehlers) oder eine andere Unstimmigkeit (beliebt sind fehlende schließende Klammern), bricht es mit einer Fehlermeldung ab und erzeugt kein PDF. In der Meldung steht meistens die Zeilennummer, in der der Fehler entdeckt wurde, diese ist jedoch nicht immer korrekt, da die eigentliche Ursache bereits früher aufgetreten sein kann.

2.2.2. Kommentare

Eine besondere Bedeutung hat das Prozentzeichen (%). Immer wenn `pdflatex` diesem Zeichen begegnet, ignoriert es alles weitere bis zum Ende der aktuellen Zeile. Text, der nach einem % steht, ist für das Programm also praktisch unsichtbar. Das ist eine einfache Möglichkeit, sich selbst Notizen in den Quellcode zu schreiben, wie `% Todo` oder `% geklaut von stackoverflow.com`. Zur Benutzung von Kommentaren siehe Abschnitt 4.

3. Ein minimales LaTeX-Dokument

Jedes LaTeX-Dokument muss mindestens die folgenden Dinge beinhalten:

Code 1: Minimales LaTeX-Dokument

```
1 \documentclass{article}
2 \begin{document}
3 % Was hier steht kommt ins Dokument.
4 Hallo Welt!
5 \end{document}
```

Hallo Welt!

Der Text in den Zeilen 3 und 4 ist natürlich austauschbar. An dieser Stelle wird der tatsächliche Inhalt des Dokuments eingefügt. Wie bereits erwähnt, erscheinen Kommentare nicht im fertigen PDF. Zeilenumbrüche, die mit \leftarrow gekennzeichnet sind, existieren nicht im Quellcode. Sie kommen nur wegen des begrenzten Platzes für die Darstellung zustande.

Zeile 1 Der Befehl `\documentclass{}` muss immer der erste LaTeX-Befehl eines Dokuments sein. Er definiert, um was für eine Art von Dokument es sich handeln soll. Die Klasse `article` z. B. steht für einen kurzen Text, wie eine wissenschaftliche Publikation in einer Fachzeitschrift (*Paper*). Für längere Aufsätze, wie Fach- und Abschlussarbeiten, eignet sich `report`. Die Wahl der Klasse hat vor allem Auswirkungen darauf, wie LaTeX das Dokument formatiert (siehe auch Kapitel 8). Für Übungsserien an der MLU eignet sich die Vorlage des FSR besonders gut.

Zeile 2 Der Befehl `\begin{document}` markiert den Anfang des eigentlichen Dokuments. Alles ab hier wird später im PDF zu sehen sein.

Zeile 3 Zwischen `\begin{document}` und `\end{document}` kann Text geschrieben, Bilder eingefügt werden u.v.m. (die sogenannte `document`-Umgebung).

Zeile 4 Der Befehl `\end{document}` signalisiert LaTeX, dass das Dokument zu Ende ist. Alles danach wird ignoriert.

Durch das Übersetzen dieses Codes erhält man eine einzelne Seite mit dem Satz „Hallo Welt!“ wie rechts neben dem LaTeX-Code dargestellt.

4. Schriftformatierung & Dokumentstruktur

4.1. Absätze und Umbrüche

Zwei Absätze werden durch eine komplett leere Zeile im Quellcode gekennzeichnet. Zeilenumbrüche müssen i. d. R. nicht von Hand eingefügt werden, da LaTeX automatische Silbentrennung beherrscht. Will man eine Zeile trotzdem an einer bestimmten Stelle umbrechen, geht dies mit `\`. Ein „einfacher“ Zeilenumbruch durch drücken der Entertaste funktioniert nicht, da LaTeX diesen nur als ein Leerzeichen ansieht. Ein Seitenumbruch lässt sich mit `\pagebreak` oder `\newpage` erzeugen.

Code 2: Umbrüche und Kommentare

```
1 Ein Absatz.
2
3 Noch ein Absatz. Und ein\\
4 Zeilenumbruch. Und kein
5 Zeilenumbruch.
```

Ein Absatz.

Noch ein Absatz. Und ein
Zeilenumbruch. Und kein Zeilenumbruch.

4.2. Schriftgröße etc.

Die Schriftgröße (voreingestellt ist meistens 12) lässt sich u. a. mit den Befehlen `\tiny` und `\huge` verändern. Zurücksetzen auf die normale Größe geht mit `\normalsize`. Die Auswirkung lässt sich auch durch geschweifte Klammern (`{}`) begrenzen.

