

L^AT_EX in a Nutshell

Ein „knapper“ Einstieg mit zahlreichen Beispielen

Stiv Nikolov

sn01@tu-clausthal.de

17. Mai 2022

Was dieses Skript sein soll

- Dieses Skript soll euch eine kondensierte Beispielsammlung darstellen, damit für die Basics kein googlen nach Anleitungen für ein Dokument (Laborprotokoll, Thesis, o.ä.) nötig ist.
- Da es sich auf die Grundlagen beschränkt könnt ihr die Beispiele dann mit weiteren Ressourcen und Dokumentationen der Pakete aus dem Netz erweitern, um das Schriftstück nach euren Bedürfnissen anzupassen.
- Begleitend ist ein vollständig durchformatiertes Dokument bereitgestellt, in dem alles hier genannte mit vielen Kommentaren Anwendung findet.

- Was ist \LaTeX ?
- Warum sollte man \LaTeX nutzen?
- Installation
- Dokumentenklassen
- Umgebungen (environments)
- Wie erstellt man ein Dokument
 - Strukturierung
 - Verzeichnisse
 - Formatierung
- Hilfreiche Pakete im Überblick
- Inhalte einfügen
 - Formeln
 - Bilder
 - Tabellen
 - Listen
 - Dateien
- Quellenverwaltung
- \LaTeX Compiler
- Tipps zum Debugging

Was ist L^AT_EX?

- MS Word und gängige Text-Editoren sind „What you see is what you get“ (WYSIWYG) Systeme, sprich das Dokument erscheint wie es dargestellt wird.
- L^AT_EX ist hingegen eine Markup-Sprache (ähnlich zu HTML) für ein Textsatzsystem. Mit Befehlen wird Form, Position, etc. vorher festgelegt und dann zu einem Dokument kompiliert. Es ist quasi Buchdruck in digital.
- L^AT_EX umfasst dabei nur den Code, für eine sinnvolle Bearbeitung sind entsprechende Editoren mit Syntaxhervorhebung nötig.

Warum sollte ich \LaTeX nutzen?

- Einheitliches Erscheinungsbild
- Globale Formatierung
- Modularbarkeit
- Unterstützung für Vektorgrafiken
- Automatisierung mit Makros möglich
- Native Quellenverwaltung
- Kostenlos & Plattformunabhängig
- Performance
- Geschriebenes bildet eure Datenbank
- **Nachteil: man muss es lernen**



\LaTeX

Installation

- \LaTeX Distributionen findet ihr unter <https://www.latex-project.org/get/>.
Für Windows wird MiKTeX empfohlen.
- Als Editor wird für die Beispiele **TeXstudio** verwendet.
- Profile für Windows und Linux mit dark-mode Syntax-Hervorhebung für TeXstudio werden zur Verfügung gestellt.
- Wer nichts installieren möchte und parallel in einem Dokument schreiben will, der kann den online Editor **overleaf** verwenden. Dort funktionieren nur nicht die später beschriebenen Tastenkürzel.

Dokumentenklassen

Für die Klassen werden hier KOMA-Skript Klassen verwendet, da sie enorm viele Einstellungen vorweg nehmen und euch Arbeit ersparen.

<code>scrartcl</code>	Zur Erstellung von Artikeln, Laborprotokollen, kurzen Berichten oder Hausarbeiten oder jegliches Dokument, welches keine Kapitel (chapter) oder Teile (parts) braucht.
<code>scrreprt</code>	Für lange Artikel, die Kapitel (chapter) enthalten, besonders geeignet für kurze Bücher oder einer Thesis.
<code>scrbook</code>	Für lange Bücher oder Ph.D Thesis mit übergreifenden Teilen (parts), die in Kapiteln (chapter) unterteilt sind.
<code>letter</code>	Für Briefköpfe, mit Positionierung von Adressaten für die klassische 1/3-Faltung. Besitzt eigene spezielle Umgebungen.
<code>beamer</code>	Zur Erstellung von Folien für Präsentationen (so wie diese hier)
<code>standalone</code>	Dokument ohne Ränder, nur so groß wie der Inhalt. Primär für Bilder nützlich.

