

L^AT_EX in a Nutshell

Ein „knapper“ Einstieg mit zahlreichen Beispielen

Stiv Nikolov

sn01@tu-clausthal.de

3. November 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
 - Dateien
- TikZ & pgfplots
- Quellenverwaltung
- \LaTeX Compiler
- Tipps zum Debugging
- Workflow

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 L^AT_EX nutzen?

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



L^AT_EX

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, da diese spezifisch für TeXstudio sind.

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, 11pt, 12pt</code>	Legt die Standardgröße im Dokument fest, ohne Angabe werden 10pt gewählt.
<code>a4paper, letterpaper, a5paper, b5paper, executivepaper, 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, 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, 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 numerierte 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.
- Mit `STRG + T` können markierte Zeilen und Code Blöcke (aus)kommentiert werden.

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 Anhang befindliche 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}
```

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.

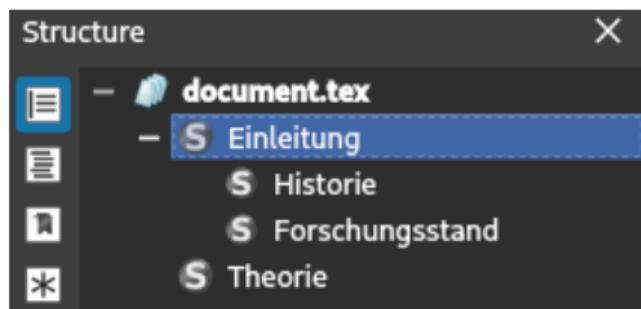
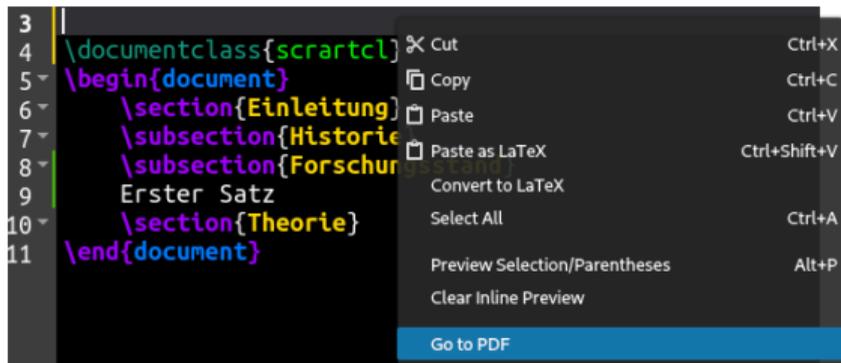


Abbildung 1: Struktur-Anzeige in TeXstudio.

- 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}
```

The image shows a code editor with a context menu open over the source code. The menu items include: Cut (Ctrl+X), Copy (Ctrl+C), Paste (Ctrl+V), Paste as LaTeX (Ctrl+Shift+V), Convert to LaTeX, Select All (Ctrl+A), Preview Selection/Parentheses (Alt+P), Clear Inline Preview, and Go to PDF (highlighted in blue).

Abbildung 2: Vom Source Code zur PDF.

1 Einleitung

1.1 Historie

1.2 Forschungsstand

Erster Satz

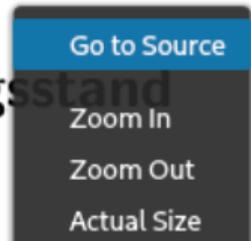


Abbildung 3: Von der PDF zum Source Code.

- 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}
```

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 des Textes zu ändern, kann er 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}
```

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 `\pagestyle{plain}`. 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 im jeweiligen Sprachformat (einstellbar durch `babel`) eingefügt.
- Die Abstände vom folgenden Beispiel sollten von euch 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:}\!\!\[0.2cm]
  Betreuer 1\!\!\[0.2cm]
  Betreuer 2\!\!\[0.5cm]
  \textbf{Geschrieben von:}\!\!\[0.2cm]
  Max Mustermann [1234567]\!\!\[0.2cm]
  Marie Musterfrau [7654321]\!\!\[0.5cm]
  \textbf{Datum:}\!\!\[0.2cm]
  \today
\end{center} \pagebreak
\end{document}
```

–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` gefolgt von \ oder ~ für ein Leerzeichen hinter dem Symbol.

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

Output:

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

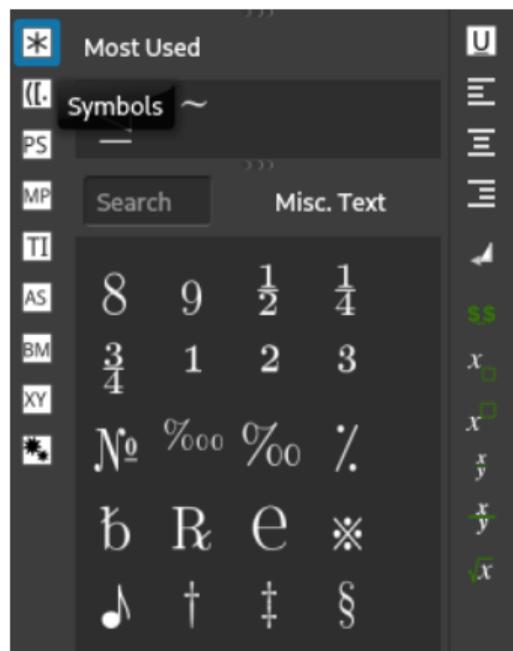
Wie erstelle ich ein Dokument - Formatierung - Sonderzeichen

- 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 In-Line 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}
```

- 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,amssymb,amsfonts}
\allowdisplaybreaks
\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^{(n)}(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 Elemente werden nicht getrennt durch Zeilenumbrüche.
- Mit `\allowdisplaybreaks` in der Präambel können Gl. über mehrere Seiten laufen.

Output:

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

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

$$T(x) = \sum_{n=1}^k \frac{f^{(n)}(a)}{n!} (x - a)^n \quad (3)$$

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

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

Hilfreiche Pakete im Überblick - geometry

- Das `geometry` Paket ermöglicht das Einstellen von den Abständen eurer Seitenränder des Dokuments.
- Mit Optionen oder einem separaten `\geometry{top = 2cm, ...}` Befehl in der Präambel werden die Abstände gesetzt.
- Die `showframe` Option macht den Seitenrand sichtbar, um zu überprüfen ob ggf. Tabellen / Bilder übertreten.

```
| \usepackage[top = 2cm, bottom = 2cm, left = 2cm, right = 2cm]{geometry}
```

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` (für Type 1 fonts) 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 (Kein Leerraum durch Text) und direkt HIER (Text wird durch Bild abgeschnitten, Position forciert). 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 des ganzen Textfelds.
- 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.

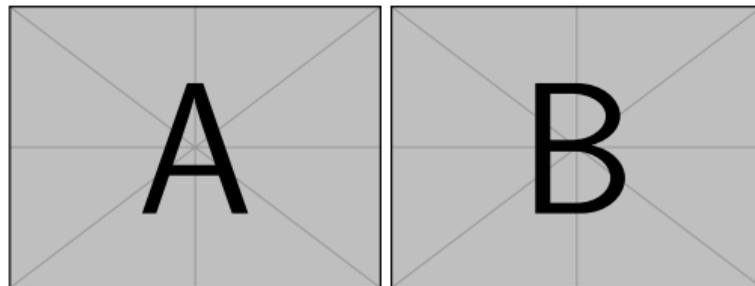
Hilfreiche Pakete im Überblick - subcaption

- Mit dem Paket `subcaption` lassen sich mehrere Abbildungen (und Tabellen) nebeneinander und untereinander anordnen mit `subfigure` Umgebungen, denen eine Breite zugewiesen wird.
- Innerhalb der `subfigure` Umgebung ist die `\linewidth` Breite vom angegebenen Wert überschrieben, sprich nutzt man dort im Nachhinein die volle Breite, füllt das Bild nur den Bereich aus.
- In den `\subcaption{}` Befehlen sind Einzelbeschriftungen möglich, mit der `\caption{}` für die Beschriftung aller Bilder zusammen als eine Abbildung.
- Die `\subcaption{}` kann auch mit einem Label für Querverweise versetzt werden.
- Im folgendem Beispiel sind die Pakete `graphicx` und `subcaption` geladen.