Code 3: Schriftgröße

```
1 Dieser Text ist normal groß.
2 Jetzt wird's \tiny winzig.
3 Groß wird es \huge jetzt.
4 Zurück zu normal: \normalsize
5 ab hier. Einzelnes {\tiny ←
  Wort}
6 geht so.
```

Dieser Text ist normal groß. Jetzt wird's
winzig. Groß wird es **jetzt**. Zurück zu
normal: ab hier. Einzelnes Wort geht
so.

Der Schrifttyp, also fett, kursiv etc., lässt sich über folgende Befehle einstellen; dabei steht „bf“ für *boldface*, „it“ für *italics* und „tt“ für *teletype* (die englischen Begriffe für fett, kursiv und Schreibmaschinenschrift):

Code 4: Schrifttyp

```
1 Dieses Wort ist ↵
   \textbf{fett},
2 \textit{kursiv},
3 \textbf{\textit{fett-kursiv}},
4 \texttt{monospace}.
```

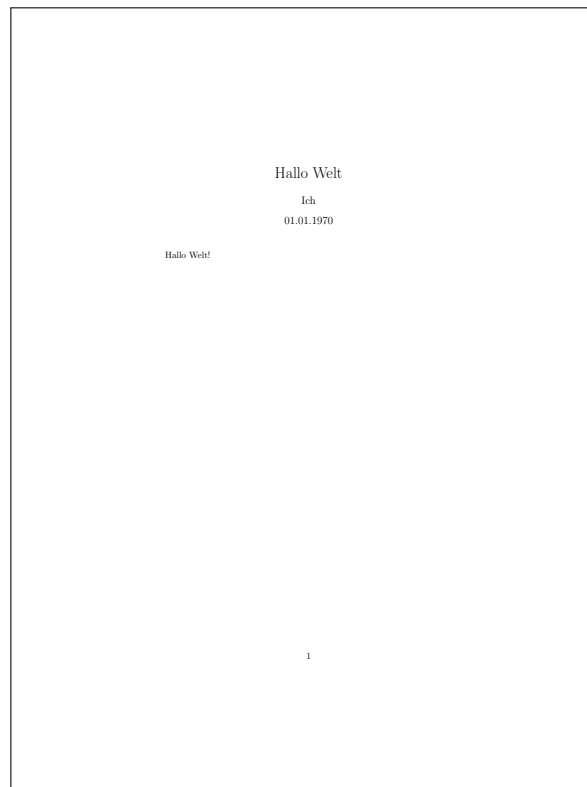
Dieses Wort ist **fett**, *kursiv*, **fett-kursiv**,
monospace.

4.3. Titel & Dokumentkopf

Das Dokument lässt sich mit einem Kopf versehen bestehend aus Titel, Autor und Datum mit Hilfe des Befehls `\maketitle`. Um zu funktionieren, müssen dafür natürlich Titel, Autor und Datum angegeben werden. Das geht mit den Befehlen `\title{Beispieltitel}`, `\author{Beispielautor}` und `\date{Beispieldatum}`. Diese müssen *vor* `\maketitle` stehen. Der Kopf wird genau da eingefügt, wo `\maketitle` im Quellcode steht (in aller Regel direkt nach `\begin{document}`). Das Hallo-Welt-Beispiel sieht mit Dokumentkopf so aus:

Code 5: Dokumentkopf

```
1 \documentclass{article}
2 \author{Ich}
3 \title{Hallo Welt}
4 \date{01.01.1970}
5 \begin{document}
6   \maketitle
7   Hallo Welt!
8 \end{document}
```



Soll kein Datum ausgegeben werden, kann `\date{}` auch leer gelassen werden. Wird `\date{}` ganz weggelassen, wird automatisch das aktuelle Datum eingefügt.

4.4. Gliederung (Zwischenüberschriften)

Die Gliederung des Dokuments (also einzelne Abschnitte mit Überschriften) kann mit den Befehlen `\section{Name}` und `\subsection{Name}` vorgenommen werden. Tiefere Gliederungsebenen gibt es auch, die benötigt man aber fast nie und werden deshalb nicht weiter erwähnt. Überschriften werden automatisch in Schriftgröße und -typ vom Text abgehoben und mit einer fortlaufenden Nummer versehen. Letzteres lässt sich durch die Varianten `\section*{Name}` und `\subsection*{Name}` unterdrücken. Mit dem Befehl `\tableofcontents` kann ein Inhaltsverzeichnis inklusive Seitenangaben erstellt werden.