Dokumentenklassen

- In der Regel werdet ihr nur den script article `scrartcl` und den script report `scrreprt` für das Meiste verwenden.
- Die Klassen geben vor welche Strukturiefen vorhanden sind bzw. welche die höchste ist. Bei der `(scr)book` Klasse sind alle vorhanden und stehen in folgender Reihenfolge: `part` > `chapter` > `section` > `subsection` > `subsubsection` > `paragraph` > `subparagraph`.
- Beim `scrreprt` entfällt `part`
- Beim `scrartcl` entfallen `part`, `chapter`, so dass hier die `section` die höchste Stufe ist.
- Man verwendet die Strukturen zur Gliederung von euren Sektionen, wie bspw. Einleitung, Theorie, Auswertung...

Dokumentenklassen

- Zur Verwendung von Dokumentenklassen und Paketen sowie Befehlen in eurem Code werden diese mit einem „\“ induziert.
- Die Idee beim Einbinden von den meisten Paketen oder Befehlen ist der Name, gefolgt von Optionen in eckigen Klammern (nicht notwendig) und in geschweiften Klammern das Argument.

```
| \documentclass[12pt]{scrartcl}
```

In diesem Beispiel würde die Schriftgröße von dem Artikel auf 12 Punkte per Option gesetzt werden.

Dokumentenklassen - Optionen

<code>10pt</code> , <code>11pt</code> , <code>12pt</code>	Legt die Standardgröße im Dokument fest, ohne Angabe werden <code>10pt</code> gewählt.
<code>a4paper</code> , <code>letterpaper</code> , <code>a5paper</code> , <code>b5paper</code> , <code>executivepaper</code> , <code>legalpaper</code>	Definiert die Papiergröße, Standard ist <code>letterpaper</code> , im EU-Raum ist <code>a4paper</code> der Standard
<code>fleqn</code>	Flush left equation - setzt Formeln linksbündig statt zentriert
<code>leqno</code>	Setzt Formelnummerierung nach links statt rechts
<code>titlepage</code> , <code>notitlepage</code>	Spezifiziert ob nach dem Titel eine neue Seite begonnen wird oder nicht; die <code>article</code> Klasse beginnt keine neue Seite, <code>report</code> und <code>book</code> hingegen schon
<code>onecolumn</code> , <code>twocolumn</code>	Ordnet \LaTeX an das Dokument ein- oder zweispaltig zu setzen (wie bspw. in Wiss. Papern)

Dokumentenklassen - Optionen

<code>oneside</code> , <code>twoside</code>	Gibt an, ob das Dok. ein-/zweiseitig ist. <code>article</code> und <code>report</code> sind per Standard einseitig, <code>book</code> hingegen zweiseitig. <i>Beachte: diese Option betrifft nur den Stil des Doks und die Option <code>twoside</code> sagt dem Drucker nicht, dass er das Dok tatsächlich zweiseitig drucken soll.</i>
<code>landscape</code>	Setzt das Dokument in Querformat
<code>draft</code>	lässt \LaTeX Probleme mit der Silbentrennung und dem Ausrichten als Blocksatz mit einem kleinen Quadrat am rechten Rand der betroffenen Zeile anzeigen, damit diese schnell von einem Menschen entdeckt werden können.

- Es gibt auch noch weitere, die sind für die meisten Anwendungen nicht relevant.

Umgebungen (environments)

- Die häufigsten Bauteile in eurem Code sind Umgebungen bzw. environments.
- Sie folgen der groben Logik der Pakete mit [Optionen] und {Argumenten}, haben jedoch einen Anfang (`begin`) und ein Ende (`end`).
- Das einfachste Dokument ist durch die Dokumentenklasse und der `document` Umgebung erstellbar.

```
\documentclass[a4paper]{scrartcl}
\begin{document}
  Hello World!
\end{document}
```

Output:

Hello World!

Umgebungen (environments)

- Die Umgebungen beschreiben häufig ihren Inhalt oder ihre Funktion
- `equation` für eine einzige Formel, `align` für mehrere, ausgerichtete Formeln, `figure` für Abbildungen, `table` für Tabellen, `enumerate` für nummerierte Aufzählungen, `center` zur Zentrierung von Text, und viele mehr.
- Einige Umgebungen lassen sich auch per Schalter (switches) mit nur einem Befehl verwenden, allerdings wirken diese dann global.
- Schalter in Umgebungen wirken nicht außerhalb dieser, bspw. `\centering`. Dazu später mehr in den Beispielen.
- Am einfachsten einzufügen mit **STRG + E** in TeXstudio.

