

L^AT_EX für Fortgeschrittene

Praktische Tipps jenseits des Textsatzes –
Was neben Bildern und Texten noch so geht

Emily Seebeck

04. Mai 2023

Gliederung des Talks

1. \LaTeX Basics & Best Practices
2. \LaTeX Tooling im Vergleich
3. Schnelle Compiles mit \LaTeX

4. Bibliographien mit \LaTeX
5. Beamer
6. TikZ

- \LaTeX ist wie das Javascript unter den Markup-Sprachen

- \LaTeX ist wie das Javascript unter den Markup-Sprachen
 - Historisch gewachsen
 - Kann sehr frustrierend sein

- \LaTeX ist wie das Javascript unter den Markup-Sprachen
 - Historisch gewachsen
 - Kann sehr frustrierend sein
 - Sehr Flexibel

- \LaTeX ist wie das Javascript unter den Markup-Sprachen
 - Historisch gewachsen
 - Kann sehr frustrierend sein
 - Sehr Flexibel
 - Es gibt immer eine Lösung

- \LaTeX ist wie das Javascript unter den Markup-Sprachen
 - Historisch gewachsen
 - Kann sehr frustrierend sein
 - Sehr Flexibel
 - Es gibt immer eine Lösung, auch, wenn sie nicht schön ist

- \LaTeX ist wie das Javascript unter den Markup-Sprachen
 - Historisch gewachsen
 - Kann sehr frustrierend sein
 - Sehr Flexibel
 - Es gibt immer eine Lösung, auch, wenn sie nicht schön ist
- \LaTeX wird La-Tech ausgesprochen, nicht Latex

- \LaTeX ist wie das Javascript unter den Markup-Sprachen
 - Historisch gewachsen
 - Kann sehr frustrierend sein
 - Sehr Flexibel
 - Es gibt immer eine Lösung, auch, wenn sie nicht schön ist
- \LaTeX wird La-Tech ausgesprochen, nicht Latex
- \LaTeX ist sehr viel learning-by-doing

- \LaTeX ist wie das Javascript unter den Markup-Sprachen
 - Historisch gewachsen
 - Kann sehr frustrierend sein
 - Sehr Flexibel
 - Es gibt immer eine Lösung, auch, wenn sie nicht schön ist
- \LaTeX wird La-Tech ausgesprochen, nicht Latex
- \LaTeX ist sehr viel learning-by-doing
- Google ist der beste Freund

- \LaTeX ist wie das Javascript unter den Markup-Sprachen
 - Historisch gewachsen
 - Kann sehr frustrierend sein
 - Sehr Flexibel
 - Es gibt immer eine Lösung, auch, wenn sie nicht schön ist
- \LaTeX wird La-Tech ausgesprochen, nicht Latex
- \LaTeX ist sehr viel learning-by-doing
- Google (oder ChatGPT) ist der beste Freund

- \LaTeX ist wie das Javascript unter den Markup-Sprachen
 - Historisch gewachsen
 - Kann sehr frustrierend sein
 - Sehr Flexibel
 - Es gibt immer eine Lösung, auch, wenn sie nicht schön ist
- \LaTeX wird La-Tech ausgesprochen, nicht Latex
- \LaTeX ist sehr viel learning-by-doing
- Google (oder ChatGPT) ist der beste Freund
- Diese Präsentation (+ mehr Materialien) werden hochgeladen
 - [Freitagsrunde](#), [GitHub](#)

L^AT_EX Basics & Best Practices

- Nutzt die Compile-Option `-output-directory=...`

- Nutzt die Compile-Option `-output-directory=...`

```
main.aux  
main.fls  
main.log  
main.out  
main.pdf  
T!main.tex
```



```
T!main.tex  
└─ target  
   └─ main.pdf  
      └─ main.log  
         └─ main.fls  
            └─ main.aux  
               └─ main.out
```

- Nutzt die Compile-Option `-output-directory=...`



- Nutzt das Paket `\usepackage[T1]{fontenc}`
 - Per default hat L^AT_EX keinen Support für Umlaute im PDF

- Nutzt die Compile-Option `-output-directory=...`



- Nutzt das Paket `\usepackage[T1]{fontenc}`
 - Per default hat L^AT_EX keinen Support für Umlaute im PDF
- Bei großen Dokumenten: Dinge modular gestalten
 - `\include`
 - Präambel von Dokument trennen
 - Einzelne Abschnitte in verschiedene Dateien auslagern

- Kennt ihr das?
 - `präsentation.tex`

- Kennt ihr das?
 - `präsentation.tex`
 - `präsentation_v2.tex`

- Kennt ihr das?
 - `präsentation.tex`
 - `präsentation_v2.tex`
 - `präsentation_final.tex`

- Kennt ihr das?
 - `präsentation.tex`
 - `präsentation_v2.tex`
 - `präsentation_final.tex`
 - `präsentation_final_final.tex`

- Kennt ihr das?
 - `präsentation.tex`
 - `präsentation_v2.tex`
 - `präsentation_final.tex`
 - `präsentation_final_final.tex`
 - `präsentation_final_final_korrigiert.tex`

- Kennt ihr das?
 - `präsentation.tex`
 - `präsentation_v2.tex`
 - `präsentation_final.tex`
 - `präsentation_final_final.tex`
 - `präsentation_final_final_korrigiert.tex`
- Es ist schwer zu wissen,
 - welche Datei die aktuellste ist
 - was sich zwischen den Versionen verändert hat
 - wer welche Änderungen vorgenommen hat

- Kennt ihr das?
 - `präsentation.tex`
 - `präsentation_v2.tex`
 - `präsentation_final.tex`
 - `präsentation_final_final.tex`
 - `präsentation_final_final_korrigiert.tex`
- Es ist schwer zu wissen,
 - welche Datei die aktuellste ist
 - was sich zwischen den Versionen verändert hat
 - wer welche Änderungen vorgenommen hat

→ Versionskontrolle verwenden

- Kennt ihr das?
 - `präsentation.tex`
 - `präsentation_v2.tex`
 - `präsentation_final.tex`
 - `präsentation_final_final.tex`
 - `präsentation_final_final_korrigiert.tex`
- Es ist schwer zu wissen,
 - welche Datei die aktuellste ist
 - was sich zwischen den Versionen verändert hat
 - wer welche Änderungen vorgenommen hat

→ Versionskontrolle verwenden: git

- Kennt ihr das?
 - `präsentation.tex`
 - `präsentation_v2.tex`
 - `präsentation_final.tex`
 - `präsentation_final_final.tex`
 - `präsentation_final_final_korrigiert.tex`
- Es ist schwer zu wissen,
 - welche Datei die aktuellste ist
 - was sich zwischen den Versionen verändert hat
 - wer welche Änderungen vorgenommen hat

→ Versionskontrolle verwenden: git, overleaf

- Kennt ihr das?
 - `präsentation.tex`
 - `präsentation_v2.tex`
 - `präsentation_final.tex`
 - `präsentation_final_final.tex`
 - `präsentation_final_final_korrigiert.tex`
- Es ist schwer zu wissen,
 - welche Datei die aktuellste ist
 - was sich zwischen den Versionen verändert hat
 - wer welche Änderungen vorgenommen hat

→ Versionskontrolle verwenden: git, overleaf, subversion

1. Paket `\usepackage{hyperref}` verwenden

1. Paket `\usepackage{hyperref}` verwenden
2. Label setzen

1. Paket `\usepackage{hyperref}` verwenden
2. Label setzen
3. Label referenzieren

1. Paket `\usepackage{hyperref}` verwenden
2. Label setzen
3. Label referenzieren
 - Falls deutsche Labels verwendet werden sollen:
`\usepackage[ngerman]{babel}`

1. Paket `\usepackage{hyperref}` verwenden
2. Label setzen
3. Label referenzieren
 - Falls deutsche Labels verwendet werden sollen:
`\usepackage[ngerman]{babel}`
 - Vor Referenzen immer ein *unbreakable space* (~) verwenden
→ `In~\cref{fig:cat}` ist eine Katze zu sehen

A	B	C
1	2	3
4	5	6
7	8	9

Tabelle 1: Dummy-Tabelle

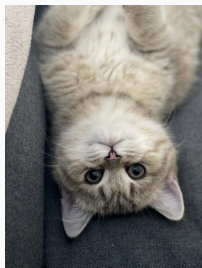


Abbildung 1: Dummy-Bild

$$1 + 1 = 2$$

(1)

\LaTeX Basics: Vergleich von Referenz-Commands (Tabellen)

Command	Output
<code>\ref{...}</code>	1
<code>\nameref{...}</code>	Dummy-Tabelle
<code>\autoref{...}</code>	Tabelle 1
<code>\hyperref[<name>]{...}</code>	<name>
<code>\vref{...}</code>	Tabelle 1 auf der vorherigen Seite
<code>\cref{...}</code>	Tabelle 1
<code>\Cref{...}</code>	Tabelle 1

L^AT_EX Basics: Vergleich von Referenz-Commands (Figuren)

Command	Output
<code>\ref{...}</code>	1
<code>\nameref{...}</code>	Dummy-Bild
<code>\autoref{...}</code>	Abbildung 1
<code>\hyperref[<name>]{...}</code>	<name>
<code>\vref{...}</code>	Abbildung 1 auf Seite 33
<code>\cref{...}</code>	Abbildung 1
<code>\Cref{...}</code>	Abbildung 1

\LaTeX Basics: Vergleich von Referenz-Commands (Gleichungen)

Command	Output
<code>\ref{...}</code>	1
<code>\nameref{...}</code>	<section-name>
<code>\autoref{...}</code>	Gleichung 1
<code>\hyperref[<name>]{...}</code>	<name>
<code>\vref{...}</code>	Gleichung (1) auf Seite 33
<code>\cref{...}</code>	Gleichung (1)
<code>\Cref{...}</code>	Gleichung (1)

- Eigentlich immer cleveref

- Eigentlich immer cleveref
- ... Aber welches Makro? `\cref` oder `\Cref`?

- Eigentlich immer cleveref
- ... Aber welches Makro? `\cref` oder `\Cref`?
 - Keinen Unterschied im *Deutschen*
 - Im Englischen macht `\cref` einen Kleinbuchstaben

- Eigentlich immer cleveref
- ... Aber welches Makro? `\cref` oder `\Cref`?
 - Keinen Unterschied im *Deutschen*
 - Im Englischen macht `\cref` einen Kleinbuchstaben
- `\autoref` geht auch, oft den gleicher Output wie cleveref

\LaTeX Basics: Wann sollte ich welchen Dash verwenden?

Dash	In \LaTeX	Deutscher Name	Englischer Name
-	-	Viertelgeviertstrich	hyphen
–	--	Halbgeviertstrich	en-dash
—	---	Geviertstrich	em-dash

\LaTeX Basics: Wann sollte ich welchen Dash verwenden?

Dash	In \LaTeX	Deutscher Name	Englischer Name
-	-	Viertelgeviertstrich	hyphen
–	--	Halbgeviertstrich	en-dash
—	---	Geviertstrich	em-dash

- **Deutsch**

\LaTeX Basics: Wann sollte ich welchen Dash verwenden?

