

# LaTeX – Kommandos und Layout

Online Tutorium 2

Florian Sihler ◦ 10.05.2019

# Die heutigen Themen

1. Abschnitte und Kapitelübersicht
2. Kopf- und Fußzeile
3. Geometrie der Seite
4. Befehle in Latex
5. Hyperlinks und Metadaten
6. Grafiken und Listen

# Abschnitte und Kapitelübersicht

## 1. Abschnitte und Kapitelübersicht

- Die wichtigsten Gliederungsbefehle
- Exkurs: Wie liest  $\text{\LaTeX}$
- Weitere Feinheiten
- Dokumenttitel, selbst gemacht

## 2. Kopf- und Fußzeile

## 3. Geometrie der Seite

## 4. Befehle in Latex

## 5. Hyperlinks und Metadaten

## 6. Grafiken und Listen

# Wie gliedert man ein Dokument?

# Wie gliedert man ein Dokument?

- Gliedern mit (absteigend):

# Wie gliedert man ein Dokument?

- Gliedern mit (absteigend): `\part`,

# Wie gliedert man ein Dokument?

- Gliedern mit (absteigend): `\part`, `\chapter`,

# Wie gliedert man ein Dokument?

- Gliedern mit (absteigend): `\part`, `\chapter`, `\section`,



# Wie gliedert man ein Dokument?

- Gliedern mit (absteigend): `\part`, `\chapter`, `\section`, `\subsection`,

# Wie gliedert man ein Dokument?

- Gliedern mit (absteigend): `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`,

# Wie gliedert man ein Dokument?

- Gliedern mit (absteigend): `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` und

# Wie gliedert man ein Dokument?

- Gliedern mit (absteigend): `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` und `\subparagraph`.

# Wie gliedert man ein Dokument?

- Gliedern mit (absteigend): `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` und `\subparagraph`.
- `\part` und `\chapter` gibt es nur in Dokumentklassen wie *report* oder *book*.

# Wie gliedert man ein Dokument?

- Gliedern mit (absteigend): `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` und `\subparagraph`.
- `\part` und `\chapter` gibt es nur in Dokumentklassen wie *report* oder *book*.
- Alle übernehmen als Argument den jeweiligen Titel, also:

# Wie gliedert man ein Dokument?

- Gliedern mit (absteigend): `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` und `\subparagraph`.
- `\part` und `\chapter` gibt es nur in Dokumentklassen wie *report* oder *book*.
- Alle übernehmen als Argument den jeweiligen Titel, also: `\section{<Titel>}`

# Wie gliedert man ein Dokument?

- Gliedern mit (absteigend): `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` und `\subparagraph`.
- `\part` und `\chapter` gibt es nur in Dokumentklassen wie *report* oder *book*.
- Alle übernehmen als Argument den jeweiligen Titel, also: `\section{<Titel>}`

```
1 | \documentclass{article}
2 |
3 | \begin{document}
4 |   \section{Ein Abschnitt}
5 |   \subsection{Ein Unterabschnitt}
6 |   Hallo Welt
7 |   \section{Ein anderer Abschnitt}
8 | \end{document}
```



# Wie gliedert man ein Dokument?

- Gliedern mit (absteigend): `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` und `\subparagraph`.
- `\part` und `\chapter` gibt es nur in Dokumentklassen wie *report* oder *book*.
- Alle übernehmen als Argument den jeweiligen Titel, also: `\section{<Titel>}`

```
1 | \documentclass{article}
2 |
3 | \begin{document}
4 |   \section{Ein Abschnitt}
5 |   \subsection{Ein Unterabschnitt}
6 |   Hallo Welt
7 |   \section{Ein anderer Abschnitt}
8 | \end{document}
```

**1    Ein Abschnitt**

1.1   Ein Unterabschnitt

Hallo Welt

**2    Ein anderer Abschnitt**

# Gliederungsübersicht

# Gliederungsübersicht

- Alle Gliederungspunkte erhält man mit `\tableofcontents`.

# Gliederungsübersicht

- Alle Gliederungspunkte erhält man mit `\tableofcontents`.
- Hierzu muss das Dokument zweimal kompiliert werden.

# Gliederungsübersicht

- Alle Gliederungspunkte erhält man mit `\tableofcontents`.
- Hierzu muss das Dokument zweimal kompiliert werden. Warum? Gleich!

# Gliederungsübersicht

- Alle Gliederungspunkte erhält man mit `\tableofcontents`.
- Hierzu muss das Dokument zweimal kompiliert werden. Warum? Gleich!
- Beispiel:

# Gliederungsübersicht

- Alle Gliederungspunkte erhält man mit `\tableofcontents`.
- Hierzu muss das Dokument zweimal kompiliert werden. Warum? Gleich!
- Beispiel:

```
1 | \documentclass{article}
2 |
3 | \begin{document}
4 |     \tableofcontents
5 |     \section{Ein Abschnitt}
6 |     \subsection{Ein Unterabschnitt}
7 |     Hallo Welt
8 |     \section{Ein anderer Abschnitt}
9 | \end{document}
```

# Gliederungsübersicht

- Alle Gliederungspunkte erhält man mit `\tableofcontents`.
- Hierzu muss das Dokument zweimal kompiliert werden. Warum? Gleich!
- Beispiel:

```
1 | \documentclass{article}
2 |
3 | \begin{document}
4 |   \tableofcontents
5 |   \section{Ein Abschnitt}
6 |   \subsection{Ein Unterabschnitt}
7 |   Hallo Welt
8 |   \section{Ein anderer Abschnitt}
9 | \end{document}
```

## Contents

1	Ein Abschnitt	1
1.1	Ein Unterabschnitt	1
2	Ein anderer Abschnitt	1
1	Ein Abschnitt	
1.1	Ein Unterabschnitt	
	Hallo Welt	
2	Ein anderer Abschnitt	



# Exkurs: Wie liest $\text{\LaTeX}$ ?

# Exkurs: Wie liest $\text{\LaTeX}$ ?

- $\text{\LaTeX}$  liest das Dokument einmal von oben nach unten.

# Exkurs: Wie liest $\text{\LaTeX}$ ?

- $\text{\LaTeX}$  liest das Dokument einmal von oben nach unten.
- Später auftretende `\section`-Befehle gibt es für `\tableofcontents` also noch gar nicht.

# Exkurs: Wie liest $\text{\LaTeX}$ ?

- $\text{\LaTeX}$  liest das Dokument einmal von oben nach unten.
- Später auftretende `\section`-Befehle gibt es für `\tableofcontents` also noch gar nicht.
- Lösung:  $\text{\LaTeX}$  schreibt sich Hilfsdateien (`.aux`, `.toc`, `.lot`, ...), die die Daten vorheriger Läufe sammeln.

# Exkurs: Wie liest $\text{\LaTeX}$ ?

- $\text{\LaTeX}$  liest das Dokument einmal von oben nach unten.
- Später auftretende `\section`-Befehle gibt es für `\tableofcontents` also noch gar nicht.
- Lösung:  $\text{\LaTeX}$  schreibt sich Hilfsdateien (`.aux`, `.toc`, `.lot`, ...), die die Daten vorheriger Läufe sammeln.
- Deswegen ist in diesem Fall mehrfaches kompilieren notwendig!

# Du siehst mich nicht!

# Du siehst mich nicht!

- Alle Gliederungsbefehle besitzen eine Variante mit Sternchen (`\section*`, ...)

# Du siehst mich nicht!

- Alle Gliederungsbefehle besitzen eine Variante mit Sternchen (`\section*`, ...)
- Diese tauchen nicht in der Inhaltsübersicht auf und werden nicht nummeriert!



# Du siehst mich nicht!

- Alle Gliederungsbefehle besitzen eine Variante mit Sternchen (`\section*`, ...)
- Diese tauchen nicht in der Inhaltsübersicht auf und werden nicht nummeriert!
- Beispiel:

# Du siehst mich nicht!

- Alle Gliederungsbefehle besitzen eine Variante mit Sternchen (`\section*`, ...)
- Diese tauchen nicht in der Inhaltsübersicht auf und werden nicht nummeriert!
- Beispiel:

```
1 | \documentclass{article}
2 |
3 | \begin{document}
4 |     \tableofcontents
5 |     \section{Ein Abschnitt}
6 |     \subsection{Ein Unterabschnitt}
7 |     Hallo Welt
8 |     \section*{Versteckt!}
9 | \end{document}
```

# Du siehst mich nicht!

- Alle Gliederungsbefehle besitzen eine Variante mit Sternchen (`\section*`, ...)
- Diese tauchen nicht in der Inhaltsübersicht auf und werden nicht nummeriert!
- Beispiel:

```
1 | \documentclass{article}
2 |
3 | \begin{document}
4 |     \tableofcontents
5 |     \section{Ein Abschnitt}
6 |     \subsection{Ein Unterabschnitt}
7 |     Hallo Welt
8 |     \section*{Versteckt!}
9 | \end{document}
```

## Contents

1	Ein Abschnitt	1
1.1	Ein Unterabschnitt	1

## 1 Ein Abschnitt

### 1.1 Ein Unterabschnitt

Hallo Welt

**Versteckt!**

# Guten Tag, ich heiße Günther!

# Guten Tag, ich heiße Günther!

- Alle Gliederungsbefehle akzeptieren ein optionales Argument in eckigen Klammern.

# Guten Tag, ich heiße Günther!

- Alle Gliederungsbefehle akzeptieren ein optionales Argument in eckigen Klammern.
- Dies ist dann der Bezeichner für `\tableofcontents`

# Guten Tag, ich heiße Günther!

- Alle Gliederungsbefehle akzeptieren ein optionales Argument in eckigen Klammern.
- Dies ist dann der Bezeichner für `\tableofcontents`
- Dann nützlich, wenn Titel sehr lang. Beispiel:

# Guten Tag, ich heiße Günther!

- Alle Gliederungsbefehle akzeptieren ein optionales Argument in eckigen Klammern.
- Dies ist dann der Bezeichner für `\tableofcontents`
- Dann nützlich, wenn Titel sehr lang. Beispiel:

```
1 | \documentclass{article}
2 |
3 | \begin{document}
4 |     \tableofcontents
5 |     \section[Von der Sonne]{Hi}
6 |     \subsection[Vom Mond!]{Ho}
7 |     Hallo Welt
8 |     \section*{Versteckt!}
9 | \end{document}
```



# Guten Tag, ich heiße Günther!

- Alle Gliederungsbefehle akzeptieren ein optionales Argument in eckigen Klammern.
- Dies ist dann der Bezeichner für `\tableofcontents`
- Dann nützlich, wenn Titel sehr lang. Beispiel:

```
1 | \documentclass{article}
2 |
3 | \begin{document}
4 |   \tableofcontents
5 |   \section[Von der Sonne]{Hi}
6 |   \subsection[Vom Mond!]{Ho}
7 |   Hallo Welt
8 |   \section*{Versteckt!}
9 | \end{document}
```

## Contents

1	Von der Sonne	1
1.1	Vom Mond! .....	1

## 1 Hi

### 1.1 Ho

Hallo Welt

Versteckt!

# Spaß mit `\maketitle`

# Spaß mit `\maketitle`

- Mittels `\maketitle` kann ein Titel generiert werden

# Spaß mit `\maketitle`

- Mittels `\maketitle` kann ein Titel generiert werden
- Dieser benötigt einen Titel (`\title`) und kann mit weiteren Befehlen wie `\author` gefüttert werden.

# Spaß mit `\maketitle`

- Mittels `\maketitle` kann ein Titel generiert werden
- Dieser benötigt einen Titel (`\title`) und kann mit weiteren Befehlen wie `\author` gefüttert werden.
- In Befehlen wie `\author` können mittels `\and` mehrere aufgeführt werden

# Spaß mit `\maketitle`

- Mittels `\maketitle` kann ein Titel generiert werden
- Dieser benötigt einen Titel (`\title`) und kann mit weiteren Befehlen wie `\author` gefüttert werden.
- In Befehlen wie `\author` können mittels `\and` mehrere aufgeführt werden
- Beispiel:

# Spaß mit `\maketitle`

- Mittels `\maketitle` kann ein Titel generiert werden
- Dieser benötigt einen Titel (`\title`) und kann mit weiteren Befehlen wie `\author` gefüttert werden.
- In Befehlen wie `\author` können mittels `\and` mehrere aufgeführt werden
- Beispiel:

```
1 | \documentclass{article}
2 |
3 | \title{Geschichten von Gestern}
4 | \author{Ich \and Du}
5 | \date{\today}
6 |
7 | \begin{document}
8 |     \maketitle
9 | \end{document}
```

# Spaß mit `\maketitle`

- Mittels `\maketitle` kann ein Titel generiert werden
- Dieser benötigt einen Titel (`\title`) und kann mit weiteren Befehlen wie `\author` gefüttert werden.
- In Befehlen wie `\author` können mittels `\and` mehrere aufgeführt werden
- Beispiel:

```
1 | \documentclass{article}
2 |
3 | \title{Geschichten von Gestern}
4 | \author{Ich \and Du}
5 | \date{\today}
6 |
7 | \begin{document}
8 |   \maketitle
9 | \end{document}
```

Geschichten von Gestern

Ich            Du

September 30, 2021



# Kopf- und Fußzeile

## 1. Abschnitte und Kapitelübersicht

## 2. Kopf- und Fußzeile

- Übersicht über die Pakete
- Mit scrlayer-scrpage
- Mit fancyhdr

## 3. Geometrie der Seite

## 4. Befehle in Latex

## 5. Hyperlinks und Metadaten

## 6. Grafiken und Listen

# Übersicht über die Pakete

# Übersicht über die Pakete

- In der Regel: `scrlayer-scrpage` oder `fancyhdr`

# Übersicht über die Pakete

- In der Regel: `scrlayer-scrpage` oder `fancyhdr`
- Hinweis: `scrpage2` ist seit 2015 veraltet!

