

L^AT_EX

Crashkurs LaTeX Teil 2: Aufbaupaket MINT
Mentoring WiSe 2017/18

Anja Wolffgramm Jeannine Darakci
Freie Universität Berlin

10. November 2017

1 Mathematikmodus

- 1.1 inline vs. abgesetzter Modus
- 1.2 Nützliche Befehle
- 1.3 Spezielle Formatierungen
- 1.4 Zahlen & Einheiten mit siunitx

2 Textstrukturierung

- 2.1 Tabellen
- 2.2 Boxen

2.3 Spaltenlayout

2.4 Minipages

3 Farben

4 Quellcode einbinden

- 4.1 Verbatim
- 4.2 Lstlistings
- 4.3 Liste der Programmcodes

5 Abspann

1 Mathematikmodus

- 1.1 inline vs. abgesetzter Modus
- 1.2 Nützliche Befehle
- 1.3 Spezielle Formatierungen
- 1.4 Zahlen & Einheiten mit siunitx

2 Textstrukturierung

- 2.1 Tabellen
- 2.2 Boxen

2.3 Spaltenlayout

2.4 Minipages

3 Farben

4 Quellcode einbinden

4.1 Verbatim

4.2 Lstlistings

4.3 Liste der Programmcodes

5 Abspann

Ist eine Umgebung, die dafür optimiert ist, mathematische Formeln und Symbole darzustellen. Benötigte Pakete:

```

\usepackage{amsmath} % MUSS vor fontspec geladen werden
\usepackage{mathtools} % modifiziert amsmath
\usepackage{amssymb} % mathematische symbole, für \ceckmarks
\usepackage{amsthm} % für proof
\usepackage{mathrsfs} % für \mathscr
\usepackage{latexsym}
\usepackage{marvosym} % zusätzliche Zeichen, z.B. Lightning
\usepackage{cancel} % für das Durchstreichen u.a. in Matheformeln mit \cancel
    
```

inline Im Fließtext wird der betreffende Ausdruck mit Dollarzeichen $\$$ umgeben. z.B.:

„Mitten im Text steht die Formel $\$E = mc^2\$$ von Einstein.“

Und so sieht es aus:

„Mitten im Text steht die Formel $E = mc^2$ von Einstein.“

inline Im Fließtext wird der betreffende Ausdruck mit Dollarzeichen $\$$ umgeben. z.B.:

„Mitten im Text steht die Formel $\$E = mc^2\$$ von Einstein.“

Und so sieht es aus:

„Mitten im Text steht die Formel $E = mc^2$ von Einstein.“

abgesetzt Um einzelilige Formeln abzusetzen, baut man eine Umgebung auf, innerhalb derer der Mathematikmodus eingeschaltet ist:

$\[E=mc^2 \]$

Führt zu:

$$E = mc^2$$

Die **align**-Umgebung

- ▶ schaltet den Mathematikmodus ein,
- ▶ zentriert Formel,
- ▶ erlaubt Zeilenumbrüche ($\backslash\backslash$),
- ▶ nummeriert die Zeilen und
- ▶ ermöglicht Ausrichtung der Zeilen zueinander (mittels $\&$).

Die **align**-Umgebung

- ▶ schaltet den Mathematikmodus ein,
- ▶ zentriert Formel,
- ▶ erlaubt Zeilenumbrüche ($\backslash\backslash$),
- ▶ nummeriert die Zeilen und
- ▶ ermöglicht Ausrichtung der Zeilen zueinander (mittels $\&$).

Beispiel:

```

\begin{align}
(a+b)^2 &\&= (a+b)(a+b) \\
&\&= a^2 + 2ab + b^2
\end{align}
    
```


Die **align**-Umgebung

- ▶ schaltet den Mathematikmodus ein,
- ▶ zentriert Formel,
- ▶ erlaubt Zeilenumbrüche ($\\\$),
- ▶ nummeriert die Zeilen und
- ▶ ermöglicht Ausrichtung der Zeilen zueinander (mittels $\&$).

Beispiel:

```

\begin{align}
(a+b)^2 &= (a+b)(a+b) \\
&= a^2 + 2ab + b^2
\end{align}
    
```

Und so sieht es aus:

$$(a+b)^2 = (a+b)(a+b) \quad (1)$$

$$= a^2 + 2ab + b^2 \quad (2)$$

Die **align**-Umgebung

- ▶ schaltet den Mathematikmodus ein,
- ▶ zentriert Formel,
- ▶ erlaubt Zeilenumbrüche ($\backslash\backslash$),
- ▶ nummeriert die Zeilen und
- ▶ ermöglicht Ausrichtung der Zeilen zueinander (mittels $\&$).

Beispiel:

```

\begin{align}
(a+b)^2 &= (a+b)(a+b) \\
&= a^2 + 2ab + b^2
\end{align}
    
```

Und so sieht es aus:

$$(a+b)^2 = (a+b)(a+b) \quad (1)$$

$$= a^2 + 2ab + b^2 \quad (2)$$

Will man keine Nummerierung, nutzt man **align***

```
\subsubsection{\texorpdfstring{Something with $\beta$ in it.}{Something with beta in it.}}
```

Quelle: <https://tex.stackexchange.com/questions/77657/how-do-i-put-special-characters-in-a-section-title>

Exponent `a2 \cdot xn+1`

$$a^2 \cdot x^{n+1}$$

Exponent `a^2 \cdot x^{n+1}`

$$a^2 \cdot x^{n+1}$$

 Indicés `x_1 + \dots + x_{k+1}`

$$x_1 + \dots + x_{k+1}$$

Exponent `a2 \cdot x{n+1}`

$$a^2 \cdot x^{n+1}$$

Indicés `x1 + \dots + x{k+1}`

$$x_1 + \dots + x_{k+1}$$

Quantoren `\forall x \in K \exists y \in K: x \cdot y = x`

$$\forall x \in K \exists y \in K : x \cdot y = x$$

Exponent

 $a^2 \cdot x^{n+1}$
 $a^2 \cdot x^{n+1}$

Indicés

 $x_1 + \dots + x_{k+1}$
 $x_1 + \dots + x_{k+1}$

Quantoren

 $\forall x \in K \exists y \in K: x \cdot y = x$
 $\forall x \in K \exists y \in K : x \cdot y = x$

Logische Op.

