

KANTONSSCHULE IM LEE WINTERTHUR

THOMAS GRAF

thomas.graf@ksimlee.ch

Ergänzungsfach / Wahlfach Informatik

19. August 2018



Kantonsschule Im Lee

Inhaltsverzeichnis

Inhaltsverzeichnis	i
0 Vorwort	1
0.1 Feedback	1
0.2 Unterrichtsmaterialien	1
0.3 Voraussetzungen und Niveau	2
1 Einleitung	3
1.1 Was ist Informatik?	3
1.2 Inhalt und Ziel des Unterrichts	4
2 Einführung in L^AT_EX	5
2.1 Vorbemerkungen	5
2.2 Installation	6
2.2.1 Online L ^A T _E X-Dienste	6
2.2.2 Lokale Installation	7
2.2.3 Aussprache des Wortes L ^A T _E X	7
2.2.4 WYSIWYG vs. WYSIWYM	8
2.3 Historische Entwicklung	9
2.4 L ^A T _E X-Grundlagen	11
2.5 Nützliche Ressourcen	19
Literaturverzeichnis	20

Kapitel 0

Vorwort

0.1 Feedback

Dieses Skript wird für das Ergänzungsfach / Wahlfach Informatik an der Kantonsschule im Lee verfasst. Die Unterrichtsmaterialien befinden sich noch im Entwicklungsstadium. Es ist uns ein grosses Anliegen, sie laufend zu verbessern und auszubauen.

Für Hinweise auf Fehler oder Ungenauigkeiten jeder Art sind wir sehr dankbar. Ebenfalls ausgesprochen willkommen sind Verbesserungsvorschläge, Ergänzungen, Kritik und Lob.

Allgemeines Feedback bitte an thomas.graf@ksimlee.ch mailen.
Alternativ, kann man einen anonymen Kommentar auf
<https://sayat.me/ThomasGraf> posten.

0.2 Unterrichtsmaterialien

Wir werden versuchen die aktuellen Unterrichtsmaterialien jeweils vor dem Unterricht zur Verfügung zu stellen.

Die aktuellen Unterrichtsmaterialien lassen sich sowohl auf dem *Moodle* der Kantonsschule Im Lee

<https://klwmoodle.tam.ch/course/view.php?id=358>

oder (alternativ) auf unserem *GitHub* Repository

https://github.com/ThomasGrafCode/EF_WF_Informatik finden.

Bitte stellt sicher, dass Ihr stets die aktuellsten Unterlagen verwendet.

Auf unserem YouTube-Kanal

www.youtube.com/channel/UCxIIM0e4oVImeJQbsphVjug/playlists

lassen sich zu vielen der Themen in diesem Kurs (hoffentlich) hilfreiche Video-Erklärungen finden.

0.3 Voraussetzungen und Niveau

Der Inhalt dieses Unterrichts setzt so gut wie keine Vorkenntnisse in der Informatik voraus. Es wird von erheblichen Unterschieden bezüglich der Vorkenntnisse unter den Schülern ausgegangen. Der Inhalt ist so gestaltet, dass alle Schüler dem Unterricht folgen können. Gleichzeitig sollen auch Schüler mit mehr Erfahrung gefordert werden und viel Neues dazu lernen.

Das Niveau dieses Fachs ist hoch (→ Maturitätsniveau):

Aktive Mitarbeit im Unterricht sowie zusätzliches Arbeiten zu Hause werden als selbstverständlich angesehen.

Wer bereit ist, sich emotional und zeitlich intensiv mit der Materie auseinanderzusetzen, wird deutlichen Fortschritt erzielen und auch mit entsprechendem Prüfungserfolg belohnt werden.

Kapitel 1

Einleitung

1.1 Was ist Informatik?

Es ist schwierig, eine genaue und abschliessende Definition einer wissenschaftlichen Disziplin zu geben.

In der Literatur lassen sich Versuche einer groben Definition der Informatik finden:

Informatik ist die „Wissenschaft von der systematischen Darstellung, Speicherung, Verarbeitung und Übertragung von Informationen, besonders der automatischen Verarbeitung mithilfe von Digitalrechnern“. [1]

Wer jedoch eine konkretere und genauere Vorstellung von der Informatik gewinnen möchte, muss bereits sein, ihre historische Entwicklung zu studieren und sich intensiv mit den Inhalten der Informatik auseinanderzusetzen.

Historisch hat sich die Informatik aus der Mathematik und den Ingenieurwissenschaften entwickelt, um den praktischen Bedarf nach der schnellen und automatischen Ausführung mathematischer Operationen (Berechnungen) zu decken.