Dash	In \LaTeX	Deutscher Name	Englischer Name
-	-	Viertelgeviertstrich	hyphen
–	--	Halbgeviertstrich	en-dash
—	---	Geviertstrich	em-dash

- **Deutsch**

- – zwischen Wörtern:
„Mein Lieblingssport ist Schach-Boxen.“

L^AT_EX Basics: Wann sollte ich welchen Dash verwenden?

Dash	In L ^A T _E X	Deutscher Name	Englischer Name
-	-	Viertelgeviertstrich	hyphen
–	--	Halbgeviertstrich	en-dash
—	---	Geviertstrich	em-dash

■ Deutsch

- – zwischen Wörtern:
„Mein Lieblingssport ist Schach-Boxen.“
- -- für Einschübe und Zahlenbereiche / Ranges:
„Meine Oma – die übrigens auch ein hervorragendes Strudelrezept hat – kommt morgen zu Besuch.“
„Die Teilnehmerzahl liegt zwischen 50–100 Personen.“

L^AT_EX Basics: Wann sollte ich welchen Dash verwenden?

Dash	In L ^A T _E X	Deutscher Name	Englischer Name
-	-	Viertelgeviertstrich	hyphen
–	--	Halbgeviertstrich	en-dash
—	---	Geviertstrich	em-dash

■ Deutsch

- - zwischen Wörtern:
„Mein Lieblingssport ist Schach-Boxen.“
- -- für Einschübe und Zahlenbereiche / Ranges:
„Meine Oma – die übrigens auch ein hervorragendes Strudelrezept hat – kommt morgen zu Besuch.“
„Die Teilnehmerzahl liegt zwischen 50–100 Personen.“
- --- wird oft als störend empfunden, kaum genutzt

\LaTeX Basics: Wann sollte ich welchen Dash verwenden?

Dash	In \LaTeX	Deutscher Name	Englischer Name
-	-	Viertelgeviertstrich	hyphen
—	--	Halbgeviertstrich	en-dash
---	---	Geviertstrich	em-dash

- **Englisch**

\LaTeX Basics: Wann sollte ich welchen Dash verwenden?

Dash	In \LaTeX	Deutscher Name	Englischer Name
-	-	Viertelgeviertstrich	hyphen
—	--	Halbgeviertstrich	en-dash
---	---	Geviertstrich	em-dash

- **Englisch**

- – zwischen Wörtern:

‘The rock-paper-scissors-lizard-Spock game is really fun.’

\LaTeX Basics: Wann sollte ich welchen Dash verwenden?

Dash	In \LaTeX	Deutscher Name	Englischer Name
-	-	Viertelgeviertstrich	hyphen
–	--	Halbgeviertstrich	en-dash
—	---	Geviertstrich	em-dash

■ Englisch

- – zwischen Wörtern:
'The rock-paper-scissors-lizard-Spock game is really fun.'
- -- für Zahlenbereiche / Ranges:
'I can count from 1–10 in five different languages.'

\LaTeX Basics: Wann sollte ich welchen Dash verwenden?

Dash	In \LaTeX	Deutscher Name	Englischer Name
-	-	Viertelgeviertstrich	hyphen
--	--	Halbgeviertstrich	en-dash
---	---	Geviertstrich	em-dash

■ Englisch

- - zwischen Wörtern:

‘The rock-paper-scissors-lizard-Spock game is really fun.’

- -- für Zahlenbereiche / Ranges:

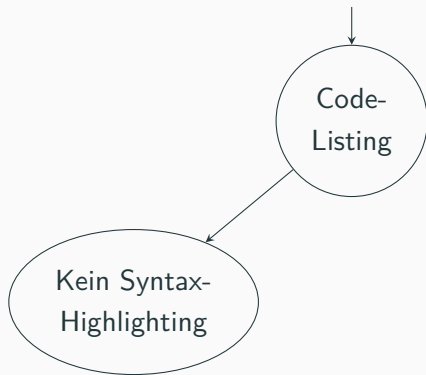
‘I can count from 1–10 in five different languages.’

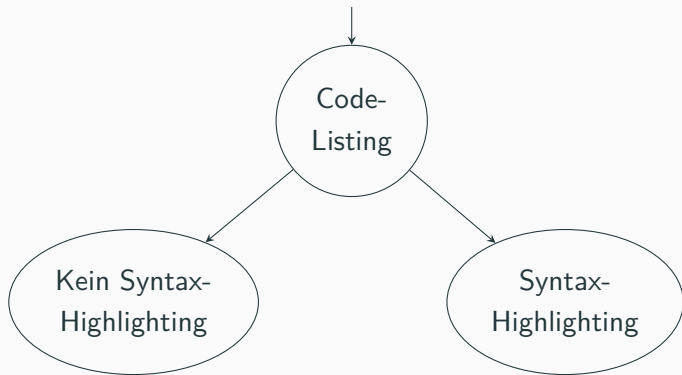
- --- für Einschübe:

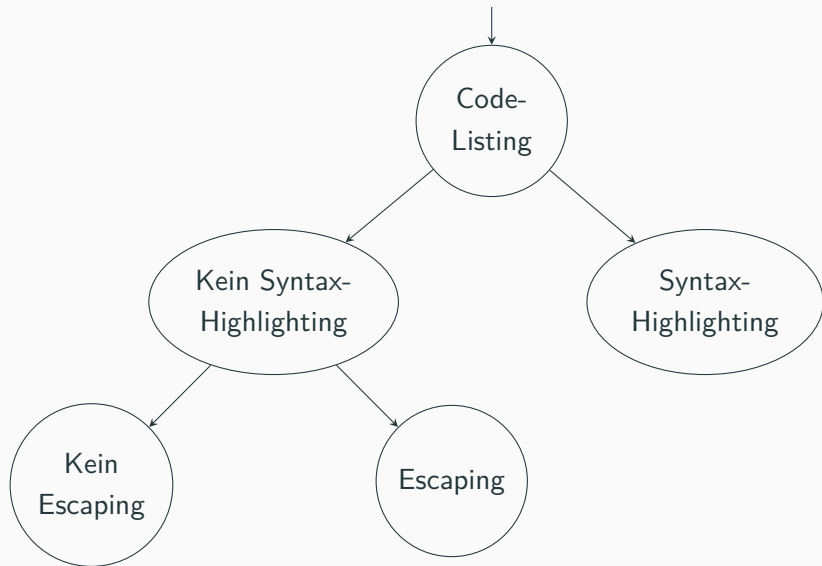
‘I absolutely adore penguins—those waddling, tuxedo-wearing birds—they always make me smile.’

→ Traditionell: heutzutage manchmal auch en-dash mit spaces









- L^AT_EX-Befehl `\texttt`

- L^AT_EX-Befehl `\texttt`

```
\texttt{print("Hello, world!")}
```



```
print("Hello, world!")
```


- L^AT_EX-Befehl `\texttt`

`\texttt{print("Hello, world!")}`



`print("Hello, world!")`

- Aber:

- L^AT_EX-Befehl `\texttt`

```
\texttt{print("Hello, world!")}
```



```
print("Hello, world!")
```

- Aber:

```
\texttt{printf("Hello, %s!", name)}
```



Fatal error occurred, no output PDF file produced!

- Automatisches Escaping

- Automatisches Escaping
- Verbatim Paket: `\usepackage{verbatim}`

- Automatisches Escaping
- Verbatim Paket: `\usepackage{verbatim}`

```
\verb|printf("Hello, %s!", name)|
```



```
printf("Hello, %s!", name)
```

- Automatisches Escaping
- Verbatim Paket: `\usepackage{verbatim}`

```
\verb|printf("Hello, %s!", name)|
```



```
printf("Hello, %s!", name)
```

- Auch als Umgebung:

- Automatisches Escaping
- Verbatim Paket: `\usepackage{verbatim}`

```
\verb|printf("Hello, %s!", name)|
```



```
printf("Hello, %s!", name)
```

- Auch als Umgebung:

```
\begin{verbatim}
```

```
    printf("Hello, %s!", name)
```

```
\end{verbatim}
```

```
    printf("Hello, %s!", name)
```

- Listings

- `minted`

L^AT_EX Basics: Code-Listings (mit Syntax Highlighting)

- Listings
 - Rudimentäres Syntax Highlighting (Keyword-Highlighting)
 - Relativ Langsam, ziemlich alt, geschrieben in T_EX
 - Muss Customized werden
 - Probleme mit Unicode
- `minted`

L^AT_EX Basics: Code-Listings (mit Syntax Highlighting)

- Listings
 - Rudimentäres Syntax Highlighting (Keyword-Highlighting)
 - Relativ Langsam, ziemlich alt, geschrieben in T_EX
 - Muss Customized werden
 - Probleme mit Unicode
- `minted`
 - Sehr feature-reich (Kontext-Highlighting)
 - Powered by Pygments
 - Unterstützt mehr Programmiersprachen
 - Benötigt System Setup
 - Python, Pygments und Compiler-Option `-shell-escape`

L^AT_EX Basics: Code-Listings (listings)

```
def greet(name: str):  
    message = f"Hello, {name}!"  
    return message  
  
def fibonacci(n: int):  
    if n <= 1:  
        return n  
    else:  
        return fibonacci(n-1) + fibonacci(n-2)  
  
def is_palindrome(s: str):  
    s = s.lower().replace(" ", "")  
    return s == s[::-1]
```

LaTeX Basics: Code-Listings (minted)

```
def greet(name: str):  
    message = f"Hello, {name}!"  
    return message  
  
def fibonacci(n: int):  
    if n <= 1:  
        return n  
    else:  
        return fibonacci(n - 1) + fibonacci(n - 2)  
  
def is_palindrome(s: str):  
    s = s.lower().replace(" ", "")  
    return s == s[::-1]
```

L^AT_EX Basics: Code-Listings (listings)

```
fn greet(name: &str) -> String {  
    format!("Hello, {name}!")  
}
```

```
fn fibonacci(n: u32) -> u32 {  
    match n {  
        0 | 1 => n,  
        _ => fibonacci(n-1) + fibonacci(n-2)  
    }  
}
```

```
fn is_palindrome(s: &str) -> bool {  
    let s = s.to_lowercase().replace(" ", "");  
    s == s.chars().rev().collect::<String>()  
}
```

L^AT_EX Basics: Code-Listings (minted)

```
fn greet(name: &str) -> String {  
    format!("Hello, {name}!")  
}
```

```
fn fibonacci(n: u32) -> u32 {  
    match n {  
        0 | 1 => n,  
        _ => fibonacci(n - 1) + fibonacci(n - 2)  
    }  
}
```

```
fn is_palindrome(s: &str) -> bool {  
    let s = s.to_lowercase().replace(" ", "");  
    s == s.chars().rev().collect::<String>()  
}
```