 $\neg x \wedge (y \vee z) \implies x$
 $\neg x \wedge (y \vee z) \implies x$

Exponent `a^2 \cdot x^{n+1}`

$$a^2 \cdot x^{n+1}$$

 Indicés `x_1 + \dots + x_{k+1}`

$$x_1 + \dots + x_{k+1}$$

 Quantoren `\forall x \in K \exists y \in K: x \cdot y = x`

$$\forall x \in K \exists y \in K : x \cdot y = x$$

 Logische Op. `\neg x \wedge (y \vee z) \implies x`

$$\neg x \wedge (y \vee z) \implies x$$

 Ungleichungen `x \leq \sqrt[4]{x} + 42`

$$x \leq \sqrt[4]{x} + 42$$

Exponent $a^2 \cdot x^{n+1}$

$$a^2 \cdot x^{n+1}$$

 Indicés $x_1 + \dots + x_{k+1}$

$$x_1 + \dots + x_{k+1}$$

 Quantoren $\forall x \in K \exists y \in K : x \cdot y = x$

$$\forall x \in K \exists y \in K : x \cdot y = x$$

 Logische Op. $\neg x \wedge (y \vee z) \implies x$

$$\neg x \wedge (y \vee z) \implies x$$

 Ungleichungen $x \leq \sqrt[4]{x} + 42$

$$x \leq \sqrt[4]{x} + 42$$

 Funktion $f : \mathbb{R}^2 \rightarrow \mathbb{N}$

$$f : \mathbb{R}^2 \rightarrow \mathbb{N}$$

Exponent `a2 \cdot x{n+1}`

$$a^2 \cdot x^{n+1}$$

 Indicés `x1 + \dots + x{k+1}`

$$x_1 + \dots + x_{k+1}$$

 Quantoren `\forall x \in K \exists y \in K: x \cdot y = x`

$$\forall x \in K \exists y \in K : x \cdot y = x$$

 Logische Op. `\neg x \wedge (y \vee z) \implies x`

$$\neg x \wedge (y \vee z) \implies x$$

 Ungleichungen `x \leq \sqrt[4]{x} + 42`

$$x \leq \sqrt[4]{x} + 42$$

 Funktion `f : \mathbb{R}^2 \to \mathbb{N}`

$$f : \mathbb{R}^2 \rightarrow \mathbb{N}$$

 Sinus `\sin \alpha`

$$\sin \alpha$$

Bruch `\frac{x+1}{2}`

$$\frac{x+1}{2}$$

Bruch `\frac{x+1}{2}`

$$\frac{x+1}{2}$$

Summe `\sum_{i=0}^{n+1} q^i`

$$\sum_{i=0}^{n+1} q^i$$

Bruch `\frac{x+1}{2}`

$$\frac{x+1}{2}$$

 Summe `\sum_{i=0}^{n+1} q^i`

$$\sum_{i=0}^{n+1} q^i$$

 Produkt `\prod_{i=1}^n i = n!`

$$\prod_{i=1}^n i = n!$$

Bruch `\frac{x+1}{2}`

$$\frac{x+1}{2}$$

 Summe `\sum_{i=0}^{n+1} q^i`

$$\sum_{i=0}^{n+1} q^i$$

 Produkt `\prod_{i=1}^n i = n!`

$$\prod_{i=1}^n i = n!$$

 Integral `\int_{0}^{\infty} e^x dx`

$$\int_0^{\infty} e^x dx$$

Bruch `\frac{x+1}{2}`

$$\frac{x+1}{2}$$

 Summe `\sum_{i=0}^{n+1} q^i`

$$\sum_{i=0}^{n+1} q^i$$

 Produkt `\prod_{i=1}^n i = n!`

$$\prod_{i=1}^n i = n!$$

 Integral `\int\limits_0^{\infty} e^x dx`

$$\int_0^{\infty} e^x dx$$

 Vereinigung `\bigcup_{i \in I} A_i`

$$\bigcup_{i \in I} A_i$$

Bruch `\frac{x+1}{2}`

$$\frac{x+1}{2}$$

 Summe `\sum_{i=0}^{n+1} q^i`

$$\sum_{i=0}^{n+1} q^i$$

 Produkt `\prod_{i=1}^n i = n!`

$$\prod_{i=1}^n i = n!$$

 Integral `\int_{\limits_0}^{\infty} e^x dx`

$$\int_0^{\infty} e^x dx$$

 Vereinigung `\bigcup_{i \in I} A_i`

$$\bigcup_{i \in I} A_i$$

 Limes `\lim_{n \to \infty} \frac{1}{n} = 0`

$$\lim_{n \rightarrow \infty} \frac{1}{n} = 0$$

Man kann sich seine eigenen Operatoren erstellen. Dazu gibt es zwei Möglichkeiten:

```

\DeclareMathOperator*{\meineOp}{\r{a}} % wird in der Präambel definiert, ermöglicht
  \limits_{...}
\newcommand{\meineOp}{\operatorname{\r{a}}} % kann überall definiert werden
    
```

Man kann sich seine eigenen Operatoren erstellen. Dazu gibt es zwei Möglichkeiten:

```
\DeclareMathOperator*{\meineOp}{\r{a}} % wird in der Präambel definiert, ermöglicht
\limits_{...}
\newcommand{\meineOp}{\operatorname{\r{a}}} % kann überall definiert werden
```

Anwendung:

```
\meineOp(x) = 2^x + 1
\meineOp_{i=0} x_i
```

$$\mathring{a}(x) = 2^x + 1 \quad (3)$$

$$\mathring{a}_{i=0}^N x_i \quad (4)$$

Es gibt verschiedene Pfeile:

Pfeil `x \rightarrow y`

$x \rightarrow y$

Es gibt verschiedene Pfeile:

Pfeil `x \rightarrow y`

 $x \rightarrow y$

Doppelpfeil `x \Rightarrow y`

 $x \Rightarrow y$

Es gibt verschiedene Pfeile:

Pfeil `x \rightarrow y`

 $x \rightarrow y$

Doppelpfeil `x \Rightarrow y`

 $x \Rightarrow y$

langer Pfeil `x \Longrightarrow y`

 $x \Longrightarrow y$

Es gibt verschiedene Pfeile:

Pfeil `x \rightarrow y`

$$x \rightarrow y$$

Doppelpfeil `x \Rightarrow y`

$$x \Rightarrow y$$

langer Pfeil `x \Longrightarrow y`

$$x \Longrightarrow y$$

Abbild `x \mapsto y`

$$x \mapsto y$$

Es gibt verschiedene Pfeile:

Pfeil `x \rightarrow y`

 $x \rightarrow y$

Doppelpfeil `x \Rightarrow y`

 $x \Rightarrow y$

langer Pfeil `x \Longrightarrow y`

 $x \Longrightarrow y$

Abbild `x \mapsto y`

 $x \mapsto y$

overset `x \overset{(1)}{\Longrightarrow} y`

 $x \overset{(1)}{\Longrightarrow} y$

Es gibt verschiedene Pfeile:

Pfeil `x \rightarrow y`

$$x \rightarrow y$$

Doppelpfeil `x \Rightarrow y`

$$x \Rightarrow y$$

langer Pfeil `x \Longrightarrow y`

$$x \Longrightarrow y$$

Abbild `x \mapsto y`

$$x \mapsto y$$

overset `x \overset{(1)}{\Longrightarrow} y`

$$x \overset{(1)}{\Longrightarrow} y$$

underset `x \underset{(1)}{\Longrightarrow} y`

$$x \underset{(1)}{\Longrightarrow} y$$

Möchte man Text in Mathematikmodus schreiben, geht dies wie folgt:

```
\[ n = 2k \implies n \text{ gerade} \]
```

So sieht es aus:

$$n = 2k \implies n \text{ gerade}$$

Möchte man Text in Mathematikmodus schreiben, geht dies wie folgt:

```
\[ n = 2k \implies n \text{ gerade} \]
```

So sieht es aus:

$$n = 2k \implies n \text{ gerade}$$

Manche Funktionalitäten unterscheiden sich im Text und Mathematikmodus:

Befehl	Textmodus	Mathematikmodus
Unterstrich	<code>_</code>	<code>_</code>
Dach	<code>\^{}</code>	<code>^</code>
Tabelle	<code>\textsuperscript</code>	<code>\superscript</code>
Table	<code>tabular</code>	<code>array</code>
Bold	<code>\textbf{}</code>	<code>\mathbf{}</code>
Backslash	<code>\textbackslash</code>	<code>\backslash</code>

$$\left(\int_1^9 \frac{x}{2} dx \right)^2 \rightarrow \text{schlecht}$$

$$\left(\int_1^9 \frac{x}{2} dx \right)^2 \rightarrow \text{gut}$$

$$\left(\int_1^9 \frac{x}{2} dx\right)^2 \rightarrow \text{schlecht}$$

$$\left(\int_1^9 \frac{x}{2} dx\right)^2 \rightarrow \text{gut}$$

Darum werden Klammern wie folgt geschrieben:

▶ `\left(... \right)` (...)

▶ `\lceil ... \rceil` [...]

▶ `\left[... \right]` [...]

▶ `\langle ... \rangle` {...}

▶ `\lfloor ... \rfloor` [...]

▶ `\left\{ ... \right\}` {...}

► Binome

```
\binom{n}{k}
```

$$\binom{n}{k}$$

► Binome

```
\binom{n}{k}
```

$$\binom{n}{k}$$

► Matrizen

```
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
\begin{theorem}
...
\end{theorem}
```

Satz

This is a theorem about right triangles and can be summarised in the next equation

$$x^2 + y^2 = z^2$$

```
\begin{corollary}
...
\end{corollary}
```

Folgerung

There's no right rectangle whose sides measure 3cm, 4cm, and 6cm.

```
\begin{lemma}
...
\end{lemma}
```

Lemma

Given two line segments whose lengths are a and b respectively there is a real number r such that $b = r \cdot a$.

```
\begin{proof}
...
\end{proof}
```

Beweis.

Mit $r = \frac{b}{a}$ gilt stets $b = r \cdot a$. □

```
\begin{definition}
...
\end{definition}
```

Definition

$$\text{fak}(n) := \begin{cases} 1, & n = 0 \\ n \cdot \text{fak}(n-1), & \text{sonst} \end{cases} \quad \forall n \in \mathbb{N}$$

Wie kann man Einheiten darstellen?

$$v = 3.14km/h \longrightarrow \text{schlecht}$$

$$v = 3,14 \frac{\text{km}}{\text{h}} \longrightarrow \text{gut}$$

Wie kann man Einheiten darstellen?

$$v = 3.14km/h \longrightarrow \text{schlecht}$$

$$v = 3,14 \frac{\text{km}}{\text{h}} \longrightarrow \text{gut}$$

Das Paket **siunitx**[21][16] bietet eine angenehme Bedienung von Einheiten:

```
\usepackage{siunitx}
```

Wie kann man Einheiten darstellen?

$$v = 3.14km/h \longrightarrow \text{schlecht}$$

$$v = 3,14 \frac{\text{km}}{\text{h}} \longrightarrow \text{gut}$$

Das Paket **siunitx**[21][16] bietet eine angenehme Bedienung von Einheiten:

```
\usepackage{siunitx}
```

Wir tätigen einige Einstellungen:

```
\sisetup{
  output-decimal-marker = {,}, % Dezimalzeichen
  per-mode = fraction, % km/s als Bruch
  list-final-separator = { und }, % Trennzeichen in Listen
  list-pair-separator = { und },
  range-phrase = { bis },
}
```

Zahlen können wie folgt dargestellt werden:

<code>\num{.1234}</code>	0,1234
<code>\num{3,141529}</code>	3,141 529
<code>\num{3.45d-4}</code>	$3,45 \times 10^{-4}$
<code>\num{-e10}</code>	-10^{10}

Zahlen können wie folgt dargestellt werden:

<code>\num{.1234}</code>	0,1234
<code>\num{3,141529}</code>	3,141 529
<code>\num{3.45d-4}</code>	$3,45 \times 10^{-4}$
<code>\num{-e10}</code>	-10^{10}

Es können auch Zahlenlisten oder Bereiche angegeben werden:

<code>\numlist{10;30;50;70}</code>	10, 30, 50 und 70
<code>\numrange{10}{30}</code>	10 bis 30

Zahlen mit Einheiten lassen sich wie folgt darstellen:

```
\SI{ZAHL}{EINHEIT}
```

Es gibt u.a. folgende vordefinierte Einheiten und Präfixe:

<code>\ampere</code>	A
<code>\degreeCelsius</code>	°C
<code>\hertz</code>	Hz
<code>\hour</code>	h
<code>\kilogram</code>	kg
<code>\metre</code>	m
<code>\newton</code>	N

<code>\ohm</code>	Ω
<code>\second</code>	s
<code>\micro</code>	μ
<code>\milli</code>	m
<code>\kilo</code>	k
<code>\mega</code>	M
<code>\giga</code>	G

Zahlen mit Einheiten lassen sich wie folgt darstellen:

```
\SI{ZAHL}{EINHEIT}
```

Es gibt u.a. folgende vordefinierte Einheiten und Präfixe:

<code>\ampere</code>	A	<code>\ohm</code>	Ω
<code>\degreeCelsius</code>	$^{\circ}\text{C}$	<code>\second</code>	s
<code>\hertz</code>	Hz	<code>\micro</code>	μ
<code>\hour</code>	h	<code>\milli</code>	m
<code>\kilogram</code>	kg	<code>\kilo</code>	k
<code>\metre</code>	m	<code>\mega</code>	M
<code>\newton</code>	N	<code>\giga</code>	G

Auch hier lassen sich Bereiche definieren:

```
\SIlist{10;30;45}{\micro\second} 10  $\mu\text{s}$ , 30  $\mu\text{s}$  und 45  $\mu\text{s}$   

\SIrange{10}{30}{\micro\second} 10  $\mu\text{s}$  bis 30  $\mu\text{s}$ 
```

Es lassen sich Einheiten auch selbst definieren:

```
\DeclareSIUnit{\NAME}{DEFINITION}
```

Beispiel:

```
\DeclareSIUnit{\kmh}{\kilo\meter\per\hour}
```


Es lassen sich Einheiten auch selbst definieren:

```
\DeclareSIUnit{\NAME}{DEFINITION}
```

Beispiel:

```
\DeclareSIUnit{\kmh}{\kilo\meter\per\hour}
```

Diese können dann wie folgt verwendet werden:

```
\[ \SI{1,079e+9}{\kmh} \]
```

So sieht es aus:

$$1,079 \times 10^9 \frac{\text{km}}{\text{h}}$$

- 1 Mathematikmodus
 - 1.1 inline vs. abgesetzter Modus
 - 1.2 Nützliche Befehle
 - 1.3 Spezielle Formatierungen
 - 1.4 Zahlen & Einheiten mit siunitx
- 2 Textstrukturierung
 - 2.1 Tabellen
 - 2.2 Boxen
 - 2.3 Spaltenlayout
 - 2.4 Minipages
- 3 Farben
- 4 Quellcode einbinden
 - 4.1 Verbatim
 - 4.2 Lstlistings
 - 4.3 Liste der Programmcodes
- 5 Abspann

Einfachste Tabelle mit:

```
\begin{tabular}{FORMAT}
  Tabelleninhalt
\end{tabular}
```

Einfachste Tabelle mit:

```
\begin{tabular}{FORMAT}
  Tabelleninhalt
\end{tabular}
```

In das Feld `{FORMAT}` wird für jede Spalte die Formatierung angegeben

- ▶ **c** – für zentrierten Text (einzeilig)
- ▶ **l** – für linksbündigen Text (einzeilig)
- ▶ **r** – für rechtsbündigen Text (einzeilig)
- ▶ **p{SIZEcm}** – Spalte soll SIZE cm breit sein und der Zelleninhalt wird in linksbündigen Flattersatz gesetzt und automatisch umgebrochen
- ▶ **|** – Pipe: zieht eine vertikale Trennlinie zwischen zwei Spalten

Beispiel:

```
\begin{tabular}{|l||c|r|}
...
\end{tabular}
```

Der Tabelleninhalt wird wie folgt gesetzt:

- ▶ `&` – trennt die Spalten voneinander
- ▶ `\\` – trennt die Zeilen voneinander, alternativ auch `\tabularnewline`
- ▶ `\hline` – zieht eine horizontale Trennlinie zwischen zwei Zeilen
- ▶ `\cline{a-b}` – zieht eine horizontale Trennlinie von Spalte a bis Spalte b

Der Tabelleninhalt wird wie folgt gesetzt:

- ▶ `&` – trennt die Spalten voneinander
- ▶ `\\` – trennt die Zeilen voneinander, alternativ auch `\tabularnewline`
- ▶ `\hline` – zieht eine horizontale Trennlinie zwischen zwei Zeilen
- ▶ `\cline{a-b}` – zieht eine horizontale Trennlinie von Spalte a bis Spalte b

Beispiel:

```

\begin{tabular}{| l || c | r |}
Spalte 1 & Spalte 2 & Spalte 3 \\
\hline % horizontale Trennlinie
1 & 2 & 3 \\
\end{tabular}
    
```

Und so sieht es aus:

Spalte 1	Spalte 2	Spalte 3
1	2	3

Der Tabelleninhalt wird wie folgt gesetzt:

- ▶ `&` – trennt die Spalten voneinander
- ▶ `\\` – trennt die Zeilen voneinander, alternativ auch `\tabularnewline`
- ▶ `\hline` – zieht eine horizontale Trennlinie zwischen zwei Zeilen
- ▶ `\cline{a-b}` – zieht eine horizontale Trennlinie von Spalte a bis Spalte b

Beispiel:

```

\begin{tabular}{| l || c | r |}
Spalte 1 & Spalte 2 & Spalte 3 \\
\hline % horizontale Trennlinie
1 & 2 & 3 \\
\end{tabular}
    
```

Und so sieht es aus:

Spalte 1	Spalte 2	Spalte 3
1	2	3

Dabei richtet sich die Tabellenbreite nach dem Inhalt.