‘Informatik’ ist ein Sammelbegriff, der sämtliche Aspekte, welche in irgend einer Form mit *Information* zu tun haben, umfasst.

Das Spektrum möglicher Aufgaben der Informatik reicht von simplen Anwendungsproblemen im Büroalltag bis hin zur Beantwortung einiger der schwierigsten Fragen der letzten Jahrhunderte ¹.

So hat z.B. das Gebiet der ‘theoretischen Informatik’, welches sich mit Themen wie der Berechenbarkeit von Funktionen, Komplexität, Theorie der Pro-

¹ Probleme wie z.B. das *P vs NP Problem*: <http://www.claymath.org/millennium-problems/p-vs-np-problem>

grammiersprachen, Algorithmik und Kryptographie beschäftigt, fast ausschliesslich mathematische / philosophische Facetten. Der Begriff ‘Informatik’ umfasst aber auch Gebiete wie das der praktischen Programmierung von Computern, der digitalen Kommunikation, des Internets, der Textverarbeitung, der digitalen Medien und noch viele mehr.

Methoden der Informatik kommen in praktisch jeder modernen Wissenschaft zum Einsatz. Dadurch verwischen die Grenzen zwischen der Informatik und anderen Disziplinen mehr und mehr. Diese Tatsache erhöht die Schwierigkeit, die Informatik aussagekräftig und in wenigen Sätzen beschreiben zu wollen, natürlich noch zusätzlich.

1.2 Inhalt und Ziel des Unterrichts

Wie in Abschnitt 1.1 bereits gesagt wurde, ist das Gebiet der Informatik äusserst umfangreich. Es ist nicht möglich, sich mit allen Bereichen ausführlich zu beschäftigen. Dies gilt natürlich auch für jede andere Wissenschaft. Deshalb werden wir eine Auswahl von Themen treffen, die klein genug ist, um jedes Thema in sinnvoller Tiefe behandeln zu können, aber immer noch gross genug ist, dass sie ermöglicht das Wesen und den Stellenwert der Informatik zu erkennen.

Viele Aspekte der Informatik entwickeln sich rasend schnell. Aus diesem Grund kann spezifisches Wissen über, zum Beispiel bestimmte Software-Produkte, oder detaillierte Kenntnisse einer bestimmten Programmiersprache, schon nach wenigen Jahren komplett veraltet sein. Deshalb ist es viel wichtiger, die grundlegenden Konzepte, welche hinter konkreten Implementierungen stehen, zu verstehen. Beispielsweise sind viele Aussagen der Informatik mathematisch beweisbar und werden für alle Zeiten gültig sein.

Unter Berücksichtigung der soeben erwähnten beiden Punkte, haben wir eine Auswahl von interessanten Themen getroffen, welche verschiedene Gebiete der Informatik repräsentiert und unser analytisches Denken fördern wird. Ziel ist es, solide Grundlagen in verschiedenen Disziplinen der Informatik aufzubauen, welche auch im späteren Leben und vor allem im Studium sehr nützlich sein werden. Dazu gehören auch gewisse “handwerkliche” Fähigkeiten, wie zum Beispiel die Textverarbeitung in \LaTeX und Grundkenntnisse über Computerarchitekturen und Programmiersprachen.

Der Fokus ist dabei immer stark auf das Verständnis der zugrundeliegenden Konzepte gerichtet.

Kapitel 2

Einführung in L^AT_EX

2.1 Vorbemerkungen

In diesem Kapitel wollen wir uns mit dem Computerprogramm ‘L^AT_EX’ beschäftigen. L^AT_EX ist ein ausgesprochen populäres Werkzeug zur Erstellung schriftlicher Arbeiten wie z.B. Maturitätsarbeiten, Bachelor- und Masterarbeiten, Büchern, wissenschaftlicher Paper, Schulunterlagen und professionell anmutender Beamer-Präsentationen. Auch dieses Skript wird in L^AT_EX verfasst. Darüber hinaus ist L^AT_EX hervorragend geeignet zur Erzeugung von Grafiken sowie zur Gestaltung imposanter Plakate und ästhetischer Bewerbungsunterlagen.

In vielen Studiengängen wird erwartet und verlangt, dass die Studierenden fähig sind, Dokumente in L^AT_EX zu verfassen. Die dazu notwendigen Kenntnisse müssen typischerweise selbstständig erarbeitet werden.

Um sich später im Studium besser auf die eigentlichen Lehrinhalte fokussieren zu können, ist es nützlich, den grundlegenden Umgang mit Werkzeugen, wie L^AT_EX, bereits in der Kantonsschule zu üben.