# Übersicht über die Pakete

- In der Regel: `scrlayer-scrpage` oder `fancyhdr`
- Hinweis: `scrpage2` ist seit 2015 veraltet!
- Wenn Koma-Skriptklasse: `scrlayer-scrpage`

# Übersicht über die Pakete

- In der Regel: `scrlayer-scrpage` oder `fancyhdr`
- Hinweis: `scrpage2` ist seit 2015 veraltet!
- Wenn Koma-Skriptklasse: `scrlayer-scrpage`
- Nützlicher Befehl innerhalb:

# Übersicht über die Pakete

- In der Regel: `scrlayer-scrpage` oder `fancyhdr`
- Hinweis: `scrpage2` ist seit 2015 veraltet!
- Wenn Koma-Skriptklasse: `scrlayer-scrpage`
- Nützlicher Befehl innerhalb: `\thepage` (aktuelle Seitenzahl)

# Übersicht über die Pakete

- In der Regel: `scrlayer-scrpage` oder `fancyhdr`
- Hinweis: `scrpage2` ist seit 2015 veraltet!
- Wenn Koma-Skriptklasse: `scrlayer-scrpage`
- Nützlicher Befehl innerhalb: `\thepage` (aktuelle Seitenzahl)
- Auch nützlich: Das Paket `lastpage`,



# Übersicht über die Pakete

- In der Regel: `scrlayer-scrpage` oder `fancyhdr`
- Hinweis: `scrpage2` ist seit 2015 veraltet!
- Wenn Koma-Skriptklasse: `scrlayer-scrpage`
- Nützlicher Befehl innerhalb: `\thepage` (aktuelle Seitenzahl)
- Auch nützlich: Das Paket `lastpage`, liefert mit `\pageref{LastPage}` Seitenzahl der letzten Seite!

# Übersicht über die Pakete

- In der Regel: `scrlayer-scrpage` oder `fancyhdr`
- Hinweis: `scrpage2` ist seit 2015 veraltet!
- Wenn Koma-Skriptklasse: `scrlayer-scrpage`
- Nützlicher Befehl innerhalb: `\thepage` (aktuelle Seitenzahl)
- Auch nützlich: Das Paket `lastpage`, liefert mit `\pageref{LastPage}` Seitenzahl der letzten Seite! (Dies benötigt ebenfalls einen weiteren Kompiliervorgang!)

# Übersicht über die Pakete

- In der Regel: `scrlayer-scrpage` oder `fancyhdr`
- Hinweis: `scrpage2` ist seit 2015 veraltet!
- Wenn Koma-Skriptklasse: `scrlayer-scrpage`
- Nützlicher Befehl innerhalb: `\thepage` (aktuelle Seitenzahl)
- Auch nützlich: Das Paket `lastpage`, liefert mit `\pageref{LastPage}` Seitenzahl der letzten Seite! (Dies benötigt ebenfalls einen weiteren Kompiliervorgang!)
- Hinweis: `memoir`-Klassen machen ihr eigenes Ding

# Übersicht über die Pakete

- In der Regel: `scrlayer-scrpage` oder `fancyhdr`
- Hinweis: `scrpage2` ist seit 2015 veraltet!
- Wenn Koma-Skriptklasse: `scrlayer-scrpage`
- Nützlicher Befehl innerhalb: `\thepage` (aktuelle Seitenzahl)
- Auch nützlich: Das Paket `lastpage`, liefert mit `\pageref{LastPage}` Seitenzahl der letzten Seite! (Dies benötigt ebenfalls einen weiteren Kompiliervorgang!)
- Hinweis: `memoir`-Klassen machen ihr eigenes Ding 😊.

# Die Freuden des Koma-Scripts

# Die Freuden des Koma-Scripts

- Hinweis: Dieses Paket kann viel mehr, wir schauen uns nur einen Teil an!

# Die Freuden des Koma-Scripts

- Hinweis: Dieses Paket kann viel mehr, wir schauen uns nur einen Teil an!
- Für doppelseitige (`\book`, ...) Dokumente: Noch mehr Befehle.

# Die Freuden des Koma-Scripts

- Hinweis: Dieses Paket kann viel mehr, wir schauen uns nur einen Teil an!
- Für doppelseitige (book, ...) Dokumente: Noch mehr Befehle.
- Sonst: `\ohead`, `\chead`, `\ihead` für



# Die Freuden des Koma-Scripts

- Hinweis: Dieses Paket kann viel mehr, wir schauen uns nur einen Teil an!
- Für doppelseitige (book, ...) Dokumente: Noch mehr Befehle.
- Sonst: `\ohead`, `\chead`, `\ihhead` für oben-links, oben-mitte und oben-rechts.

# Die Freuden des Koma-Scripts

- Hinweis: Dieses Paket kann viel mehr, wir schauen uns nur einen Teil an!
- Für doppelseitige (book, ...) Dokumente: Noch mehr Befehle.
- Sonst: `\ohead`, `\chead`, `\ihead` für oben-links, oben-mitte und oben-rechts. analog `\ofoot`, `\cfoot`, `\ifoot` für unten.

# Die Freuden des Koma-Scripts

- Hinweis: Dieses Paket kann viel mehr, wir schauen uns nur einen Teil an!
- Für doppelseitige (book, ...) Dokumente: Noch mehr Befehle.
- Sonst: `\ohead`, `\chead`, `\ihead` für oben-links, oben-mitte und oben-rechts. analog `\ofoot`, `\cfoot`, `\ifoot` für unten.
- Löschen bisheriger Einstellung: `\clearscrheadfoot`.

# Die Freuden des Koma-Scripts

- Hinweis: Dieses Paket kann viel mehr, wir schauen uns nur einen Teil an!
- Für doppelseitige (book, ...) Dokumente: Noch mehr Befehle.
- Sonst: `\ohead`, `\chead`, `\ihead` für oben-links, oben-mitte und oben-rechts. analog `\ofoot`, `\cfoot`, `\ifoot` für unten.
- Löschen bisheriger Einstellung: `\clearscrheadfoot`.
- Da `scrlayer-scrpage` mit Markern arbeitet, gibt es auch `\pagemark` (Seitenzahl) und viele weitere

# Die Freuden des Koma-Scripts

- Hinweis: Dieses Paket kann viel mehr, wir schauen uns nur einen Teil an!
- Für doppelseitige (book, ...) Dokumente: Noch mehr Befehle.
- Sonst: `\ohead`, `\chead`, `\ihead` für oben-links, oben-mitte und oben-rechts. analog `\ofoot`, `\cfoot`, `\ifoot` für unten.
- Löschen bisheriger Einstellung: `\clearscrheadfoot`.
- Da `scrlayer-scrpage` mit Markern arbeitet, gibt es auch `\pagemark` (Seitenzahl) und viele weitere
- Layout wird durch: `\pagestyle{scrheadings}` erst aktiv!

# Ein Beispiel

# Ein Beispiel

```
1 | \documentclass{article}
2 | \usepackage{scrlayer-scrpage}
3 | \ofoot{\pagemark}
4 | \ifoot{Florian Dieter}
5 |
6 | \pagestyle{scrheadings}
7 |
8 | \begin{document}
9 |     \section{Willkommen in Narnia}
10 |     Hi
11 | \end{document}
```

# Ein Beispiel

```
1 | \documentclass{article}
2 | \usepackage{scrlayer-scrpage}
3 | \ofoot{\pagemark}
4 | \ifoot{Florian Dieter}
5 |
6 | \pagestyle{scrheadings}
7 |
8 | \begin{document}
9 |   \section{Willkommen in Narnia}
10 |   Hi
11 | \end{document}
```

*Florian Dieter*

1



# Die Freuden der transparenten Blätter

# Die Freuden der transparenten Blätter

- Hinweis: Dieses Paket kann viel mehr, wir schauen uns nur einen Teil an!

# Die Freuden der transparenten Blätter

- Hinweis: Dieses Paket kann viel mehr, wir schauen uns nur einen Teil an!
- Für doppelseitige (`book`, ...) Dokumente: Noch mehr Optionen.

# Die Freuden der transparenten Blätter

- Hinweis: Dieses Paket kann viel mehr, wir schauen uns nur einen Teil an!
- Für doppelseitige (`book, ...`) Dokumente: Noch mehr Optionen.
- Sonst: `\fancyhead` bzw. `\fancyfoot`. Erhält in eckigen Klammern ob L (links), C (mittig) oder R (rechts).

# Die Freuden der transparenten Blätter

- Hinweis: Dieses Paket kann viel mehr, wir schauen uns nur einen Teil an!
- Für doppelseitige (book, ...) Dokumente: Noch mehr Optionen.
- Sonst: `\fancyhead` bzw. `\fancyfoot`. Erhält in eckigen Klammern ob L (links), C (mittig) oder R (rechts).
- Löschen bisheriger Einstellung: `\fancyhf`.

# Die Freuden der transparenten Blätter

- Hinweis: Dieses Paket kann viel mehr, wir schauen uns nur einen Teil an!
- Für doppelseitige (`book`, ...) Dokumente: Noch mehr Optionen.
- Sonst: `\fancyhead` bzw. `\fancyfoot`. Erhält in eckigen Klammern ob L (links), C (mittig) oder R (rechts).
- Löschen bisheriger Einstellung: `\fancyhf`.
- Da auch `fancyhdr` mit Markern arbeitet, gibt es auch `\leftmark` (oberer Gliederungsname), `\rightmark` (untere Gliederungsname) und viele mehr.

# Die Freuden der transparenten Blätter

- Hinweis: Dieses Paket kann viel mehr, wir schauen uns nur einen Teil an!
- Für doppelseitige (`book, ...`) Dokumente: Noch mehr Optionen.
- Sonst: `\fancyhead` bzw. `\fancyfoot`. Erhält in eckigen Klammern ob L (links), C (mittig) oder R (rechts).
- Löschen bisheriger Einstellung: `\fancyhf`.
- Da auch `fancyhdr` mit Markern arbeitet, gibt es auch `\leftmark` (oberer Gliederungsname), `\rightmark` (untere Gliederungsname) und viele mehr.
- Layout wird durch: `\pagestyle{fancy}` erst aktiv!

# Ein Beispiel



# Ein Beispiel

```
1 | \documentclass{article}
2 | \usepackage{fancyhdr}
3 | \fancyhf{}
4 | \fancyfoot[R]{\thepage}
5 | \fancyfoot[L]{\leftmark}
6 |
7 | \pagestyle{fancy}
8 |
9 | \begin{document}
10 |     \section{Willkommen in Narnia}
11 |     Hi
12 | \end{document}
```

# Ein Beispiel

```
1 | \documentclass{article}
2 | \usepackage{fancyhdr}
3 | \fancyhf{}
4 | \fancyfoot[R]{\thepage}
5 | \fancyfoot[L]{\leftmark}
6 |
7 | \pagestyle{fancy}
8 |
9 | \begin{document}
10 |   \section{Willkommen in Narnia}
11 |   Hi
12 | \end{document}
```

1 WILLKOMMEN IN NARNIA

1

# Geometrie der Seite

1. Abschnitte und Kapitelübersicht
2. Kopf- und Fußzeile
3. Geometrie der Seite
  - Das geometry-Paket
  - Margins und marginpar
4. Befehle in Latex
5. Hyperlinks und Metadaten
6. Grafiken und Listen

# Mädchen für alles: Das geometry-Paket

# Mädchen für alles: Das geometry-Paket

- Alle folgenden Argumente lassen sich dem Paket direkt übergeben.

# Mädchen für alles: Das geometry-Paket

- Alle folgenden Argumente lassen sich dem Paket direkt übergeben.
- Die meisten lassen sich danach mittels `\newgeometry` anwenden.

# Mädchen für alles: Das geometry-Paket

- Alle folgenden Argumente lassen sich dem Paket direkt übergeben.
- Die meisten lassen sich danach mittels `\newgeometry` anwenden. (Hiervon ausgenommen sind Befehle die das Papierformat verhunzen etc.)

# Mädchen für alles: Das geometry-Paket

- Alle folgenden Argumente lassen sich dem Paket direkt übergeben.
- Die meisten lassen sich danach mittels `\newgeometry` anwenden. (Hiervon ausgenommen sind Befehle die das Papierformat verhunzen etc.)
- Eine gesamte Liste aller Optionen liefert die tolle Dokumentation.