Wie erstellt man ein Dokument

- \LaTeX erzeugt einige Hilfsdateien beim Kompilieren, es wird stark empfohlen pro Dokument einen eigenen Ordner anzulegen.
- Euer Source Code muss in einer "Name".tex Datei gespeichert werden. Dies übernimmt idR euer Editor.
- Sämtlicher Code, der vor „`\begin{document}`“ steht, wird Präambel (preamble) genannt. Er enthält alle Pakete und Einstellungen, die sich auf das gesamte Dokument auswirken.
- Das im ersten Beispiel gezeigte Dokument benötigt zahlreiche Pakete, um die volle Kontrolle über Inhalte und Formatierung des Erscheinungsbildes zu erhalten.
- Jede Präambel **muss** mit der Dokumentenklasse beginnen.

Wie erstellt man ein Dokument - Strukturen

- Wie bereits in den Dokumentenklassen erwähnt, stehen euch Strukturen zur Verfügung, um eure Sektionen sinnvoll zu gliedern und damit den Text besonders überschaubar zu gestalten.

```
\documentclass[a4paper]{scrartcl}
\begin{document}
\section{Einleitung}
\subsection{Historie}
\subsection{Forschungsstand}
Erster Satz
\section{Theorie}
\end{document}
```

Output:

1 Einleitung

1.1 Historie

1.2 Forschungsstand

Erster Satz

2 Theorie

Wie erstellt man ein Dokument - Strukturen

- Die Strukturen tauchen auch in TeXstudio auf, so dass man dort schnell zu den entsprechenden Stellen springen kann.

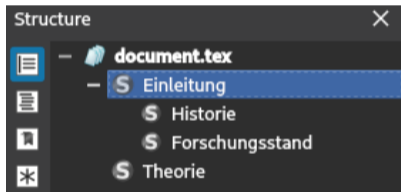
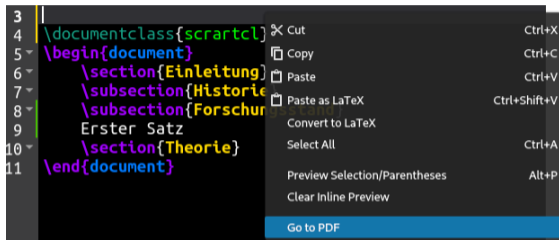


Abbildung 1: Struktur-Anzeige in TeXstudio.

- Man erhält per Rechtsklick im Code und in der kompilierten PDF die Möglichkeit an die jeweilige Stelle zu springen.

Wie erstellt man ein Dokument - Strukturen



```
3 |
4 | \documentclass{scrartcl}
5 | \begin{document}
6 |   \section{Einleitung}
7 |   \subsection{Historie}
8 |   \subsection{Forschung}
9 |   Erster Satz
10 |   \section{Theorie}
11 | \end{document}
```

A context menu is open over the code, listing actions such as Cut, Copy, Paste, Paste as LaTeX, Convert to LaTeX, Select All, Preview Selection/Parentheses, Clear Inline Preview, and Go to PDF.

Abbildung 2: Vom Source Code zur PDF.

1 Einleitung

1.1 Historie

1.2 Forschungsstand

Erster Satz

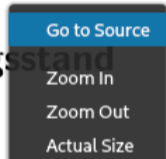


Abbildung 3: Von der PDF zum Source Code.

- Man erkennt, die Strukturen helfen euch nicht nur zur Gliederung von den Inhalten, sondern auch des Codes. Selbst in der PDF kann man sie zum Springen an die jeweiligen Inhalte erscheinen lassen durch Hilfe des Pakets [hyperref](#).

Wie erstellt man ein Dokument - Verzeichnisse

- Sobald Strukturen verwendet wurden, lässt sich ein Inhaltsverzeichnis sehr leicht erstellen mit dem Befehl `\tableofcontents`.

```
\documentclass[a4paper]{scrartcl}
\begin{document}
\tableofcontents
\section{Einleitung}
\subsection{Historie}
\subsection{Forschungsstand}
Erster Satz
\section{Theorie}
\end{document}
```

Output:

Contents

1	Einleitung	1
1.1	Historie	1
1.2	Forschungsstand	1
2	Theorie	1

1 Einleitung

1.1 Historie

1.2 Forschungsstand

Erster Satz

2 Theorie

Wie erstellt man ein Dokument - Verzeichnisse

- Das Inhaltsverzeichnis ist in diesem Fall noch auf standardmäßigem Englisch, dies wird später durch das Paket `babel` korrigiert.
- Es taucht direkt dort auf, wo man es aufruft im Code, man hat die Möglichkeit es auf eine eigene Seite zu setzen und auch die Seitenzahl vom Rest des Dokuments zu entkoppeln.
- Analog dazu lassen sich für Abbildungen und Tabellen eigene Verzeichnisse mittels `\listoffigures` und `\listoftables` erzeugen. Hierzu müssen im Dokument natürlich erst einmal Bilder und Tabellen vorhanden sein, entsprechend dazu später mehr.