- Mit dem $\$$ wird der *inline* Mathe-Modus eingeleitet

- Mit dem `$` wird der *inline* Mathe-Modus eingeleitet

`$a = -b$` wird zu $a = -b$

- Mit dem `$` wird der *inline* Mathe-Modus eingeleitet

$$\texttt{\$a = -b\$} \text{ wird zu } a = -b$$

- Der *display* Mathe Modus wird mit `\[... \]` eingeleitet

$$\texttt{\[a = -b \]} \qquad a = -b$$

- Mit dem `$` wird der *inline* Mathe-Modus eingeleitet

$$\texttt{\$a = -b\$} \text{ wird zu } a = -b$$

- Der *display* Mathe Modus wird mit `\[... \]` eingeleitet

$$\texttt{\[a = -b \]} \qquad a = -b$$

- Statt `$$... $$` lieber `\[... \]` verwenden

- Mit dem `$` wird der *inline* Mathe-Modus eingeleitet

$$\texttt{\$a = -b\$} \text{ wird zu } a = -b$$

- Der *display* Mathe Modus wird mit `\[... \]` eingeleitet

$$\texttt{\[a = -b \]} \qquad a = -b$$

- Statt `$$... $$` lieber `\[... \]` **verwenden**
- Durch `\usepackage{amsmath}` auch durch `align` möglich

- Mit dem `$` wird der *inline* Mathe-Modus eingeleitet

$$\texttt{\$a = -b\$} \text{ wird zu } a = -b$$

- Der *display* Mathe Modus wird mit `\[... \]` eingeleitet

$$\texttt{\[a = -b \]} \qquad a = -b$$

- Statt `$$... $$` lieber `\[... \]` **verwenden**
- Durch `\usepackage{amsmath}` auch durch `align` möglich

$$\begin{array}{rcl} \texttt{\backslashbegin{align}} & & \\ a \texttt{ \&= } b \texttt{ \backslash\backslash} & a = b & (2) \\ \texttt{\backslashsqrt{2} \&= } c & \sqrt{2} = c & (3) \\ \texttt{\backslashend{align}} & & \end{array}$$

- Verschiedene Kategorien
 - Gleichzeichen: $=$, \neq (`\neq`), \approx (`\approx`)
 - Griechische Buchstaben: α (`\alpha`), β (`\beta`)
 - Mathematische Operatoren: \sum (`\sum`), \int (`\int`)
 - Zahlenmengen: \mathbb{N} (`\mathbb{N}`) oder \mathbb{N} (`\mathds{N}`)
 - Pfeile: \rightarrow (`\to`), \Rightarrow (`\Rightarrow`), \mapsto (`\mapsto`)

- Verschiedene Kategorien
 - Gleichzeichen: $=$, \neq (`\neq`), \approx (`\approx`)
 - Griechische Buchstaben: α (`\alpha`), β (`\beta`)
 - Mathematische Operatoren: \sum (`\sum`), \int (`\int`)
 - Zahlenmengen: \mathbb{N} (`\mathbb{N}`) oder \mathbb{N} (`\mathds{N}`)
 - Pfeile: \rightarrow (`\to`), \Rightarrow (`\Rightarrow`), \mapsto (`\mapsto`)
- Falls dein Zeichen hier nicht dabei war: [Detexify](#)

- Verschiedene Kategorien
 - Gleichzeichen: $=$, \neq (`\neq`), \approx (`\approx`)
 - Griechische Buchstaben: α (`\alpha`), β (`\beta`)
 - Mathematische Operatoren: \sum (`\sum`), \int (`\int`)
 - Zahlenmengen: \mathbb{N} (`\mathbb{N}`) oder \mathbb{N} (`\mathds{N}`)
 - Pfeile: \rightarrow (`\rightarrow`), \Rightarrow (`\Rightarrow`), \mapsto (`\mapsto`)
- Falls dein Zeichen hier nicht dabei war: [Detexify](#)
- Welches der Symbole findet ihr besser? \mathbb{N} oder \mathbb{N}

- Verschiedene Kategorien
 - Gleichzeichen: $=$, \neq (`\neq`), \approx (`\approx`)
 - Griechische Buchstaben: α (`\alpha`), β (`\beta`)
 - Mathematische Operatoren: \sum (`\sum`), \int (`\int`)
 - Zahlenmengen: \mathbb{N} (`\mathbb{N}`) oder \mathbb{N} (`\mathds{N}`)
 - Pfeile: \rightarrow (`\to`), \Rightarrow (`\Rightarrow`), \mapsto (`\mapsto`)
- Falls dein Zeichen hier nicht dabei war: [Detexify](#)
- Welches der Symbole findet ihr besser? \mathbb{N} oder \mathbb{N}
 - \mathbb{N} ist eher handschriftlich

- Verschiedene Kategorien
 - Gleichzeichen: $=$, \neq (`\neq`), \approx (`\approx`)
 - Griechische Buchstaben: α (`\alpha`), β (`\beta`)
 - Mathematische Operatoren: \sum (`\sum`), \int (`\int`)
 - Zahlenmengen: \mathbb{N} (`\mathbb{N}`) oder \mathbb{N} (`\mathds{N}`)
 - Pfeile: \rightarrow (`\to`), \Rightarrow (`\Rightarrow`), \mapsto (`\mapsto`)
- Falls dein Zeichen hier nicht dabei war: [Detexify](#)
- Welches der Symbole findet ihr besser? \mathbb{N} oder \mathbb{N}
 - \mathbb{N} ist eher handschriftlich
 - Falls euch etwas vorgeschrieben wird, nehmt das

- Verschiedene Kategorien
 - Gleichzeichen: $=$, \neq (`\neq`), \approx (`\approx`)
 - Griechische Buchstaben: α (`\alpha`), β (`\beta`)
 - Mathematische Operatoren: \sum (`\sum`), \int (`\int`)
 - Zahlenmengen: \mathbb{N} (`\mathbb{N}`) oder \mathbb{N} (`\mathds{N}`)
 - Pfeile: \rightarrow (`\to`), \Rightarrow (`\Rightarrow`), \mapsto (`\mapsto`)
- Falls dein Zeichen hier nicht dabei war: [Detexify](#)
- Welches der Symbole findet ihr besser? \mathbb{N} oder \mathbb{N}
 - \mathbb{N} ist eher handschriftlich
 - Falls euch etwas vorgeschrieben wird, nehmt das
 - Ansonsten: Nehmt das was ihr möchtet.

Aber haltet euch dran!

$$\{x \in \mathbb{N} | x > 5\}$$

$$\{x \in \mathbb{N} | x > 5\} \quad \{x \in \mathbb{N} \mid x > 5\}$$

- Das Spacing zwischen dem Strich ist zu wenig
- Wie geht es besser? `\mid`

$$\{x \in \mathbb{N} | x > 5\} \quad \{x \in \mathbb{N} \mid x > 5\}$$

$$a := 5$$

$$a := 5 \qquad a := 5$$

- Das Spacing zwischen dem Strich ist zu wenig
- Wie geht es besser? `\mid`
- Das `=` und der Doppelpunkt sind nicht aligned
- Wie geht es besser? `\coloneq`

$$a := 5 \qquad a \coloneq 5$$

$$\sin(x)$$

$$\sin(x)$$

$$\sin(x)$$

L^AT_EX Mathe-Basics: Was ist hier falsch?

- Das Spacing zwischen dem Strich ist zu wenig
- Wie geht es besser? `\mid`
- Das `=` und der Doppelpunkt sind nicht aligned
- Wie geht es besser? `\coloneq`
- Das *sin* sollte nicht kursiv geschrieben sein
- Wie geht es besser? `\sin`

sin(*x*)

`\sin`(*x*)

- Ausgehend von **NIST** ...

L^AT_EX Mathe-Basics: Wann sollte was kursiv sein?

- Ausgehend von NIST ...
- Mathematische Funktionen sollten *nicht* kursiv sein
→ *tan*(*x*) vs. tan(*x*), *exp*(*x*) vs. exp(*x*)

L^AT_EX Mathe-Basics: Wann sollte was kursiv sein?

- Ausgehend von NIST ...
- Mathematische Funktionen sollten *nicht* kursiv sein
→ $\tan(x)$ vs. $\tan(x)$, $\exp(x)$ vs. $\exp(x)$
- Variablen und Funktionen mit 1 Buchstaben sollten kursiv sein
→ $x, y, z, t, r, \lambda, f(x)$

- Ausgehend von **NIST** ...
- Mathematische Funktionen sollten *nicht* kursiv sein
→ $\tan(x)$ vs. $\tan(x)$, $\exp(x)$ vs. $\exp(x)$
- Variablen und Funktionen mit 1 Buchstaben sollten kursiv sein
→ $x, y, z, t, r, \lambda, f(x)$
- Einheiten sollten *nicht* kursiv sein
→ $t = 3\text{s}$, $r = 11\text{cm}$, $\lambda = 420\text{nm}$

L^AT_EX Mathe-Basics: Wann sollte was kursiv sein?

- Ausgehend von [NIST](#) ...
- Mathematische Funktionen sollten *nicht* kursiv sein
→ $\tan(x)$ vs. $\tan(x)$, $\exp(x)$ vs. $\exp(x)$
- Variablen und Funktionen mit 1 Buchstaben sollten kursiv sein
→ $x, y, z, t, r, \lambda, f(x)$
- Einheiten sollten *nicht* kursiv sein
→ $t = 3\text{s}$, $r = 11\text{cm}$, $\lambda = 420\text{nm}$
- Mathematische Konstanten werden nicht kursiv geschrieben
→ $e^{i\cdot\pi} = -1$

L^AT_EX Mathe-Basics: Wann sollte was kursiv sein?

- Ausgehend von NIST ...
- Mathematische Funktionen sollten *nicht* kursiv sein
→ $\tan(x)$ vs. $\tan(x)$, $\exp(x)$ vs. $\exp(x)$
- Variablen und Funktionen mit 1 Buchstaben sollten kursiv sein
→ $x, y, z, t, r, \lambda, f(x)$
- Einheiten sollten *nicht* kursiv sein
→ $t = 3\text{s}$, $r = 11\text{cm}$, $\lambda = 420\text{nm}$
- Mathematische Konstanten werden nicht kursiv geschrieben
→ $e^{i\cdot\pi} = -1$
- Der Differentialoperator sollte nicht kursiv sein
→ $\int_a^b x^2 dx$ vs. $\int_a^b x^2 \mathrm{d}x$, $f'(x) = \frac{df(x)}{dx}$ vs. $f'(x) = \frac{\mathrm{d}f(x)}{\mathrm{d}x}$

\LaTeX Tooling im Vergleich

ΛT_EX Tooling: Wie installieren?

- Linux
 - T_EXLive durch den Paketmanager installieren
 - `sudo apt install texlive-latex-extra`
 - `sudo pacman -S texlive-most`
 - `sudo dnf install texlive-scheme-medium`

ΛT_EX Tooling: Wie installieren?

- Linux
 - T_EXLive durch den Paketmanager installieren
 - `sudo apt install texlive-latex-extra`
 - `sudo pacman -S texlive-most`
 - `sudo dnf install texlive-scheme-medium`
- macOS
 - MacTeX
 - `brew cask install mactex`

L^AT_EX Tooling: Wie installieren?