# Mädchen für alles: Das geometry-Paket

- Alle folgenden Argumente lassen sich dem Paket direkt übergeben.
- Die meisten lassen sich danach mittels `\newgeometry` anwenden. (Hiervon ausgenommen sind Befehle die das Papierformat verhunzen etc.)
- Eine gesamte Liste aller Optionen liefert die tolle Dokumentation.  
(<https://ftp.uni-erlangen.de/ctan/macros/latex/contrib/geometry/geometry.pdf>)

# Mädchen für alles: Das geometry-Paket

- Alle folgenden Argumente lassen sich dem Paket direkt übergeben.
- Die meisten lassen sich danach mittels `\newgeometry` anwenden. (Hiervon ausgenommen sind Befehle die das Papierformat verhunzen etc.)
- Eine gesamte Liste aller Optionen liefert die tolle Dokumentation.  
(<https://ftp.uni-erlangen.de/ctan/macros/latex/contrib/geometry/geometry.pdf>)
- Die Auswahl des Papierformats mittels: `a0paper`, ..., `a4paper`, ..., `a6paper`,

# Mädchen für alles: Das geometry-Paket

- Alle folgenden Argumente lassen sich dem Paket direkt übergeben.
- Die meisten lassen sich danach mittels `\newgeometry` anwenden. (Hiervon ausgenommen sind Befehle die das Papierformat verhunzen etc.)
- Eine gesamte Liste aller Optionen liefert die tolle Dokumentation.  
(<https://ftp.uni-erlangen.de/ctan/macros/latex/contrib/geometry/geometry.pdf>)
- Die Auswahl des Papierformats mittels: `a0paper`, ..., `a4paper`, ..., `a6paper`, analog die anderen Formate `b4paper`, `c3paper`, ..., `letterpaper`, `legalpaper`, ....

# Mädchen für alles: Das geometry-Paket

- Alle folgenden Argumente lassen sich dem Paket direkt übergeben.
- Die meisten lassen sich danach mittels `\newgeometry` anwenden. (Hiervon ausgenommen sind Befehle die das Papierformat verhunzen etc.)
- Eine gesamte Liste aller Optionen liefert die tolle Dokumentation.  
(<https://ftp.uni-erlangen.de/ctan/macros/latex/contrib/geometry/geometry.pdf>)
- Die Auswahl des Papierformats mittels: `a0paper`, ..., `a4paper`, ..., `a6paper`, analog die anderen Formate `b4paper`, `c3paper`, ..., `letterpaper`, `legalpaper`, ....
- Grenzen nach außen: `top`, `left`, `right` und `bottom`,

# Mädchen für alles: Das geometry-Paket

- Alle folgenden Argumente lassen sich dem Paket direkt übergeben.
- Die meisten lassen sich danach mittels `\newgeometry` anwenden. (Hiervon ausgenommen sind Befehle die das Papierformat verhunzen etc.)
- Eine gesamte Liste aller Optionen liefert die tolle Dokumentation.  
(<https://ftp.uni-erlangen.de/ctan/macros/latex/contrib/geometry/geometry.pdf>)
- Die Auswahl des Papierformats mittels: `a0paper`, ..., `a4paper`, ..., `a6paper`, analog die anderen Formate `b4paper`, `c3paper`, ..., `letterpaper`, `legalpaper`, ....
- Grenzen nach außen: `top`, `left`, `right` und `bottom`, oder: Definition der Textbreite mittels `textheight` und `textwidth`.

# Mädchen für alles: Das geometry-Paket

- Alle folgenden Argumente lassen sich dem Paket direkt übergeben.
- Die meisten lassen sich danach mittels `\newgeometry` anwenden. (Hiervon ausgenommen sind Befehle die das Papierformat verhunzen etc.)
- Eine gesamte Liste aller Optionen liefert die tolle Dokumentation.  
(<https://ftp.uni-erlangen.de/ctan/macros/latex/contrib/geometry/geometry.pdf>)
- Die Auswahl des Papierformats mittels: `a0paper`, ..., `a4paper`, ..., `a6paper`, analog die anderen Formate `b4paper`, `c3paper`, ..., `letterpaper`, `legalpaper`, ....
- Grenzen nach außen: `top`, `left`, `right` und `bottom`, oder: Definition der Textbreite mittels `textheight` und `textwidth`. (Es gibt noch andere Möglichkeiten.)

# Mädchen für alles: Das geometry-Paket

- Alle folgenden Argumente lassen sich dem Paket direkt übergeben.
- Die meisten lassen sich danach mittels `\newgeometry` anwenden. (Hiervon ausgenommen sind Befehle die das Papierformat verhunzen etc.)
- Eine gesamte Liste aller Optionen liefert die tolle Dokumentation.  
(<https://ftp.uni-erlangen.de/ctan/macros/latex/contrib/geometry/geometry.pdf>)
- Die Auswahl des Papierformats mittels: `a0paper`, ..., `a4paper`, ..., `a6paper`, analog die anderen Formate `b4paper`, `c3paper`, ..., `letterpaper`, `legalpaper`, ....
- Grenzen nach außen: `top`, `left`, `right` und `bottom`, oder: Definition der Textbreite mittels `textheight` und `textwidth`. (Es gibt noch andere Möglichkeiten.)
- Auch möglich:

# Mädchen für alles: Das geometry-Paket

- Alle folgenden Argumente lassen sich dem Paket direkt übergeben.
- Die meisten lassen sich danach mittels `\newgeometry` anwenden. (Hiervon ausgenommen sind Befehle die das Papierformat verhunzen etc.)
- Eine gesamte Liste aller Optionen liefert die tolle Dokumentation.  
(<https://ftp.uni-erlangen.de/ctan/macros/latex/contrib/geometry/geometry.pdf>)
- Die Auswahl des Papierformats mittels: `a0paper`, ..., `a4paper`, ..., `a6paper`, analog die anderen Formate `b4paper`, `c3paper`, ..., `letterpaper`, `legalpaper`, ....
- Grenzen nach außen: `top`, `left`, `right` und `bottom`, oder: Definition der Textbreite mittels `textheight` und `textwidth`. (Es gibt noch andere Möglichkeiten.)
- Auch möglich: Kombination mit `total={w,h}`



# Mädchen für alles: Das geometry-Paket

- Alle folgenden Argumente lassen sich dem Paket direkt übergeben.
- Die meisten lassen sich danach mittels `\newgeometry` anwenden. (Hiervon ausgenommen sind Befehle die das Papierformat verhunzen etc.)
- Eine gesamte Liste aller Optionen liefert die tolle Dokumentation.  
(<https://ftp.uni-erlangen.de/ctan/macros/latex/contrib/geometry/geometry.pdf>)
- Die Auswahl des Papierformats mittels: `a0paper`, ..., `a4paper`, ..., `a6paper`, analog die anderen Formate `b4paper`, `c3paper`, ..., `letterpaper`, `legalpaper`, ....
- Grenzen nach außen: `top`, `left`, `right` und `bottom`, oder: Definition der Textbreite mittels `textheight` und `textwidth`. (Es gibt noch andere Möglichkeiten.)
- Auch möglich: Kombination mit `total={w,h}` welches die effektive Text-Dimensionen definiert.

# Ein Beispiel

# Ein Beispiel

```
1 | \documentclass{article}
2 |
3 | \usepackage[a4paper, left=10cm,
4 |   right=8cm]{geometry}
5 |
6 | \begin{document}
7 |   Blah blah blah, ich bin langer Text.
8 |   Du du du, Ich füll die Seite jetzt!
9 | \end{document}
```

# Ein Beispiel

```
1 | \documentclass{article}
2 |
3 | \usepackage[a4paper, left=10cm,
4 |   right=8cm]{geometry}
5 |
6 | \begin{document}
7 |   Blah blah blah, ich bin langer Text.
8 |   Du du du, Ich füll die Seite jetzt!
9 | \end{document}
```

Blah blah blah,  
ich bin langer Text.  
Du du du, Ich füll  
die Seite jetzt!

# Ein Beispiel

```
1 | \documentclass{article}
2 |
3 | \usepackage[a4paper, left=10cm,
4 |   right=8cm]{geometry}
5 |
6 | \begin{document}
7 |   Blah blah blah, ich bin langer Text.
8 |   Du du du, Ich füll die Seite jetzt!
9 | \end{document}
```

Blah blah blah,  
ich bin langer Text.  
Du du du, Ich füll  
die Seite jetzt!

# Ein Beispiel

```
1 | \documentclass{article}
2 |
3 | \usepackage[a4paper, left=10cm,
4 |   right=8cm]{geometry}
5 |
6 | \begin{document}
7 |   Blah blah blah, ich bin langer Text.
8 |   Du du du, Ich füll die Seite jetzt!
9 | \end{document}
```

Blah blah blah,  
ich bin langer Text.  
Du du du, Ich füll  
die Seite jetzt!

- Hinweis: Hier wurde nur die Höhe der Seite beschnitten und die Seite skaliert.

# Das geometry-Paket

# Das geometry-Paket

- Margins erlauben Randkommentare.



# Das geometry-Paket

- Margins erlauben Randkommentare.
- Beispielhaft durch `\marginpar`.

# Das geometry-Paket

- Margins erlauben Randkommentare.
- Beispielhaft durch `\marginpar`.
- Aktiviert durch `includemp`.

# Das geometry-Paket

- Margins erlauben Randkommentare.
- Beispielhaft durch `\marginpar`.
- Aktiviert durch `includemp`. (`include marginp`)

# Das geometry-Paket

- Margins erlauben Randkommentare.
- Beispielhaft durch `\marginpar`.
- Aktiviert durch `includemp`. (`include marginp`)
- Breite durch `marginparwidth`, Abstand durch `marginparsep`.

# Das geometry-Paket

- Margins erlauben Randkommentare.
- Beispielhaft durch `\marginpar`.
- Aktiviert durch `includemp`. (`include marginp`)
- Breite durch `marginparwidth`, Abstand durch `marginparsep`.
- Kann auch für den zweiseitigen Druck angepasst werden.

# Das geometry-Paket

- Margins erlauben Randkommentare.
- Beispielhaft durch `\marginpar`.
- Aktiviert durch `includemp`. (`include marginp`)
- Breite durch `marginparwidth`, Abstand durch `marginparsep`.
- Kann auch für den zweiseitigen Druck angepasst werden. (Beachte hierfür auch das optionale Argument.)

# Das geometry-Paket

- Margins erlauben Randkommentare.
- Beispielhaft durch `\marginpar`.
- Aktiviert durch `includemp`. (`include marginp`)
- Breite durch `marginparwidth`, Abstand durch `marginparsep`.
- Kann auch für den zweiseitigen Druck angepasst werden. (Beachte hierfür auch das optionale Argument.)
- *Hinweis:* Im folgenden Dokument verwende ich `blindtext`. Es handelt sich um Platzhalter-Text

# Ein Beispiel



# Ein Beispiel

```
1 | \documentclass{article}
2 |
3 | \usepackage{blindtext}
4 | \usepackage[a4paper,
5 |     includemp]{geometry}
6 |
7 | \begin{document}
8 |     \marginpar{Hallo Welt, wie gehts?}
9 |     \blindtext
10 |     \marginpar{Sonnenschein}
11 | \end{document}
```