Wie erstellt man Dokumente - Formatierung - Schriftgröße

- Die Schriftgröße (font size) lässt sich mit einfachen Schaltern im Text ändern.
- Es sei angemerkt, dass die Größen von der durch euch angegebenen Standardgröße abhängen.

Befehl	<code>\tiny</code>	<code>\scriptsize</code>	<code>\footnotesize</code>	<code>\small</code>	<code>\normalsize</code>
Größe	Text	Text	Text	Text	Text
Befehl	<code>\large</code>	<code>\Large</code>	<code>\LARGE</code>	<code>\huge</code>	<code>\Huge</code>
Größe	Text	Text	Text	Text	Text

Wie erstellt man Dokumente - Formatierung - Schriftgröße

- Hier ein Überblick wie die jeweiligen Befehle basierend von der Standardgröße die Schriftgröße skalieren.

Größe	10pt	11pt	12pt
<code>\tiny</code>	5	6	6
<code>\scriptsize</code>	7	8	8
<code>\footnotesize</code>	8	9	10
<code>\small</code>	9	10	10,95
<code>\normalsize</code>	10	10,95	12
<code>\large</code>	12	12	14,4
<code>\Large</code>	14,4	14,4	17,28
<code>\LARGE</code>	17,28	17,28	20,74
<code>\huge</code>	20,74	20,74	24,88
<code>\Huge</code>	24,88	24,88	24,88

Wie erstellt man Dokumente - Formatierung - Schriftstil

- Um den Stil eures Textes zu ändern, kann der Text markiert und mit **STRG + B** in fetter Schrift dargestellt werden durch den Befehl `\textbf{}` (text bold font).
- Für kursiv kann **STRG + I** für `\textit{}` (text italics) verwendet werden.
- Mittels `\underline{}` wird Text unterstrichen.
- Zentrierung von Text kann entweder durch die `center` Umgebung oder den Schalter `\centering` durchgeführt werden.
- Damit Schalter (wie `\centering` oder später auch Farbschalter `\color{red}`) nicht für das ganze Dokument wirken, können sie mit geschweiften Klammern eingegrenzt werden.
- Eine neue Zeile kann mit **STRG + ENTER** für „`\\`“ oder durch `\newline` erzeugt werden.

Wie erstellt man Dokumente - Formatierung - Schriftstil

```
\begin{document}
```

```
Dieser Text ist ein \textbf{Blocksatz}.
```

```
\begin{center}
```

```
  Dieser ist durch die \textit{Umgebung}  
  zentriert.
```

```
\end{center}
```

```
Dieser ist wieder ein \textbf{Blocksatz}.
```

```
{\centering
```

```
  Der hier ist zentriert durch den \underline  
  {Schalter}.\}
```

```
Der letzte soll nicht zentriert sein, {\} sei  
Dank.
```

```
\end{document}
```

Output:

Dieser Text ist ein **Blocksatz**.

Dieser ist durch die *Umgebung* zentriert.

Dieser ist wieder ein **Blocksatz**.

Der hier ist zentriert durch den Schalter.

Der letzte soll nicht zentriert sein, {} sei Dank.

Wie erstellt man Dokumente - Formatierung - Seitenstil

- Weitere Umgebungen, um Text zu positionieren sind `\flushleft` und `\flushright`, die ihn jeweils nach links oder rechts ausrichten.
- Standardmäßig besitzen Seiten den Seitenstil . Den Seitenstil kann man für **eine** einzige Seite mit `\thispagestyle{}` und für das **ganze** Dokument mit `\pagestyle{}` gesetzt werden.
- Die basic Stile sind `empty`, damit keine Seitenzahl angezeigt wird, `plain` für eine einfache Seitenzahl und `headings`, damit liegt die Seitenzahl oben in der Ecke der Kopfzeile.
- Kopf- und Fußzeilen werden später durch das Paket `fancyhdr` (fancy header) gehandhabt für simple aber schicke Umsetzung sowie Positionierung der Seitenzahl.