- Linux
 - T_EXLive durch den Paketmanager installieren
 - `sudo apt install texlive-latex-extra`
 - `sudo pacman -S texlive-most`
 - `sudo dnf install texlive-scheme-medium`
- macOS
 - MacTeX
 - `brew cask install mactex`
- Windows
 - MiK_TE_X / T_EXLive

LaTeX Tooling: MiKTeX vs. TeXLive

	MiKTeX	TeXLive
Anzahl Pakete	Wenig	Viel
„on-the-fly“ Pakete installieren	✓	✗
Open-Source	✓	✓
Plattform	(Primär) Windows	Alle
Maintainer	Christian Schenk	TeX Users Group

LaTeX Tooling: MiKTeX vs. TeXLive

	MiKTeX	TeXLive
Anzahl Pakete	Wenig	Viel
„on-the-fly“ Pakete installieren	✓	✗
Open-Source	✓	✓
Plattform	(Primär) Windows	Alle
Maintainer	Christian Schenk	TeX Users Group

Was ist schneller?

LaTeX Tooling: MiKTeX vs. TeXLive

	MiKTeX	TeXLive
Anzahl Pakete	Wenig	Viel
„on-the-fly“ Pakete installieren	✓	✗
Open-Source	✓	✓
Plattform	(Primär) Windows	Alle
Maintainer	Christian Schenk	TeX Users Group

Was ist schneller? → TeXLive

- Online Editor

- Online Editor
 - Overleaf

- Online Editor
 - Overleaf
- Lokaler Editor

- Online Editor
 - Overleaf
- Lokaler Editor
 - Dedizierte IDE
 - T_EXStudio
 - T_EXMaker
 - LyX

- Online Editor
 - Overleaf
- Lokaler Editor
 - Dedizierte IDE
 - T_EXStudio
 - T_EXMaker
 - LyX
 - IDE mit LSP / Plugin support
 - Visual Studio Code
 - IntelliJ IDEA
 - vim / emacs

- Nicht alle PDF viewer sind gleich!

- Nicht alle PDF viewer sind gleich!
- Auto-refresh

- Nicht alle PDF viewer sind gleich!
- Auto-refresh
- Forward / Backward Search

- Nicht alle PDF viewer sind gleich!
- Auto-refresh
- Forward / Backward Search
- Reverse-Jump

- Nicht alle PDF viewer sind gleich!
- Auto-refresh
- Forward / Backward Search
- Reverse-Jump
- SyncTeX (zur relevanten Seite scrollen)

- Nicht alle PDF viewer sind gleich!
- Auto-refresh
- Forward / Backward Search
- Reverse-Jump
- SyncTeX (zur relevanten Seite scrollen)
- Platform

LaTeX Tooling: Vergleich von PDF-Viewern

	Auto- Refresh	F/B- Search	Reverse- Jump	SyncTeX	Platform
Okular	✓	✓	✓	✓	Alle
Evince	✓	✓	✓	✓	Alle
SumatraPDF	✓	✓	✓	✓	Windows
Overleaf	✓	✓	✗	✓	Alle
Chromium	✗	✗	✗	✗	Alle
Firefox	✗	✗	✗	✗	Alle
Preview	✗	✗	✓	✗	macOS
Skim	✓	✓	✓	✓	macOS

Schnelle Compiles mit \LaTeX

Schnelle Compiles: Warum?

- \LaTeX wird kompiliert, es ist nicht WYSIWYG

Schnelle Compiles: Warum?

- \LaTeX wird kompiliert, es ist nicht WYSIWYG
- Bis das Dokument kompiliert ist, gibt es kein Feedback

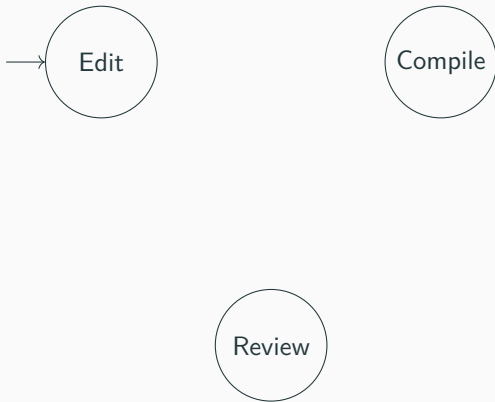
Schnelle Compiles: Warum?

- \LaTeX wird kompiliert, es ist nicht WYSIWYG
- Bis das Dokument kompiliert ist, gibt es kein Feedback
 - Positionierung von Bildern
 - TikZ Grafiken
 - ...

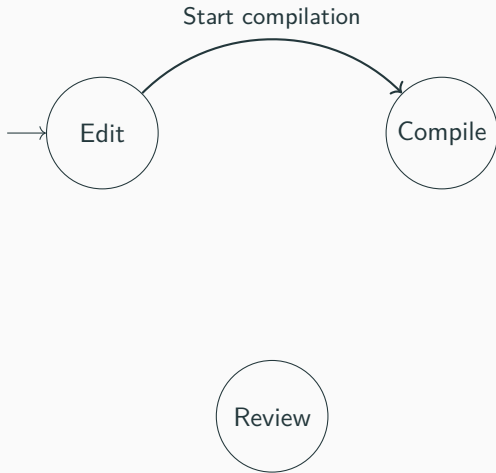
Schnelle Compiles: Warum?

- \LaTeX wird kompiliert, es ist nicht WYSIWYG
- Bis das Dokument kompiliert ist, gibt es kein Feedback
 - Positionierung von Bildern
 - TikZ Grafiken
 - ...
- Feedback Cycle

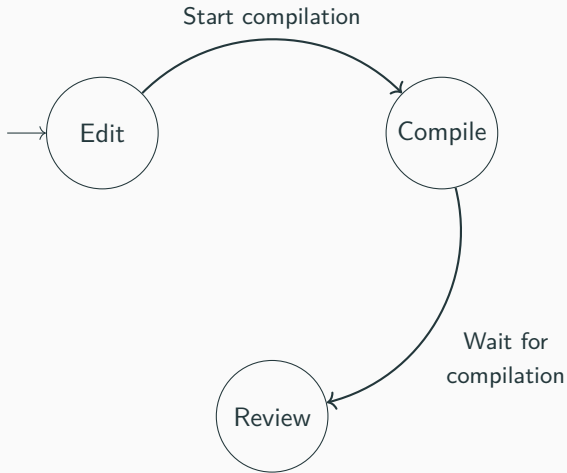
Schnelle Compiles: Warum?



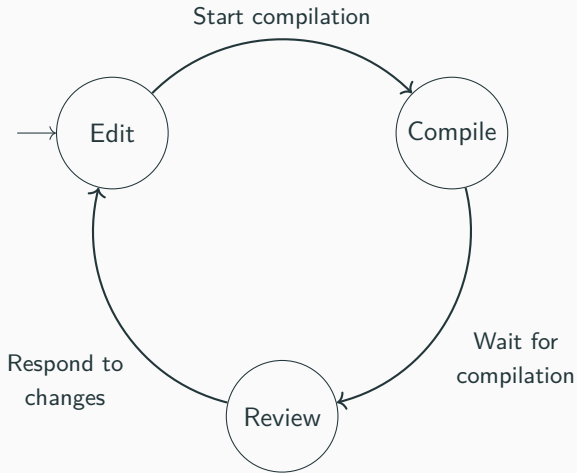
Schnelle Compiles: Warum?



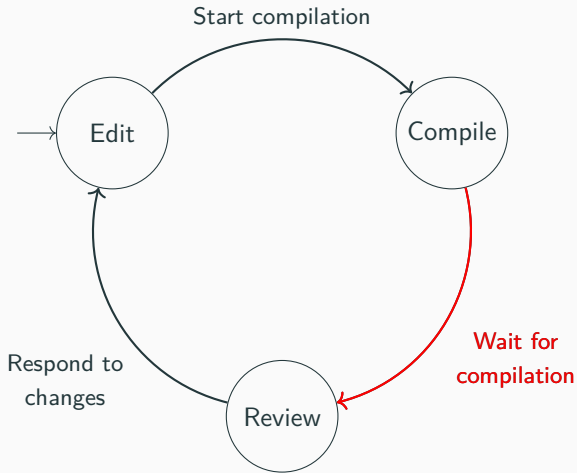
Schnelle Compiles: Warum?



Schnelle Compiles: Warum?



Schnelle Compiles: Warum?



Schnelle Compiles: Warum?

- \LaTeX wird kompiliert, es ist nicht WYSIWYG
- Bis das Dokument kompiliert ist, gibt es kein Feedback
 - Positionierung von Bildern
 - TikZ Grafiken
 - ...
- Feedback Cycle

Schnelle Compiles: Warum?

- \LaTeX wird kompiliert, es ist nicht WYSIWYG
- Bis das Dokument kompiliert ist, gibt es kein Feedback
 - Positionierung von Bildern
 - TikZ Grafiken
 - ...
- Feedback Cycle
- Verschiedene Optimierungsdimensionen

Schnelle Compiles: Warum?

- \LaTeX wird kompiliert, es ist nicht WYSIWYG
- Bis das Dokument kompiliert ist, gibt es kein Feedback
 - Positionierung von Bildern
 - TikZ Grafiken
 - ...
- Feedback Cycle
- Verschiedene Optimierungsdimensionen
 1. Beschleunigen des Codes

Schnelle Compiles: Warum?

- \LaTeX wird kompiliert, es ist nicht WYSIWYG
- Bis das Dokument kompiliert ist, gibt es kein Feedback
 - Positionierung von Bildern
 - TikZ Grafiken
 - ...
- Feedback Cycle
- Verschiedene Optimierungsdimensionen
 1. Beschleunigen des Codes
 2. Reduzieren der zu verarbeitenden Code-Menge

Schnelle Compiles: Warum?