# Ein Beispiel

```
1 | \documentclass{article}
2 |
3 | \usepackage{blindtext}
4 | \usepackage[a4paper,
5 |   includemp]{geometry}
6 |
7 | \begin{document}
8 |   \marginpar{Hallo Welt, wie gehts?}
9 |   \blindtext
10 |   \marginpar{Sonnenschein}
11 | \end{document}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultricies augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Hallo Welt,  
wie gehts?

Sonnenschein

# Befehle in Latex

1. Abschnitte und Kapitelübersicht

2. Kopf- und Fußzeile

3. Geometrie der Seite

4. Befehle in Latex

- Exkurs: Wie funktioniert ein Befehl?
- `\newcommand` und `\renewcommand`
- Optionale Argumente
- Tipps und Tricks
- Umgebungen in Latex

5. Hyperlinks und Metadaten

6. Grafiken und Listen

# Exkurs: Wie funktioniert ein Befehl?

# Exkurs: Wie funktioniert ein Befehl?

- $\text{\LaTeX}$  unterteilt Zeichen in „aktive“ und „andere“.

# Exkurs: Wie funktioniert ein Befehl?

- $\text{\LaTeX}$  unterteilt Zeichen in „aktive“ und „andere“. (und noch andere aber egal 😊)

# Exkurs: Wie funktioniert ein Befehl?

- $\text{\LaTeX}$  unterteilt Zeichen in „aktive“ und „andere“. (und noch andere aber egal 😊)
- Liest es ein Backslash „`\`“,

# Exkurs: Wie funktioniert ein Befehl?

- $\text{\LaTeX}$  unterteilt Zeichen in „aktive“ und „andere“. (und noch andere aber egal 😊)
- Liest es ein Backslash „`\`“, so wechselt es in den Befehlsmodus und liest,



# Exkurs: Wie funktioniert ein Befehl?

- $\text{\LaTeX}$  unterteilt Zeichen in „aktive“ und „andere“. (und noch andere aber egal 😊)
- Liest es ein Backslash „`\`“, so wechselt es in den Befehlsmodus und liest, bis ein Zeichen kommt, das (Faustregel) kein Buchstabe ist.

# Exkurs: Wie funktioniert ein Befehl?

- $\text{\LaTeX}$  unterteilt Zeichen in „aktive“ und „andere“. (und noch andere aber egal 😊)
- Liest es ein Backslash „ $\backslash$ “, so wechselt es in den Befehlsmodus und liest, bis ein Zeichen kommt, das (Faustregel) kein Buchstabe ist.
- Danach sieht  $\text{\LaTeX}/\text{\TeX}$  nach, welche Signatur der Befehl besitzt.

# Exkurs: Wie funktioniert ein Befehl?

- $\text{\LaTeX}$  unterteilt Zeichen in „aktive“ und „andere“. (und noch andere aber egal 😊)
- Liest es ein Backslash „ $\backslash$ “, so wechselt es in den Befehlsmodus und liest, bis ein Zeichen kommt, das (Faustregel) kein Buchstabe ist.
- Danach sieht  $\text{\LaTeX}$ / $\text{\TeX}$  nach, welche Signatur der Befehl besitzt.
- Beispiel:

# Exkurs: Wie funktioniert ein Befehl?

- $\text{\LaTeX}$  unterteilt Zeichen in „aktive“ und „andere“. (und noch andere aber egal 😊)
- Liest es ein Backslash „ $\backslash$ “, so wechselt es in den Befehlsmodus und liest, bis ein Zeichen kommt, das (Faustregel) kein Buchstabe ist.
- Danach sieht  $\text{\LaTeX}/\text{\TeX}$  nach, welche Signatur der Befehl besitzt.
- Beispiel:

$\backslash\text{textit}$  {Hallo Welt}  
Befehl      Erstes Argument

## Exkurs: Wie funktioniert ein Befehl?

- $\text{\LaTeX}$  unterteilt Zeichen in „aktive“ und „andere“. (und noch andere aber egal 😊)
- Liest es ein Backslash „ $\backslash$ “, so wechselt es in den Befehlsmodus und liest, bis ein Zeichen kommt, das (Faustregel) kein Buchstabe ist.
- Danach sieht  $\text{\LaTeX}/\text{\TeX}$  nach, welche Signatur der Befehl besitzt.
- Beispiel:

```
\textit {Hallo Welt}
```

Befehl      Erstes Argument

- Liest Latex eine geschwungene Klammer (Gruppe genannt),

# Exkurs: Wie funktioniert ein Befehl?

- $\text{\LaTeX}$  unterteilt Zeichen in „aktive“ und „andere“. (und noch andere aber egal 😊)
- Liest es ein Backslash „ $\backslash$ “, so wechselt es in den Befehlsmodus und liest, bis ein Zeichen kommt, das (Faustregel) kein Buchstabe ist.
- Danach sieht  $\text{\LaTeX}/\text{\TeX}$  nach, welche Signatur der Befehl besitzt.
- Beispiel:

$\backslash\text{textit}\{\text{Hallo Welt}\}$   
Befehl      Erstes Argument

- Liest Latex eine geschwungene Klammer (Gruppe genannt), so wird bis zur schließenden Klammer gelesen.

# Exkurs: Wie funktioniert ein Befehl?

- $\text{\LaTeX}$  unterteilt Zeichen in „aktive“ und „andere“. (und noch andere aber egal 😊)
- Liest es ein Backslash „ $\backslash$ “, so wechselt es in den Befehlsmodus und liest, bis ein Zeichen kommt, das (Faustregel) kein Buchstabe ist.
- Danach sieht  $\text{\LaTeX}/\text{\TeX}$  nach, welche Signatur der Befehl besitzt.
- Beispiel:

$\backslash\text{textit}\{\text{Hallo Welt}\}$   
Befehl      Erstes Argument

- Liest Latex eine geschwungene Klammer (Gruppe genannt), so wird bis zur schließenden Klammer gelesen.
- Sonst wird das erste nicht-Leerfeld genommen.

# Exkurs: Wie funktioniert ein Befehl?

- $\text{\LaTeX}$  unterteilt Zeichen in „aktive“ und „andere“. (und noch andere aber egal 😊)
- Liest es ein Backslash „ $\backslash$ “, so wechselt es in den Befehlsmodus und liest, bis ein Zeichen kommt, das (Faustregel) kein Buchstabe ist.
- Danach sieht  $\text{\LaTeX}/\text{\TeX}$  nach, welche Signatur der Befehl besitzt.
- Beispiel:

$\underbrace{\text{\texttt{\textcolor{blue}{textit}}}}_{\text{Befehl}} \underbrace{\text{\texttt{\textcolor{blue}{}}} \text{Hallo Welt} \text{\texttt{\textcolor{blue}{}}}}_{\text{Erstes Argument}}$

- Liest Latex eine geschwungene Klammer (Gruppe genannt), so wird bis zur schließenden Klammer gelesen.
- Sonst wird das erste nicht-Leerfeld genommen.
- In Wirklichkeit noch aufwändiger 😊.



# Exkurs: Wie funktioniert ein Befehl?

# Exkurs: Wie funktioniert ein Befehl?

- $\text{T}_{\text{E}}\text{X}$  expandiert dann den Befehl.

# Exkurs: Wie funktioniert ein Befehl?

- $\text{\TeX}$  expandiert dann den Befehl.
- Das bedeutet, er wird durch seine Definition ersetzt.

# Exkurs: Wie funktioniert ein Befehl?

- $\text{T}_{\text{E}}\text{X}$  expandiert dann den Befehl.
- Das bedeutet, er wird durch seine Definition ersetzt.
- Es gibt  $\text{T}_{\text{E}}\text{X}$ -Befehle wie `\uppercase`, die nicht expandieren können!

# Exkurs: Wie funktioniert ein Befehl?

- $\text{T}_{\text{E}}\text{X}$  expandiert dann den Befehl.
- Das bedeutet, er wird durch seine Definition ersetzt.
- Es gibt  $\text{T}_{\text{E}}\text{X}$ -Befehle wie `\uppercase`, die nicht expandieren können!
- Genaueres gleich 😊.

# Einen Befehl definieren

# Einen Befehl definieren

- Normalerweise werden sie in der Präambel definiert.

# Einen Befehl definieren

- Normalerweise werden sie in der Präambel definiert. (Dies muss aber so nicht sein.)



# Einen Befehl definieren

- Normalerweise werden sie in der Präambel definiert. (Dies muss aber so nicht sein.)
- Hierfür gibt es `\newcommand` und `\renewcommand`.

# Einen Befehl definieren

- Normalerweise werden sie in der Präambel definiert. (Dies muss aber so nicht sein.)
- Hierfür gibt es `\newcommand` und `\renewcommand`. Letzterer wird dann verwendet, wenn ein Befehl verändert werden soll.

# Einen Befehl definieren

- Normalerweise werden sie in der Präambel definiert. (Dies muss aber so nicht sein.)
- Hierfür gibt es `\newcommand` und `\renewcommand`. Letzterer wird dann verwendet, wenn ein Befehl verändert werden soll.
- Wir definieren:

# Einen Befehl definieren

- Normalerweise werden sie in der Präambel definiert. (Dies muss aber so nicht sein.)
- Hierfür gibt es `\newcommand` und `\renewcommand`. Letzterer wird dann verwendet, wenn ein Befehl verändert werden soll.
- Wir definieren:

```
\newcommand{\Dieter}{Ich heiße Dieter}
```

# Einen Befehl definieren

- Normalerweise werden sie in der Präambel definiert. (Dies muss aber so nicht sein.)
- Hierfür gibt es `\newcommand` und `\renewcommand`. Letzterer wird dann verwendet, wenn ein Befehl verändert werden soll.
- Wir definieren:

```
\newcommand{\Dieter}{Ich heiße Dieter}
```

Also einen Befehl mit dem Namen „Dieter“, der zu „Ich heiße Dieter“ expandiert.

# Ein Beispiel

# Ein Beispiel

```
1 | \documentclass{article}
2 | \usepackage[T1]{fontenc}
3 | \usepackage[utf8]{inputenc}
4 |
5 | \newcommand{\Dieter}{Ich heie Dieter}
6 |
7 | \begin{document}
8 |   Hallo, ich bin \Dieter und mag das.
9 | \end{document}
```

# Ein Beispiel

```
1 | \documentclass{article}
2 | \usepackage[T1]{fontenc}
3 | \usepackage[utf8]{inputenc}
4 |
5 | \newcommand{\Dieter}{Ich heie Dieter}
6 |
7 | \begin{document}
8 |   Hallo, ich bin \Dieter und mag das.
9 | \end{document}
```

Hallo, ich bin Ich heie Dieter und mag das.



# Ein Beispiel

```
1 | \documentclass{article}
2 | \usepackage[T1]{fontenc}
3 | \usepackage[utf8]{inputenc}
4 |
5 | \newcommand{\Dieter}{Ich heie Dieter}
6 |
7 | \begin{document}
8 |   Hallo, ich bin \Dieter und mag das.
9 | \end{document}
```

Hallo, ich bin Ich heie Dieter und mag das.

# Ein Beispiel

```
1 | \documentclass{article}
2 | \usepackage[T1]{fontenc}
3 | \usepackage[utf8]{inputenc}
4 |
5 | \newcommand{\Dieter}{Ich heie Dieter}
6 |
7 | \begin{document}
8 |   Hallo, ich bin \Dieter und mag das.
9 | \end{document}
```

Hallo, ich bin Ich heie Dieter und mag das.

- Hinweis: Das Leerfeld wird verschluckt, da  $\text{\LaTeX}$  immer einmal nach Argumenten sucht!

# Ein Beispiel

```
1 | \documentclass{article}
2 | \usepackage[T1]{fontenc}
3 | \usepackage[utf8]{inputenc}
4 |
5 | \newcommand{\Dieter}{Ich heie Dieter}
6 |
7 | \begin{document}
8 |   Hallo, ich bin \Dieter und mag das.
9 | \end{document}
```

Hallo, ich bin Ich heie Dieter und mag das.

- Hinweis: Das Leerfeld wird verschluckt, da  $\text{\LaTeX}$  immer einmal nach Argumenten sucht!
- Wir schreiben deswegen: `...\Dieter{}` ...

# Befehle mit Argumenten

# Befehle mit Argumenten

- Wir möchten natürlich auch Befehle mit Argumenten. Hierzu:

# Befehle mit Argumenten

- Wir möchten natürlich auch Befehle mit Argumenten. Hierzu:

```
\newcommand{\< Name >} [< Anzahl >] {\< Rumpf >}
```

# Befehle mit Argumenten

- Wir möchten natürlich auch Befehle mit Argumenten. Hierzu:

```
\newcommand{\< Name >} [< Anzahl >] {< Rumpf >}
```

- Die Argumente werden mit #< Nummer > von eins an nummeriert.

# Befehle mit Argumenten

- Wir möchten natürlich auch Befehle mit Argumenten. Hierzu:

```
\newcommand{\< Name >}[\< Anzahl >]{\< Rumpf >}
```

- Die Argumente werden mit #< Nummer > von eins an nummeriert.
- Wir definieren:



# Befehle mit Argumenten

- Wir möchten natürlich auch Befehle mit Argumenten. Hierzu:

```
\newcommand{\< Name >}[\< Anzahl >]{\< Rumpf >}
```

- Die Argumente werden mit #< Nummer > von eins an nummeriert.
- Wir definieren:

```
\newcommand{\Dieter}[2]{%  
  {Ich heie #1, Ich bin \(<#2\> \textit{Jahre} alt.}}
```

# Befehle mit Argumenten

- Wir möchten natürlich auch Befehle mit Argumenten. Hierzu:

```
\newcommand{\< Name >}[< Anzahl >]{< Rumpf >}
```

- Die Argumente werden mit #< Nummer > von eins an nummeriert.
- Wir definieren:

```
\newcommand{\Dieter}[2]{%  
  {Ich heie #1, Ich bin \(<#2\> \textit{Jahre} alt.}}
```

- *Hinweis:* die neue Zeile ist nur zur Übersicht!

# Befehle mit Argumenten

- Wir möchten natürlich auch Befehle mit Argumenten. Hierzu:

```
\newcommand{\< Name >}[< Anzahl >]{< Rumpf >}
```

- Die Argumente werden mit #< Nummer > von eins an nummeriert.
- Wir definieren:

```
\newcommand{\Dieter}[2]%  
  {Ich heie #1, Ich bin \(<#2\> \textit{Jahre} alt.)}
```

- *Hinweis:* die neue Zeile ist nur zur Übersicht!
- Wir können die Befehle also auch verschachteln.

# Noch ein Beispiel

# Noch ein Beispiel

```
1 | \documentclass{article}
2 | \usepackage[T1]{fontenc}
3 | \usepackage[utf8]{inputenc}
4 |
5 | \newcommand{\Dieter}[2]%
6 |     {Ich heie #1, Ich bin \((#2\) ←
       \textit{Jahre} alt.}
7 |
8 | \begin{document}
9 |     Hallo, \Dieter{Wolfgang}{200} Das ←
       ist super!
10 | \end{document}
```

# Noch ein Beispiel