Wie erstellt man Dokumente - Formatierung - Abstände

- Für ein Deckblatt benötigt man nur eine Kombination von Zeilenumbrüchen, Zentrierungen und Schriftgrößen. Mit `\pagebreak` wird ein Seitenumbruch erzeugt.
- Die Abstände von Zeilenumbrüchen lassen sich durch in eckige Klammern gesetzte Abstände zusätzlich steuern, bspw. `\[1cm]`.
- Weitere horizontale Abstände lassen sich durch `\hspace{}` und vertikale Abstände mit `\vspace{}` einfügen.
- Als Einheiten für die Abstände stehen `pt,mm,cm,in` und paar weitere zur Verfügung.
- Mit dem Befehl `\today` wird das heutige Datum automatisch eingefügt.
- Die Abstände vom folgenden Beispiel sollten angepasst werden.

Wie erstellt man Dokumente - Formatierung - Abstände

```
\begin{document}
\begin{center}
  \large \thispagestyle{empty}
  \textbf{--Deckblatt--}\!\!\! [0.2cm]
  \textbf{Titel}\!\!\! [0.2cm]
  \textbf{Betreut durch:}\!\!\!
  Betreuer 1\!\!\!
  Betreuer 2\!\!\! [0.5cm]
  \textbf{Geschrieben von:}\!\!\!
  Max Mustermann [1234567]\!\!\!
  Marie Musterfrau [7654321]\!\!\! [0.5cm]
  \textbf{Datum:}\!\!\!
  \today
\end{center} \pagebreak
\end{document}
```

Output:
–Deckblatt–

Titel

Betreut durch:

Betreuer 1

Betreuer 2

Geschrieben von:

Max Mustermann [1234567]

Marie Musterfrau [7654321]

Datum:

May 16, 2022

Wie erstelle ich ein Dokument - Formatierung - Sonderzeichen

- Sämtliche Zeichen, die nicht standardmäßig im Englischen Alphabet sind, sind Sonderzeichen. Dazu zählen wegen der codebasierten Natur von \LaTeX auch `%`, `&`, `\`, `_`, `~`, `\`, `{`, `}`.
- Damit `%`, `&`, `\`, `_` im Text auftauchen statt vom Compiler ausgewertet zu werden, müssen diese mit einem `\` voraus geschrieben werden. Für alle anderen Symbole wie `\` oder `~` benötigt man für das spezielle Sonderzeichen einen Befehl `\textbackslash`, `\textasciitilde`.

Hier `\textbackslash` könnte `\&` Ihre `\%` `\{Werbung\}` `_`stehen`\textasciitilde`.

Output:

Hier `\`könnte `&` Ihre `%` `{Werbung}` `_`stehen`~`.

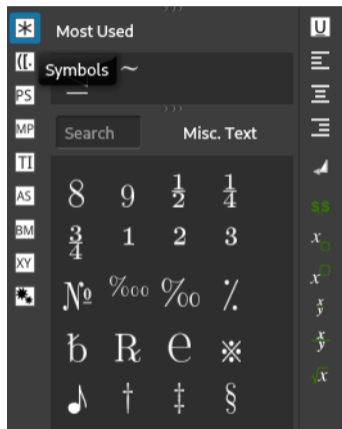
Wie erstelle ich ein Dokument - Formatierung - Sonderzeichen

- Eine Liste an Symbolen findet ihr bei TeXstudio in der Symbolleiste in der Side Bar links.
- Die Symbole unter „Misc. Math“ erfordern den Mathe Modus, entweder durch entsprechende Umgebungen oder den Mathe-Delimiter $\$ \$$.

| Eine $E = \hbar \cdot \nu$ Gleichung.

Output:

Eine $E = \hbar \cdot \nu$ Gleichung.



Wie erstelle ich ein Dokument - Formatierung - Aufzählungen

- Man kann Aufzählungen mit den `itemize` und `enumerate` Umgebungen umsetzen.
- `enumerate` numeriert mit Zahlen, `itemize` setzt nur Punkte.
- Für jedes Objekt in der Aufzählung ist ein `\item` Befehl vorläufig.
- Es spielt keine Rolle was hinter dem Item steht, es kann neben ganzen Paragraphen auch ein Bild, eine Tabelle, eine Gleichung etc. sein.
- Die Label (Zahlen oder Punkte) lassen sich mit dem Paket `enumitem` einstellen.