L^AT_EX unterscheidet sich grundlegend von Textverarbeitungsprogrammen, wie z.B. Microsoft Word (siehe Unterabschnitt 2.2.4). Daher kann das Arbeiten mit L^AT_EX zu Beginn recht gewöhnungsbedürftig sein. Dennoch ist es nicht schwierig, den Einstieg in das Arbeiten mit dieser Software zu finden.

Zu diesem Thema lassen sich Video-Erklärungen auf unserem YouTube-Kanal

www.youtube.com/channel/UCxIIM0e4oVImeJQbsphVjug/playlists
finden.

2.2 Installation

Im Folgenden werden wir verschiedene Tools vorstellen, die es ermöglichen mit \LaTeX zu arbeiten. Grundsätzlich kann man entweder mit einem online \LaTeX -Tool oder einer lokalen Installation arbeiten.

Wir ziehen online Tools den lokalen Installationen ganz klar vor.

2.2.1 Online \LaTeX -Dienste

Im Internet werden verschiedene Webdienste angeboten, die es erlauben, mit \LaTeX online im Browser zu arbeiten. Es wird keine lokale Installation auf dem Computer benötigt, und man kann deshalb an jedem Computer mit Internetzugang sofort zu arbeiten beginnen. Beispiele solcher online Dienste sind Papeeria, Overleaf, ShareLaTeX, Datazar, und LaTeX base.

Wir können den online \LaTeX -Dienst ***Overleaf*** sehr empfehlen. Der Service ist kostenlos und man muss sich lediglich mit einer E-Mail-Adresse auf der Overleaf-Webseite:

<https://www.overleaf.com/>

registrieren. Wir benutzen Overleaf selber (z.B. um dieses Dokument zu schreiben), weil wir dieses Tool enorm praktisch finden.

Auf der Webseite

<https://www.latex-project.org/get/>

sind in der Sparte *Online* nebst dem Link zur Overleaf-Webseite auch die Links zu allen anderen oben genannten online Tools angegeben.

Um die \LaTeX -Beispiele in diesem Skript und in den Übungen fehlerfrei ausführen zu können, werden wir diverse \LaTeX -Zusatzpakete (siehe 2.4) benötigen. **Da dieses gesamte Skript und alle Übungen in Overleaf werden, können wir garantieren, dass sämtliche in diesem Kurs benötigten Zusatzpakete in Overleaf zur Verfügung stehen.**

Ihr seid alle dazu eingeladen (anstelle von Overleaf), ein anderes online Tool Eurer Wahl zu benutzen oder eine lokale Installation (siehe 2.2.2) zu verwenden.

Dabei gilt es allerdings zu beachten, dass Ihr eventuell fehlende Zusatzpakete manuell installieren müsst. Je nach Wahl Eures Setups kann dies sehr einfach oder recht schwierig sein. Bei mehreren dutzend, in das Ergänzungsfach / Wahlfach Informatik eingeschriebenen Personen, würde es einen enormen Aufwand bedeuten, allen bei der individuellen Problembehebung zu helfen.

Daher empfehlen wir nur erfahrenen Usern sich für einen anderen Setup als Overleaf zu entscheiden.

2.2.2 Lokale Installation

Natürlich kann man \LaTeX auch lokal auf dem eigenen Rechner installieren. Um effizient arbeiten zu können, müssen dazu zwei Sachen installiert werden:

- eine \TeX -Distribution
- ein \LaTeX -Editor.

\TeX -Distribution

\TeX -Distributionen können z.B. von dieser Webseite heruntergeladen werden (für GNU/Linux, Microsoft Windows und macOS):

<https://www.latex-project.org/get/>.

Dort lassen sich auch ausführliche Installationsanleitungen finden, mit deren Hilfe die Installation reibungslos ablaufen dürfte.

\LaTeX -Editor

Die \LaTeX -Editoren **TEXMAKER** (<http://www.xm1math.net/texmaker/>) und **TeXstudio** (<https://www.texstudio.org/>) sind sehr gut.

Eventuell müssen zur fehlerfreien Ausführung der \LaTeX -Beispiele in diesem Skript und den Übungen noch Zusatzpakete installiert werden.

2.2.3 Aussprache des Wortes \LaTeX

Der Buchstabe ‘X’ in dem Wört \LaTeX steht nicht für den lateinischen Buchstaben ‘X’ so wie beispielsweise in dem Wort ‘Xylophon’, sondern für den grossgeschriebenen griechischen Buchstaben ‘Chi’ (χ), welcher wie das harte

‘ch’ im deutschen Wort ‘ach’ ausgesprochen wird. Damit lautet die Aussprache nicht ‘Latex’ sondern ‘Latech’.