```
1 | \documentclass{article}
2 | \usepackage[T1]{fontenc}
3 | \usepackage[utf8]{inputenc}
4 |
5 | \newcommand{\Dieter}[2]%
6 |     {Ich heie #1, Ich bin \textit{Jahre} alt.}
7 |
8 | \begin{document}
9 |     Hallo, \Dieter{Wolfgang}{200} Das
10 |         ist super!
    \end{document}
```

Hallo, Ich heie Wolfgang, Ich bin 200 *Jahre* alt. Das ist super!

# Ein (sinnvolleres) Beispiel

# Ein (sinnvolleres) Beispiel

```
1 | \documentclass{article}
2 |
3 | \usepackage{xcolor}
4 | \newcommand{\borange}[1]{\textcolor{↵
   |     orange}{(\textcolor{orange}{})}}
5 |
6 | \begin{document}
7 |     Hallo, ich bin \borange{wunderbar} ↵
   |     und mag das.
8 | \end{document}
```



# Ein (sinnvolleres) Beispiel

```
1 | \documentclass{article}
2 |
3 | \usepackage{xcolor}
4 | \newcommand{\borange}[1]{\textcolor{↵
    orange}{(#1\textcolor{orange}{})}}
5 |
6 | \begin{document}
7 |   Hallo, ich bin \borange{wunderbar} ↵
    und mag das.
8 | \end{document}
```

Hallo, ich bin (wunderbar) und mag das.

# Ein (sinnvolleres) Beispiel

```
1 | \documentclass{article}
2 |
3 | \usepackage{xcolor}
4 | \newcommand{\borange}[1]{\textcolor{↵
    orange}{(#1\textcolor{orange}{})}}
5 |
6 | \begin{document}
7 |   Hallo, ich bin \borange{wunderbar} ↵
    und mag das.
8 | \end{document}
```

Hallo, ich bin (wunderbar) und mag das.

# Ein (sinnvolleres) Beispiel

```
1 | \documentclass{article}
2 |
3 | \usepackage{xcolor}
4 | \newcommand{\borange}[1]{\textcolor{↵
   |     orange}{(#1\textcolor{orange}{})}}
5 |
6 | \begin{document}
7 |     Hallo, ich bin \borange{wunderbar} ↵
   |     und mag das.
8 | \end{document}
```

Hallo, ich bin (wunderbar) und mag das.

- Erspart schon einmal Schreibarbeit 😊.

# Beispiel zu Verschachtelungen

# Beispiel zu Verschachtelungen

```
1 | \documentclass{article}
2 | \usepackage{xcolor}
3 | \newcommand{\bcolor}[2]{\textcolor{#2}
   |   {}#1\textcolor{#2}{}}}
4 |
5 | \newcommand{\borange}[1]
6 |   {\bcolor{#1}{orange}}
7 | \newcommand{\bteal}[1]
8 |   {\bcolor{#1}{teal}}
9 | \newcommand{\bred}[1]{\bcolor{#1}{red}}
10 |
11 | \begin{document}
12 |   Hallo, ich \bteal{bin \borange{
   |     wunderbar} und} mag das.
13 | \end{document}
```

# Beispiel zu Verschachtelungen

```
1 | \documentclass{article}
2 | \usepackage{xcolor}
3 | \newcommand{\bcolor}[2]{\textcolor{#2}
   |   {}#1\textcolor{#2}{}}}
4 |
5 | \newcommand{\borange}[1]
6 |   {\bcolor{#1}{orange}}
7 | \newcommand{\bteal}[1]
8 |   {\bcolor{#1}{teal}}
9 | \newcommand{\bred}[1]{\bcolor{#1}{red}}
10 |
11 | \begin{document}
12 |   Hallo, ich \bteal{bin \borange{
   |     wunderbar} und} mag das.
13 | \end{document}
```

Hallo, ich (bin (wunderbar) und) mag das.

# Beispiel zu Verschachtelungen

```
1 | \documentclass{article}
2 | \usepackage{xcolor}
3 | \newcommand{\bcolor}[2]{\textcolor{#2}
   |   {}#1\textcolor{#2}{}}}
4 |
5 | \newcommand{\borange}[1]
6 |   {\bcolor{#1}{orange}}
7 | \newcommand{\bteal}[1]
8 |   {\bcolor{#1}{teal}}
9 | \newcommand{\bred}[1]{\bcolor{#1}{red}}
10 |
11 | \begin{document}
12 |   Hallo, ich \bteal{bin \borange{
   |     wunderbar} und} mag das.
13 | \end{document}
```

Hallo, ich (bin (wunderbar) und) mag das.

# Beispiel zu Verschachtelungen

```
1 | \documentclass{article}
2 | \usepackage{xcolor}
3 | \newcommand{\bcolor}[2]{\textcolor{#2}
   |   {}(#1\textcolor{#2}{})}}
4 |
5 | \newcommand{\borange}[1]
6 |   {\bcolor{#1}{orange}}
7 | \newcommand{\bteal}[1]
8 |   {\bcolor{#1}{teal}}
9 | \newcommand{\bred}[1]{\bcolor{#1}{red}}
10 |
11 | \begin{document}
12 |   Hallo, ich \bteal{bin \borange{
   |     wunderbar} und} mag das.
13 | \end{document}
```

Hallo, ich (bin (wunderbar) und) mag das.

- Solangsam wird die Faulheit eines Programmiers deutlich 😊.



# Optionale Argumente

# Optionale Argumente

- Manchmal muss ein Befehl nur manchmal ein Argument erhalten.

# Optionale Argumente

- Manchmal muss ein Befehl nur manchmal ein Argument erhalten.
- Optionale Argumente werden in Latex in der Regel mittels eckiger Klammern eingeschlossen.

# Optionale Argumente

- Manchmal muss ein Befehl nur manchmal ein Argument erhalten.
- Optionale Argumente werden in Latex in der Regel mittels eckiger Klammern eingeschlossen.
- Bei `\newcommand` setzen wir hierzu einfach nochmal eine eckige Klammer dahinter

# Optionale Argumente

- Manchmal muss ein Befehl nur manchmal ein Argument erhalten.
- Optionale Argumente werden in Latex in der Regel mittels eckiger Klammern eingeschlossen.
- Bei `\newcommand` setzen wir hierzu einfach nochmal eine eckige Klammer dahinter und schreiben dort hin, was das erste Argument sein soll,

# Optionale Argumente

- Manchmal muss ein Befehl nur manchmal ein Argument erhalten.
- Optionale Argumente werden in Latex in der Regel mittels eckiger Klammern eingeschlossen.
- Bei `\newcommand` setzen wir hierzu einfach nochmal eine eckige Klammer dahinter und schreiben dort hin, was das erste Argument sein soll, wenn es nicht angegeben wird.

# Optionale Argumente

- Manchmal muss ein Befehl nur manchmal ein Argument erhalten.
- Optionale Argumente werden in Latex in der Regel mittels eckiger Klammern eingeschlossen.
- Bei `\newcommand` setzen wir hierzu einfach nochmal eine eckige Klammer dahinter und schreiben dort hin, was das erste Argument sein soll, wenn es nicht angegeben wird.
- Beispiel:

# Optionale Argumente

- Manchmal muss ein Befehl nur manchmal ein Argument erhalten.
- Optionale Argumente werden in Latex in der Regel mittels eckiger Klammern eingeschlossen.
- Bei `\newcommand` setzen wir hierzu einfach nochmal eine eckige Klammer dahinter und schreiben dort hin, was das erste Argument sein soll, wenn es nicht angegeben wird.
- Beispiel:

```
\newcommand{\Dieter}[3][Dieter]{%  
  {Ich heie #1, bin #2 Jahre alt und #3.}}
```



# Beispiel zu optionalen Argumenten

# Beispiel zu optionalen Argumenten

```
1 | \documentclass{article}
2 | \usepackage[T1]{fontenc}
3 | \usepackage[utf8]{inputenc}
4 |
5 | \newcommand{\Dieter}[3][Dieter]{Ich hei-
   | ße #1, bin #2 Jahre alt und #3.}
6 |
7 | \begin{document}
8 |   Hallo, \Dieter{32}{cool} \\
9 |   Hallo, \Dieter[Jens]{5}{eine Ente}
10 | \end{document}
```

# Beispiel zu optionalen Argumenten

```
1 | \documentclass{article}
2 | \usepackage[T1]{fontenc}
3 | \usepackage[utf8]{inputenc}
4 |
5 | \newcommand{\Dieter}[3][Dieter]{Ich hei-
   | ße #1, bin #2 Jahre alt und #3.}
6 |
7 | \begin{document}
8 |   Hallo, \Dieter{32}{cool} \\
9 |   Hallo, \Dieter[Jens]{5}{eine Ente}
10 | \end{document}
```

Hallo, Ich heiße Dieter, bin 32 Jahre alt und cool.  
Hallo, Ich heiße Jens, bin 5 Jahre alt und eine Ente.

# Beispiel zu optionalen Argumenten

```
1 | \documentclass{article}
2 | \usepackage[T1]{fontenc}
3 | \usepackage[utf8]{inputenc}
4 |
5 | \newcommand{\Dieter}[3][Dieter]{Ich hei-
   | ße #1, bin #2 Jahre alt und #3.}
6 |
7 | \begin{document}
8 |   Hallo, \Dieter{32}{cool} \\
9 |   Hallo, \Dieter[Jens]{5}{eine Ente}
10 | \end{document}
```

Hallo, Ich heiße Dieter, bin 32 Jahre alt und cool.  
Hallo, Ich heiße Jens, bin 5 Jahre alt und eine Ente.

# Beispiel zu optionalen Argumenten

```
1 | \documentclass{article}
2 | \usepackage[T1]{fontenc}
3 | \usepackage[utf8]{inputenc}
4 |
5 | \newcommand{\Dieter}[3][Dieter]{Ich hei-
   | ße #1, bin #2 Jahre alt und #3.}
6 |
7 | \begin{document}
8 |   Hallo, \Dieter{32}{cool} \\
9 |   Hallo, \Dieter[Jens]{5}{eine Ente}
10 | \end{document}
```

Hallo, Ich heiße Dieter, bin 32 Jahre alt und cool.  
Hallo, Ich heiße Jens, bin 5 Jahre alt und eine Ente.

- So funktioniert zum Beispiel `\usepackage`.

# Ein paar Tipps

# Ein paar Tipps

- Befehle, deren Argumente nur aus wenigen Worten oder gar einem Zeichen bestehen,

# Ein paar Tipps

- Befehle, deren Argumente nur aus wenigen Worten oder gar einem Zeichen bestehen, (oder gar keine Befehle wollen),



# Ein paar Tipps

- Befehle, deren Argumente nur aus wenigen Worten oder gar einem Zeichen bestehen, (oder gar keine Befehle wollen), sollten mit einem Sternchen definiert werden. Also: `\newcommand*` beziehungsweise `\renewcommand*`.

# Ein paar Tipps

- Befehle, deren Argumente nur aus wenigen Worten oder gar einem Zeichen bestehen, (oder gar keine Befehle wollen), sollten mit einem Sternchen definiert werden. Also: `\newcommand*` beziehungsweise `\renewcommand*`. Das hilft bei der Fehlersuche!

# Ein paar Tipps

- Befehle, deren Argumente nur aus wenigen Worten oder gar einem Zeichen bestehen, (oder gar keine Befehle wollen), sollten mit einem Sternchen definiert werden. Also: `\newcommand*` beziehungsweise `\renewcommand*`. Das hilft bei der Fehlersuche! (Stichwort: `\long`)

# Ein paar Tipps

- Befehle, deren Argumente nur aus wenigen Worten oder gar einem Zeichen bestehen, (oder gar keine Befehle wollen), sollten mit einem Sternchen definiert werden. Also: `\newcommand*` beziehungsweise `\renewcommand*`. Das hilft bei der Fehlersuche! (Stichwort: `\long`)
- Man muss es mit Befehlen nicht übertreiben!

# Ein paar Tipps

- Befehle, deren Argumente nur aus wenigen Worten oder gar einem Zeichen bestehen, (oder gar keine Befehle wollen), sollten mit einem Sternchen definiert werden. Also: `\newcommand*` beziehungsweise `\renewcommand*`. Das hilft bei der Fehlersuche! (Stichwort: `\long`)
- Man muss es mit Befehlen nicht übertreiben! Für die meisten nützlichen Dinge gibt es bereits ein Paket.

# Ein paar Tipps

- Befehle, deren Argumente nur aus wenigen Worten oder gar einem Zeichen bestehen, (oder gar keine Befehle wollen), sollten mit einem Sternchen definiert werden. Also: `\newcommand*` beziehungsweise `\renewcommand*`. Das hilft bei der Fehlersuche! (Stichwort: `\long`)
- Man muss es mit Befehlen nicht übertreiben! Für die meisten nützlichen Dinge gibt es bereits ein Paket.
- Für viele Fälle gibt es vermutlich auch noch andere Mechaniken...

# Definieren einer Umgebung

# Definieren einer Umgebung

- `\newenvironment` funktioniert analog zu `\newcommand`,



# Definieren einer Umgebung

- `\newenvironment` funktioniert analog zu `\newcommand`, `\renewenvironment` analog zu `\renewcommand`.

# Definieren einer Umgebung

- `\newenvironment` funktioniert analog zu `\newcommand`, `\renewenvironment` analog zu `\renewcommand`.
- Die Argumentdefinition verläuft analog.

# Definieren einer Umgebung

- `\newenvironment` funktioniert analog zu `\newcommand`, `\renewenvironment` analog zu `\renewcommand`.
- Die Argumentdefinition verläuft analog.
- Einziger Unterschied:

# Definieren einer Umgebung

- `\newenvironment` funktioniert analog zu `\newcommand`, `\renewenvironment` analog zu `\renewcommand`.
- Die Argumentdefinition verläuft analog.
- Einziger Unterschied: Rumpf gliedert sich in zwei Teile,

# Definieren einer Umgebung

- `\newenvironment` funktioniert analog zu `\newcommand`, `\renewenvironment` analog zu `\renewcommand`.
- Die Argumentdefinition verläuft analog.
- Einziger Unterschied: Rumpf gliedert sich in zwei Teile, „am Anfang“ und „am Ende“.

# Definieren einer Umgebung

- `\newenvironment` funktioniert analog zu `\newcommand`, `\renewenvironment` analog zu `\renewcommand`.
- Die Argumentdefinition verläuft analog.
- Einziger Unterschied: Rumpf gliedert sich in zwei Teile, „am Anfang“ und „am Ende“.
- *Und*: Die Definition wird ohne Backslash angeführt!

# Definieren einer Umgebung

- `\newenvironment` funktioniert analog zu `\newcommand`, `\renewenvironment` analog zu `\renewcommand`.
- Die Argumentdefinition verläuft analog.
- Einziger Unterschied: Rumpf gliedert sich in zwei Teile, „am Anfang“ und „am Ende“.
- *Und*: Die Definition wird ohne Backslash angeführt!
- Beispiel:

# Definieren einer Umgebung

- `\newenvironment` funktioniert analog zu `\newcommand`, `\renewenvironment` analog zu `\renewcommand`.
- Die Argumentdefinition verläuft analog.
- Einziger Unterschied: Rumpf gliedert sich in zwei Teile, „am Anfang“ und „am Ende“.
- *Und*: Die Definition wird ohne Backslash angeführt!
- Beispiel:

```
\newenvironment{episch}[1][Titel:]{%  
  {\textit{#1}}\}% Start  
  {Das wars!}% Ende
```



# Beispiel zu Umgebungen

# Beispiel zu Umgebungen

```
1 | \documentclass{article}
2 |
3 | \newenvironment{episch}[1][Titel:]{%
4 |     {\textit{#1}}\}% Start
5 |     {Das wars!}% Ende
6 |
7 | \begin{document}
8 |     \begin{episch}
9 |         Ich bin eine Umgebung!
10 |    \end{episch} \\
11 |    \begin{episch}[Sonnenschein]
12 |        Bin ich nicht toll?
13 |    \end{episch}
14 | \end{document}
```

# Beispiel zu Umgebungen

```
1 | \documentclass{article}
2 |
3 | \newenvironment{episch}[1][Titel:]{%
4 |     {\textit{#1}}\}% Start
5 |     {Das wars!}% Ende
6 |
7 | \begin{document}
8 |     \begin{episch}
9 |         Ich bin eine Umgebung!
10 |    \end{episch} \\
11 |    \begin{episch}[Sonnenschein]
12 |        Bin ich nicht toll?
13 |    \end{episch}
14 | \end{document}
```

*Titel:*  
Ich bin eine Umgebung! Das wars!  
*Sonnenschein*  
Bin ich nicht toll? Das wars!

# Hyperlinks und Metadaten

1. Abschnitte und Kapitelübersicht

2. Kopf- und Fußzeile

3. Geometrie der Seite

4. Befehle in Latex

5. Hyperlinks und Metadaten

- Ich linke also bin ich!
- Farbige Konfiguration
- Metadaten setzen

6. Grafiken und Listen

# Anker und Sprünge

# Anker und Sprünge

- Ein Paket für alles: `hyperref`.

# Anker und Sprünge

- Ein Paket für alles: `hyperref`. Das bloße einbinden macht den `\tableofcontents` anklickbar.

# Anker und Sprünge

- Ein Paket für alles: `hyperref`. Das bloße einbinden macht den `\tableofcontents` anklickbar.
- Im Dokument: `\label{\< AnkerName >}` um einen Sprunganker/Sprungmarker zu setzen.



# Anker und Sprünge

- Ein Paket für alles: `hyperref`. Das bloße einbinden macht den `\tableofcontents` anklickbar.
- Im Dokument: `\label{< AnkerName >}` um einen Sprunganker/Sprungmarker zu setzen.
- Dann `\hyperref[< AnkerName >]{< Text >}`.

# Anker und Sprünge

- Ein Paket für alles: `hyperref`. Das bloße einbinden macht den `\tableofcontents` anklickbar.
- Im Dokument: `\label{< AnkerName >}` um einen Sprunganker/Sprungmarker zu setzen.
- Dann `\hyperref[< AnkerName >]{< Text >}`. Dies macht „Text“ anklickbar.

# Anker und Sprünge

- Ein Paket für alles: `hyperref`. Das bloße einbinden macht den `\tableofcontents` anklickbar.
- Im Dokument: `\label{< AnkerName >}` um einen Sprunganker/Sprungmarker zu setzen.
- Dann `\hyperref[< AnkerName >]{< Text >}`. Dies macht „Text“ anklickbar. Springt dann zu `< AnkerName >`.

# Anker und Sprünge

- Ein Paket für alles: `hyperref`. Das bloße einbinden macht den `\tableofcontents` anklickbar.
- Im Dokument: `\label{< AnkerName >}` um einen Sprunganker/Sprungmarker zu setzen.
- Dann `\hyperref[< AnkerName >]{< Text >}`. Dies macht „Text“ anklickbar. Springt dann zu `< AnkerName >`.
- Es gibt auch noch `\pageref{< AnkerName >}` dies expandiert zur Seitenzahl des Ziels und

# Anker und Sprünge

- Ein Paket für alles: `hyperref`. Das bloße einbinden macht den `\tableofcontents` anklickbar.
- Im Dokument: `\label{< AnkerName >}` um einen Sprunganker/Sprungmarker zu setzen.
- Dann `\hyperref[< AnkerName >]{< Text >}`. Dies macht „Text“ anklickbar. Springt dann zu `< AnkerName >`.
- Es gibt auch noch `\pageref{< AnkerName >}` dies expandiert zur Seitenzahl des Ziels und `\autoref{< AnkerName >}` versucht „sinnvoll“ Bezeichner dazu zu setzen.

# Anker und Sprünge

- Ein Paket für alles: `hyperref`. Das bloße einbinden macht den `\tableofcontents` anklickbar.
- Im Dokument: `\label{< AnkerName >}` um einen Sprunganker/Sprungmarker zu setzen.
- Dann `\hyperref[< AnkerName >]{< Text >}`. Dies macht „Text“ anklickbar. Springt dann zu `< AnkerName >`.
- Es gibt auch noch `\pageref{< AnkerName >}` dies expandiert zur Seitenzahl des Ziels und `\autoref{< AnkerName >}` versucht „sinnvoll“ Bezeichner dazu zu setzen. (Auch einen Blick wert: `\ref`)

# Anker und Sprünge

- Ein Paket für alles: `hyperref`. Das bloße einbinden macht den `\tableofcontents` anklickbar.
- Im Dokument: `\label{< AnkerName >}` um einen Sprunganker/Sprungmarker zu setzen.
- Dann `\hyperref[< AnkerName >]{< Text >}`. Dies macht „Text“ anklickbar. Springt dann zu `< AnkerName >`.
- Es gibt auch noch `\pageref{< AnkerName >}` dies expandiert zur Seitenzahl des Ziels und `\autoref{< AnkerName >}` versucht „sinnvoll“ Bezeichner dazu zu setzen. (Auch einen Blick wert: `\ref`)
- Es gibt viele weitere Pakete, die die Funktionalität erweitern.

# Anker und Sprünge

- Ein Paket für alles: `hyperref`. Das bloße einbinden macht den `\tableofcontents` anklickbar.
- Im Dokument: `\label{< AnkerName >}` um einen Sprunganker/Sprungmarker zu setzen.
- Dann `\hyperref[< AnkerName >]{< Text >}`. Dies macht „Text“ anklickbar. Springt dann zu `< AnkerName >`.
- Es gibt auch noch `\pageref{< AnkerName >}` dies expandiert zur Seitenzahl des Ziels und `\autoref{< AnkerName >}` versucht „sinnvoll“ Bezeichner dazu zu setzen. (Auch einen Blick wert: `\ref`)
- Es gibt viele weitere Pakete, die die Funktionalität erweitern.
- Sollte das letzte Paket sein, das geladen wird (überschreibt andere Befehle).



# Beispiel zu Links

# Beispiel zu Links

```
1 | \documentclass{article}
2 | \usepackage{hyperref}
3 |
4 | \begin{document}
5 |     \label{txt:sun}Hallo Welt ich mag ↵
        die Sonne!\\
6 |     \hyperref[txt:sun]{Wirklich sehr!}
7 |     (\pageref{txt:sun})
8 | \end{document}
```

# Beispiel zu Links

```
1 | \documentclass{article}
2 | \usepackage{hyperref}
3 |
4 | \begin{document}
5 |     \label{txt:sun}Hallo Welt ich mag ↵
        die Sonne!\\
6 |     \hyperref[txt:sun]{Wirklich sehr!}
7 |     (\pageref{txt:sun})
8 | \end{document}
```

Hallo Welt ich mag die Sonne!  
Wirklich sehr! (1)

# Beispiel zu Links

```
1 | \documentclass{article}
2 | \usepackage{hyperref}
3 |
4 | \begin{document}
5 |     \label{txt:sun}Hallo Welt ich mag ↵
        die Sonne!\\
6 |     \hyperref[txt:sun]{Wirklich sehr!}
7 |     (\pageref{txt:sun})
8 | \end{document}
```

Hallo Welt ich mag die Sonne!  
Wirklich sehr! (1)

# Beispiel zu Links

```
1 | \documentclass{article}
2 | \usepackage{hyperref}
3 |
4 | \begin{document}
5 |     \label{txt:sun}Hallo Welt ich mag ↵
        die Sonne!\\
6 |     \hyperref[txt:sun]{Wirklich sehr!}
7 |     (\pageref{txt:sun})
8 | \end{document}
```

Hallo Welt ich mag die Sonne!  
Wirklich sehr! (1)

- Es empfiehlt sich (wie `txt:`) die Anker thematisch zu kategorisieren.

# Beispiel zu Links

```
1 | \documentclass{article}
2 | \usepackage{hyperref}
3 |
4 | \begin{document}
5 |     \label{txt:sun}Hallo Welt ich mag ↵
        die Sonne!\\
6 |     \hyperref[txt:sun]{Wirklich sehr!}
7 |     (\pageref{txt:sun})
8 | \end{document}
```

Hallo Welt ich mag die Sonne!  
Wirklich sehr! (1)

- Es empfiehlt sich (wie `txt:`) die Anker thematisch zu kategorisieren.
- *Hinweis:* Hier nicht optisch dargestellt, da es standardmäßig am PDF-Viewer liegt.

# Ein Kessel buntes

# Ein Kessel bunt

- Alle Optionen mittels `\hypersetup`.



# Ein Kessel bunt

- Alle Optionen mittels `\hypersetup`.
- Die wichtigsten: `colorlinks`, `breaklinks`, `linkcolor`, `citecolor`, `filecolor` und `urlcolor`.

# Ein Kessel bunt

- Alle Optionen mittels `\hypersetup`.
- Die wichtigsten: `colorlinks`, `breaklinks`, `linkcolor`, `citecolor`, `filecolor` und `urlcolor`.
- Man kann auch URLs mittels `\url` angeben, und vieles mehr 😊.

# Ein Kessel bunt

- Alle Optionen mittels `\hypersetup`.
- Die wichtigsten: `colorlinks`, `breaklinks`, `linkcolor`, `citecolor`, `filecolor` und `urlcolor`.
- Man kann auch URLs mittels `\url` angeben, und vieles mehr 😊.
- *Hinweis: `colorlinks` ist standardmäßig aktiv, im Kontext dieser Präsentation aber von mir abgeschaltet.*

# Beispiel zu farbigen Links

# Beispiel zu farbigen Links

```
1 | \documentclass{article}
2 | \usepackage{xcolor}
3 | \usepackage{hyperref}
4 |
5 | \hypersetup{colorlinks=true,
6 |   linkcolor=teal,urlcolor=orange}
7 |
8 | \begin{document}
9 |   \label{txt:sun}Hallo Welt\\
10 |   \hyperref[txt:sun]{Wirklich sehr!}
11 |   (\pageref{txt:sun})
12 |   \url{https://moodle.uni-ulm.de}
13 | \end{document}
```

# Beispiel zu farbigen Links