Wie erstelle ich ein Dokument - Formatierung - Aufzählungen

```
\begin{document}
  \begin{itemize}
    \item Dies
    \item ist
    \item eine
    \item Aufzählung
  \end{itemize}
  \begin{enumerate}
    \item Dies
    \item ist
    \item eine
    \item Aufzählung
  \end{enumerate}
\end{document}
```

Output:

- Dies
 - ist
 - eine
 - Aufzählung
-
1. Dies
 2. ist
 3. eine
 4. Aufzählung

Hilfreiche Pakete im Überblick

- \LaTeX hat ohne Weiteres nur sehr beschränkte Möglichkeiten für spezielle Formatierungen oder mit externen Inhalten wie Bildern oder PDFs umzugehen.
- Die Funktionalitäten können *enorm* durch Einbinden von Paketen erweitert werden.
- Ausführliche Dokumentationen zu den jeweiligen Paketen findet man im Comprehensive \TeX Archive Network ([CTAN](#)).
- Hier wird eine Auswahl von hilfreichen Paketen mit den wichtigsten Befehlen vorgestellt, damit man ein vernünftiges Laborprotokoll oder eine Thesis erstellen kann.
- Konkretere Anwendungen findet ihr in der beigefügten Vorlage.

Hilfreiche Pakete im Überblick - amsmath, amssymb, amsfonts

- Das Tripplett der Pakete von der American Math Society bieten umfangreiche Symbole und neue Umgebungen für Mathe Modi und Formeln.
- Mit der `equation` Umgebung lässt sich eine zentrierte Formel samt Nummer einfügen.
- Mit der `align` Umgebung lassen sich viele, ausgerichtete Formeln untereinander einfügen, für bspw. Rechenwege. Ausrichtung am `&` Zeichen.
- Die Formelnummern kann man für Querverweise im Dokument verwenden, um nicht per Hand bei jeder kleinen Änderung bspw. Gleichung (3) dann in Gl. (5) ändern zu müssen wenn sich die Anzahl der Gleichungen ändert.
- Dies wird durch `\label{}` umgesetzt, sie können für alles was durch `\LaTeX` nummeriert wird gesetzt werden (Beschriftungen von Abbildungen & Tabellen, Sektionen,...)

Hilfreiche Pakete im Überblick - amsmath, amssymb, amsfonts

```
\documentclass[a4paper,12pt]{scrartcl}
\usepackage{amsmath}
\begin{document}
\begin{equation}
  a + b = c \label{eq:einfach}
\end{equation}
\begin{align}
  \int_a^b f(x) dx &= [\ln(x)]_a^b \label{eq:medium} \\
  T(x) &= \sum_{n=1}^k \frac{f'(a)}{n!} (x-a)^n \\
  \label{eq:schwer}
\end{align}
In Gl.~\eqref{eq:einfach} ist's simpel,
dann kommt Gl.~\eqref{eq:medium} und Gl.~\eqref{eq:schwer}.
\end{document}
```

Hilfreiche Pakete im Überblick - amsmath, amssymb, amsfonts

- Querverweise setzt ihr mit `\ref{}`, hier setzt `\eqref{}` automatisch die Klammern.
- Es ist sinnvoll Label kategorisch mit `eq`, `fig`, `tab` zu unterteilen.
- Die Tilde im Code ist ein geschütztes Leerzeichen, die zwei Symbole/Wörter links/rechts werden nicht getrennt durch Zeilenumbrüche.

Output:

$$a + b = c \quad (1)$$

$$\int_a^b f(x)dx = [\ln(x)]_a^b \quad (2)$$

$$T(x) = \sum_{n=1}^k \frac{f'(a^n)}{n!} (x - a)^n \quad (3)$$

In Gl. (1) ist's simpel, dann kommt Gl. (2) und Gl. (3).

- Mit `equation*` und `align*` gibt es keine Nummerierung der Gleichungen.

Hilfreiche Pakete im Überblick - inputenc, fontenc, babel

- Damit LaTeX Deutsche Sonderzeichen wie ä, ö, ü, ß auch einlesen kann ohne Befehle zu schreiben, verwendet man das Paket `inputenc` (input encoding) mit der Option `utf8`.