2.2.4 WYSIWYG vs. WYSIWYM

Textverarbeitungsprogramme wie z.B. Microsoft Word oder LibreOffice Writer bezeichnet man als WYSIWYG “**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et” - man sieht den formatierten Text und die endgültige Form des Dokuments stets auf dem Bildschirm.

Im Gegensatz dazu stehen WYSIWYM “**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **M**ean” Textverarbeitungsprogramme wie z.B. \LaTeX . In \LaTeX wird mit Computerbefehlen gearbeitet, welche die Struktur und Optik des zu erstellenden Dokumentes beschreiben. Die im \LaTeX -File gespeicherten Befehle (Commands) müssen zuerst kompiliert werden ^{1,2}, um das finale Produkt (typischerweise ein PDF-Dokument) zu erhalten (siehe 2.4).

Das endgültige Resultat eines \LaTeX -Dokuments ist also nicht das \LaTeX -Dokument selber, sondern vielmehr das bei seiner Kompilation entstehende Output-File.

Das folgende Beispiel soll den Unterschied zwischen WYSIWYG und WYSIWYM Programmen illustrieren:

Nehmen wir an, wir möchten eine linguistische Arbeit verfassen und dabei den griechischen kleingeschriebenen Buchstaben α (alpha) erwähnen. Um dies in Microsoft Word zu erreichen, könnte man die in Word integrierte Sammlung von Symbolen öffnen und das Symbol für α an der entsprechenden Stelle im Text einfügen. In \LaTeX würde man anstatt ein α -Symbol einzufügen, lediglich den Command `$\backslash\alpha$` hinschreiben. Dieser Befehl hat optisch gesprochen natürlich mit dem gewünschten α -Symbol überhaupt nichts zu tun. Das tatsächliche α -Symbol erscheint nicht im \LaTeX -Dokument, sondern erst im PDF-Dokument, welches bei der Ausführung der Befehle im \LaTeX -Dokument erstellt wird.

¹ Der Prozess des Kompilierens umfasst das Übersetzen des \LaTeX -Files (Files mit der Dateierdung `.tex`) zu verschiedenen Output Formaten (über verschiedene Zwischenstufen).

² Die Details der Kompilation gehen deutlich über den Rahmen dieses Kapitel hinaus. Interessierte finden auf der Webseite https://www.sharelatex.com/learn/Choosing_a_LaTeX_Compiler gibt einen guten ersten Überblick über den Ablauf der Kompilierung.

WSIWYYG-like Tools

Der Vollständigkeit halber soll noch erwähnt werden, dass es sogenannte *WSIWYYG-like* (WSIWYYG ähnliche) Tools für L^AT_EX gibt, welche z.B. Symbole wie das oben erwähnte α direkt in als eigentliche Symbole in dem L^AT_EX-Dokument und nicht erst im Output-File (PDF-Dokument) erscheinen lassen.

Beispiele solcher Tools sind

GNU TeXmacs: <http://www.texmacs.org/tmweb/home/welcome.en.html>
(nur für lokale Installationen),

BaKoMa T_EX: <http://www.bakoma-tex.com/>
(nur für lokale Installationen)

oder der *Rich Text* Modus von Overleaf. Wir verwenden keines dieser WSIWYYG-like Tools und haben von den soeben erwähnten lediglich das Letzte (Overleaf Rich Text-Modus) etwas eingehender betrachtet. Wir finden, dass dieses sehr schön gemacht ist und sicherlich bei vielen Nutzern (auch bei erfahrenen Nutzern) Anklang finden wird. Andere Online Tools verfügen über recht ähnliche Features (→ Ausprobieren lohnt sich).

2.3 Historische Entwicklung

T_EX

T_EX ist ein im Jahre 1978 veröffentlichtes Computerprogramm zur Erstellung von digitalen Texten, welches hauptsächlich von dem berühmten US-amerikanischen Informatiker Donald E. Knuth geschrieben wurde. Sein Hauptziel war es, mit T_EX ein Werkzeug zu schaffen, mit dessen Hilfe hochqualitative Bücher mit wenig Aufwand verfasst werden können. Des weiteren sollte T_EX auf jedem Computer und zu jeder Zeit identische Resultate (Texte) produzieren ³.