```
1 | \documentclass{article}
2 | \usepackage{xcolor}
3 | \usepackage{hyperref}
4 |
5 | \hypersetup{colorlinks=true,
6 |   linkcolor=teal,urlcolor=orange}
7 |
8 | \begin{document}
9 |   \label{txt:sun}Hallo Welt\\
10 |   \hyperref[txt:sun]{Wirklich sehr!}
11 |   (\pageref{txt:sun})
12 |   \url{https://moodle.uni-ulm.de}
13 | \end{document}
```

Hallo Welt  
Wirklich sehr! (1) <https://moodle.uni-ulm.de>

# Na klar ist das von mir

# Na klar ist das von mir

- Alle Optionen wieder mittels `\hypersetup`.



# Na klar ist das von mir

- Alle Optionen wieder mittels `\hypersetup`.
- Wir können weiter setzen: `pdftitle`, `pdfauthor`, `pdfsubject`, `pdfcreator` und `pdflang`.

# Na klar ist das von mir

- Alle Optionen wieder mittels `\hypersetup`.
- Wir können weiter setzen: `pdftitle`, `pdfauthor`, `pdfsubject`, `pdfcreator` und `pdflang`.
- Dies setzt dann die Metadaten des Dokuments.

# Na klar ist das von mir

- Alle Optionen wieder mittels `\hypersetup`.
- Wir können weiter setzen: `pdftitle`, `pdfauthor`, `pdfsubject`, `pdfcreator` und `pdflang`.
- Dies setzt dann die Metadaten des Dokuments.
- Beispiel:

# Na klar ist das von mir

- Alle Optionen wieder mittels `\hypersetup`.
- Wir können weiter setzen: `pdftitle`, `pdfauthor`, `pdfsubject`, `pdfcreator` und `pdflang`.
- Dies setzt dann die Metadaten des Dokuments.
- Beispiel:

```
\hypersetup{pdftitle=Geschichten von Gestern, pdfauthor=Ich, ↵  
             pdfcreator=pdflatex}
```

# Na klar ist das von mir

- Beispiel:

```
\hypersetup{pdftitle=Geschichten von Gestern, pdfauthor=Ich, ↵  
pdfcreator=pdflatex}
```

# Na klar ist das von mir

- Beispiel:

```
\hypersetup{pdftitle=Geschichten von Gestern, pdfauthor=Ich, ↵  
pdfcreator=pdflatex}
```

- Ergibt:

# Na klar ist das von mir

- Beispiel:

```
\hypersetup{pdftitle=Geschichten von Gestern, pdfauthor=Ich, ↵  
pdfcreator=pdflatex}
```

- Ergibt:

# Na klar ist das von mir

- Beispiel:

```
\hypersetup{pdftitle=Geschichten von Gestern, pdfauthor=Ich, ↵  
pdfcreator=pdflatex}
```

- Ergibt: (Oh mein Gott ein Foto!!!!!!)



# Na klar ist das von mir

- Beispiel:

```
\hypersetup{pdftitle=Geschichten von Gestern, pdfauthor=Ich, ↵  
pdfcreator=pdflatex}
```

- Ergibt: (Oh mein Gott ein Foto!!!!!!)

Seitengröße: 216 × 279 mm (Hochformat Letter / ANSI A)

Dateigröße: 26,1 KiB

Titel: Geschichten von Gestern

Autor: Ich

Erstellt von: pdflatex

Ersteller: pdfTeX-1.40.19

Seiten: 1

Erstellt am: Mittwoch, 20. November 2019 22:09:02 UTC

Geändert am: Mittwoch, 20. November 2019 22:09:02 UTC

# Grafiken und Listen

1. Abschnitte und Kapitelübersicht
2. Kopf- und Fußzeile
3. Geometrie der Seite
4. Befehle in Latex
5. Hyperlinks und Metadaten
6. **Grafiken und Listen**
  - Grafiken einbinden
  - Untertitel für Bilder (und Tabellen)
  - Grafiken und Tabellen auflisten

# Ist es nicht wunderschön?

# Ist es nicht wunderschön?

- Wichtiges Paket: `graphicx`.

# Ist es nicht wunderschön?

- Wichtiges Paket: `graphicx`.
- Einzig wichtiger Befehl: `\includegraphics{<Datei>}`

# Ist es nicht wunderschön?

- Wichtiges Paket: `graphicx`.
- Einzig wichtiger Befehl: `\includegraphics{<Datei>}`
- Kann Optionen wie `width` und `height` annehmen:

# Ist es nicht wunderschön?

- Wichtiges Paket: `graphicx`.
- Einzig wichtiger Befehl: `\includegraphics{<Datei>}`
- Kann Optionen wie `width` und `height` annehmen:  
`\includegraphics[<Optionen>]{<Datei>}`

# Ist es nicht wunderschön?

- Wichtiges Paket: `graphicx`.
- Einzig wichtiger Befehl: `\includegraphics{<Datei>}`
- Kann Optionen wie `width` und `height` annehmen:  
`\includegraphics[<Optionen>]{<Datei>}`
- Die Datei kann ohne Endung angegeben werden.



# Ist es nicht wunderschön?

- Wichtiges Paket: `graphicx`.
- Einzig wichtiger Befehl: `\includegraphics{<Datei>}`
- Kann Optionen wie `width` und `height` annehmen:  
`\includegraphics[<Optionen>]{<Datei>}`
- Die Datei kann ohne Endung angegeben werden. (Die Angabe kann manchmal zu Problemen führen!)

# Beispiel mit Bildern

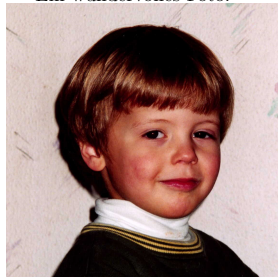
# Beispiel mit Bildern

```
1 | \documentclass{article}
2 |
3 | \usepackage{graphicx}
4 |
5 | \begin{document}
6 |     Ein wundervolles Foto:\\
7 |     \includegraphics[width=5cm]{me}
8 | \end{document}
```

# Beispiel mit Bildern

```
1 | \documentclass{article}
2 |
3 | \usepackage{graphicx}
4 |
5 | \begin{document}
6 |     Ein wundervolles Foto:\\
7 |     \includegraphics[width=5cm]{me}
8 | \end{document}
```

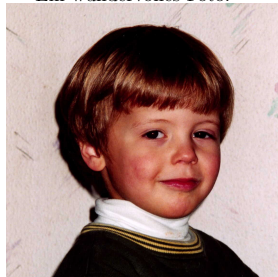
Ein wundervolles Foto:



# Beispiel mit Bildern

```
1 | \documentclass{article}
2 |
3 | \usepackage{graphicx}
4 |
5 | \begin{document}
6 |     Ein wundervolles Foto:\\
7 |     \includegraphics[width=5cm]{me}
8 | \end{document}
```

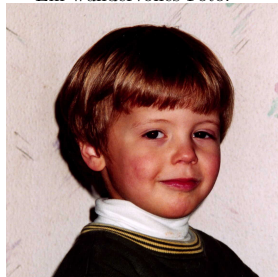
Ein wundervolles Foto:



# Beispiel mit Bildern

```
1 | \documentclass{article}
2 |
3 | \usepackage{graphicx}
4 |
5 | \begin{document}
6 |     Ein wundervolles Foto:\\
7 |     \includegraphics[width=5cm]{me}
8 | \end{document}
```

Ein wundervolles Foto:

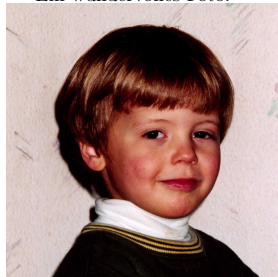


- Nützlich: `\linewidth` entspricht der Breite einer Zeile.

# Beispiel mit Bildern

```
1 | \documentclass{article}
2 |
3 | \usepackage{graphicx}
4 |
5 | \begin{document}
6 |     Ein wundervolles Foto:\\
7 |     \includegraphics[width=5cm]{me}
8 | \end{document}
```

Ein wundervolles Foto:



- Nützlich: `\linewidth` entspricht der Breite einer Zeile.
- So kann mit `0.5\linewidth` die Breite einer halben Zeile definiert werden.

# Untertitel - Was soll das sein?



# Untertitel - Was soll das sein?

- Wichtige Umgebung: *figure*.

# Untertitel - Was soll das sein?

- Wichtige Umgebung: *figure*. (Komplett analog: *table* für Tabellen)

# Untertitel - Was soll das sein?

- Wichtige Umgebung: *figure*. (Komplett analog: *table* für Tabellen)
- Erlaubt innerhalb mittels `\caption` Benennung der Grafik.

## Untertitel - Was soll das sein?

- Wichtige Umgebung: *figure*. (Komplett analog: *table* für Tabellen)
- Erlaubt innerhalb mittels `\caption` Benennung der Grafik.
- Optionales Argument, erlaubt Positionierungs-Wünsche (`t` für *top*, `b` für *bottom*, `h` für *here*, ...).

# Untertitel - Was soll das sein?

- Wichtige Umgebung: *figure*. (Komplett analog: *table* für Tabellen)
- Erlaubt innerhalb mittels `\caption` Benennung der Grafik.
- Optionales Argument, erlaubt Positionierungs-Wünsche (`t` für *top*, `b` für *bottom*, `h` für *here*, ...). Auch mehrere Angaben möglich (und üblich).

# Untertitel - Was soll das sein?

- Wichtige Umgebung: *figure*. (Komplett analog: *table* für Tabellen)
- Erlaubt innerhalb mittels `\caption` Benennung der Grafik.
- Optionales Argument, erlaubt Positionierungs-Wünsche (`t` für *top*, `b` für *bottom*, `h` für *here*, ...). Auch mehrere Angaben möglich (und üblich).
- Man kann zu Beginn `\centering` verwenden um die Grafik zu zentrieren.

# Beispiel mit Bildunterschrift

# Beispiel mit Bildunterschrift

```
1 | \documentclass{article}
2 |
3 | \usepackage{graphicx}
4 |
5 | \begin{document}
6 |     Ein wundervolles Foto:
7 |     \begin{figure}[tbh]
8 |         \centering
9 |         \includegraphics[width=5cm]{me}
10 |        \caption{Das bin ich.}
11 |    \end{figure}
12 | \end{document}
```



# Beispiel mit Bildunterschrift

```
1 | \documentclass{article}
2 |
3 | \usepackage{graphicx}
4 |
5 | \begin{document}
6 |     Ein wundervolles Foto:
7 |     \begin{figure}[tbh]
8 |         \centering
9 |         \includegraphics[width=5cm]{me}
10 |        \caption{Das bin ich.}
11 |    \end{figure}
12 | \end{document}
```

Ein wundervolles Foto:



Figure 1: Das bin ich.

# Beispiel mit Bildunterschrift

```
1 | \documentclass{article}
2 |
3 | \usepackage{graphicx}
4 |
5 | \begin{document}
6 |     Ein wundervolles Foto:
7 |     \begin{figure}[tbh]
8 |         \centering
9 |         \includegraphics[width=5cm]{me}
10 |         \caption{Das bin ich.}
11 |     \end{figure}
12 | \end{document}
```

Ein wundervolles Foto:



Figure 1: Das bin ich.

# Beispiel mit Bildunterschrift

```
1 | \documentclass{article}
2 |
3 | \usepackage{graphicx}
4 |
5 | \begin{document}
6 |     Ein wundervolles Foto:
7 |     \begin{figure}[tbh]
8 |         \centering
9 |         \includegraphics[width=5cm]{me}
10 |         \caption{Das bin ich.}
11 |     \end{figure}
12 | \end{document}
```

Ein wundervolles Foto:



Figure 1: Das bin ich.

- *Hinweis:* Die Benennung kann mit dem Paket `caption` geändert werden.

# Beispiel mit Bildunterschrift

```
1 | \documentclass{article}
2 |
3 | \usepackage{graphicx}
4 |
5 | \begin{document}
6 |     Ein wundervolles Foto:
7 |     \begin{figure}[tbh]
8 |         \centering
9 |         \includegraphics[width=5cm]{me}
10 |        \caption{Das bin ich.}
11 |    \end{figure}
12 | \end{document}
```

Ein wundervolles Foto:



Figure 1: Das bin ich.

- *Hinweis:* Die Benennung kann mit dem Paket `caption` geändert werden.
- Auch nützlich: Das Paket `wrapfig`. (Ermöglicht Grafiken und Tabellen im Textfluss)

# Und was waren jetzt noch mal alle Fotos?

# Und was waren jetzt noch mal alle Fotos?

- Alle Grafiken/Tabellen

# Und was waren jetzt noch mal alle Fotos?

- Alle Grafiken/Tabellen (die durch *figure/ table* mittels `\caption` benannt wurden)

# Und was waren jetzt noch mal alle Fotos?

- Alle Grafiken/Tabellen (die durch *figure/ table* mittels `\caption` benannt wurden) können mittels `\listoffigures/ \listoftables` aufgelistet werden.



# Und was waren jetzt noch mal alle Fotos?

- Alle Grafiken/Tabellen (die durch *figure/ table* mittels `\caption` benannt wurden) können mittels `\listoffigures/ \listoftables` aufgelistet werden.
- Beispiel

# Und was waren jetzt noch mal alle Fotos?

- Alle Grafiken/Tabellen (die durch *figure/ table* mittels `\caption` benannt wurden) können mittels `\listoffigures/ \listoftables` aufgelistet werden.
- Beispiel (`\noindent` entfernt den horizontalen Abstand):

# Und was waren jetzt noch mal alle Fotos?

- Alle Grafiken/Tabellen (die durch *figure/table* mittels `\caption` benannt wurden) können mittels `\listoffigures/ \listoftables` aufgelistet werden.
- **Beispiel** (`\noindent` entfernt den horizontalen Abstand):