5. Funktion erweitern mit Paketen

Obwohl in LaTeX standardmäßig schon viel Funktionalität enthalten ist, reicht diese nicht immer aus. Für das Einfügen von Bildern oder komplexen Formelsatz (um nur zwei Beispiele zu nennen) muss sie erweitert werden. Das geht durch das explizite Laden von Erweiterungen („Pakete“) mit dem Befehl `\usepackage{}`, dem der Paketname übergeben wird, also z. B. `\usepackage{graphicx}` zum Laden des Pakets *graphicx*. Der Befehl muss nach `\documentclass{}`, aber vor `\begin{document}` gegeben werden. Dieser Bereich heißt auch *Präambel*. Manche Pakete erwarten auch Optionen. Diese können wie üblich in eckigen Klammern ([]) angegeben werden, wie in `\usepackage[utf8]{inputenc}`. Pakete können neue Befehle zur Verfügung stellen, schon bestehende modifizieren oder generell das Verhalten von LaTeX beeinflussen.

Alle Pakete, die hier vorgestellt werden, sind in der FSR-Vorlage bereits enthalten. Wenn diese also als Dokumentklasse verwendet wird, müssen sie nicht extra geladen werden. Für das Erstellen eigener Dokumente ohne die Vorlage ist eine Liste mit den wichtigsten Paketen in Anhang A zu finden.

6. Mathematik

Im normalen Textmodus sind in LaTeX nur einige wenige Rechenzeichen erlaubt, wie + und =. Alles andere würde einen Fehler erzeugen. Deswegen gibt es noch spezielle Mathemodi für das Setzen von mathematischen Ausdrücken.

6.1. Mathematische Formeln setzen

Der einfachste Mathemodus ist der sogenannte *inline-Modus*. Aktiviert wird er mit `\(` und mit `\)` beendet. (`$ $` funktioniert auch). Wie unten zu sehen ist, gelten im Mathemodus andere Regeln, weshalb darin kein Text gesetzt werden sollte (z.B. werden

Leerzeichen ignoriert und eine andere Schriftart verwendet). Wie sich Text trotzdem richtig darstellen lässt, wird weiter unten bei der Einführung der `align`-Umgebung erklärt.

Code 6: Text- vs. Mathemodus

```
1 Hier Textmodus, jetzt kommt
2 eine Formel im Mathemodus:
3 \(\sum_{i=1}^n i^2\).
4 Hier ist wieder Text.
5
6 Noch mal Mathe:
7 $\int_{i=1}^n i^2$
8
9 \(\text{Text im Mathemodus} \leftarrow
   \text{sieht nicht gut aus!}\)
```

Hier Textmodus, jetzt kommt eine Formel im Mathemodus: $\sum_{i=1}^n i^2$. Hier ist wieder Text.

Noch mal Mathe: $\int_{i=1}^n i^2$

TextimMathemodussiehtnichtgutaus!

Mit dem inline-Modus werden besonders große Symbole (wie oben das Summenzeichen Σ) extra kompakt dargestellt um besser in den Fließtext zu passen. Um z. B. eine wichtige Formel besser aus dem Text heraustechen zu lassen, gibt es noch den *abgesetzten* Mathemodus. Er wird mit `\[` und `\]` ein- bzw. ausgeschaltet. Damit können Ausdrücke zentriert gesetzt werden, mit extra Platz darüber und darunter.

Code 7: Abgesetzter Mathemodus

```
1 Hier ist Text. Jetzt folgt
2 die gleiche Formel wie oben:
3 \[ \sum_{i=1}^n i^2 \]
4 Hier ist wieder Text.
```

Hier ist Text. Jetzt folgt die gleiche Formel wie oben:

$$\sum_{i=1}^n i^2$$

Hier ist wieder Text.

Häufig möchte man besonders wichtige Formeln mit einer Nummer versehen, um sich im Text darauf beziehen zu können. Das übernimmt die `equation`-Umgebung.