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

- Um eure Schrift auch glatt vektorisiert zu generieren empfiehlt es sich `fontenc` mit der `T1` Option zu nutzen.

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

- Für die Silbentrennung und Übersetzung von automatisch generierten Texten bei Beschriftungen (Figure, Table, Datum von `\today`) kann `babel` mit der Option `ngerman` (n steht für neue Rechtschreibung) benutzt werden.

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

Hilfreiche Pakete im Überblick - float, graphicx

- Das `float` Paket ermöglicht eine Option für Bilder und Tabellen exakt dort zu positionieren, wo sie im Code auftauchen. \LaTeX setzt sie sonst automatisch „günstig“, um Leerstellen zu minimieren.
- Das `graphicx` Paket stellt den Befehl `\includegraphics[options]{file}` zur Verfügung, mit dem sich JPG, PNG und PDF Dateien einbinden lassen. Wird der Befehl so eingesetzt taucht das Bild in der gleichen Zeile wie der Text auf. Um das Bild vom Text zu separieren sollte der Befehl innerhalb einer `figure` Umgebung gesetzt werden.
- Optionen an der `figure` (oder `table`) Umgebung zur Positionierung sind `[t|b|h|H]` für top, bottom, ungefähr hier und direkt HIER. Letzte Option existiert nur durch das `float` Paket.

Hilfreiche Pakete im Überblick - float, graphicx

```
\documentclass[a4paper]{scrartcl}
\usepackage{float}
\usepackage{graphicx}
\usepackage[ngerman]{babel}
\begin{document}
  Hier könnte Ihre Werbung stehen.
  \begin{figure}[H]
    \centering
    \includegraphics[width=0.4\linewidth]{example-image-a}
    \caption{Dies ist eine Bildbeschriftung.}\label{fig:Bild}
  \end{figure}
  In Abbildung~\ref{fig:Bild} ist ein zentriertes Beispielbild zu sehen.
\end{document}
```

Hilfreiche Pakete im Überblick - float, graphicx

- Der `\caption{}` Befehl setzt Beschriftungen für Bilder und Tabellen. Da er eine Nummer generiert kann man Label für Querverweise setzen.
- Als Option wurde hier die Breite anhand der Seitenbreite skaliert. Der `\linewidth` Befehl enthält stets automatisch die exakte Strecke.
- Ohne vorlaufenden Wert oder mit `1\linewidth` erstreckt sich das Bild von Rand zu Rand.

Output:

Hier könnte Ihre Werbung stehen.



Abbildung 1: Dies ist eine Bildbeschriftung.

In Abbildung 1 ist ein zentriertes Beispielbild zu sehen.

Hilfreiche Pakete im Überblick - fancyhdr

- Das `fancyhdr` Paket bietet eine neue `\pagestyle{fancy}` Option an, mit der man leicht eigene Kopfzeilen (header) und Fußzeilen (footer) konfigurieren kann.
- Die Kopf- und Fußzeilen besitzen jeweils 3 Positionen, links, zentriert und rechts. Alle 3 lassen sich separat mit eignen Text oder der Seitenzahl füllen, so dass man Kontrolle über dessen Position und Format erhält.
- Die Seitenzahl ist im Befehl `\thepage` enthalten und zeigt die aktuelle Seitenzahl innerhalb des Dokuments an (analog wie `\today` funktioniert)
- Der Zähler für die Seitenzahl, dessen Information `\thepage` pro Seite ausgibt, ist im counter `page` enthalten. Ändern lässt er sich mittels `\setcounter{page}{zahl}`.

Hilfreiche Pakete im Überblick - fancyhdr

```
\documentclass[a4paper]{scrartcl}
\usepackage{fancyhdr}
\pagestyle{fancy}
\fancyhf{}
\lhead{Versuch 1.5}
\chead{Experiment}
\rhead{Seite \thepage}
\lfoot{Das}
\cfoot{ist}
\rfoot{cool}
\begin{document}
  \setcounter{page}{5}
  Eine Demo des fancy pagestyles.
\end{document}
```

Output:

Versuch 1.5	Experiment	Seite 5
-------------	------------	---------

Eine Demo des fancy pagestyles.

Das	ist	cool
-----	-----	------

Bald geht es weiter

Fortsetzung folgt...