Möchte man, dass die Tabelle eine feste Breite hat und Zelleninhalt automatisch umgebrochen wird, benutzt man **tabularx**

```

\usepackage{tabularx}

\begin{document}
  \begin{tabularx}{\linewidth}{| X | X | p{2cm} |}
    % Die Gesamtbreite wird zu gleichen Teilen auf alle Spalten, die mit \textbf{X}
    % deklariert wurden, verteilt.
    \hline
    ... & ... & ... \\
    \hline
  \end{tabularx}
\end{document}
    
```


Zeilen und Spalten zu einer verschmelzen:

```
\usepackage{multirow} % für multirow in tabulars
```

Zeilen und Spalten zu einer verschmelzen:

```
\usepackage{multirow} % für multirow in tabulars
```

Beispiel:

```
\begin{tabular}{|c|c|c|} \hline
\multicolumn{2}{|c|}{eine Spalte} & Hefe \\ \hline
Zeit & \multirow{2}{*}{eine Zeile} & Zucker \\ \cline{1-1}\cline{3-3} % vertikale
Trennlinie
Sieb & & Mehl \\ \hline
\end{tabular}
```

So sieht es aus:

	eine Spalte	Hefe
Zeit	eine Zeile	Zucker
Sieb		Mehl

Tabellen können auch geschachtelt werden:

```

\begin{tabular}{|l|} \hline
erste Zelle &
{ % Tabelle muss eingefasst werden
\begin{tabular}{|c|} \hline
Zeile 1 \\ \hline
Zeile 2 \\ \hline
\end{tabular}
} % Ende der Einfassung
\\ \hline
\end{tabular}
  
```

So sieht es aus:

erste Zelle	Zeile 1
	Zeile 2

Hinweis: Die innere Tabelle muss mit `{ }` eingefasst werden.

- ▶ ermöglicht eine Beschriftung der Tabelle
- ▶ wird im Tabellenverzeichnis automatisch aufgeführt
- ▶ Beispiel:

```

\begin{table}
  \centering
  \caption[meine Tabelle (LOT)]{meine Tabelle -- lange Beschriftung}
  \begin{tabular}{|l|l|} \hline
    Zelle 1 & Zelle 2 \\ \hline
  \end{tabular}
\end{table}
    
```

- ▶ So sieht es aus:

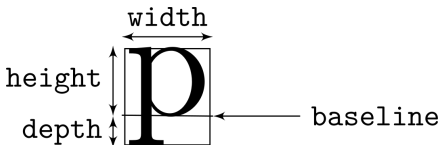
Tabelle: meine Tabelle – lange Beschriftung

Zelle 1	Zelle 2
---------	---------

- ▶ ein Tabellenverzeichnis wird mit `\listoftables` erstellt.

In $\text{T}_{\text{E}}\text{X}$ wird jedes Element – auch einzelne Lettern – in Boxen verpackt. Boxen haben dreidimensionale Eigenschaften: `[boxes]`

Abbildung: Box



Höhe: Länge zwischen der Grundlinie (`baseline`) und der Oberkante der Box

Tiefe: Länge zwischen der Grundlinie (`baseline`) und dem unteren Ende der Box

Breite: Länge, welche die Box auf Grund des Inhalts einnimmt

Text lässt sich mit einer **fbx** sehr einfach einrahmen.

Text lässt sich mit einer `\textbf{fbx}` sehr `\fbx{einfach einrahmen}`.

Text lässt sich mit einer **fbx** sehr einfach einrahmen.

Text lässt sich mit einer `\textbf{fbx}` sehr `\fbx{einfach einrahmen}`.

Hinweis: Allerdings werden werden auf diese Weise eingerahmte Abschnitte nicht umgebrochen.

Text lässt sich mit einer **fbox** sehr einfach einrahmen.

```
Text lässt sich mit einer \textbf{fbox} sehr \fbox{einfach einrahmen}.
```

Hinweis: Allerdings werden werden auf diese Weise eingerahmte Abschnitte nicht umgebrochen.

Die **parbox** wird eingesetzt, damit der Text nicht über das Layout hinaus ragt. Ihr kann die Breite als optionales Argument übergeben werden. Will man auch hier einen Rahmen oder farbig hinterlegen, setzt man die *parbox* in eine der oben genannten Boxen.

```
\fbox{% umgebende Box  
\parbox{\dimexpr\textwidth-2\fboxsep}{% Textbreite abzüglich 2*Rahmenbreite  
...  
}%  
}
```


Objekte können mittels **rotatebox** gegen den Uhrzeigersinn gedreht werden:

```
\rotatebox[OPTION]{DEGREE}{CONTENT}
```

Optionales Argument ist der Punkt, um den das Objekt gedreht wird. Beispiel:

```
Rotation \rotatebox[origin=c]{45}{um 45°}
% Drehung um den Mittelpunkt c
```

Rotation
um 45°

Inhalte können mittels **raisebox** vertikal verschoben werden:

```
\raisebox{LIFT}[HEIGHT][DEPTH]{CONTENT}
```

LIFT: Größe, um die der Inhalt nach oben verschoben wird

HEIGHT: gibt darüber zusätzlichen Abstand

DEPTH: gibt darunter zusätzlichen Abstand

Beispiel:

```
Einige Zei\raisebox{2pt}{ch}en ragen aus dem Text he\raisebox{-2pt}{rau}s.
```

Einige Zei^{ch}en ragen aus dem Text he_{rau}s.