T_EX ist insbesondere eine Programmiersprache. Diese ermöglicht die Definition diverser Befehle, welche den Computer z.B. beauftragen, die Schriftgrösse anzupassen, Abstände im Text einzubauen oder die Schriftart zu ändern. Eine T_EX-Distribution enthält viele solcher bereits definierten primitive Befehle. Ein Beispiel eines solchen Befehls ist `\baselineskip = 24pt`, welcher bewirkt, dass zwischen den einzelnen Zeilen im Text ein Abstand von 24 Einheitspunkten gesetzt wird (in T_EX beginnen Befehle mit einem Backslash ‘\’).

³ Details zur Geschichte von T_EX findet man hier: <https://www.tug.org/whatis.html>

Nehmen wir an, wir möchten nun ein Textdokument wie dieses hier schreiben.

Konzentrieren wir uns einmal auf die Überschrift “Historische Entwicklung” dieses Abschnitts (Section). Wie ist diese wohl entstanden?

Um eine solche Überschrift in T_EX zu erstellen, müssten dem Computer diverse Instruktionen (Befehle) übergeben werden:

- Befehle zur Anpassung der Schriftgrösse und Dicke der Schrift
- diverse Befehle zum Einfügen bestimmter Abstände vor und nach der Überschrift
- Anweisungen zur Nummerierung der verschiedenen Abschnitte
- usw.

Dies bedeutet, dass wir zahlreiche T_EX-Befehle geschickt kombinieren müssten, um einen neuen Abschnitt deklarieren zu können. Dies erscheint recht unpraktisch zu sein. Was wäre jedoch, wenn jemand sich die Mühe machen würde, so einen komplexen Befehl für das Erstellen eines neuen Textabschnittes aus primitiven T_EX-Befehlen zusammenzusetzen? Diese tatkräftige Person könnte der Kombination aller dafür benötigten primitiven Befehle einen neuen Namen, wie z.B. `\section{}` geben. Dieser Befehl steht dann als Stellvertreter für die Kombination aller Befehle, welche zur Erstellung eines neuen Abschnitts notwendig sind. Man bezeichnet den dadurch entstandenen neuen Befehl als *Makro* (griechisch *μακρός* makros = ‘gross’, ‘weit’, ‘lang’).

Die Idee hinter Makros ist simpel und effektiv: man fasst alle kleinen (mikro) Befehle, welche benötigt werden, um z.B. einen neuen Abschnitt (Section) zu erstellen, zu einem neuen, grossen (makro) Befehl zusammen und gibt diesem einen Namen, wie beispielsweise `\section{}`. Dass dieses Vorgehen möglich ist, dürfte nicht erstaunen, da T_EX wie bereits erwähnt, eine vollwertige Programmiersprache ist, mit welcher natürlich solche neuen Befehle definiert werden können.

L^AT_EX

Der beflissene US-amerikanische Informatiker Leslie B. Lamport baute am Anfang der 1980er Jahre zahlreiche höchst nützliche Makro-Befehle aus primitiven T_EX-Befehlen zusammen. Das daraus resultierende Softwarepaket, welches die Benutzung von T_EX mit Hilfe von Makros stark vereinfacht, nennt sich L^AT_EX (kurz für **L**amport **T**e**X**). L^AT_EX ist also nichts anderes als eine auf T_EX aufbauende (und in T_EX geschriebene) Erweiterung von T_EX. Mittlerweile gibt es tausende von T_EX-Paketten, welche von diversen Leuten

verfasst wurden und eine ungeheure Vielfalt an nützlicher Funktionalität zur \TeX / \LaTeX Basisdistribution hinzufügen ⁴. \LaTeX ist eine sogenannte *freie Software* und unterliegt den Regelungen der LaTeX Project Public License (LPPL) ⁵.

2.4 \LaTeX -Grundlagen

In diesem Abschnitt werden wir lediglich eine handvoll \LaTeX Befehle und Techniken kennen lernen. Ihr werdet aber schnell sehen, dass diese limitierten Kenntnisse bereits ausreichen um direkt in \LaTeX loslegen zu können: Sobald man diese essentiellen Grundlagen kennt, kann alles Weitere bei Bedarf im Internet gefunden werden.

Diese essentiellen Grundlagen wollen wir hier kennenlernen.

Erstes \LaTeX -Dokument

Das einfachste \LaTeX Dokument sieht so aus:

```
1 \documentclass{article}
2 \begin{document}
3 Dies ist mein erstes \LaTeX-Dokument!
4 \end{document}
```

Wenn man dieses kompiliert (wird bei Overleaf automatisch nach jeder Änderung des \LaTeX -Files gemacht), wird ein PDF-File mit folgendem Inhalt generiert:

Dies ist mein erstes \LaTeX -Dokument!