Hilfreiche Pakete im Überblick - subcaption

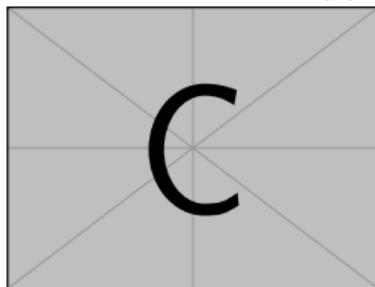
```
\begin{figure}[h] \centering
  \begin{subfigure}{0.4\linewidth} \centering
    \includegraphics[width=\linewidth]{example-image-a}
    \subcaption{Bild 1}
  \end{subfigure} % <- Prozent = nebeneinander, ohne = untereinander
  \begin{subfigure}{0.4\linewidth} \centering
    \includegraphics[width=\linewidth]{example-image-b}
    \subcaption{Bild 2}
  \end{subfigure}
  \begin{subfigure}{0.4\linewidth} \centering
    \includegraphics[width=\linewidth]{example-image-c}
    \subcaption{Bild 3}
  \end{subfigure}
  \caption{Drei Bilder neben- und untereinander.}
\end{figure}
```

Hilfreiche Pakete im Überblick - subcaption



(a) Bild 1

(b) Bild 2



(c) Bild 3

Figure 1: Drei Bilder neben- und untereinander.

Hilfreiche Pakete im Überblick - wrapfig

- Das Paket `wrapfig` erlaubt es Bilder von Text umranden zu lassen mit der `wrapfigure` Umgebung.
- Mit den Argumenten `R,L` lassen sie sich rechts, links und bei `twoside` Dokumenten mit `I,O` innen oder außen vom Binderand setzen.
- Der folgende Code Block geht **über 2 Folien**

```
\documentclass[a4paper,12pt]{scrartcl}
\usepackage{wrapfig}
\usepackage{graphicx}
\parindent0mm
\begin{document}
  \begin{wrapfigure}{R}{0.3\linewidth}
    \centering
    \includegraphics[width=0.25\linewidth]{example-image-a}
```

Hilfreiche Pakete im Überblick - wrapfig

```
        \caption{Musterbild} \label{fig:a}  
\end{wrapfigure}
```

Text ...

```
\begin{wrapfigure}{L}{0.3\linewidth}  
\centering  
\includegraphics[width=0.25\linewidth]{example-image-b}  
\caption{Musterbild} \label{fig:b}  
\end{wrapfigure}
```

Text ...

```
\end{document}
```

Hilfreiche Pakete im Überblick - wrapfig

Text Text Text Text Text Text Text Text Text
Text Text Text Text Text Text Text Text Text
Text Text Text Text Text Text Text Text Text
Text Text Text Text Text Text Text Text Text
Text Text Text Text Text Text Text Text Text
Text Text Text Text Text Text Text Text Text
Text Text Text Text Text Text Text Text Text
Text Text Text Text Text Text Text Text Text
Text Text Text Text Text Text Text Text Text



Figure 1: Musterbild

Text Text



Figure 2: Musterbild

Text Text Text Text Text Text Text Text Text
Text Text Text Text Text Text Text Text Text
Text Text Text Text Text Text Text Text Text
Text Text Text Text Text Text Text Text Text
Text Text Text Text Text Text Text Text Text
Text Text Text Text Text Text Text Text Text
Text Text Text Text Text Text Text Text Text
Text Text Text Text Text Text Text Text Text
Text Text Text Text Text Text Text Text Text

Text Text Text Text Text Text Text Text Text

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. **Es sollte nach `geometry` geladen werden!**
- Die Kopf- und Fußzeilen besitzen jeweils 3 Positionen, links, zentriert und rechts. Alle 3 lassen sich separat mit eigenem 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 lokal 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}
```

Versuch 1.5 Experiment Seite 5

Eine Demo des fancy pagestyles.

Das ist cool

Hilfreiche Pakete im Überblick - tocbibind, bm

- Das Paket `tocbibind` lässt neben Sektionen auch Verzeichnisse (Inhaltsverzeichnis, Tabellenverzeichnis, Abbildungsverzeichnis,...) auch im Inhaltsverzeichnis auftauchen.
- Mit den Optionen `[nottoc, notbib, notlof,notlot]` werden die jeweiligen Verzeichnisse nicht aufgenommen.
- Das Paket `bm` (bold math) erzeugt fette Schrift im Mathe-Modus.

```
| $ \alpha + \beta = \bm{\alpha + \beta} $
```

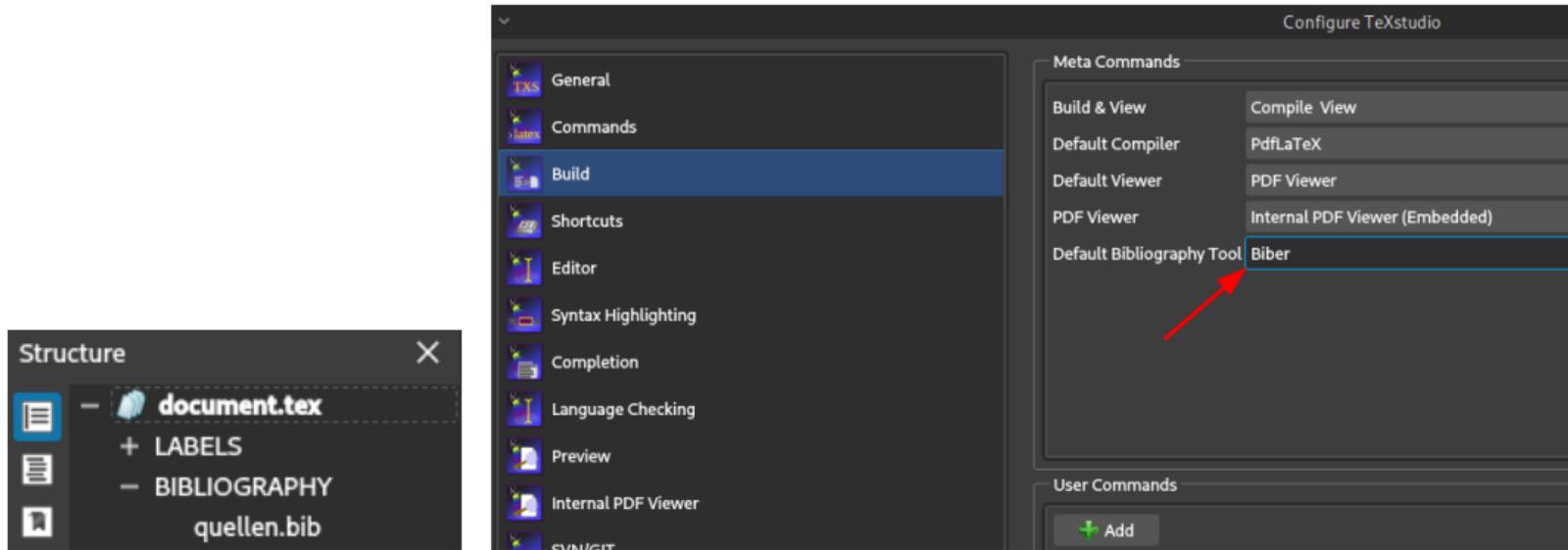
Output:

$$\alpha + \beta = \boldsymbol{\alpha + \beta}$$

Hilfreiche Pakete im Überblick - biblatex, biber

- **biblatex** verwaltet für euch Zitate im Dokument und den Zitierstil, dessen Daten in einer **Name.bib** Datei gespeichert sind.
- Ihr könnt von Papern Zitate exportieren in diesem Format und eure Quelldatenbank sehr einfach erweitern.
- Um zitieren zu können sollte **biber** als Backend auch in TeXstudio unter Optionen → Konfigurieren → Erzeugen eingestellt werden. Stellt sicher, dass **biber** auch installiert ist.
- Eine häufige Fehlerquelle für Errors sind Sonderzeichen (&, _,...) in Titeln, im Abstract oder in den URLs von Quellen, diese müssten manuell mit \ oder den jeweiligen Befehlen \LaTeX -kompatibel umgeschrieben werden.

Hilfreiche Pakete im Überblick - biblatex, biber



- Eure verwendete Datei erscheint in der Side Bar links im Editor unter Struktur, wo die Sektionen stehen unter BIBLIOGRAPHY

Hilfreiche Pakete im Überblick - biblatex, biber

Inhalt einer `quellen.bib` Datei:

```
@book{Citekey,  
  title = {Cooler Titel},  
  author = {Max Mustermann},  
  publisher = {Springer Verlag},  
  year = {2022},  
  month = {06},  
  edition = {4}  
}
```