Code 8: `equation`-Umgebung

```
1 Hier ist Text. Jetzt folgt
2 noch mal die gleiche Formel
3 wie oben, automatisch mit
4 einer Nummer versehen:
5 \begin{equation}
6   \sum_{i=1}^n i^2
7 \end{equation}
8 Hier ist wieder Text.
```

Hier ist Text. Jetzt folgt noch mal die gleiche Formel wie oben, automatisch mit einer Nummer versehen:

$$\sum_{i=1}^n i^2 \tag{1}$$

Hier ist wieder Text.

Die `equation`-Umgebung kann nur eine einzelne Zeile darstellen. Für lange, mehrzeilige Passagen, wie Formelumstellungen und -herleitungen, gibt es die `align`-Umgebung aus dem Paket *amsmath*. Sie kann außerdem die Zeilen aneinander ausrichten, sodass

alles schön untereinander steht (*align* = ausrichten). Wie oben beschrieben, muss vor ihrer Benutzung das Paket *amsmath* mit `\usepackage{amsmath}` eingebunden werden.

Code 9: align-Umgebung

```
1 % vor \begin{document}:
2 % \usepackage{amsmath}
3 \begin{align}
4     1+2+\dots+n
5     &= \sum_{i=1}^n i \\
6     &= \frac{n(n+1)}{2}
7 \end{align}
```

$$1 + 2 + \dots + n = \sum_{i=1}^n i \quad (2)$$

$$= \frac{n(n+1)}{2} \quad (3)$$

Dabei muss jede Zeile mit `\\` beendet werden. Für die korrekte Ausrichtung muss jede Zeile ein `&` enthalten. Die Zeilen werden dann so gesetzt, dass alle `&` direkt untereinander stehen. Die automatische Nummerierung kann ausgeschaltet werden durch Ersetzen von `align` durch `align*`.

Hilfreich ist auch die Zeichenfolge `&&`: Damit wird der Rest der Zeile so weit nach rechts gerückt wie möglich. So lassen sich Anmerkungen einfügen, wie z. B. gemachte Umformungen. Damit auch Text im Mathemodus korrekt gesetzt wird, muss er durch den Befehl `\text{}` eingeschlossen werden (ebenfalls aus dem *amsmath*-Paket).

Code 10: align-Umgebung (II)

```
1 \begin{align*}
2     1+2+\dots+n
3     &= \sum_{i=1}^n i
4     && \text{\text{|erste n Zahlen} \leftarrow}
        \text{Zahlen}} \\
5     &= \frac{n(n+1)}{2}
6     && \text{\text{|nach Gauß}}
7 \end{align*}
```

$$1 + 2 + \dots + n = \sum_{i=1}^n i \quad \text{|erste n Zahlen}$$

$$= \frac{n(n+1)}{2} \quad \text{|nach Gauß}$$

Häufig möchte man auch eine ganze Textpassage in eine Herleitung einfügen. Damit dabei nicht die Ausrichtung kaputt geht, gibt es den Befehl `\intertext{}`. Damit geht die Ausrichtung nach dem Einschub weiter:

Code 11: align-Umgebung (III)

```

1 Das ist die Summe der ↵
   ersten n Zahlen:
2 \begin{align*}
3   1+2+...+n
4   &= \sum_{i=1}^n i
5   \intertext{Nun formen ↵
   wir um nach Gauss:}
6   &= \frac{n(n+1)}{2}
7 \end{align*}

```

Das ist die Summe der ersten n Zahlen:

$$1 + 2 + \dots + n = \sum_{i=1}^n i$$

Nun formen wir um nach Gauss:

$$= \frac{n(n+1)}{2}$$

6.2. Den richtigen Befehl zu einem Symbol finden

Die Befehle für einige mathematische Symbole sind einfach die englischen Bezeichnungen wie `\sum` für Summe, oder eine Abkürzung dafür wie `\int` für Integral. Viele sind aber auch einfach nur kryptisch wie `\succcurlyeq` (\succcurlyeq). Anhang B enthält einige Links, mit denen man den entsprechenden Befehl zu einem gesuchten Symbol finden kann.