```
1 | \documentclass{article}
2 | \usepackage{graphicx}
3 | \begin{document}
4 |     \listoffigures
5 |
6 |     \noindent Ein wundervolles Foto:
7 |     \begin{figure}[tbh]
8 |         \centering
9 |         \includegraphics[width=5cm]{me}
10 |        \caption{Das bin ich.}
11 |    \end{figure}
12 | \end{document}
```

# Und was waren jetzt noch mal alle Fotos?

- Alle Grafiken/Tabellen (die durch *figure/table* mittels `\caption` benannt wurden) können mittels `\listoffigures`/`\listoftables` aufgelistet werden.
- Beispiel (`\noindent` entfernt den horizontalen Abstand):

```
1 | \documentclass{article}
2 | \usepackage{graphicx}
3 | \begin{document}
4 |   \listoffigures
5 |
6 |   \noindent Ein wundervolles Foto:
7 |   \begin{figure}[tbh]
8 |     \centering
9 |     \includegraphics[width=5cm]{me}
10 |    \caption{Das bin ich.}
11 |  \end{figure}
12 | \end{document}
```

## List of Figures

1    Das bin ich. .... 1  
Ein wundervolles Foto:



Figure 1: Das bin ich.

# Informationen

# Informationen

- Dies benötigt, analog zu `\tableofcontents` wieder zwei Durchläufe.

# Informationen

- Dies benötigt, analog zu `\tableofcontents` wieder zwei Durchläufe.
- Info: Wie `\section` etc. kann `\caption` mit einem optionalen Argument den Titel in der LOF (**L**ist **o**f **F**igures) ändern.

# Informationen

- Dies benötigt, analog zu `\tableofcontents` wieder zwei Durchläufe.
- Info: Wie `\section` etc. kann `\caption` mit einem optionalen Argument den Titel in der LOF (**L**ist **o**f **F**igures) ändern.
- *figure* und *table* erzeugen sogenannte „Floats“.



# Informationen

- Dies benötigt, analog zu `\tableofcontents` wieder zwei Durchläufe.
- Info: Wie `\section` etc. kann `\caption` mit einem optionalen Argument den Titel in der LOF (**L**ist **o**f **F**igures) ändern.
- *figure* und *table* erzeugen sogenannte „Floats“. Sie können mittels dem Paket `float` durch die Platzierungsoption „H“ an Ort und stelle forciert werden.

# Informationen

- Dies benötigt, analog zu `\tableofcontents` wieder zwei Durchläufe.
- Info: Wie `\section` etc. kann `\caption` mit einem optionalen Argument den Titel in der LOF (**L**ist **o**f **F**igures) ändern.
- *figure* und *table* erzeugen sogenannte „Floats“. Sie können mittels dem Paket `float` durch die Platzierungsoption „H“ an Ort und stelle forciert werden.
- Das Paket `newfloat` erlaubt durch `\DeclareFloatingEnvironment` die Definition einer eigenen *floating* Umgebungen.

$\text{\LaTeX}$  ist schon toll...



# Aussicht: Was noch geht

# Aussicht: Was noch geht

- Grafiken (so wie Enten) und eigentlich alles mittels `tikz`.

# Aussicht: Was noch geht

- Grafiken (so wie Enten) und eigentlich alles mittels `tikz`.
- Bibliographien, Glossare, Indizes

# Aussicht: Was noch geht

- Grafiken (so wie Enten) und eigentlich alles mittels `tikz`.
- Bibliographien, Glossare, Indizes
- Tolle Boxen/Theoreme

# Aussicht: Was noch geht

- Grafiken (so wie Enten) und eigentlich alles mittels `tikz`.
- Bibliographien, Glossare, Indizes
- Tolle Boxen/Theoreme
- Mit  $\text{\LaTeX}$ 3 (`exp13`) noch viel mehr.



# Aussicht: Was noch geht

- Grafiken (so wie Enten) und eigentlich alles mittels `tikz`.
- Bibliographien, Glossare, Indizes
- Tolle Boxen/Theoreme
- Mit  $\text{\LaTeX}3$  (`exp13`) noch viel mehr.
- Zähler (`\newcounter`), Längen (`\newlength`, `\newdimen`)

# Aussicht: Was noch geht

- Grafiken (so wie Enten) und eigentlich alles mittels `tikz`.
- Bibliographien, Glossare, Indizes
- Tolle Boxen/Theoreme
- Mit  $\text{\LaTeX}$ 3 (`expl3`) noch viel mehr.
- Zähler (`\newcounter`), Längen (`\newlength`, `\newdimen`)
- Bedingungen (`\if`, `\ifx`, `\ifnum`, `\ifthenelse`, ...)

# Aussicht: Was noch geht

- Grafiken (so wie Enten) und eigentlich alles mittels `tikz`.
- Bibliographien, Glossare, Indizes
- Tolle Boxen/Theoreme
- Mit  $\text{\LaTeX}3$  (`expl3`) noch viel mehr.
- Zähler (`\newcounter`), Längen (`\newlength`, `\newdimen`)
- Bedingungen (`\if`, `\ifx`, `\ifnum`, `\ifthenelse`, ...)
- Berechnungen (`\pgfmathparse`, ...)

# Aussicht: Was noch geht

- Grafiken (so wie Enten) und eigentlich alles mittels `tikz`.
- Bibliographien, Glossare, Indizes
- Tolle Boxen/Theoreme
- Mit  $\text{\LaTeX}3$  (`expl3`) noch viel mehr.
- Zähler (`\newcounter`), Längen (`\newlength`, `\newdimen`)
- Bedingungen (`\if`, `\ifx`, `\ifnum`, `\ifthenelse`, ...)
- Berechnungen (`\pgfmathparse`, ...)
- Etliches...(Befehle wie `\def`, `\hfill`, `\vfill`, ...)

# Einige Beispiele



## Florian Sihler

### KURZBESCHREIBUNG

Hallo Welt: Hallo Welt ich mag Züge und finde, dass diese Texttext wirklich eine Auszeichnung für Hervorragendes Texting verdient. Du nicht auch? Ich bin geil, danke fürs Lesen

### KONTAKT

+49 900 00 100 100  
florian.sihler@gmail.de  
Niemandsland 42, 12345 Kaffee-  
gen

### PERSÖNLICHE FERTIGKEITEN

- Klähheit
- Tarzen
- Intelligenz
- Verzweifelt
- KungFuMaster

### SOZIALE NETZWERKE

- @egloputoe
- halbschuet
- ropetope
- who shares dat?

### AUSBILDUNG

- Grundschule / Volksschule, Kaffigen 2006 – 2010  
Grundschulabschluss
- Gymnasium / Erich-Kleiner-Gemeinschaft, Kaffigen 2010 – 2018  
Schulabschluss: Allgemeine Hochschulreife (Abitur) (Abitur: 1,0)
- Universität / Universität Kaffigen 2018 – heute  
Bachelorstudium: Informatik mit Mathematik und Psychologie. Um auch ein bisschen zu erklären was ich hier gemacht habe: schreibe ich diesen etwas längeren Text der zeigen soll wie geil ich bin und wie gut ich bin in der Informatik.

### FERTIGKEITEN UND KENNNTNISSE

- Klähpötte
- Stylwork
- Milado
- Siedel von Cuten
- Tenzen
- Tut & Freunde

Alle Daten entstammen den eigenen Beobachtungen.

### BERUFLICHER WERDEGANG

- Logistik Manager / Kaffigen & Kaffigen 2011 – heute  
Tägliche Besuche beim Dönermann meines Vertrauens, wo ich lernen muss:
  - Zuverlässigkeit, da er mir den Döner täglich frisch zubereitet.
  - Pünktlichkeit, da ich den Döner ja noch warm möchte.
- Gut-Arbeiter / Kaffigen 1999 – heute  
Die größte Freude: Erreichen des ersten Meilensteins in der Psychologie, meinen Zeichen der eigenen Natur und Eigenart, tagtäglich aus. Die Vorteile eines derartigen charakteristischen Auftretens liegen insbesondere auf der Hand, umgeben von der Effizienz der Döner- und Haremswelt gleichermassen in den Bann zu ziehen.
- Professioneller Getränkeberater / Kaffigen 2018 – heute  
Mit dem Erhalt der eigenen Wohnung lerne ich die Gesellschaft nicht nur ein unglaublich großes Maß an Verantwortung zu tragen sondern auch regelmäßig frische Getränke den ersten Eindruck darauf um das menschliche Bedürfnis des Durstes zu stillen.

### MOTIVATION

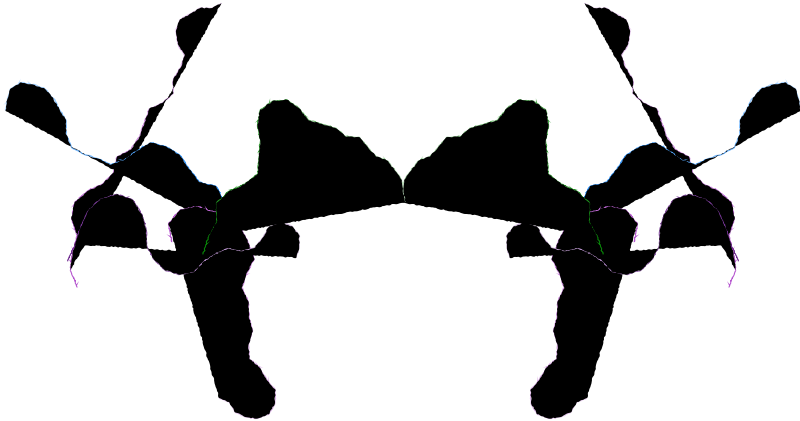
Ich möchte dir doch, demnach auch das Gold Erlebens (Döner) hier ist ein Blendend zum ersten von Tausenden. Wie diesen Text les, so selbstschulend. Der Text gibt lediglich den Grobwert der Schrift an, ist das wirklich so? Ist es gleichgültig, ob ich schreibe „Das ist ein Blendend“ oder „Kaffigen gefahren“? Nicht – ist ein Blendend. Ein Blendend lesst mir wichtige Informationen. Auch wenn ich die Leichtigkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und profit, wie breit oder schmal sie sind. Ein Blendend sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalgröße gut lesbar sein. Er muss können: Sinn ergeben, sollte aber leichter sein. Fremdsprachliche Texte wie „Lernen braun“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Annutung vermitteln.

Mit freundlichen Grüßen

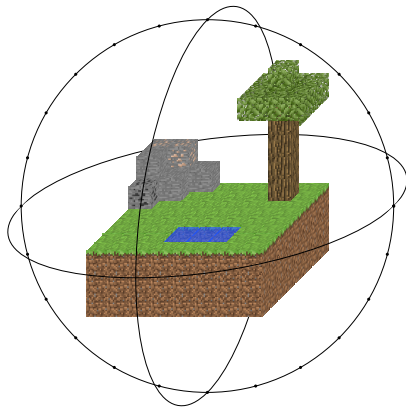
*Florian Sihler*  
Florian Sihler, 4. Oktober 2019



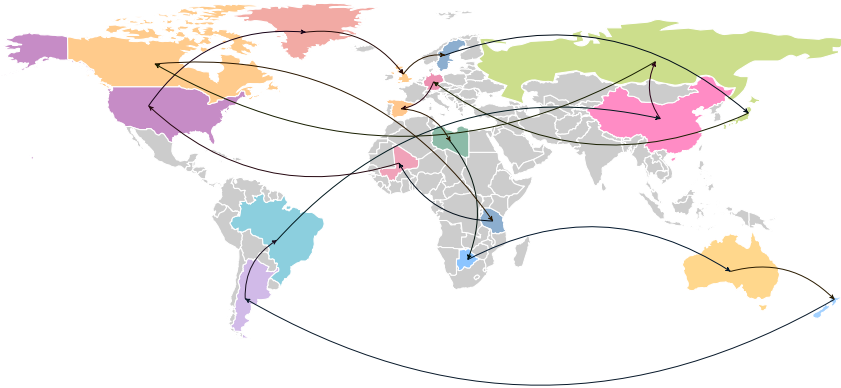
# Einige Beispiele



## Einige Beispiele



# Einige Beispiele





# Einige Beispiele