```
\documentclass{scrartcl}  
\usepackage[%  
backend=biber,  
citestyle=ieee]{biblatex}  
\addbibresource{quellen.bib}  
\begin{document}  
  Siehe Quelle \cite{Citekey}.  
  \printbibliography  
\end{document}
```

- Mehrere Autoren werden mit „and“ getrennt, da Kommata Nachname, Vorname Formationen trennen.
- Eine Übersicht von einigen verschiedenen Zitierstilen findet ihr [hier](#).

Output:

Siehe Quelle [1].

Literatur

[1] Max Mustermann. *Cooler Titel*.
4. Aufl. Springer Verlag, Juni 2022.

- Die Sprache von der Überschrift wird über `babel` gesteuert.
- Im Code wird der cite key zitiert, ändert den zu was sinnvollem (bspw. Autor2022-Thema).
- `\addbibresource{}` bindet eure Quelldatei ein.
- `\printbibliography` gibt euer Literaturverzeichnis im angegebenen Zitierstil aus.
- Die Typen der Quellen (`@article`, `@book`, `@online`,...) geben automatisch die Formatierung vor.

Hilfreiche Pakete im Überblick - biblatex, biber

- IdR findet man für alle Paper eine Exportfunktion für Zitate, das ist für Bücher nicht häufig die Norm. Eine eigene Quelle ist schnell angelegt mit der im Beispiel gezeigten Struktur. Folgende Infofelder gibt es:

author	Autor(en)
title	Titel (vom Kapitel oder Artikel)
year, month, day	Veröffentlichungsdatum
publisher	Verlag
edition	Auflage
doi	DOI Nummer
isbn	ISBN Nummer
eprint	eBook Link

Hilfreiche Pakete im Überblick - biblatex, biber

url	Weblink zum Werk
issue	Ausgabe
volume	Band
pages	Seiten (Angabe mit Bereich, 10-20 bspw.)
address	Adresse
school	Bildungsstätte
editor	Redakteur
institution	Institution (bspw. Firma)
organization	Organisation (bspw. Stiftung)
journal	Journal, in dem veröffentlicht wurde
abstract	Enthält das Abstract

Hilfreiche Pakete im Überblick - biblatex, biber

note	Eure Notizen und Kommentare, kein Output
booktitle	Buchtitel
number	Nummer (bspw. von technischen Anleitungen)
addendum	Ein Anhang, bspw. Zugriff auf einen Weblink = <code>{\newline Letzter Zugriff: 02.03.2022}</code>

- Eine Übersicht der 14 Eintragstypen (@book, @article...) findet man [hier](#).

Hilfreiche Pakete im Überblick - setspace

- Mit dem Paket `setspace` kann der Zeilenabstand mit vorgefertigten Abständen oder einer neuen `spacing` Umgebung nach eigenen Bedürfnissen eingestellt werden.
- Für Zeilenabstände stehen `\onehalfspacing` für 1.5-fachen und `\doublespacing` für doppeltem Zeilenabstand zur Verfügung.
- Der 1.5 Zeilenabstand in `MS Word` ist anders (eigentlich falsch, siehe [hier](#)) berechnet als `\onehalfspacing`, um den gleichen Effekt zu erzielen nutzt man die `spacing` Umgebung mit dem Argument 1.5.
- Ohne KOMA-Script Dokumentenklassen wird Standardmäßig die erste Zeile eines Absatzes im Englischen eingerückt, um dies auszuschalten kann der Befehl `\parindent0mm` in der Präambel verwendet werden.

Hilfreiche Pakete im Überblick - setspace

```
\documentclass{scrartcl}
```

```
\usepackage{setspace}
```

```
\parindent0mm
```

```
\begin{document}
```

```
  \begin{spacing}{2.61}
```

Hier könnte Ihre Werbung stehen. Hier könnte Ihre Werbung stehen. Hier könnte Ihre Werbung stehen.

```
  \end{spacing}
```

```
\end{document}
```

Hier könnte Ihre Werbung stehen. Hier könnte Ihre Werbung stehen. Hier könnte Ihre Werbung stehen.

Hilfreiche Pakete im Überblick - mhchem, enumitem

- Das `mhchem` Paket bietet die Möglichkeit chemische Reaktionen samt Pfeile einfach in Mathe-Umgebungen einzugeben mit dem `\ce{}` Befehl.
- Das `enumitem` Paket gibt die Möglichkeit die Label von Aufzählungen beliebig anzupassen. Ladet die Pakete in eure Präambel für das Beispiel.

```
$\ce{x Na(NH4)HPO4 ->[\Delta] (NaPO3)_x + x NH3 ^ + x H2O}$
```



```
\begin{enumerate}[label=\textbf{\alph*}]  
  \item Eins  
  \item Zwei  
  \item Drei  
\end{enumerate}
```

a) Eins

b) Zwei

c) Drei

Hilfreiche Pakete im Überblick - hyperref

- Das **hyperref** Paket verlinkt sämtliche Verweise (Inhaltsverzeichnis, Überschriften, Querverweise von Formeln, Abbildungen & Tabellen, Zitate,...) in eurem Dokument, so dass man durch Anklicken dieser Links direkt zur Stelle in der PDF springt.
- Die Gliederung samt Sektionen wird durch das Paket in der Navigation links eines PDF-Readers angezeigt (so wie von dieser Präsentation).
- Es gibt Optionen, mit denen Titel, Thema, und Metadaten eurer PDF (bspw. Autor) angegeben werden können.
- Da die Features interaktiv sind, können nur die Farben im Beispiel sichtbar gemacht werden von den Links der Überschriften im Dokument.
- **Es sollte als letztes Paket geladen werden!**

Hilfreiche Pakete im Überblick - hyperref

- Ohne Optionen sind alle Links mit einem Kasten umgeben, **dies wird beim Ausdrucken nicht mitgedruckt!**
- Damit es keine hässlichen Kästen sind, sondern die Nummern bzw. Überschriften selbst farbig **und ausdrückbar** werden, gibt es Optionen für den `\hypersetup{}` Befehl.
- `hyperref` Gibt einem ebenfalls einen Befehl, um URL Weblinks in ein Wort oder einen Satz zu hinterlegen mittels `\href{URL}{Text}`
- Der Code von folgendem Beispiel **geht über 2 Folien!**

Hilfreiche Pakete im Überblick - hyperref

```
\documentclass[a4paper,12pt]{scrartcl}
\usepackage{amsmath}
\usepackage{hyperref}
\hypersetup{% Einstellen der hyperref Optionen
  pdftitle={Titel/Name von der Datei}, %- title (PDF meta)
  pdfsubject={Thema der Arbeit}, %- subject (PDF meta)
  pdfauthor={Autor}, %- author (PDF meta)
  colorlinks=true, %- farbige Dokumentenlinks?
  pdfborder={0 0 0}, %- Entfernen des Kastens
  breaklinks=true, %- Zeilenumbrüche von URLs erlauben
  bookmarksnumbered=true,
  bookmarksopen=true,
  linkcolor=cyan, % Farbe der Dokumentenlinks
  citecolor=red % Farbe der Zitate von \cite{} Befehlen
}
```

Hilfreiche Pakete im Überblick - hyperref