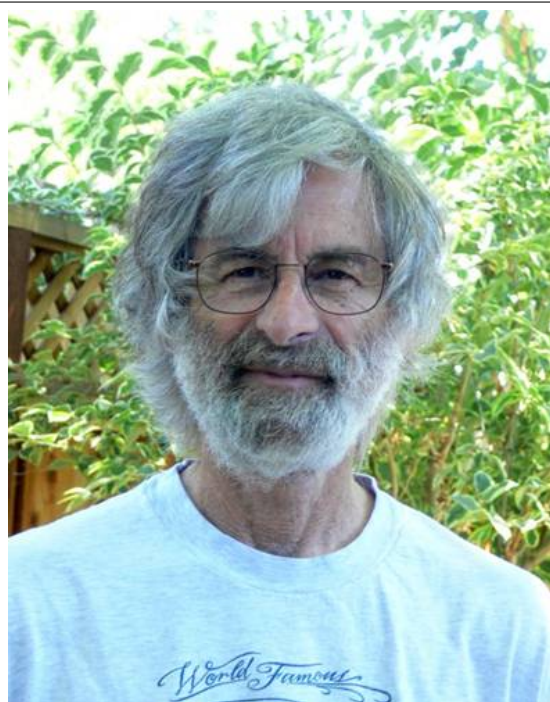
7. Abbildungen und Tabellen (Gleitumgebungen)

7.1. Abbildungen

L^AT_EX ist auch in der Lage, Bilddateien in das Dokument einzubinden. Das `graphicx`-Paket bietet dafür den Befehl `\includegraphics{}` an. Im muss lediglich der Dateiname des einzufügenden Bildes übergeben werden. Unterstützt werden Bildformate wie PNG, JPG und PDF. Zusätzlich lassen sich Optionen wie Höhe, Breite etc. angeben. Hier ist ein Foto des L^AT_EX-Entwicklers LESLIE LAMPORT (https://de.wikipedia.org/wiki/Leslie_Lamport#/media/Datei:Leslie_Lamport.jpg):

Code 12: Bild einfügen

```
1 % vor \begin{document}:
2 % \usepackage{graphicx}
3 \includegraphics[
4     width=\linewidth
5 ]{lamport.jpg}
```



7.2. Tabellen

Für die übersichtliche Darstellung von Daten sind Tabellen unerlässlich. Natürlich bietet LaTeX auch dafür umfangreiche Möglichkeiten. Hier sollen aber nur die Grundlagen erklärt werden.

Erstellt werden Tabellen mit Hilfe der `tabular`-Umgebung. Dabei müssen folgende Dinge angegeben werden: Anzahl der Spalten, Ausrichtung des Zelleninhalts (linksbündig, zentriert, rechtsbündig), Rahmenlinien und natürlich der Tabelleninhalt. Abmessungen wie die Spaltenbreite werden automatisch optimal gesetzt. Eine einfache Tabelle mit zwei Spalten, zwei Zeilen und ohne Rahmenlinien sieht so aus:

Code 13: Tabelle

```
1 \begin{tabular}{ll}
2     Spalte 1 & Spalte 2 \\
3     a & 1 \\
4     b & 2 \\
5 \end{tabular}
```

Spalte 1	Spalte 2
a	1
b	2

Die gleiche Tabelle sieht mit Rahmenlinien und zentriertem Inhalt so aus:

Code 14: Tabelle (II)

```

1 Text vor der Tabelle.
2
3 \begin{tabular}{|c|c|}
4   \hline
5   Spalte 1 & Spalte 2 \\
6   \hline
7   a & 1 \\
8   \hline
9   b & 2 \\
10  \hline
11 \end{tabular}
12
13 Text nach der Tabelle.
```

Text vor der Tabelle.	
Spalte 1	Spalte 2
a	1
b	2
Text nach der Tabelle.	

Anzahl der Spalten Die Anzahl der Buchstaben in den eckigen Klammern direkt nach `\begin{tabular}` (wie `{11}`) definieren die Spaltenanzahl.

Ausrichtung der Zellen Dabei steht jeder Buchstabe für die Ausrichtung der Zellen innerhalb der jeweiligen Spalte: `left`, `center`, `right`.

Rahmenlinien Die vertikalen Striche (`|`, „Pipe“) fügen vertikale Rahmenlinien ein (wie in `{|c|c|}`). Horizontale Linien können mit `\hline` erzeugt werden.

Inhalt Der Inhalt wird zeilenweise angegeben. Dabei markiert `&` den Übergang von einer Spalte zur nächsten. Wie üblich bedeutet `\\` das Ende einer Zeile.