Unserem ersten \LaTeX -Dokument entnehmen wir, dass Commands in \LaTeX mit einem Backslash ‘\’ beginnen (siehe auch Abschnitt 2.3). Der Command `\documentclass` legt fest, welche Art von Dokument wir schreiben möchten. Damit der Command wissen kann, welche Art von Dokument wir uns wünschen, müssen wir ihm die Zusatzoption `article` übergeben. Anstelle von `article` könnte man z.B. auch `book` oder `letter` angeben. Die allgemeine Form eines Commands ohne weitere Optionen oder mit einer Option ist also

⁴ umfangreiche Sammlung von \TeX -Paketen: <https://ctan.org/?lang=en>

⁵ Details zur LPPL-Lizenz: <https://www.latex-project.org/lppl/>

```

1 \dosomething % ohne Zusatzoptionen
2 \dosomething{} % mit einer Zusatzoption.

```

Das Prozent Symbol % markiert den Anfang eines Kommentars (Comment). Alles, was nach diesem Symbol steht, wird ist ein Kommentar und wird bei der Kompilierung vom Computer ignoriert. Mit Kommentaren, lässt sich in Worten beschreiben, was im Code gemacht wird. Kommentierter Code ist meist (besonders für andere Leute) viel leserlicher als Code ohne Kommentar. Im obigen Beispiel ist `\LaTeX` ein Command ohne Zusatzoption. Der Befehl macht nichts anderes als das Wort ‘LaTeX’ im Design des L^AT_EXLogos zu schreiben.

Das Konstrukt

```

\begin{document}
...
\end{document}

```

definiert eine Umgebung (Environment), welches den Beginn und das Ende des Dokuments markiert. Dazwischen wird der eigentliche Inhalt des Dokuments geschrieben. Die allgemeine Form eines Environments ist:

```

1 \begin{something}
2 Hier soll etwas Interessantes geschehen!
3 \end{something}

```

Zusatzpakete importieren

Unser erstes L^AT_EX-Dokument 2.4 enthielt nur sehr elementare Befehle. Häufig möchte man dem Dokument aber weitere Funktionalität hinzufügen. Diese Funktionalität wird durch das Einfügen von Zusatzpaketen erreicht. Dazu muss das Zusatzpaket allerdings am Richtigen Ort auf dem Rechner installiert und gespeichert sein (bei lokalen Installation) oder von dem online Tool zur Verfügung gestellt werden (ansonsten muss man das Paket manuell installieren bzw. zur online Umgebung hinzufügen). Falls das Paket vorhanden ist, kann es durch den Befehl

```
\usepackage{NamesDesPakets}
```

in das L^AT_EX-File eingefügt werden. Wichtig ist, dass der Befehl vor dem

```

\begin{document}
...
\end{document}

```

Environment steht.

Dazu ein Beispiel mit einem Zusatzpaket:

```
1 \usepackage{xcolor} % stellt diverse Farben /  
   ↳ Schattierungen zur Verfügung  
2  
3 \begin{document}  
4 % verwende nun das Paket 'xcolor' um in Farbe zu schreiben:  
5 \textcolor{red}{Dieser Text soll in roter Farbe geschrieben  
   ↳ werden.}  
6 \end{document}
```

ergibt:

Dieser Text soll in roter Farbe geschrieben werden.

Wenn man viele Pakete importiert, ist es stilistisch am besten, wenn man alle `\usepackage{...}`-Commands in einem separaten `.tex`-File (nennen wir es `preamble.tex`) speichert.

Der Vorteil liegt darin, dass man sein Hauptdokument nicht mit etlichen `\usepackage{...}`-Commands beginnen muss, sondern diese Commands mit dem Befehl `\input{preamble}` (muss vor `\begin{document}` stehen) einfügen kann.

Das Präambel-File der Autoren (Name: `preamble.tex`) dieses Skripts wird auf Moodle oder GitHub zur Verfügung gestellt. **Es muss in demselben Verzeichnis liegen wie das Haupt- \LaTeX -File.** Dadurch kann garantiert werden, dass Ihr alle Beispiele in diesem Skript und den Übungen ohne Fehlermeldungen kompilieren könnt.

Sections und subsections

Die Einteilung des Dokuments in Abschnitte und Unterabschnitte (oder auch Kapitel, Unterkapitel etc.) trägt stark zur Übersichtlichkeit und Struktur des Dokuments bei (siehe Beispiel 2.4).