```
\begin{document}
\tableofcontents
\section{Einleitung}
\subsection{Thema}
\begin{equation}
  a+b=2 \label{eq:gleichung}
\end{equation}
Hier ist ein klickbarer Querverweis zu Gl. \eqref{eq:gleichung}.\
Und zu Wikipedia findet man \href{https://en.wikipedia.org}{hier}.
\end{document}
```

Output:

Contents

1	Einleitung	1
1.1	Thema	1

1 Einleitung

1.1 Thema

$$a + b = 2 \tag{1}$$

Hier ist ein klickbarer Querverweis zu Gl. (1).

Und zu Wikipedia findet man [hier](#).

Hilfreiche Pakete im Überblick - xcolor

- Wenn man schon bei Farben ist, das Paket `xcolor` stellt einem den Befehl `\definecolor{Name}{Farbmodell}{Farbcode-nach-Modell}` zur Verfügung, mit dem eigene Farben definiert werden können.
- Die Farbmodelle sind unter anderem `RGB`, `CMYK`, `HTML`. Es gibt auch weitere, diese dürften jedoch die häufigste Anwendung finden.
- Ist eine Farbe mit dem Befehl definiert worden, kann diese nur noch mit dem Namen (oder vordefinierte Farben per Name) über den Befehl `\color{}` zum Einfärben genutzt werden.
- Der `\color{}` Befehl ist ein Schalter, sollte entsprechend entweder nur in Umgebungen benutzt werden oder falls er im Fließtext benutzt wird mit geschwungenen `{}` Klammern eingegrenzt werden, damit nicht der ganze Rest farbig wird.

Hilfreiche Pakete im Überblick - xcolor

- Es gibt verschiedene vordefinierte Farben in Namenslisten, bspw. `dvipsnames`, die als Option zum Paket geladen werden. Diese besitzen eine Großschreibung, die exakt so berücksichtigt werden muss für die Namen. In der Dokumentation gibt es auch weitere Namenslisten.
- Damit auch Tabellen zeilenweise oder zellenweise eingefärbt werden können, sollte die `table` Option neben Namenslisten als Option gesetzt werden.
- Farbmischung ist möglich mittels Ausrufezeichen, bspw. 60 % rot und 40 % gelb wäre `\color{red!60!yellow}` für diese **Farbe**.
- Vor dem Beispiel wird ein Ausschnitt der Dokumentation für die `dvipsnames` Namensliste beigefügt.

Hilfreiche Pakete im Überblick - xcolor

4.1 Base colors (always available)

 black	 darkgray	 lime	 pink	 violet
 blue	 gray	 magenta	 purple	 white
 brown	 green	 olive	 red	 yellow
 cyan	 lightgray	 orange	 teal	

4.2 Colors via dvipsnames option

 Apricot	 Cyan	 Mahogany	 ProcessBlue	 SpringGreen
 Aquamarine	 Dandelion	 Maroon	 Purple	 Tan
 Bittersweet	 DarkOrchid	 Melon	 RawSienna	 TealBlue
 Black	 Emerald	 MidnightBlue	 Red	 Thisle
 Blue	 ForestGreen	 Mulberry	 RedOrange	 Turquoise
 BlueGreen	 Fuchsia	 NavyBlue	 RedViolet	 Violet
 BlueViolet	 Goldenrod	 OliveGreen	 Rhodamine	 VioletRed
 BrickRed	 Gray	 Orange	 RoyalBlue	 White
 Brown	 Green	 OrangeRed	 RoyalPurple	 WildStrawberry
 BurntOrange	 GreenYellow	 Orchid	 RubineRed	 Yellow
 CadetBlue	 JungleGreen	 Peach	 Salmon	 YellowGreen
 CarnationPink	 Lavender	 Periwinkle	 SeaGreen	 YellowOrange
 Cerulean	 LimeGreen	 PineGreen	 Sepia	
 CornflowerBlue	 Magenta	 Plum	 SkyBlue	

Hilfreiche Pakete im Überblick - xcolor

```
\documentclass[a4paper,12pt]{scrartcl}
\usepackage[dvipsnames,table]{xcolor}
\definecolor{my-col}{RGB}{255, 150, 0}
\definecolor{my-html}{HTML}{00af87}
\begin{document}
  Dies hier ist meine {\color{my-col} Farbe}.\
  Durch die Klammern ist \color{my-html} es eingegrenzt.\
  Ohne {\} Klammern wird der \color{RubineRed} Rest vom Text farbig.
\end{document}
```

Output:

Dies hier ist meine **Farbe**.

Durch die Klammern ist **es eingegrenzt**.

Ohne {} Klammern wird der **Rest vom Text farbig**.

Hilfreiche Pakete im Überblick - microtype

- Das Paket `microtype` steuert die Abstände der Buchstaben (s.g. kerning) und verbessert damit massiv den Satzspiegel. Eine Animation gibts [hier](#).

Output OHNE microtype:

Aluminium and its alloys possess many attractive characteristics for engineering purposes including light weight, paramagnetic nature, high resistance to corrosion, reasonably high mechanical strength with good ductility and easy fabrication. From all those properties the key characteristic of aluminium is its low density, which is about one-third of steels and copper alloys.

Output MIT microtype:

Aluminium and its alloys possess many attractive characteristics for engineering purposes including light weight, paramagnetic nature, high resistance to corrosion, reasonably high mechanical strength with good ductility and easy fabrication. From all those properties the key characteristic of aluminium is its low density, which is about one-third of steels and copper alloys.

Hilfreiche Pakete im Überblick - siunitx

- Physikalische Größen mit Exponenten und Einheiten lassen sich auch standardmäßig im reinen Mathemodus eingeben, jedoch ist dies bei vielen irgendwann zu umständlich die Standard Kursivschrift zu umgehen.
- Das `siunitx` Paket übernimmt nicht nur einfache Eingabe von Einheiten, sondern auch sehr umfangreiche Verarbeitung von Zahlen, Rundungen, Kommasetzung, Exponenten, etc.
- Mit dieser Funktionalität wird später ein ganzer Datensatz in einer Tabelle automatisch formatiert, was eine qualitativ höchstwertige Darstellung für verhältnismäßig wenig Aufwand erzeugt.
- Die drei primären Befehle sind `\num{Zahl}` für reine Zahlen und `\qty{Zahl}{Einheiten}` (quantity) für Größen und `\unit{Einheiten}` nur für Einheiten.

Hilfreiche Pakete im Überblick - siunitx