- \LaTeX wird kompiliert, es ist nicht WYSIWYG
- Bis das Dokument kompiliert ist, gibt es kein Feedback
 - Positionierung von Bildern
 - TikZ Grafiken
 - ...
- Feedback Cycle
- Verschiedene Optimierungsdimensionen
 1. Beschleunigen des Codes
 2. Reduzieren der zu verarbeitenden Code-Menge
 3. TikZ

Schnelle Compiles: Beschleunigen des Codes

- Vermeiden von unnötigen Paketen
 - TikZ
 - Bib(La)TeX
 - minted

Schnelle Compiles: Beschleunigen des Codes

- Vermeiden von unnötigen Paketen
 - TikZ
 - Bib(La)TeX
 - minted
- Trojanische Pferde vermeiden
 - Pakete, die als Dependency ein langsames Paket haben

Schnelle Compiles: Beschleunigen des Codes

- Vermeiden von unnötigen Paketen
 - TikZ
 - Bib(La)T_EX
 - minted
- Trojanische Pferde vermeiden
 - Pakete, die als Dependency ein langsames Paket haben
- Caches nutzen: Build Artefakte nicht löschen

Schnelle Compiles: Optimieren von Grafiken

- Auflösung anpassen
 - Reduzieren der Auflösung (4k \rightarrow 1080p)

\rightarrow Schnellere Verarbeitung

Schnelle Compiles: Optimieren von Grafiken

- Auflösung anpassen
 - Reduzieren der Auflösung (4k \rightarrow 1080p)
 - \rightarrow Schnellere Verarbeitung
- In .pdf konvertieren
 - Natives Format für \LaTeX -Ausgabe
 - Keine zusätzliche Formatkonvertierung
 - \rightarrow Kürzere Kompilierungszeit

Schnelle Compiles: Optimieren von Grafiken

- Auflösung anpassen
 - Reduzieren der Auflösung (4k \rightarrow 1080p)
 - \rightarrow Schnellere Verarbeitung
- In .pdf konvertieren
 - Natives Format für \LaTeX -Ausgabe
 - Keine zusätzliche Formatkonvertierung
 - \rightarrow Kürzere Kompilierungszeit

- Kann automatisch gemacht werden:

```
find . -type f -exec file --mime-type {} \+ |  
awk -F: '{if ($2 ~ /image\/\//) print $1}' |  
xargs -P "$(nproc)" -I it \                  \  
sh -c 'img2pdf $1 -o ${1%.*}.pdf' -- it $argv
```

Schnelle Compiles: Reduzieren der Code-Menge

- Beispiel: Dokument von 100 Seiten
- Meist wird nur 1 aktiv bearbeitet, warum alle kompilieren?

Schnelle Compiles: Reduzieren der Code-Menge

- Beispiel: Dokument von 100 Seiten
- Meist wird nur 1 aktiv bearbeitet, warum alle kompilieren?
- `\usepackage{subfiles}`: Zerteilen des Projekts
- Jedes Teildokument kann separat kompiliert werden

Schnelle Compiles: Reduzieren der Code-Menge

- Beispiel: Dokument von 100 Seiten
- Meist wird nur 1 aktiv bearbeitet, warum alle kompilieren?
- `\usepackage{subfiles}`: Zerteilen des Projekts
- Jedes Teildokument kann separat kompiliert werden
- Jedes Teildokument ist ein vollständiges \LaTeX -Dokument
- Die main Datei bekommt den Inhalt der Subfile
- Die Subfile bekommt die Präambel der main Datei

Schnelle Compiles: Reduzieren der Code-Menge

- Beispiel: Dokument von 100 Seiten
- Meist wird nur 1 aktiv bearbeitet, warum alle kompilieren?
- `\usepackage{subfiles}`: Zerteilen des Projekts
- Jedes Teildokument kann separat kompiliert werden
- Jedes Teildokument ist ein vollständiges \LaTeX -Dokument
- Die main Datei bekommt den Inhalt der Subfile
- Die Subfile bekommt die Präambel der main Datei
- Wird transparent von `\documentclass{subfiles}` gemacht
- Minimaler Overhead

Schnelle Compiles: Precompiled Header

- Die Präambel des Dokuments ändert sich (fast) nie
- Warum wird die immer wieder aufs neue kompiliert?

Schnelle Compiles: Precompiled Header

- Die Präambel des Dokuments ändert sich (fast) nie
- Warum wird die immer wieder aufs neue kompiliert?
- Abspeichern des Zustands am Ende der Präambel
 - Kann bei der nächsten Kompilation geladen werden

Schnelle Compiles: Precompiled Header

- Die Präambel des Dokuments ändert sich (fast) nie
- Warum wird die immer wieder aufs neue kompiliert?
- Abspeichern des Zustands am Ende der Präambel
 - Kann bei der nächsten Kompilation geladen werden
- Mit dem Paket `mylatexformat` möglich:

```
pdflatex -ini -jobname="main" "&pdflatex" \  
mylatexformat.ltx main.tex
```


Schnelle Compiles: Precompiled Header

- Die Präambel des Dokuments ändert sich (fast) nie
- Warum wird die immer wieder aufs neue kompiliert?
- Abspeichern des Zustands am Ende der Präambel
 - Kann bei der nächsten Kompilation geladen werden
- Mit dem Paket `mylatexformat` möglich:

```
pdflatex -ini -jobname="main" "&pdflatex" \mylatexformat.ltx main.tex
```
- Heraus kommt eine ~10MB Binärdatei
- Kann über den Header `%&main` eingebunden werden

Schnelle Compiles: Externalisierung von TikZ Bildern

- Beispiel: Dokument mit 20 schönen TikZ Bildern.
- Keins ist unnötig, alle sollen gerendert werden

Schnelle Compiles: Externalisierung von TikZ Bildern

- Beispiel: Dokument mit 20 schönen TikZ Bildern.
- Keins ist unnötig, alle sollen gerendert werden
→ Externalisierung nutzen

Schnelle Compiles: Externalisierung von TikZ Bildern

- Beispiel: Dokument mit 20 schönen TikZ Bildern.
- Keins ist unnötig, alle sollen gerendert werden
→ Externalisierung nutzen
- TikZ-Bilder in separate PDFs speichern

Schnelle Compiles: Externalisierung von TikZ Bildern

- Beispiel: Dokument mit 20 schönen TikZ Bildern.
- Keins ist unnötig, alle sollen gerendert werden
→ Externalisierung nutzen
- TikZ-Bilder in separate PDFs speichern
- Einfügen der PDFs in Hauptdokument

Schnelle Compiles: Externalisierung von TikZ Bildern

- Beispiel: Dokument mit 20 schönen TikZ Bildern.
- Keins ist unnötig, alle sollen gerendert werden
→ Externalisierung nutzen
- TikZ-Bilder in separate PDFs speichern
- Einfügen der PDFs in Hauptdokument
- Beschleunigung durch Vermeidung von Neuberechnungen

Schnelle Compiles: Externalisierung von TikZ Bildern

- Beispiel: Dokument mit 20 schönen TikZ Bildern.
- Keins ist unnötig, alle sollen gerendert werden
→ Externalisierung nutzen
- TikZ-Bilder in separate PDFs speichern
- Einfügen der PDFs in Hauptdokument
- Beschleunigung durch Vermeidung von Neuberechnungen
- Aktualisierung nur bei Änderungen erforderlich

Schnelle Compiles: Externalisierung von TikZ Bildern

- Beispiel: Dokument mit 20 schönen TikZ Bildern.
- Keins ist unnötig, alle sollen gerendert werden
→ Externalisierung nutzen
- TikZ-Bilder in separate PDFs speichern
- Einfügen der PDFs in Hauptdokument
- Beschleunigung durch Vermeidung von Neuberechnungen
- Aktualisierung nur bei Änderungen erforderlich
- Mit `[list and make]` ist parallele Verarbeitung möglich!
→ Bisschen komplizierter, [zum lesen](#)

Schnelle Compiles: Fallbeispiel

- Diese Präsentation ist mit \LaTeX -Beamer erstellt

Schnelle Compiles: Fallbeispiel

- Diese Präsentation ist mit \LaTeX -Beamer erstellt
- Es dauert 32s um die Präsentation *vollständig* zu kompilieren

Schnelle Compiles: Fallbeispiel

- Diese Präsentation ist mit \LaTeX -Beamer erstellt
- Es dauert 32s um die Präsentation *vollständig* zu kompilieren
→ 3 compile Runs + 1 biber Run

Schnelle Compiles: Fallbeispiel

- Diese Präsentation ist mit \LaTeX -Beamer erstellt
- Es dauert 32s um die Präsentation *vollständig* zu kompilieren
→ 3 compile Runs + 1 biber Run
- Ein compile Run dauert 7s

Schnelle Compiles: Fallbeispiel

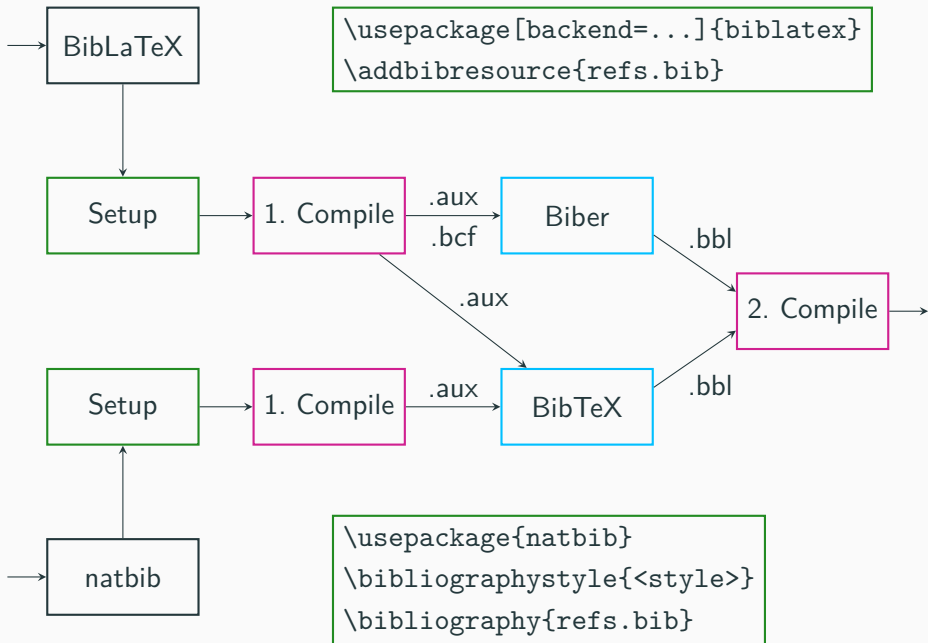
- Diese Präsentation ist mit \LaTeX -Beamer erstellt
- Es dauert 32s um die Präsentation *vollständig* zu kompilieren
→ 3 compile Runs + 1 biber Run
- Ein compile Run dauert 7s
- Mit Format-File (Template) 6s 500ms

Schnelle Compiles: Fallbeispiel



- Diese Präsentation ist mit \LaTeX -Beamer erstellt
- Es dauert 32s um die Präsentation *vollständig* zu kompilieren
→ 3 compile Runs + 1 biber Run
- Ein compile Run dauert 7s
- Mit Format-File (Template) 6s 500ms

Subfile	Ohne Template	Mit Template
0	1s 300ms	600ms
1	3s	2s 200ms
2	1s 400ms	730ms
3	2s	1s 200ms
4	1s 400ms	700ms
5	1s 300ms	600ms
6	3s 300ms	2s 500ms