7.3. Gleitumgebungen

Die bis jetzt gezeigten Möglichkeiten, Abbildungen und Tabellen einzufügen, sind recht unflexibel was die Positionierung innerhalb des Dokuments betrifft. Um LaTeX dabei mehr Freiheit zu bieten, gibt es sogenannte *Gleitumgebungen* (`figure` für Bilder, `table` für Tabellen). Der Name beschreibt sehr gut, wozu sie gedacht sind: Sie ermöglichen einem Bild oder einer Tabelle, durch das Dokument *zu gleiten* und von LaTeX automatisch dort positioniert zu werden, wo sie am besten hinpassen. Wenn einem also zugunsten des Gesamtlayouts nicht so wichtig ist, dass ein Bild oder eine Tabelle exakt dort erscheint, wo sie in der Quelldatei definiert wird, sollte man zu diesen Umgebungen greifen.

Außerdem bringen sie einige nützliche Features mit: der Befehl `\caption{}` bringt an der Umgebung eine Beschreibung an, die automatisch mit dem Namen „Abbildung“ oder „Tabelle“ und der Nummer der Umgebung versehen wird.

Code 15: figure-Umgebung

```
1 Hier ist Text vor der ↵  
    Gleitumgebung.  
2  
3 \begin{figure}[bth]  
4 \includegraphics[  
5     width=\linewidth  
6 ]{lamport.jpg}  
7 \caption{Foto von Leslie ↵  
    Lamport}  
8 \end{figure}  
9  
10 Hier Text nach der ↵  
    Gleitumgebung.
```

Hier ist Text vor der Gleitumgebung.

Hier ist Text nach der Gleitumgebung.

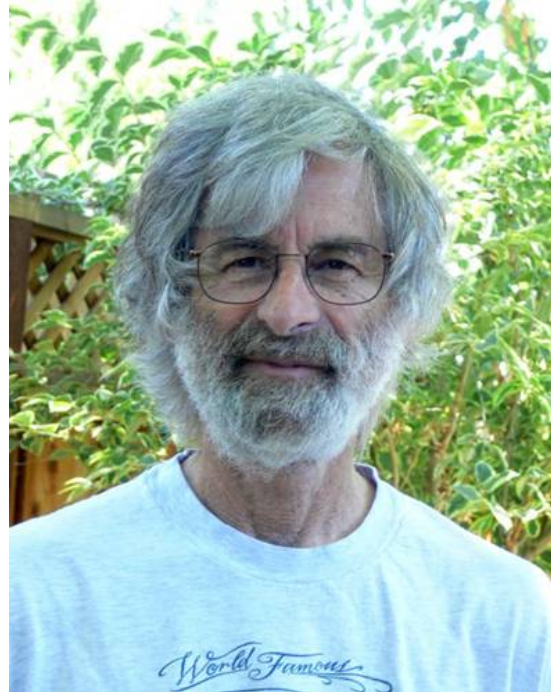


Abbildung 1: Foto von Leslie Lamport

Das optionale [bth] gibt in absteigender Priorität die Positionen an, an denen LaTeX versucht, die Umgebung auf der Seite zu platzieren: **b**ottom, **t**op und **h**ere. Die Positionen werden durchprobiert, bis die ganze Seite aus typographischer Sicht *schön* ist. Obwohl der untere Absatz in der Quelldatei *nach* der Gleitumgebung steht, wird er im fertigen Dokument *davor* angezeigt.

Mit dem Befehl \centering kann der Inhalt der Umgebung zentriert werden. Ein weiteres Feature von LaTeX sind die Befehle \label{} und \ref{}, die übrigens nicht nur im Zusammenhang mit Gleitumgebungen funktionieren:

Code 16: table-Umgebung

```

1 \begin{table}[htb]
2 \centering
3 \begin{tabular}{|c|c|}
4     \hline
5     Spalte 1 & Spalte 2 \\
6     \hline
7     a & 1 \\
8     \hline
9     b & 2 \\
10    \hline
11 \end{tabular}
12 \caption{Eine Tabelle.}
13 \label{tab_a}
14 \end{table}
15
16 Siehe Tabelle \ref{tab_a}.
```

Spalte 1	Spalte 2
a	1
b	2

Tabelle 1: Eine Tabelle.

Siehe Tabelle 1.