```
\documentclass{scrartcl}
\usepackage{siunitx}
\sisetup{locale=DE, % Locale setzt die Formatierung für die Trennzeichen
per-mode=fraction}
\begin{document}
Die Erdbeschleunigung beträgt \qty{9.81}{\kg\per\s\squared}.\\
Die Stefan-Boltzmann Konstante ist  $\sigma = \frac{2\pi^5 k_B^4}{15c^2 h^3} =$ 
\qty{5.67e-8}{\watt\per\meter\squared\per\kelvin\tothe{4}}$\\
Gerundete Zahl auf 3 Nachkommastellen per Optionen:
\num[round-mode=places,round-precision=3]{3.27812}
\end{document}
```

Die Erdbeschleunigung beträgt $9,81 \frac{\text{kg}}{\text{s}^2}$.

Die Stefan-Boltzmann Konstante ist $\sigma = \frac{2\pi^5 k_B^4}{15c^2 h^3} = 5,67 \cdot 10^{-8} \frac{\text{W}}{\text{m}^2 \text{K}^4}$

Gerundete Zahl auf 3 Nachkommastellen per Optionen: 3,278

Hilfreiche Pakete im Überblick - siunitx

- Zahlreiche Einheiten besitzen eine Kurzschreibweise, damit nicht lange Wörter ausgeschrieben werden müssen. Hier sind einige Beispiele für die häufiger genutzten. Für eine vollständige Liste siehe [Dokumentation](#). Die meisten Präfixe wie centi, milli, nano, etc. gelten für die Kurzformen.

Einheit	Kurzschreibweise
<code>\meter</code>	<code>\m</code>
<code>\micro\meter</code>	<code>\um</code>
<code>\milli\second</code>	<code>\ms</code>
<code>\joule</code>	<code>\J</code>
<code>\volt</code>	<code>\V</code>
<code>\electronvolt</code>	<code>\eV</code>

Einheit	Kurzschreibweise
<code>\ampere</code>	<code>\A</code>
<code>\coulomb</code>	<code>\C</code>
<code>\watt</code>	<code>\W</code>
<code>\henry</code>	<code>\H</code>
<code>\farad</code>	<code>\F</code>
<code>\hertz</code>	<code>\Hz</code>

Hilfreiche Pakete im Überblick - siunitx

Einheit	Kurzschreibweise
<code>\kilo\watt\hour</code>	<code>\kWh</code>
<code>\newton</code>	<code>\N</code>
<code>\pascal</code>	<code>\Pa</code>
<code>\kilo\gram</code>	<code>\kg</code>
<code>\liter</code>	<code>\L</code>
<code>\kelvin</code>	<code>\K</code>

Hilfreiche Pakete im Überblick - csvsimple-13

- Tabellen sind für Anfänger die schwierigsten Elemente. Mit dem Paket `csvsimple-13` können ganze Datensätze aus CSV Dateien eingelesen werden. In der ersten Zeile steht idR der s.g. Header mit Beschriftungen.
- Ein Datensatz als CSV kann nicht nur für eine Tabelle eingelesen werden, sondern auch mittels dem Paket `pgfplots` visualisiert werden. So kann eine Datei gleichzeitig als Tabelle und Plot im Dokument auftauchen.
- Ein konkretes Beispiel wird im späteren Abschnitt für Tabellen gezeigt, da zunächst Verständnis wie eine Tabelle konstruiert wird nötig ist.
- Der wesentliche Befehl von diesem Paket ist
`\csvreader[Option]{Datei.csv}{Spaltenzuweisung}{Spaltreihenfolge}`

Hilfreiche Pakete im Überblick - tikz

- TikZ ist eine Abkürzung für „TikZ ist kein Zeichenprogramm“, es ist extrem umfangreich um Vektorgrafiken mit Befehlen zu erzeugen. Es ist Teil von PGF (portable graphics format). Es nutzt die `tikzpicture` Umgebung.
- Der große Vorteil ist die Beschriftung eurer Zeichnungen sowie mittels weniger Optionen das Erscheinungsbild nach eigenen Bedürfnissen exakt anzupassen zu können.
- Die grundlegende Idee ist die Verwendung von **Koordinaten**, die durch **Pfade** verbunden sind und mit **nodes** beschriftet werden.
- Damit eine Intuition für die Syntax entsteht werden schrittweise die Konzepte in einem eigenen Abschnitt (**Folie 95**) mit vielen Beispielen zu den zahlreichen Optionen gegeben, die die meisten Fälle abdecken dürften, ohne die Dokumentation (mit über 1300 Seiten) zu Rate ziehen zu müssen.

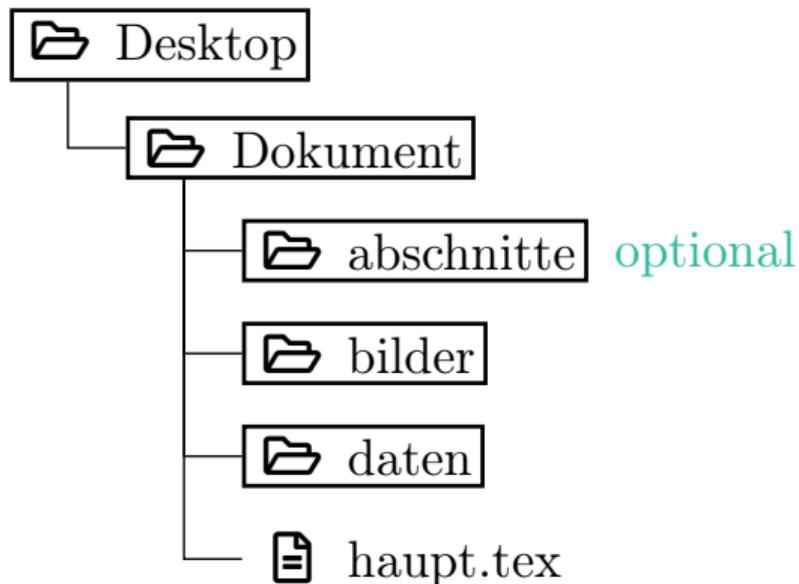
Hilfreiche Pakete im Überblick - pgfplots

- Das Paket `pgfplots` basiert auf `tikz` um vereinfacht Achsen, Funktionen und Datensätze zu plotten innerhalb einer `axis` Umgebung.
- Maßgeblich für euren Plot sind die Optionen an den Achsen für das Erscheinungsbild sowie den Optionen für die `\addplot{}`; Befehle.
- Es erlaubt das direkte Einlesen von Datensätzen (bspw. CSV Dateien), um diese zu plotten.
- Wenn `tikz` oder `pgfplots` verwendet wird, berücksichtigt es eure Einstellungen für Schriftart und Schriftgröße, so dass die erzeugten Abbildungen einheitlich zum gesamten Rest vom Dokument passen.
- Ähnlich zu `tikz` werden Beispiele und Optionen in einem eigenen Abschnitt dargestellt.

- In den nächsten Abschnitten wird kurz und knapp mit Beispielen gezeigt wie die Inhalte eingefügt werden können.
- Es wird dabei nicht erklärt welcher Befehl von welchem Paket explizit stammt, dazu schaut in den Überblick der Pakete.
- Für Dateien (für Bildern, ggf. Tabellen und direkte Einbindung) gilt es nicht nur den Namen, sondern ggf. auch den Dateipfad anzugeben, wenn sie sich nicht im gleichen Verzeichnis wie die `Haupt.tex` Datei befinden.
- Es ist höchst empfehlenswert für Bilder, Dateien für Tabellen & Plots und Abschnitte einen eigenen Ordner anzulegen, um alles übersichtlicher zu halten.

Inhalte einfügen

- Hier ein Beispiel für eine Ordnerstruktur.

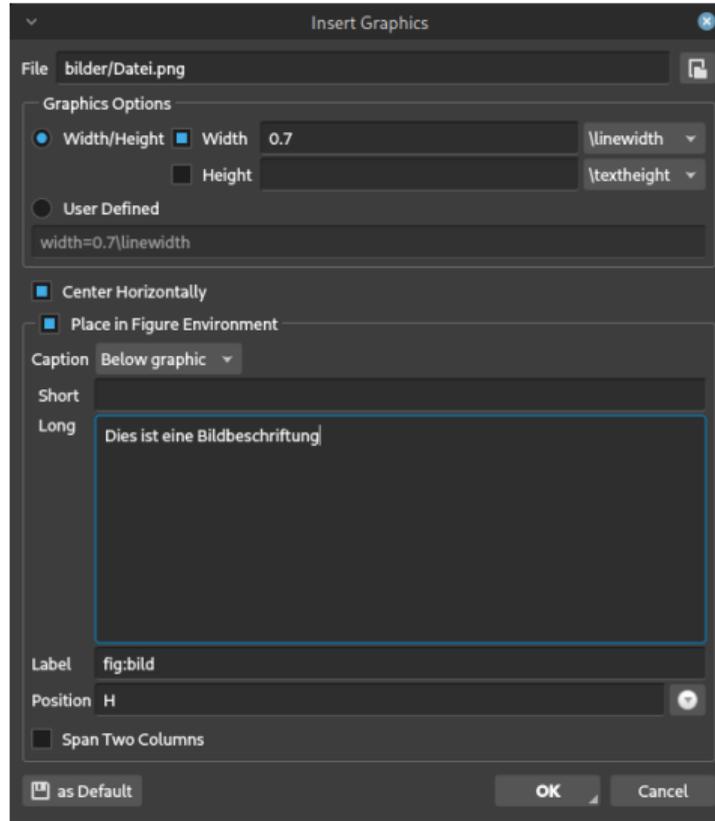
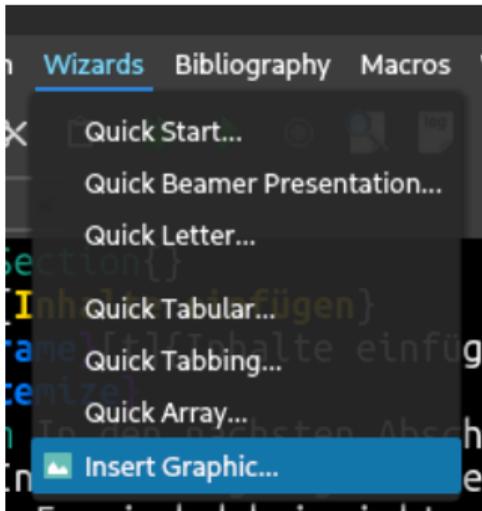


Einfügen - Bilder

- TeXstudio besitzt Wizards, um einfach Bilder und Tabellen zu erstellen. Stellt nur sicher entsprechende Pakete `graphicx`, `float`, `subcaption` zu verwenden.
- Für mehrere Bilder neben- oder untereinander schaut bitte in die Übersicht von `subcaption`. Für Bilder inmitten von Text siehe `wrapfig`.
- Folgender Code wurde mit dem Wizard, der ganz oben in der Leiste von TeXstudio zu finden ist (siehe nächste Folie), generiert:

```
\begin{figure}[H]
  \centering
  \includegraphics[width=0.7\linewidth]{bilder/Datei}
  \caption{Dies ist eine Bildbeschriftung}
  \label{fig:bild}
\end{figure}
```

Einfügen - Bilder



Einfügen - Tabellen

- Um eine Tabelle zu erstellen wird für die Trennung vom Fließtext mit einem float äquivalent zur `figure` Umgebung eine `table` benötigt.
- Die eigentliche Tabelle wird innerhalb einer `tabular` Umgebung erzeugt. Diese braucht als Argument die Anzahl der Spalten und dessen Orientierung links, zentriert, oder rechts `{l,c,r}`. Optional lassen sich hier mit dem `|` Symbol vertikale Linien setzen.
- Auch hierfür gibt es einen Wizard (Quick Tabular...), der euch das Leben einfacher macht zur Erstellung. Setzt dies innerhalb der `table` Umgebung.
- Jede Reihe muss mit `\\` terminiert werden. Spalten trennt man mit dem `&` Symbol. Optional kann mit `\\hline` eine horizontale Linie gesetzt werden.

Einfügen - Tabellen

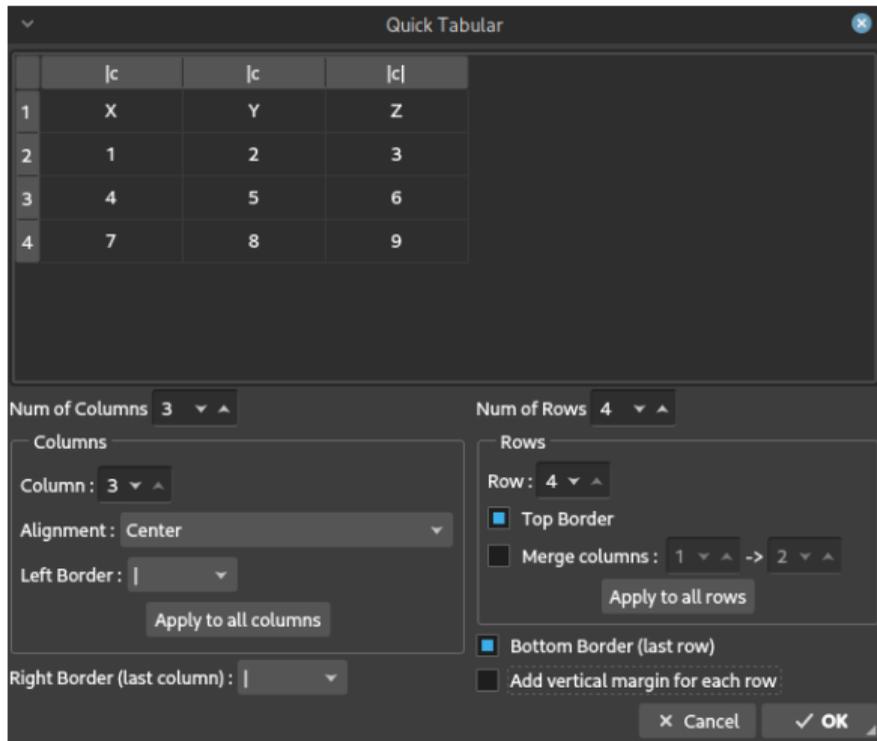


Tabelle 2: Tabellenüberschrift.

X	Y	Z
1	2	3
4	5	6
7	8	9

Einfügen - Tabellen

```
\begin{table}[H] % <- nicht vom Wizard
  \centering % <- nicht vom Wizard
  \caption{Tabellenüberschrift.} \label{tab:inhalt} % <- nicht vom Wizard
  \begin{tabular}{|c|c|c|}
    \hline
    X & Y & Z \\
    \hline
    1 & 2 & 3 \\
    \hline
    4 & 5 & 6 \\
    \hline
    7 & 8 & 9 \\
    \hline
  \end{tabular}
\end{table} % <- nicht vom Wizard
```

Einfügen - Tabellen - Zellen verbinden

- Was wenn man nun mehrere Zellen kombinieren möchte? Hierfür gibt es den `\multicolumn{Anzahl-Spalten}{Ausrichtung}{Inhalt}` Befehl, um Zellen horizontal zu kombinieren.
- Für das Äquivalent dazu, vertikal Zellen zu kombinieren, wird ein Paket namens `multirow` für den gleichnamigen Befehl `\multirow{Anzahl-Reihen}{Breite}{Inhalt}` verwendet. Die Breite wird automatisch gewählt mittels `*`.
- Verwendet man eine `\multirow`, so würde die `\hline` die kombinierte Zelle durchstreichen. Stattdessen verwendet man `\cline{i-j}`, um von der i-ten bis zur j-ten Spalte die Linie zu setzen.

Einfügen - Tabellen - Zellen verbinden

```
\begin{tabular}{|c|c|c|}  
  \hline  
  \multicolumn{2}{|c|}{XY} & Z \\  
  \hline  
  1 & 2 & 3 \\  
  \hline  
  4 & 5 & 6 \\  
  \hline  
\end{tabular}
```

XY		Z
1	2	3
4	5	6

```
\begin{tabular}{|c|c|c|}  
  \hline  
  X & Y & Z \\  
  \hline  
  \multirow{2}{*}{14} & 2 & 3 \\  
  \cline{2-3}  
  & 5 & 6 \\  
  \hline  
\end{tabular}
```

X	Y	Z
14	2	3
	5	6

Einfügen - Tabellen - Zellen färben, Reihe vergrößern

- Die Breite der Spalten von Tabellen wird abhängig vom Inhalt automatisch angepasst.
- Die Höhe der Reihen ist von dem Abstand `\arraystretch` gesteuert und hat den Standardwert 1. Um diesen Schalter zu ändern und eure Reihen vertikal auszubreiten nutzt bspw. `\renewcommand{\arraystretch}{1.5}`
- Die Zahl kann beliebige Werte annehmen
- Im nächsten Beispiel werden Farben für die Tabelle verwendet mit zwei Befehlen, der erste färbt die allererste Reihe mit der Betitelung der Spalten `\rowcolor{}` und `\rowcolors{Ab-Reihe-X}{Farbe-1}{Farbe-2}`
- Eine einzige Zelle kann mit `\cellcolor{}` gefärbt werden.

Einfügen - Tabellen - Zellen färben

```
\documentclass[a4paper,12pt]{scrartcl}
\usepackage[dvipsnames,table]{xcolor}
\begin{document}
  \begin{table}[h] \centering
    \caption{Farbige Tabelle.} \label{tab:farbig}
    \rowcolors{2}{blue!20}{red!20}
    \begin{tabular}{l|c|r}
      \rowcolor{Goldenrod!20} X & Y & Z \\ \hline
      1 & 2 & 3 \\ \hline
      4 & \cellcolor{violet}5 & 6 \\ \hline
      7 & 8 & 9 \\ \hline
      100 & 110 & 120 \\ \hline
    \end{tabular}
  \end{table}
\end{document}
```

Tabelle 1: Farbige Tabelle.

X	Y	Z
1	2	3
4	5	6
7	8	9
100	110	120

Einfügen - Tabellen - Nebeneinander platzieren

- Um Tabellen nebeneinander zu platzieren gibt es vom `subcaption` Paket äquivalent zur `subfigure` eine `subtable` Umgebung.

```
\begin{table}[H] \centering \caption{Zwei Tabellen}
  \begin{subtable}{0.5\linewidth} \centering \caption{Tabelle 1}
    \begin{tabular}{|c|c|c|}
      \hline A & B & C \\ \hline
    \end{tabular}
  \end{subtable}% <- Prozent = nebeneinander, ohne = untereinander
  \begin{subtable}{0.5\linewidth} \centering \caption{Tabelle 2}
    \begin{tabular}{|c|c|c|}
      \hline D & E & F \\ \hline
    \end{tabular}
  \end{subtable}
\end{table}
```

Einfügen - Tabellen - Nebeneinander platzieren

Tabelle 3: Zwei Tabellen

(a) Tabelle 1

A	B	C
---	---	---

(b) Tabelle 2

D	E	F
---	---	---

- Wichtig ist wie bei der **subfigure** Umgebung nebeneinander zu setzende Umgebungen durch ein Prozentzeichen % zu trennen. Ein normaler Zeilenumbruch bewirkt auch hier analog eine Platzierung untereinander.
- Je nach Aufteilung der `\linewidth` Anteile und Anzahl der **subtable** Umgebungen (mit % getrennt) können beliebig viele Tabellen nebeneinander platziert werden.

Einfügen - Tabellen - Einlesen von Datensätzen

- Für sehr kleine Tabellen kann man diese wie oben beschrieben „händisch“ einfügen, jedoch wird es für viele Zahlen und Spalten zu umständlich.
- Man betrachte folgenden Inhalt der Datei `datensatz.csv` im Ordner `daten` (siehe Ordnerstruktur auf Folie 77). Diese CSV wird mit der Syntax des `\csvreader` Befehls vom Paket `csvsimple-l3` eingelesen.