Bibliographien mit \LaTeX



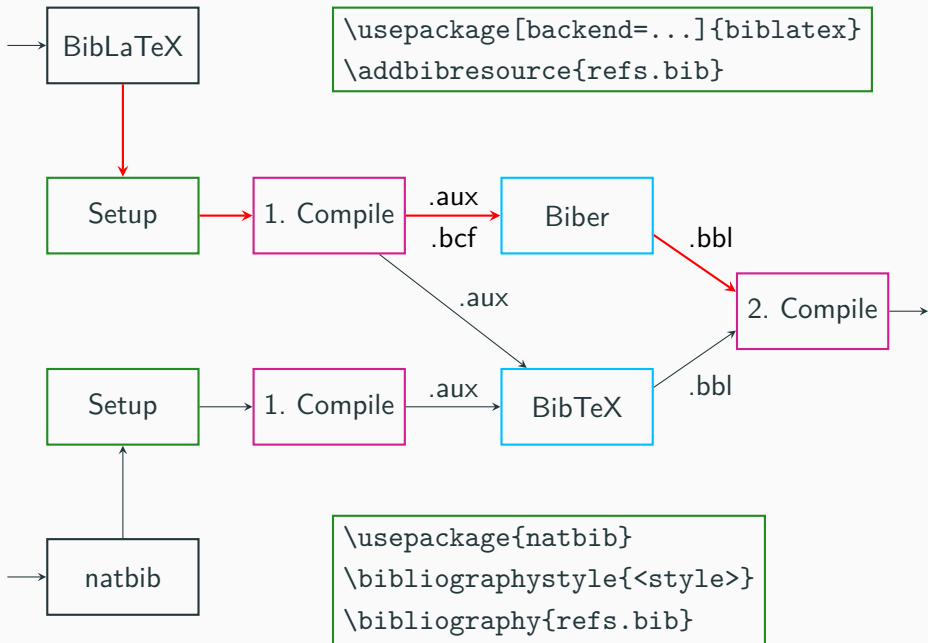
Bibliographien mit \LaTeX : Backends im Vergleich

	Bib \TeX	biber
Programmiersprache	TeX	Perl
Geschwindigkeit	Schnell	Langsam
Unicode support		
Datumsformat	Eingeschränkt	Flexibel
Sortierung	Einfach	Benutzerdefiniert
Namensformatierung	Begrenzt	Erweitert

Bibliographien mit \LaTeX : Backends im Vergleich

	Bib \TeX	biber
Programmiersprache	TeX	Perl
Geschwindigkeit	Schnell	Langsam
Unicode support		
Datumsformat	Eingeschränkt	Flexibel
Sortierung	Einfach	Benutzerdefiniert
Namensformatierung	Begrenzt	Erweitert

Was soll ich verwenden? → [biber](#)



Bibliographien mit \LaTeX : Wie zitiere ich?

Command	Output
<code>\cite{...}</code>	[1]
<code>\textcite{...}</code>	Kime, Wemheuer und Lehman [2]
<code>\parencite{...}</code>	[3]
<code>\autocite{...}</code>	[1]
<code>\footcite{...}</code>	1
<code>\footfullcite{...}</code>	2

Tabelle 2: Zitationen im Stil der IEEE

¹1.

²T. Tantau, „**TikZ and PGF Manual**,“ (15. Jan. 2023), Adresse:
<https://ftp.gwdg.de/pub/ctan/graphics/pgf/base/doc/pgfmanual.pdf>.

Bibliographien mit \LaTeX : Wie zitiere ich?

Command	Output
<code>\cite{...}</code>	[2]
<code>\textcite{...}</code>	Kime, Wemheuer und Lehman [1]
<code>\parencite{...}</code>	[4]
<code>\autocite{...}</code>	[2]
<code>\footcite{...}</code>	¹
<code>\footfullcite{...}</code>	²

Tabelle 2: Zitationen im Stil der ACM

¹2.

²T. Tantau. *TikZ and PGF Manual*. 15. Jan. 2023. URL: <https://ftp.gwdg.de/pub/ctan/graphics/pgf/base/doc/pgfmanual.pdf>.

Bibliographien mit \LaTeX : Wie zitiere ich?

Command	Output
<code>\cite{...}</code>	Knuth 1984
<code>\textcite{...}</code>	Kime, Wemheuer und Lehman (2023)
<code>\parencite{...}</code>	(Tantau, Wright und Miletić 2023)
<code>\autocite{...}</code>	(Knuth 1984)
<code>\footcite{...}</code>	1
<code>\footfullcite{...}</code>	2

Tabelle 2: Zitationen im Stil von Chicago-Authordate

1. Knuth 1984.

2. Till Tantau. 2023. „TikZ and PGF Manual“, 15. Januar 2023.
<https://ftp.gwdg.de/pub/ctan/graphics/pgf/base/doc/pgfmanual.pdf>.

Bibliographien mit \LaTeX : Wie zitiere ich?

Command	Output
<code>\cite{...}</code>	Knuth, 1984
<code>\textcite{...}</code>	Kime et al. (2023)
<code>\parencite{...}</code>	(Tantau et al., 2023)
<code>\autocite{...}</code>	(Knuth, 1984)
<code>\footcite{...}</code>	¹
<code>\footfullcite{...}</code>	²

Tabelle 2: Zitationen im Stil der APA

¹Knuth, 1984.

²Tantau, T. (2023, 15. Januar). *TikZ and PGF Manual*.
<https://ftp.gwdg.de/pub/ctan/graphics/pgf/base/doc/pgfmanual.pdf>

Bibliographien mit \LaTeX : Quellenverzeichnis

- [1] D. E. Knuth, *The TeXbook, A complete user's guide to computer typesetting with TEX*. Addison-Wesley, 1984, ISBN: 0-201-13447-0.
- [2] P. Kime, M. Wemheuer und P. Lehman, „**The biblatex Package, Programmable Bibliographies and Citations,**“ (5. März 2023), Adresse: <https://ctan.kako-dev.de/macros/latex/contrib/biblatex/doc/biblatex.pdf>.
- [3] T. Tantau, J. Wright und V. Miletić, „**The beamer class User Guide for version 3.69,**“ (20. Feb. 2023), Adresse: <https://tug.ctan.org/macros/latex/contrib/beamer/doc/beameruserguide.pdf>.
- [4] T. Tantau, „**TikZ and PGF Manual,**“ (15. Jan. 2023), Adresse: <https://ftp.gwdg.de/pub/ctan/graphics/pgf/base/doc/pgfmanual.pdf>.

Beamer



1. Dokumentklasse setzen: `\documentclass{beamer}`

1. Dokumentklasse setzen: `\documentclass{beamer}`
2. Stil wählen: `\usetheme{<stil>}`
 - metropolis
 - Copenhagen
 - Berlin

Präsentationen mit \LaTeX

1. Dokumentklasse setzen: `\documentclass{beamer}`

2. Stil wählen: `\usetheme{<stil>}`

- metropolis
- Copenhagen
- Berlin

3. Folien erstellen

```
\begin{frame}{<Titel>}
```

...

```
\end {frame}
```

Präsentationen mit L^AT_EX

1. Dokumentklasse setzen: `\documentclass{beamer}`

2. Stil wählen: `\usetheme{<stil>}`

- metropolis
- Copenhagen
- Berlin

3. Folien erstellen

```
\begin{frame}{<Titel>}
```

...

```
\end {frame}
```

- Falls Verbatim verwendet wird: Option `[fragile]` verwenden:

```
\begin{frame}[fragile]{<Titel>}
```

Präsentationen mit \LaTeX : Animationen

- Möglich mit dem Befehl `\pause`
- Letzter Schritt vor dem Finalisieren der Präsentation

Präsentationen mit \LaTeX : Animationen

- Möglich mit dem Befehl `\pause`
- Letzter Schritt vor dem Finalisieren der Präsentation

```
\begin{frame}{Pause-Beispiel}  
  \begin{itemize}  
    \item Erster Punkt  
    \pause  
    \item Zweiter Punkt  
    \pause  
    \item Dritter Punkt  
  \end{itemize}  
\end{frame}
```

Pause-Beispiel

- Erster Punkt

Pause-Beispiel

- Erster Punkt
- Zweiter Punkt

Pause-Beispiel

- Erster Punkt
- Zweiter Punkt
- Dritter Punkt

Präsentationen mit \LaTeX : Animationen

- **Aber:** `\pause` kann nur Dinge einblenden, nicht ausblenden

Präsentationen mit \LaTeX : Animationen

- **Aber:** `\pause` kann nur Dinge einblenden, nicht ausblenden
- Mehr **Commands**
 - `\uncover<2->`
 - `\only<3, 4>`

Präsentationen mit \LaTeX : Animationen

- **Aber:** `\pause` kann nur Dinge einblenden, nicht ausblenden
- Mehr **Commands**
 - `\uncover<2->`
 - `\only<3, 4>`

```
\begin{frame}{Animations-Beispiel}  
  \begin{itemize}  
    \uncover<1->{\item Erster Punkt}  
    \only<2-3>{\item Zweiter Punkt}  
    \uncover<3->{\item Dritter Punkt}  
    \only<5-6>{\item Vierter Punkt}  
  \end{itemize}  
\end{frame}
```

- Erster Punkt

Animations-Beispiel

- Erster Punkt
- Zweiter Punkt

Animations-Beispiel

- Erster Punkt
- Zweiter Punkt
- Dritter Punkt

- Erster Punkt
- Dritter Punkt

Animations-Beispiel

- Erster Punkt
- Dritter Punkt
- Vierter Punkt

TikZ



TikZ (ist kein Zeichenprogramm)

- Mit TikZ wird nicht gezeichnet, es wird programmiert

TikZ (ist kein Zeichenprogramm)

- Mit TikZ wird nicht gezeichnet, es wird programmiert

```
\begin{tikzpicture}  
    \node at (0, 0) {Hello, World!};  
\end{tikzpicture}
```

TikZ (ist kein Zeichenprogramm)

- Mit TikZ wird nicht gezeichnet, es wird programmiert

```
\begin{tikzpicture}  
  \node at (0, 0) {Hello, World!};  
\end{tikzpicture}
```

- Sehr gutes [Manual](#): 1300 Seiten, 24 Seiten Inhaltsverzeichnis

TikZ (ist kein Zeichenprogramm)

- Mit TikZ wird nicht gezeichnet, es wird programmiert

```
\begin{tikzpicture}  
  \node at (0, 0) {Hello, World!};  
\end{tikzpicture}
```

- Sehr gutes [Manual](#): 1300 Seiten, 24 Seiten Inhaltsverzeichnis
- Erweiterbarkeit: `\usetikzlibrary{...}`
 - positioning
 - calc
 - shapes

TikZ (ist kein Zeichenprogramm)

- Befehle für das Zeichnen

TikZ (ist kein Zeichenprogramm)

- Befehle für das Zeichnen
 - `\node`
 - `\draw`
 - `\fill`

TikZ (ist kein Zeichenprogramm)

- Befehle für das Zeichnen
 - `\node`
 - `\draw`
 - `\fill`
- Koordinaten
 - Normal: (x, y)
 - Polar: $(a:b:r)$

TikZ (ist kein Zeichenprogramm)

- Befehle für das Zeichnen

- `\node`
- `\draw`
- `\fill`

- Koordinaten

- Normal: `(x, y)`
- Polar: `(a:b:r)`

- Positionierung neben anderen Nodes

```
\node (hello) at (0, 0) {Hello, World!};  
\node[left=of hello] {Bonjour};
```

TikZ Optionen

- Dem `tikzpicture` können optionen mitgegeben werden

TikZ Optionen

- Dem `tikzpicture` können optionen mitgegeben werden
- Im Nachhinein kann die Größe des Bildes angepasst werden
→ `\begin{tikzpicture}[scale=2]`