`\label{mein_label}` bringt an einem Objekt den internen Namen „mein_label“ an, auf den sich an einer beliebigen Stelle im Dokument mit `\ref{mein_label}` bezogen werden kann. Dort wird dann automatisch die Nummer des Objekts eingefügt. Das funktioniert nicht nur mit Gleitumgebungen, sondern mit allen Objekten, die nummeriert werden, also z. B. Überschriften wie `\section{}` und Gleichungen mit `equation`. Neben `\ref{}` gibt es noch `\pageref{}`, wodurch statt der Nummer die Zahl der Seite eingefügt wird, auf der das Objekt steht.

Um aus Abbildungen und Tabellen gleitende Objekte zu machen, werden sie also einfach in die entsprechende Gleitumgebung eingefügt. Anders als im ersten Beispiel verwendet gelten dabei die Positionen `[htb]` als Konvention.

8. Dokumentklassen

Wie am Anfang erwähnt, bestimmt die Dokumentklasse weite Teile des fertigen Layouts. Die wichtigsten Standardklassen sind `article` und `report`. Von ihnen existieren auch die moderneren Versionen `scrartcl` und `scrreprt` („script article“ und „script report“), welche einige Verbesserungen bringen. Ihre Verwendung sollte man auf jeden Fall in Betracht ziehen. Sie sind Teil der Erweiterung KOMA-Script¹⁰. Die Vorlage des FSR ist ebenfalls eine Dokumentklasse, die alles speziell für Übungsserien an der MLU Benötigte zur Verfügung stellt.

¹⁰<https://www.ctan.org/pkg/koma-script>

Anhang

A. Die wichtigsten Pakete

Eine Übersicht der wichtigsten LaTeX-Pakete und ihrer Funktion.

Paket	Funktion	Wichtige Optionen
inputenc	Interpretiere Quellcode als UTF8-kodiert	[utf8]
fontenc	Bessere Darstellung von Umlauten etc.	[T1]
babel	Landestypische Silbentrennung etc.	[ngerman]
csquotes	Landestypische Anführungszeichen	[style=german]
lmodern	Schöne Schrift	
amsmath, amsfonts, amssymb	Erweiterter Formelsatz	
geometry	Einstellung der Seitenränder	
graphicx	Einbinden von Bildern	
hyperref	Klickbare Links	
listings	Formatieren von Programmcode	

B. Wo gibt's Hilfe?

Eine Liste ausgewählter Links zu Anleitungen und Hilfen rund ums Thema LaTeX.

www.overleaf.com/learn Umfangreiche Einführung in LaTeX und das Arbeiten mit Overleaf.

<https://tex.stackexchange.com/> LaTeX-Community des Stack Exchange Networks

<https://www.informatik.uni-halle.de/studium/informatik-treff/> Informatik-Treff des Instituts für Informatik der MLU - von Studenten für Studenten

https://de.wikipedia.org/wiki/Liste_mathematischer_Symbole Gut sortierte Liste mathematischer Symbole mit entsprechenden LaTeX-Befehlen.

<http://detexify.kirelabs.org/classify.html> Diese Website erlaubt das Zeichnen eines Symbols und liefert Vorschläge für Befehle, um dieses Symbol in LaTeX zu setzen.

www.weinelt.de/latex/ Beschreibung aller Befehle von Standard-LaTeX (nicht mehr ganz aktuell, aber immer noch sehr hilfreich und auf Deutsch).

www.ctan.org Comprehensive TeX Archive Network, zentrale Anlaufstelle für Fragen zu einem bestimmten Paket.

<https://fachschaft.mathinf.uni-halle.de/kontakt/> Solltet ihr trotz allem mal nicht weiter kommen, könnt ihr euch natürlich auch jederzeit an uns, den FSR Mathe/Info, wenden.

Es gibt auch eine Vielzahl von Fachbüchern über L^AT_EX, folgende zwei aus dem Bestand der Universitäts- und Landesbibliothek können wir Einsteigern besonders empfehlen:

H. Kopka: L^AT_EX <http://search.ebscohost.com/login.aspx?direct=true&db=cat06365a&AN=ulb.518786056&lang=de&site=eds-live&scope=site>

M. Kohm, J.-U. Morawski: KOMA-Script <http://search.ebscohost.com/login.aspx?direct=true&db=cat06365a&AN=ulb.68547044X&lang=de&site=eds-live&scope=site>