Der Befehl `\` erzwingt einen Zeilenumbruch. Die Nummerierung kann durch das Hinzufügen des Symbols `*` unterdrückt werden:

`\section*{...}`, `\subsection*{...}`.

```
1 \section{Abschnitt}
2 Habe nun, ach! Philosophie,\
3 Juristerei und Medizin,\
4 Und leider auch Theologie\
5 Durchaus studiert, mit heißem Bemühn.
6 \subsection{Unterabschnitt}
7 Da steh ich nun, ich armer Tor!
8 \subsubsection{Unterunterabschnitt}
9 Und bin so klug als wie zuvor;
10 \subsection{noch ein Unterabschnitt}
11 Heiße Magister, heiße Doktor gar\
12 Und ziehe schon an die zehen Jahr\
13 Herauf, herab und quer und krumm\
14 Meine Schüler an der Nase herum-\
15 Und sehe, daß wir nichts wissen können!
```

kompiliert zu



1 Abschnitt

Habe nun, ach! Philosophie,
Juristerei und Medizin,
Und leider auch Theologie
Durchaus studiert, mit heißem Bemühn.

1.1 Unterabschnitt

Da steh ich nun, ich armer Tor!

1.1.1 Unterunterabschnitt

Und bin so klug als wie zuvor;

1.2 noch ein Unterabschnitt

Heiße Magister, heiße Doktor gar
Und ziehe schon an die zehen Jahr
Herauf, herab und quer und krumm
Meine Schüler an der Nase herum-
Und sehe, daß wir nichts wissen können!

Listen, Aufzählungen

Listen können ganz einfach wie folgt erstellt werden

```
1 \begin{itemize}
2 \item erstens
3 \item zweitens
4 \item ...
5 \end{itemize}
```

erzeugt folgende Liste

- erstens
- zweitens
- ...

Aufzählungen lassen sich sehr ähnlich erstellen:

```
1 \begin{enumerate}
2 \item erstens
3 \item zweitens
4 \item ...
5 \end{enumerate}
```

erzeugt die Aufzählung

1. erstens
2. zweitens
3. ...

Mit dem Parameter z.B. `\begin{enumerate}[label=\roman*.]` erfolgt die Aufzählung mit römischen Zahlen.

Formeln, Gleichungen, mathematische Symbole

L^AT_EX brilliert ganz besonders beim Verfassen mathematischer Texte und Aufschreiben von Gleichungen.

Schauen wir uns folgendes, umfangreiches Beispiel an:

```

1 Betrachte die quadratische Gleichung
2 \begin{equation}
3 \label{quadratische_gleichung}
4 ax^2+bx+c=0,
5 \end{equation}
6 mit  $a, b, c \in \mathbb{R}$  und  $a \neq 0$ .\\
7
8 \noindent
9 Definiere die Diskriminante  $\Delta = b^2 - 4ac$ .\\
10 Dann hat die quadratische Gleichung
11    $\rightarrow$  \ref{quadratische_gleichung} folgende Lösungen in den
12    $\rightarrow$  reellen Zahlen:
13 \begin{enumerate}[label=\roman*.]
14 \item die zwei verschiedene reellen Lösungen
15 \begin{equation}
16 x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a},
17 \end{equation}
18 falls  $\Delta > 0$ ,
19 \item genau eine reelle Lösung
20 \begin{equation}
21 x = -\frac{b}{2a},
22 \end{equation}
23 falls  $\Delta = 0$ ,
24 \item keine reellen Lösungen, falls  $\Delta < 0$ .

```

Dieser L^AT_EX-Code wird uns folgenden Output liefern:

Betrachte die quadratische Gleichung

$$ax^2 + bx + c = 0, \quad (2.1)$$

mit $a, b, c \in \mathbb{R}$ und $a \neq 0$.

Definiere die Diskriminante $\Delta = b^2 - 4ac$.

Dann hat die quadratische Gleichung 2.1 folgende Lösungen in den reellen Zahlen:

- i. die zwei verschiedene reellen Lösungen

$$x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a}, \quad (2.2)$$

falls $\Delta > 0$,

- ii. genau eine reelle Lösung

$$x = -\frac{b}{2a}, \quad (2.3)$$

falls $\Delta = 0$,

- iii. keine reellen Lösungen, falls $\Delta < 0$.

Hier haben wir recht viele neue Konzepte benutzt. Schauen wir uns das Beispiel mit der quadratischen Gleichung etwas genauer an. Wir verwenden hier das `equation`-Environment, welches Anfang und Ende einer neuen Gleichung (oder eines allgemeinen mathematischen Ausdrucks) kennzeichnet. Wir geben dieser Gleichung das Label `\label{quadratische_gleichung}` um später im Text auf diese Gleichung mit `\ref{quadratische_gleichung}` Referenz nehmen zu können. Referenzen sind ungeheuer wichtig in wissenschaftlichen Arbeiten. Man beachte auch, dass die Gleichung durch das `equation`-Environment eine Nummer erhalten hat. Damit hat die Gleichung eine ‘Namen’ bekommen.