TikZ Optionen

- Dem `tikzpicture` können optionen mitgegeben werden
- Im Nachhinein kann die Größe des Bildes angepasst werden
→ `\begin{tikzpicture}[scale=2]`
- Auch nodes sowie die meisten Befehle unterstützen das
 - Rotation: `[rotate=45]`
 - Farben: `[red]`, `[blue]`, `[orange]`
 - Shift: `[xshift=1cm]` / `[yshift=0.5mm]`

TikZ Optionen

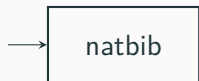
- Dem tikzpicture können optionen mitgegeben werden
- Im Nachhinein kann die Größe des Bildes angepasst werden
→ `\begin{tikzpicture}[scale=2]`
- Auch nodes sowie die meisten Befehle unterstützen das
 - Rotation: `[rotate=45]`
 - Farben: `[red]`, `[blue]`, `[orange]`
 - Shift: `[xshift=1cm]` / `[yshift=0.5mm]`
- Klassischerweise: y -Achse im Koordinatensystem nach oben
- Kann geändert werden!
→ `\begin{tikzpicture}[yscale=-1]`

TikZ Optionen

- Dem tikzpicture können optionen mitgegeben werden
- Im Nachhinein kann die Größe des Bildes angepasst werden
→ `\begin{tikzpicture}[scale=2]`
- Auch nodes sowie die meisten Befehle unterstützen das
 - Rotation: `[rotate=45]`
 - Farben: `[red]`, `[blue]`, `[orange]`
 - Shift: `[xshift=1cm]` / `[yshift=0.5mm]`
- Klassischerweise: y -Achse im Koordinatensystem nach oben
- Kann geändert werden!
→ `\begin{tikzpicture}[yscale=-1]`
- TikZ unterstützt auch animationen!
`\uncover<1-2>{\node at (0, 0) {Hello, World!};}`

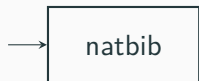

```
\node (biblatex) {BibLaTeX};  
\node[below=6cm of biblatex] (natbib) {natbib};
```

TikZ *in action*



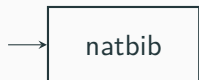
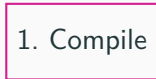
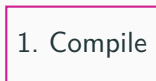
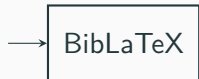
```
\node (biblatex) {BibLaTeX};  
\node[below=6cm of biblatex] (natbib) {natbib};  
  
\node[below=of biblatex] (s biblatex) {Setup};  
\node[above=of natbib] (s natbib) {Setup};
```

TikZ *in action*



```
\node (biblatex) {BibLaTeX};  
\node[below=6cm of biblatex] (natbib) {natbib};  
  
\node[below=of biblatex] (s biblatex) {Setup};  
\node[above=of natbib] (s natbib) {Setup};  
  
\node[right=of s biblatex] (1c biblatex) {1.Compile};  
\node[right=of s natbib] (1c natbib) {1.Compile};
```

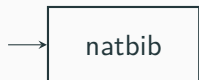
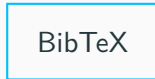
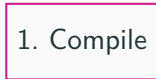
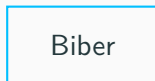
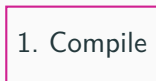
TikZ *in action*



TikZ *in action*

```
\node (biblatex) {BibLaTeX};  
\node[below=6cm of biblatex] (natbib) {natbib};  
  
\node[below=of biblatex] (s biblatex) {Setup};  
\node[above=of natbib] (s natbib) {Setup};  
  
\node[right=of s biblatex] (1c biblatex) {1.Compile};  
\node[right=of s natbib] (1c natbib) {1.Compile};  
  
\node[right=of 1c biblatex] (biber) {Biber};  
\node[right=of 1c natbib] (bibtex) {BibTeX};
```

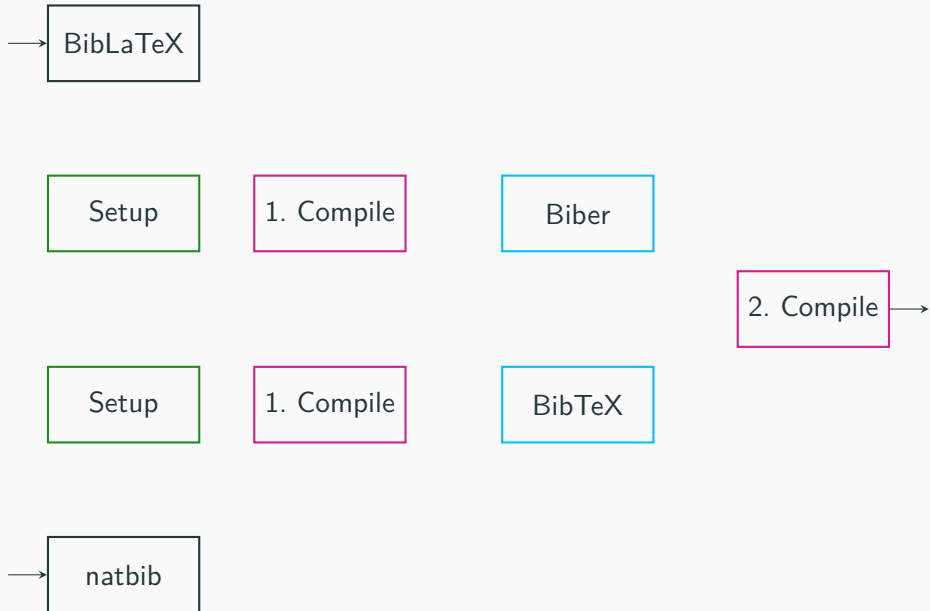
TikZ *in action*



TikZ *in action*

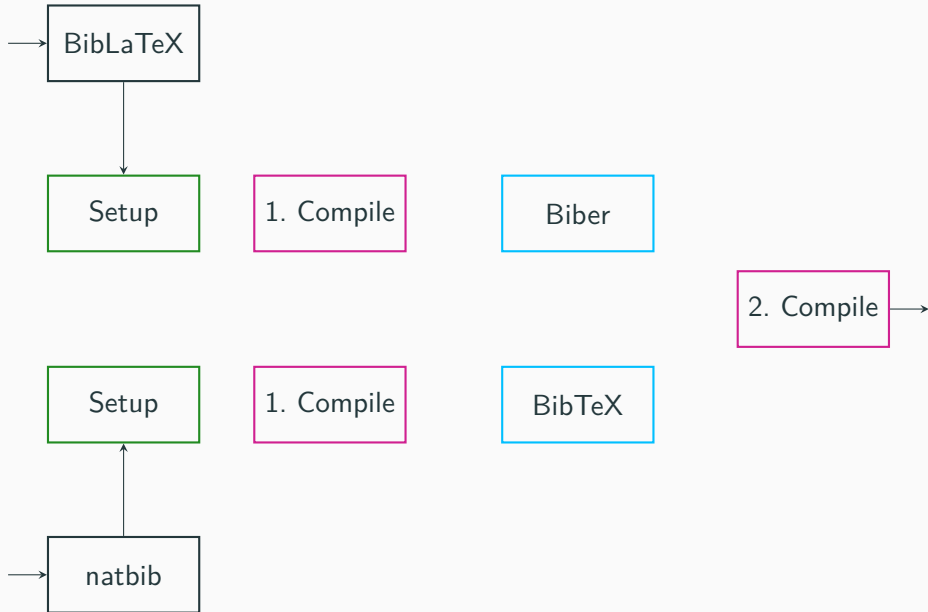
```
\node (biblatex) {BibLaTeX};  
\node[below=6cm of biblatex] (natbib) {natbib};  
  
\node[below=of biblatex] (s biblatex) {Setup};  
\node[above=of natbib] (s natbib) {Setup};  
  
\node[right=of s biblatex] (1c biblatex) {1.Compile};  
\node[right=of s natbib] (1c natbib) {1.Compile};  
  
\node[right=of 1c biblatex] (biber) {Biber};  
\node[right=of 1c natbib] (bibtex) {BibTeX};  
  
\node[right=of $(biber)!0.5!(bibtex)$] (2 compile)  
  ↪ {2. Compile};
```

TikZ *in action*



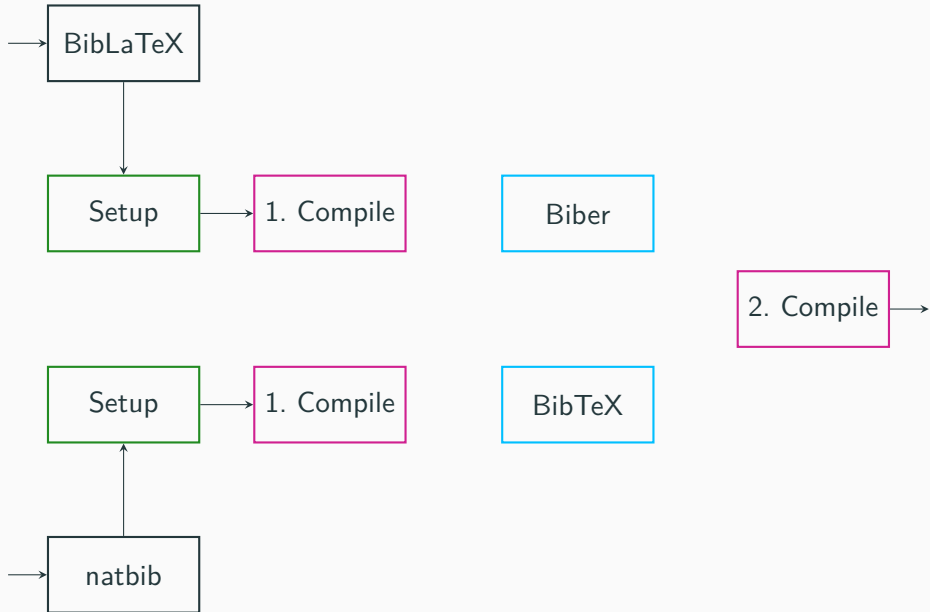
```
\draw[->] (biblatex) -- (setup biblatex);  
\draw (natbib) -> (setup natbib);
```