- ▶ Einzelne Bereiche einer Seite können in mehrere Spalten aufgeteilt werden.
- ▶ Dafür wird das **multicol**-Paket benötigt.

```

\usepackage{multicol}

\begin{document}
  \begin{multicols}{ANZAHL}
  Inhalt
  \end{multicols}
  ...
\end{document}

```

- ▶ Einzelne Bereiche einer Seite können in mehrere Spalten aufgeteilt werden.
- ▶ Dafür wird das **multicol**-Paket benötigt.

```

\usepackage{multicol}

\begin{document}
  \begin{multicols}{ANZAHL}
  Inhalt
  \end{multicols}
  ...
\end{document}
  
```

Der Abstand zwischen den Zeilen kann wie folgt angepasst werden:

```

\setlength{\columnsep}{1em}
  
```

Minipages geben – neben Spalten und Tabellen – die Möglichkeit, Inhalte zusammengehörig in einer Art **Container** zu anderen Inhalten auszurichten und ihnen eine feste Breite zu geben.

```
\begin{minipage}[ÄUSSERE POSITION][HÖHE][INNERE POSITION]{BREITE}  
Inhalt  
\end{minipage}
```

- ▶ ÄUSSERE POSITION (optional) – richtet die Minipage relativ zur aktuellen Grundlinie aus:
 - c center
 - t top
 - b bottom
- ▶ HÖHE (optional), ist eine gültige Längenangabe – durch welche die Gesamthöhe der Minipage bestimmt wird.
- ▶ INNERE POSITION (optional) – richtet den Inhalt der Minipage innerhalb der angegebenen HÖHE aus.
- ▶ BREITE (obligatorisch) – Gesamtbreite der Minipage

Beispiel:

```

\begin{minipage}{0.55\linewidth}
  Inhalt 1
\end{minipage}
\hfill % schiebt die nachfolgende Minipage an den rechten Layoutrand
\begin{minipage}{0.4\linewidth}
  Inhalt 2
\end{minipage}
  
```

Hinweis: Die Minipages richten sich standardmäßig vertikal mittig zueinander aus. Der Zwischenraum beträgt im Beispielcode 5% der Layoutbreite. Die Gesamtbreite alle Minipages sollte kleiner als 1 sein. Auch dieser Bereich ist noch zusammengehörig zur ersten Minipage.



Abbildung: Strand

- 1 Mathematikmodus
 - 1.1 inline vs. abgesetzter Modus
 - 1.2 Nützliche Befehle
 - 1.3 Spezielle Formatierungen
 - 1.4 Zahlen & Einheiten mit siunitx
- 2 Textstrukturierung
 - 2.1 Tabellen
 - 2.2 Boxen
- 2.3 Spaltenlayout
- 2.4 Minipages
- 3 Farben**
- 4 Quellcode einbinden
 - 4.1 Verbatim
 - 4.2 Lstlistings
 - 4.3 Liste der Programmcodes
- 5 Abspann

Zuerst muss das entsprechende Paket eingebunden werden:

```
\usepackage[cmyk,table]{xcolor} % um Farben zu benutzen
```


Zuerst muss das entsprechende Paket eingebunden werden:

```
\usepackage[cmym,table]{xcolor} % um Farben zu benutzen
```

Es können eigene Farben definiert werden:

Definition

```
\definecolor{Name}{Farbmodell}{Werte} % definiert eigene Farbe
```

Beispiel

```
\definecolor{magicpink}{cmym}{0,73, 0, 0}
```

Text farbig darstellen:

Befehl `\textcolor{Farbe}{Text}`

Beispiel `\textcolor{magicpink}{roter Text}`

So sieht's aus **roter Text**

Text farbig hinterlegen:

Befehl `\colorbox{Hintergrundfarbe}{Text}`

Beispiel `\colorbox{cyan}{cyan hinterlegter Text}`

So sieht's aus **cyan hinterlegter Text**

Text farbig umrahmen und hinterlegen:

Befehl `\fcolorbox{Rahmenfarbe}{Hintergrundfarbe}{Text}`

Beispiel `\fboxrule1mm % setzt Rahmenbreite fest`
`\fcolorbox{green}{yellow}{Text in Rahmen mit Hintergrundfarbe}`

So sieht's aus Text in Rahmen mit Hintergrundfarbe

Mit einer **colorbox** lassen sich Objekte farblich hinterlegen .

Mit einer `\textbf{colorbox}` lassen sich Objekte `\colorbox{blue!20!red!20}`{farblich hinterlegen}.

Mit einer **colorbox** lassen sich Objekte farblich hinterlegen .

Mit einer `\textbf{colorbox}` lassen sich Objekte `\colorbox{blue!20!red!20}`{farblich hinterlegen}.

Eine **Framebox** erlaubt es, Text farblich einzurahmen und zu hinterlegen .

`\setlength \fboxrule{2pt}` % setzt die Rahmenbreite auf 2pt
 Eine `\textbf{Framebox}` erlaubt es, Text `\fcolorbox{blue!50!red!50}`{blue!20!red!20}{farblich einzurahmen und zu hinterlegen}.

Zellen, Zeilen und Spalten in Tabellen färben:

Befehle

```

\cellcolor{Farbe} % ändert die Hintergrundfarbe einer Zelle
\rowcolor{Farbe} % ändert die Hintergrundfarbe einer Zeile
\columncolor{Farbe} % ändert die Hintergrundfarbe einer Spalte
  
```

Beispiel

```

\begin{tabular}{>{\columncolor{cyan}} c c c}
\rowcolor{yellow} Text & Text & Text \\
Text & Text & Text \\
Text & Text & \cellcolor{green} Text \\
Text & Text & Text \\
\end{tabular}
  
```

So sieht's aus

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

- 1 Mathematikmodus
 - 1.1 inline vs. abgesetzter Modus
 - 1.2 Nützliche Befehle
 - 1.3 Spezielle Formatierungen
 - 1.4 Zahlen & Einheiten mit siunitx
- 2 Textstrukturierung
 - 2.1 Tabellen
 - 2.2 Boxen
- 2.3 Spaltenlayout
- 2.4 Minipages
- 3 Farben
- 4 **Quellcode einbinden**
 - 4.1 Verbatim
 - 4.2 Lstlistings
 - 4.3 Liste der Programmcodes
- 5 Abspann

Wenn Text nicht vom L^AT_EX-Compiler interpretiert werden soll, kann man ihn in ein **verb**-Befehl setzen:

```
\verb| nicht interpretierter Text |
```

Dabei muss dem Befehl mitgeteilt werden, wann dies endet. Dies geschieht durch ein Zeichen, das den Bereich einleitet und abschließt.