```
x,y,z  
0,0,1  
0.4,0.04,0.9210609940028851  
0.8,0.08,0.6967067093471654  
1.2,0.12,0.3623577544766734  
1.6,0.16,-0.02919952230128882  
2,0.2,-0.4161468365471424
```

Einfügen - Tabellen - Einlesen von Datensätzen

```
\begin{table} \centering
\begin{tabular}{|c|c|c|}
\hline  $x$  &  $y$  &  $z$  \\ \hline
\csvreader[late after line = \\ \hline]{daten/datensatz.csv}{1=\a,2=\b,3=\c}
{\a & \b & \c}
\end{tabular}
\end{table}
```

- Die `late after line` Option gibt an, was am Ende einer Reihe passieren soll (hier wird die Reihe terminiert und eine horizontale Linie unterhalb gezogen). Besitzt eure Tabelle **keinen Header**, wird zusätzlich die Option `no head` benötigt, sonst wird die 1. Zeile nicht angezeigt.
- Im letzten Argument werden die zugewiesenen Spalten in die gewünschte Reihenfolge gebracht.

Einfügen - Tabellen - Einlesen von Datensätzen

- Man erkennt kleine Problematiken ohne Zahlenformatierung einzulesen.
- Das `siunitx` Paket bietet einem flexible Optionen zur Verarbeitung von Zahlen, hier sogar ganzen Spalten. Durch den Schalter `\sisetup{}` kann innerhalb der Tabelle für alle Spalten die gleiche Formatierung vorgenommen werden, oder man setzt sie pro Spalte einzeln nach Bedarf als Optionen.

x	y	z
0	0	1
0.4	0.04	0.9210609940028851
0.8	0.08	0.6967067093471654
1.2	0.12	0.3623577544766734
1.6	0.16	-0.02919952230128882
2	0.2	-0.4161468365471424

Einfügen - Tabellen - Einlesen von Datensätzen

```
\begin{table} \centering
\sisetup{round-mode=places,round-precision=2,table-format=1.2}
\begin{tabular}{|c|c|c|}
  \hline $x$ & $y$ & $z$ \\ \hline
  \csvreader[late after line = \\ \hline]{daten/datensatz.csv}{1=\a,2=\b,3=\c}
  {\tablenum{\a} &
\tablenum{\b} &
\tablenum[table-format=-1.2]{\c}}
\end{tabular}
\end{table}
```

- Die Option `table-format` gibt an ob ein Vorzeichen, Stellen vor und nach dem Komma für die Zentrierung an Platz berücksichtigt werden. Bei (x -stelligen) Exponenten wäre es bspw. `table-format=1.2e x` .

Einfügen - Tabellen - Einlesen von Datensätzen

- Man achte auch hier auf Vorzeichen, diese ebenfalls im `table-format` anzugeben falls der Exponent negativ sein sollte für korrekte Ausrichtung.
- Durch die Optionen wurden alle Zahlen der Spalten auf 2 Nachkommastellen gerundet. Diese (und viele weitere) Optionen können für jeden `\tablename{}` Befehl individuell gesetzt werden für maximale Flexibilität pro Spalte.

x	y	z
0.00	0.00	1.00
0.40	0.04	0.92
0.80	0.08	0.70
1.20	0.12	0.36
1.60	0.16	-0.03
2.00	0.20	-0.42

Einfügen - Dateien

- Um sehr große Schriftstücke wie eine Thesis oder ein Buch übersichtlich zu schreiben, können Abschnitte und Kapitel in eigene Dateien gesetzt werden.
- Dadurch ist auskommentieren in der Hauptdatei äußerst einfach und man kann sich das Kompilieren von bereits fertiggestellten Abschnitten sparen.
- Wesentlich wird der `\input{}` Befehl benutzt, der auch einen relativen Dateipfad berücksichtigen kann. Der in der Datei enthaltene Code wird eingelesen, als stünde er so direkt in der Hauptdatei.
- In den besagten Abschnitten darf dann entsprechend nicht die Präambel oder die `document` Umgebung vorhanden sein.

Einfügen - Dateien

- Inhalt der Datei `Einleitung.tex`:

Hier steht eine Formel `$a+b=c$.`

- Eingebunden in `Haupt.tex`, aus dem Unterordner `Abschnitte`
(siehe Ordnerstruktur auf Folie 77)

```
\documentclass[a4paper,12pt]{scrartcl}
\begin{document}
  \input{Abschnitte/Einleitung}
  Da oben steht der ganze Abschnitt.
\end{document}
```

Hier steht eine Formel $a + b = c$.
Da oben steht der ganze Abschnitt.

Einfügen - Dateien

- So kann im Hauptdokument nur der Rahmen (Pakete, Stile, Optionen, Konfigurationen) stehen und in den Abschnitten jeweils nur noch der Inhalt.

```
\documentclass[a4paper,12pt]{scrartcl}
Pakete, Optionen, Stile...
\begin{document}
  \input{Abschnitte/Einleitung}
  \input{Abschnitte/Theorie}
  \input{Abschnitte/Methoden}
  \input{Abschnitte/Ergebnisse und Auswertung}
  \input{Abschnitte/Konklusion}
  \printbibliography
  \input{Abschnitte/Anhang}
\end{document}
```

- Die grundlegenden Bausteine für die `tikzpicture` Umgebungen sind **Pfade**, **Koordinaten** und **Nodes**.
- Koordinaten setzen Punkte im Raum fest. Sie können benannt werden.
- Pfade verbinden Koordinaten.
- Nodes beinhalten Beschriftungen, die sich entlang von Pfaden oder Koordinaten ausrichten lassen.
- Der immense Vorteil ist neben optischer Konsistenz zum restlichen Dokument, dass Nodes sich relativ entlang von Pfaden setzen lassen.
- Es empfiehlt sich sehr eure Grafiken in einem eigenen Dokument mit der `standalone` Dokumentenklasse anzufertigen, um Kompilierzeit zu sparen und einfacher Fehler zu finden.

Vektorgrafiken mit TikZ

- Optionen wie Farben, Stile von Strichen, Dicke, Markierungen, Pfeilspitzen, Verschiebungen etc. sind extrem umfangreich, bieten einem jedoch enorme Flexibilität. Daher zeige ich euch ausreichend kleine Beispiele, um viele Fälle grob abzudecken und ein Verständnis für die Syntax zu erhalten bzw. Herangehensweisen.
- Für komplexe Abbildungen ist es ratsam sich ggf. auf Papier eine Skizze zu setzen, als Denkstütze für den Code. Die größte Herausforderung im Umgang mit TikZ ist man muss sich anfangs mehr räumlich vorstellen bevor Ergebnisse sichtbar werden.

Pakete und Ladereihenfolge

- Damit TikZ auch erweiterte (`dvipsnames`) und selbst definierte Farben anhand der Farbnamen verwenden kann muss das Paket `xcolor` vorher geladen werden. Für mathematische Symbole (bei Beschriftungen) sind u.U. `amsmath`, `amssymb` benötigt.
- Alle Beispiele werden mit der `standalone` Klasse und neben o.g. mit dem `tikz` Paket erstellt. Auf folgender Folie ist der Codeblock für die Beispiele zur Referenz angegeben. Dieser Codeblock wird in den Beispielen aus Platzgründen nicht weiter auftauchen.
- Dort wo `-> Beispiel <-` steht müssen die gegebenen Beispiele einfach nur reinkopiert werden.

TikZ - Pakete für die Beispiele

```
\documentclass{standalone}
\usepackage{amsmath,amssymb,amsfonts}
\usepackage[dvipsnames]{xcolor}
\usepackage{tikz}
\begin{document}
  \begin{tikzpicture}
    -> Beispiel <-
  \end{tikzpicture}
\end{document}
```

- Sofern weitere Pakete verwendet oder zusätzliche Bibliotheken (libraries) von TikZ geladen werden, wird entsprechend darauf hingewiesen.

TikZ - Einstieg

- Für den Anfang verbindet man zwei Punkte mit absoluten Koordinaten mit einer einfachen Linie. Die Reihenfolge der Koordinaten ist stets x für horizontale, y für vertikale Abstände, z für Tiefe. Extrem wichtig hierbei ist, dass **alle tikz-Befehle mit einem Semikolon ; abschließen müssen!** Standardeinheit der Koordinaten ist **cm**.

```
\draw (0,0) -- (2,0.5);  
\draw (2.5,0) -- (2.5,0.3) (3,0.5) -- (3.5,0);
```



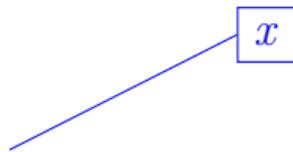
- Ohne Linie mit `--` zwischen Punkten wird zum nächsten „gesprungen“.
- Möchte man einen geschlossenen Pfad erzeugen, so dass der letzte Strich mit dem ersten verbunden wird (statt zu überlappen), wird `cycle` statt der Koordinate verwendet.

```
\draw[red] (0,0) -- (2,1)  
-- (3,0) -- cycle;
```



- Der Code muss nicht auf einer Zeile sein. Wichtig ist nur keine leeren Zeilen inmitten von Pfaden, Befehlen oder Optionen zu hinterlassen.
- Mit der Option `red` ließ sich zusätzlich der Stil des Pfades zu roter Farbe ändern. Platziert man Nodes entlang des Pfades nehmen diese ohne eigene Optionen die Farbe des Pfades an.