Durch die Dollar-Symbole kann innerhalb des Textflusses in die Mathematik-Umgebung (*math environment*) gewechselt werden.

Der Command `\frac{a}{b}` ergibt den Bruch $\frac{a}{b}$ und `\sqrt{a}` die Quadratwurzel (*square root*) \sqrt{a} .

Das `enumerate`-Environment haben wir bereits in 2.4 kennengelernt. Die Bedeutung der anderen Commands sollte aus dem Kontext klar sein.

Makros

Zum Schluss wollen wir uns noch der Erstellung von Makros widmen, welche bereits in 2.3 kurz angesprochen wurden.

Die nächsten zwei Beispiele dienen der Illustration der Funktionsweise von Makros in L^AT_EX.

Beispiel 1 (ohne Argumente)

Angenommen du musst in einer Arbeit in der Chemie die Zusammensetzung des Sprengstoffs *TNT* untersuchen. In deinem Laborbericht möchtest du ab und zu den vollständigen Namen von TNT, nämlich **2-Methyl-1,3,5-trinitrobenzen** erwähnen (in Fettschrift). Dazu müsste man an den entsprechenden Stellen jeweils den Command

`\textbf{2-Methyl-1,3,5-trinitrobenzen}` hinschreiben. Wenn man dies oft tun muss, ist es einfach, diesen Command durch ein Makro zu ersetzen, welchem wir den Namen `\TNT` geben möchten:

```
\newcommand{\TNT}{\textbf{2-Methyl-1,3,5-trinitrobenzen}}
```

Von nun an, kann in L^AT_EX einfach `\TNT` geschrieben werden. Bei der Kompilierung wird jedes `\TNT` durch **2-Methyl-1,3,5-trinitrobenzen** ersetzt werden. Dies macht das Schreiben einfacher und das L^AT_EX-File bleibt übersichtlicher.

Beispiel 2 (mit Argumenten)

In einer Arbeit über Geometrie sind wir es leid, für viele verschiedene Zahlen a und b den Ausdruck $\sqrt{a^2 + b^2}$ aus dem Satz des Pythagoras aufzuschreiben. Wir schreiben lieber ein Makro mit dem Namen `\pythagoras` schreiben:

```
\newcommand{\pythagoras}[2]{\sqrt{#1^2+#2^2}}
```

Von nun an können wir anstelle von `\sqrt{3^2+4^2}` einfach `\pythagoras$` schreiben um $\sqrt{3^2 + 4^2}$ zu erhalten. Man beachte, dass wir dem Makro eine Anzahl von zwei Makros (`$[2]$`) als Input übergeben.

Die allgemeine Struktur eines solchen Makros ist also:

```
\newcommand{\NameMakro}[Anzahl Argumente]{Definition}
```

2.5 Nützliche Ressourcen

Templates (Vorlagen)

Wir empfehlen, neue L^AT_EX-Projekte mit einer bereits bestehenden Vorlage (sogenannten *Templates*) zu beginnen.

Gute Quellen für L^AT_EX-Templates sind:

- <https://www.overleaf.com/latex/templates/>
- <https://www.latextemplates.com/>

Nachschlagewerke

- **Internet-Suchmaschinen**
- ausführliches Tutorial:
<http://web.mit.edu/jgross/Public/latex/lshort.pdf>
- kurzes Tutorial:
<http://www.ipcms.unistra.fr/wp-content/uploads/2018/04/latex.pdf>
- Symbol-Erkennungstool:
<http://detexify.kirelabs.org/classify.html>
- bei Fragen und Problemen:
<https://tex.stackexchange.com/>
- kurze Übersicht mathematischer Symbole:
https://oeis.org/wiki/List_of_LaTeX_mathematical_symbols
- riesige Sammlung von L^AT_EX-Symbolen: <http://tug.ctan.org/info/symbols/comprehensive/symbols-a4.pdf>
- Wikibook
<https://en.wikibooks.org/wiki/LaTeX>

Übungsaufgaben

- <https://guides.nyu.edu/c.php?g=601858&p=4168140>

Literaturverzeichnis

- [1] Volker Claus und Andreas Schwill. *Duden Informatik A-Z: Fachlexikon für Studium, Ausbildung und Beruf*, volume 4. Bibliographisches Institut & F. A. Brockhaus, 2006. 3