Wenn Text nicht vom L^AT_EX-Compiler interpretiert werden soll, kann man ihn in ein **verb**-Befehl setzen:

```
\verb| nicht interpretierter Text |
```

Dabei muss dem Befehl mitgeteilt werden, wann dies endet. Dies geschieht durch ein Zeichen, das den Bereich einleitet und abschließt.

Für mehrere Zeilen gibt es die **verbatim**-Umgebung:

```
\usepackage{verbatim}

\begin{document}
  \begin{verbatim}
  diese Zeilen
  sollen nicht interpretiert
  werden _ & \ -- % Kommentar
  \end{verbatim}
\end{document}
```

Um Programmcode in das Dokument einzubinden, möchte man sprachspezifisches **Syntax-Highlighting** und Zeilenummerierungen haben.

Dafür gibt es das **listings**-Paket:

```
\usepackage{listings}
\begin{lstlisting}[language=LAN, caption={Beschriftung}]
...
\end{lstlisting}
```

Um Programmcode in das Dokument einzubinden, möchte man sprachspezifisches **Syntax-Highlighting** und Zeilennummerierungen haben.

Dafür gibt es das **listings**-Paket:

```
\usepackage{listings}
\begin{lstlisting}[language=LAN, caption={Beschriftung}]
...
\end{lstlisting}
```

- ▶ Eine Liste der Listings wird automatisch mit folgendem Begehl erstellt:
`\lstlistoflistings`
- ▶ Es werden sehr viele Programmier-Sprachen unterstützt. Einige wenige davon sind:
 - ▶ bash
 - ▶ C, C++
 - ▶ Haskell
 - ▶ HTML
 - ▶ Java
 - ▶ Matlab
 - ▶ PHP
 - ▶ Python
 - ▶ SQL
- ▶ Nun müssen noch weitere Einstellungen vorgenommen werden.

Quelle: <http://www.texdoc.net/texmf-dist/doc/latex/listings/listings.pdf>

```

\lstset{
  aboveskip=2ex,                % Abstand zum darüber liegenden Element
  basicstyle=\small\fontfamily{fvm}\selectfont, % \scriptsize the size of the fonts that are
  used for the code
  backgroundcolor = \color{bgcolour}, % legt Farbe der Box fest
  breakatwhitespace=false,      % sets if automatic breaks should only happen at
  whitespace
  captionpos=t,                % sets the caption-position to bottom, t for top
  commentstyle=\color{codeblue}\ttfamily, % comment style
  flexiblecolumns=false,       % selects the most recently selected flexible
  column format
  frame=single,                % adds a frame around the code
  keepspaces=true,             % keeps spaces in text, useful for keeping
  indentation of code (possibly needs columns=flexible)
  keywordstyle=\bfseries\color{blue}, % keyword style
  numberblanklines=false,      % leere Zeilen werden nicht mitnummeriert
  numbers=left,                % where to put the line-numbers; possible values
  are (none, left, right)
  numberstyle=\tiny\color{codegreen}, % the style that is used for the line-numbers
  numbersep=5pt,               % how far the line-numbers are from the code
  stepnumber=1,                % nummeriert nur jede i-te Zeile
  showspaces=false,            % show spaces everywhere adding particular
  underscores; it overrides 'showstringspaces'

```

```

% ...
showtabs=false,                % show tabs within strings adding particular
underscores                    % string literal style
stringstyle=\color{orange}\ttfamily, % Abstand zum linken Layoutrand
xleftmargin=1.2em,             % Abstand zum rechten Layoutrand
xrightmargin=0.4em,
}

```

Beispiel:

```
\begin{lstlisting}[language=Haskell, caption={meine Haskell Funktion}]  
fak :: [Integer] -> [Integer]  
fak 0 = 1  
fak n = n * fak(n-1)  
\end{lstlisting}
```

Und so sähe Haskell-Code aus:

Listing 1: meine Haskell Funktion

```
1 fak :: [Integer] -> [Integer]  
2 fak 0 = 1  
3 fak n = n * fak(n-1)
```

In der Regel programmiert man in einer Datei und möchte Teile dieser im $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Dokument einbinden. Dies geht mittels **lstinputstring**:

Listing 2: File einbinden

```
\lstinputstring[language=LAN, caption={Beschriftung}]{PATH/FILENAME.TYPE}
```

In der Regel programmiert man in einer Datei und möchte Teile dieser im \LaTeX -Dokument einbinden. Dies geht mittels **lstinputstring**:

Listing 3: File einbinden

```
\lstinputstring[language=LAN, caption={Beschriftung}]{PATH/FILENAME.TYPE}
```

Wenn nur einige Zeilen eingebunden werden sollen:

```
\lstinputstring[language=latex, firstline=23, lastline=42]{path/filename.tex}
% alternativ:
\lstinputstring[language=latex, linerange=23-42, firstnumber=23]{path/filename.tex}
```

linerange Bereich im Quellcode

firstline Beginn des einzubindenden Quellcodes

lastline Ende des einzubindenden Quellcodes

firstnumber Beginn der Zeilen-Nummerierung

Eine Liste der **lstlistings** wird wie folgt erstellt:

```
\lstlistoflistings
```

- 1 Mathematikmodus
 - 1.1 inline vs. abgesetzter Modus
 - 1.2 Nützliche Befehle
 - 1.3 Spezielle Formatierungen
 - 1.4 Zahlen & Einheiten mit siunitx
- 2 Textstrukturierung
 - 2.1 Tabellen
 - 2.2 Boxen
- 2.3 Spaltenlayout
- 2.4 Minipages
- 3 Farben
- 4 Quellcode einbinden
 - 4.1 Verbatim
 - 4.2 Lstlistings
 - 4.3 Liste der Programmcodes
- 5 Abspann

Templates Um euch den Einstieg etwas zu erleichtern, haben wir für euch **Templates** erstellt. Dieses könnt ihr für die Abgabe von Übungszetteln, zum Schreiben von Skripten oder einfach nur zum Üben benutzen.