```
\draw[blue] (0,0) -- (2,1) node[right,draw]{$x$};
```



- Nodes werden mittig an Koordinaten platziert, mit Optionen lassen sich diese am Rand ausrichten. Die `draw` Option macht den Rand der Node sichtbar.

TikZ - Koordinaten

- Möchte man Koordinaten mehrfach verwenden (bspw. um einen Würfel zu zeichnen), lassen sich diese innerhalb eines Pfad-Befehls benennen und auch die Abstände skalieren. Punkt B liegt hier also bei 3 cm Entfernung zu A.
- Mit Labels lassen sich die definierten Punkte zur Übersicht markieren als Stütze bei der Visualisierung.

```
\path[scale=1.5]
(0,0) coordinate [label=above:A] (A)
(2,0) coordinate [label=above:B] (B)
;
\draw (A) -- (B);
```



A horizontal line segment with point A at the left end and point B at the right end.

TikZ - Relative Koordinaten

- Man kann von einer absoluten Koordinate aus sich relativ zur letzten die nächste setzen, dies wird mit zwei Plus Zeichen umgesetzt.

```
\draw[<->] (1,1) -- ++ (0,-1) -- ++ (2,0);
```

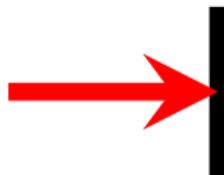


- Hier wird von der Anfangskoordinate (1,1) aus um 1 cm in negativer y-Richtung gegangen und danach 2 cm nach rechts in x-Richtung.
- Die Option der größer und kleiner Zeichen erlaubt es Pfeilspitzen am Anfang [$<-$] und Ende [$->$] oder beidseitig [$<->$] zu setzen.

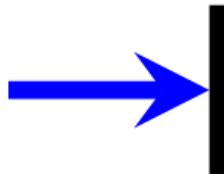
TikZ - Pfade

- Die Pfade, die durch `draw` Befehle Koordinaten verbinden, enden standardmäßig immer **zentral** auf den Koordinaten. Zeichnet man einen Pfeil orthogonal zu einer Linie trifft diese mittig auf den Pfad, der eine Breite hat.
- Damit der Pfad auf der **Kante** der Linie sitzt, kann der Pfad am Ende (oder Anfang) mit der Option `shorten >= 0.2pt` bzw. für den Anfang mit `shorten <= 0.2pt` gekürzt werden. Die Standardlinienbreite ist `0.4pt`.

```
\draw (0.5,0.5) -- (0.5,-0.5);  
\draw[->,>=stealth,red] (0,0) -- (0.5,0);
```



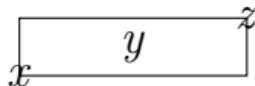
```
\draw (0.5,0.5) -- (0.5,-0.5);  
\draw[->,>=stealth,blue,shorten >= 0.2pt]  
(0,0) -- (0.5,0);
```



TikZ - Pfade - Einfache Formen

- Um Rechtecke, Kreise und Ellipsen zu zeichnen benötigt man keine komplexen Koordinatenangaben.
- Für ein Rechteck wird nur die untere linke und obere rechte Ecke benötigt.
- Ein Kreis benötigt nur Mittelpunkt und Radius.
- Eine Ellipse benötigt einen Mittelpunkt und zwei Radien.
- Nodes sind bei den Formen zentral positioniert. Beim Rechteck geht es auch zu den Koordinaten der angegebenen Ecken.

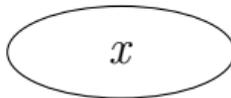
```
\draw (0,0) node{$x$} rectangle node{$y$} (2,0.5) node{$z$};
```



```
\draw (0,0) circle (0.5) node{$x$};
```



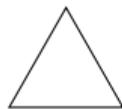
```
\draw (0,0) ellipse (1 and 0.5) node{$x$};
```



TikZ - Pfade - Winkel und Kreisbögen

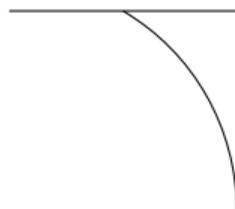
- Mit (relativen) Koordinaten lassen sich leicht trigonometrische Formen durch Winkel und Längen umsetzen. In diesem Fall wird ein gleichseitiges Dreieck gezeichnet durch die relative Koordinate zur Spitze oben, die vom Punkt rechts um 120 Grad gedreht eine Linie der Länge 1 cm zieht.

```
\draw (0,0) -- (1,0) -- ++ (120:1) -- cycle;
```



- Kreisbögen benötigen eine Koordinate, Startwinkel, Endwinkel und Radius. Sie beginnen stets an der Koordinate.

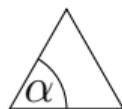
```
\draw (-1,0) -- (1,0);  
\draw (0,0) arc (60:0:2);
```



TikZ - Pfade - Winkel und Kreisbögen

- Möchte man nun bei bspw. dem gleichseitigen Dreieck den Kreisbogen als Winkel einzeichnen, muss der Startpunkt entlang der Seite verschoben werden. Hierzu sind relative Koordinaten ideal. Hier wird vom Ursprung aus der nächste Punkt *relativ* um 0.5 cm im 60 Grad Winkel gesetzt für den Kreisbogen.

```
\draw (0,0) -- (1,0) -- ++ (120:1) -- cycle;  
\draw (0,0) ++ (60:0.5) arc (60:0:0.5);  
\path (0,0) ++ (30:0.3) node{\alpha};
```

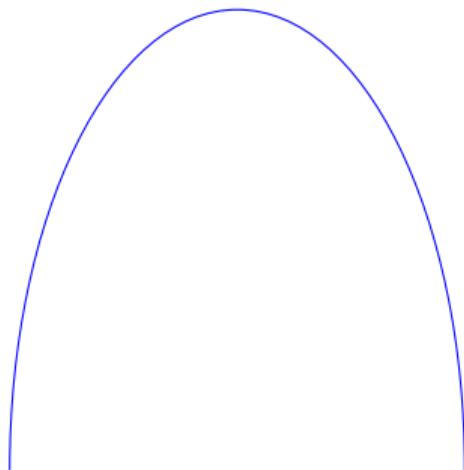


- Es lassen sich beliebig viele Koordinaten setzen, erst mit `--`, `to` bzw. `arc` werden sie auch mit Linien entlang der Pfade verbunden.
- `\path` Befehle werden nie gezeichnet, erleichtern Platzierungen von nodes.

TikZ - Elliptische Bögen

- Aus einer Kombination von der Syntax für `arc` und den Angaben zweier Radien einer Ellipse lassen sich elliptische Bögen zeichnen. Benötigt werden also Koordinate, Startwinkel, Endwinkel, horizontaler Radius und vertikaler Radius.

```
\draw[blue] (0,0) arc (180:0:2 and 4);
```



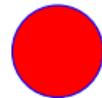
TikZ - Pfade - Flächen füllen

- Flächen lassen sich mit der gleichen Farbe der Linien durch die `fill` Option füllen. Man kann durch die `draw` Option separat Linien und Füllfarben angeben. Mit der Option `opacity` kann Undurchsichtigkeit für Transparenz gesetzt werden (für Linien UND Füllung) oder `fill opacity` nur für die Füllung. Auch Pfade (wie das Dreieck) lassen sich füllen, nicht nur Formen.

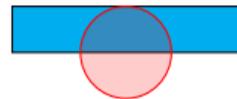
```
| \draw[fill] (0,0) circle (0.5);
```



```
| \draw[draw=blue,fill=red] (0,0) circle (0.5);
```



```
| \draw[fill=cyan] (-1,0) rectangle (1,0.5);  
| \draw[red,fill,fill opacity=0.2] (0,0) circle (0.5);
```

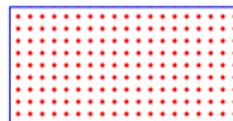


TikZ - Pfade - Flächen füllen mit Mustern

- Neben Farben können auch Muster gesetzt werden. Vordefinierte Muster sind durch die Bibliothek `patterns` verfügbar, um sie zu nutzen muss die Bibliothek in der Präambel geladen werden:

```
\usepackage{tikz}
\usetikzlibrary{patterns}
```

```
\draw[blue, pattern = dots, pattern color = red]
(-1,-0.5) rectangle (1,0.5);
\draw[red, pattern = north east lines] (2,0) circle (0.5);
```



- Die Muster übernehmen nicht die Farbvorgabe des Pfads. Farben werden mit der Option `pattern color = Farbname` gesetzt.
- Eine Liste von verfügbaren Mustern findet man in der Dokumentation [hier](#). 111/113