TikZ in action



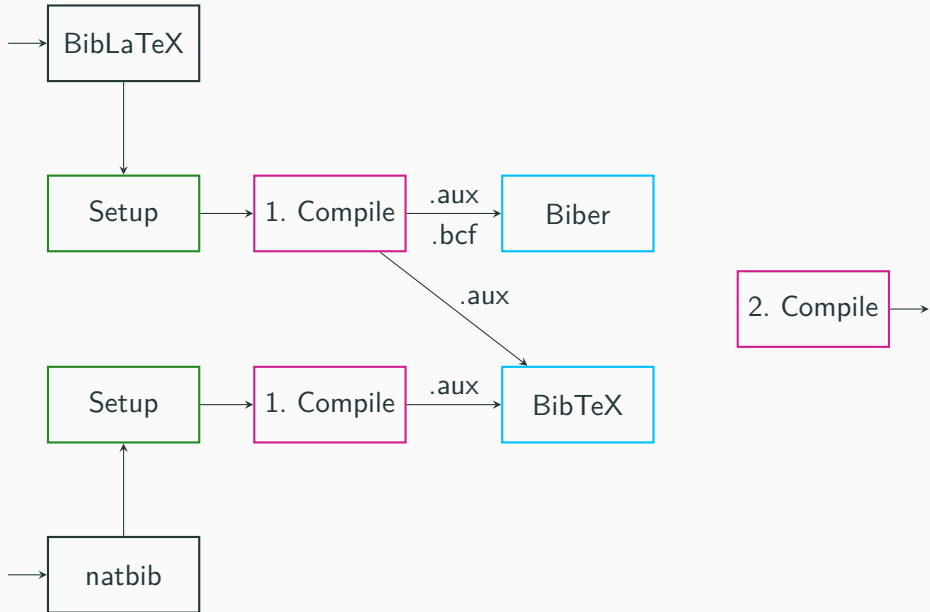
```
\draw[->] (biblatex) -- (setup biblatex);  
\draw (natbib) -> (setup natbib);  
  
\draw[->] (setup biblatex) -- (1c biblatex);  
\draw[->] (setup natbib) -- (1c natbib);
```

TikZ in action



```
\draw[->] (biblatex) -- (setup biblatex);  
\draw (natbib) -> (setup natbib);  
  
\draw[->] (setup biblatex) -- (1c biblatex);  
\draw[->] (setup natbib) -- (1c natbib);  
  
\draw[->] (1c natbib) -- node[above] {.aux} (bibtex);  
\draw[->] (1c biblatex) -- node[above] {.aux}  
  ↪ node[below] {.bcf} (biber);  
\draw[->] (1c biblatex) -- node[above right=-0.1]  
  ↪ {.aux} (bibtex);
```

TikZ in action



TikZ *in action*

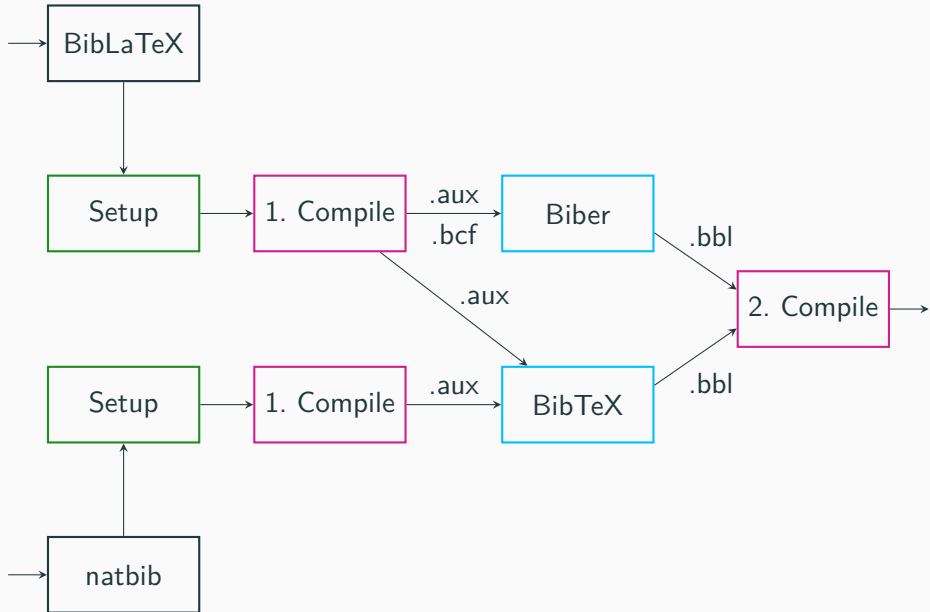
```
\draw[->] (biblatex) -- (setup biblatex);
\draw (natbib) -> (setup natbib);

\draw[->] (setup biblatex) -- (1c biblatex);
\draw[->] (setup natbib) -- (1c natbib);

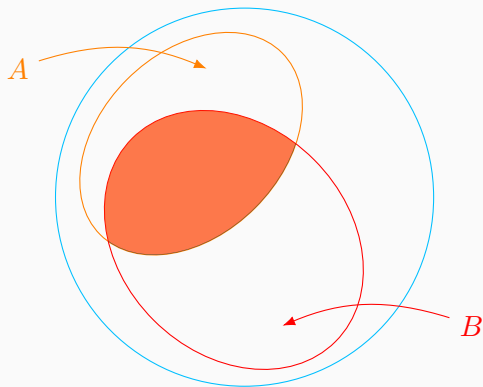
\draw[->] (1c natbib) -- node[above] {.aux} (bibtex);
\draw[->] (1c biblatex) -- node[above] {.aux}
  ↪ node[below] {.bcf} (biber);
\draw[->] (1c biblatex) -- node[above right=-0.1]
  ↪ {.aux} (bibtex);

\draw[->] ($ (biber.east)!0.5!(biber.south east)$) --
  ↪ ($ (2 compile.west)!0.5!(2 compile.north west)$);
\draw[->] ($ (bibtex.east)!0.5!(bibtex.north east)$)
  ↪ -- ($ (2 compile.west)!0.5!(2 compile.south
  ↪ west)$);
```

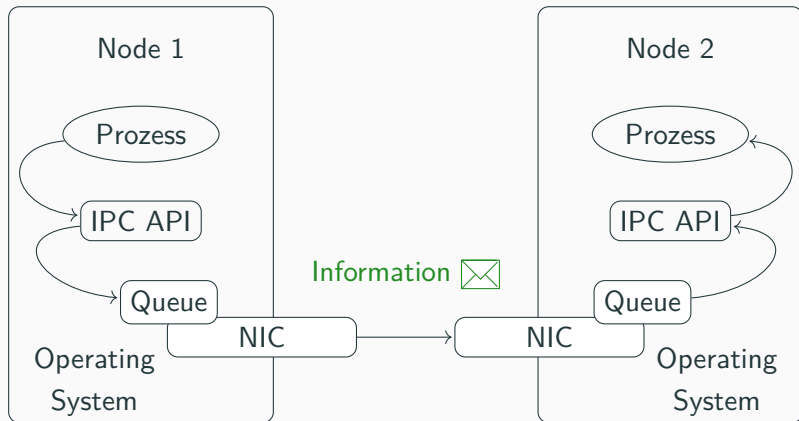
TikZ in action



Was TikZ so alles kann: Intersections

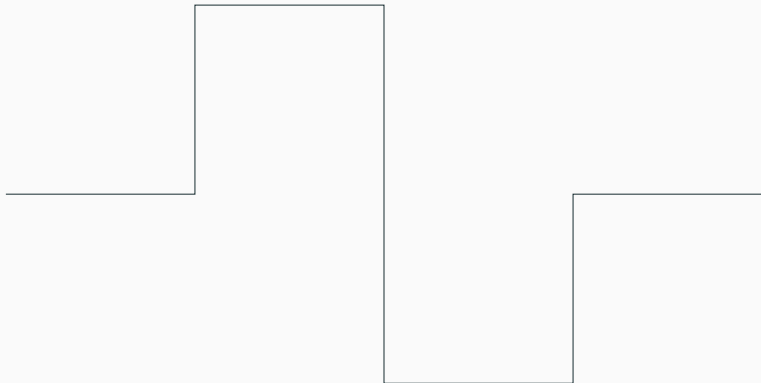


Was TikZ so alles kann: Prozessdiagramme

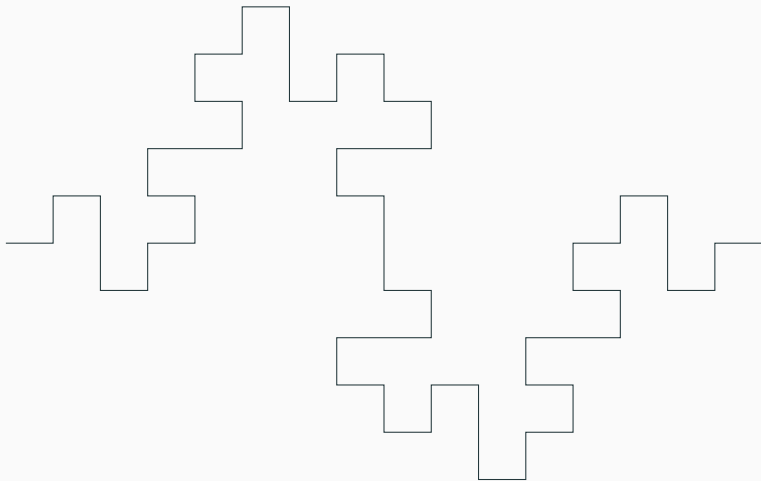


Bonuspunkte, wenn ihr wisst wo dieses Bild herkommt ;)

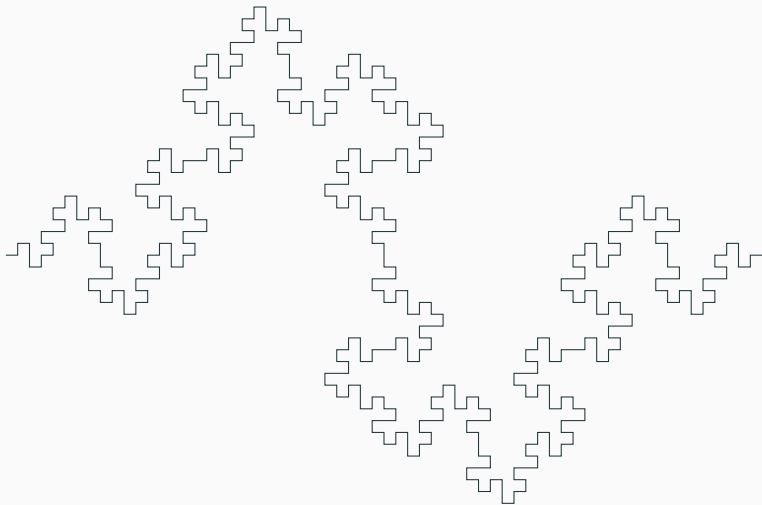
Was TikZ so alles kann: Fraktale



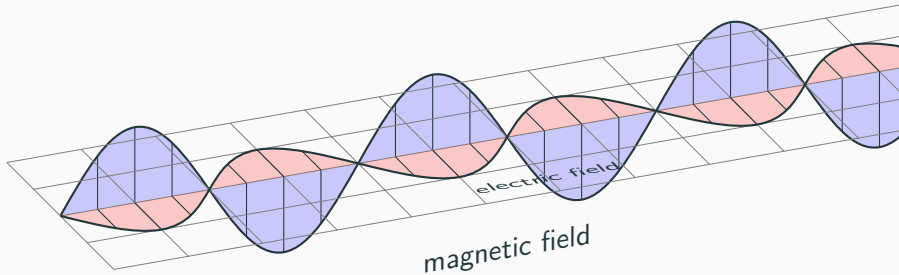
Was TikZ so alles kann: Fraktale



Was TikZ so alles kann: Fraktale



Was TikZ so alles kann: 3d Bilder



Was TikZ so alles kann: Mindmaps