- ▶ **Mathe/(Bio-)Informatik**

- <https://www.overleaf.com/read/szfymnrfdksc>

- ▶ **Pyhsik** <https://www.overleaf.com/read/pvmmgzchwysx>

Foliensatz Die Beamer-Folien gibt es online:

- ▶ Teil 1: <https://www.overleaf.com/read/wpfwmpnzfxtx>

- ▶ Teil 2: <https://www.overleaf.com/read/zqcjsybmzpyx>

Teil 3 folgt am 17. November 2017

Die Freie Universität Berlin bietet regelmäßig einen L^AT_EX-ABV-Kurs an:

<http://latex.userpage.fu-berlin.de/>

- [1] Jacques Crémer. *A very minimal introduction to TikZ**. Webseite. März 2011. URL: <http://cremeronline.com/LaTeX/minimaltikz.pdf> (besucht am 12.05.2017).
- [2] diabonas. *Drawing Graphs in L^AT_EX*. Webseite. Juni 2012. URL: <http://tex.stackexchange.com/questions/45734/drawing-graphs-in-latex> (besucht am 25.05.2017).
- [3] Matthias Heinkenschloss. *L^AT_EX Mathematical Symbols*. Webseite. URL: <http://www.caam.rice.edu/~heinken/latex/symbols.pdf> (besucht am 12.05.2017).
- [4] K. Kittel. *Matrizen*. Webseite. Mai 2016. URL: <http://www.kkittel.de/wiki/doku.php?id=mathematik:matrizen> (besucht am 12.05.2017).
- [5] Stefan Kottwitz. *Example: A simple Tree*. Webseite. Aug. 2015. URL: <http://www.texample.net/tikz/examples/tree/> (besucht am 30.05.2017).

- [6] Philipp Lehman. *The biblalex Package*. Webseite. Mai 2014. URL: <http://texdoc.net/texmf-dist/doc/latex/biblalex/biblalex.pdf> (besucht am 12. 05. 2017).
- [7] o.A. *L^AT_EX/Hyperlinks*. Webseite. Feb. 2017. URL: <https://en.wikibooks.org/wiki/LaTeX/Hyperlinks> (besucht am 26. 05. 2017).
- [8] o.A. *LaTeX Templates*. Webseite. 2017. URL: <https://www.latextemplates.com/> (besucht am 07. 06. 2017).
- [9] o.A. *LaTeX/Labels and Cross-referencing*. Webseite. Dez. 2016. URL: https://en.wikibooks.org/wiki/LaTeX/Labels_and_Cross-referencing (besucht am 26. 05. 2017).
- [10] o.A. *LaTeX/Presentations*. Webseite. Apr. 2017. URL: <https://en.wikibooks.org/wiki/LaTeX/Presentations> (besucht am 04. 11. 2017).
- [11] o.V. *T_EX Cookbook*. Webseite. 1989. URL: http://www.stat.missouri.edu/~liufei/Resources_files/TeX%2520cookbook.pdf (besucht am 12. 05. 2017).

- [12] o.V. *Einstiegshilfe in L^AT_EX*. Webseite. Dez. 2008. URL: <http://page.mi.fu-berlin.de/rhschulz/Studienberatung/latexinfo.html> (besucht am 12.05.2017).
- [13] o.V. *o.T.* Webseite. TikZ examples. URL: <http://www.texample.net> (besucht am 12.05.2017).
- [14] Matthias Pospiech. *Erstellung von Bachelor und Masterarbeiten mit L^AT_EX*. Webseite. Dez. 2011. URL: <http://www.matthiaspospiech.de/blog/2011/12/09/erstellung-von-bachelor-und-masterarbeiten-mit-latex/> (besucht am 12.05.2017).
- [15] Matthias Pospiech. *L^AT_EX Variablen, If Abfragen und Schleifen*. Webseite. Apr. 2008. URL: <http://www.matthiaspospiech.de/blog/2008/04/13/latex-variablen-if-abfragen-und-schleifen/> (besucht am 12.05.2017).
- [16] Thomas Quaritsch. *Anwendung für Fortgeschrittene*. Webseite. Nov. 2013. URL: <http://latex.tugraz.at/latex/fortgeschrittene> (besucht am 26.05.2017).

- [17] Andrew Stacey. *The tikzmark package*. Webseite. Version 1.2. Apr. 2016. URL: <http://ctan.mackichan.com/graphics/pgf/contrib/tikzmark/tikzmark.pdf> (besucht am 12.05.2017).
- [18] Hauke Stieler. *TikZ für Automaten*. Webseite. Version 0.2de_DE. Apr. 2015. URL: <http://hauke-stieler.de/public/tikz-for-state-machines.pdf> (besucht am 12.05.2017).
- [19] Herbert Voß. *LaTeX Referenz der Umgebungen, Makros, Längen und Zähler*. Bd. 1. Dez. 2013. URL: <https://www.lehmanns.de/page/latexreferenz> (besucht am 01.06.2017).
- [20] Herbert Voß. *Math mode*. Webseite. Jan. 2014. URL: <http://tug.ctan.org/obsolete/info/math/voss/mathmode/Mathmode.pdf> (besucht am 12.05.2017).
- [21] Joseph Wright. *siunitx – A comprehensive (SI)units package*. Webseite. Sep. 2011. URL: <https://www.dpg-physik.de/dpg/gliederung/junge/rg/wuerzburg/Archiv/WS%202011-12/LaTeX/siunitx.pdf> (besucht am 30.05.2017).