

Diplomarbeit schreiben mit Typst

Clemens Koza

2024

TGM, HIT

made with Typst and Touying



<https://github.com/TGM-HIT/typst-thesis-workshop/blob/main/slides/typst-workshop-handout.pdf>

Anforderungen an die Diplomarbeit

Inhalt

- aussagekräftig
- nachvollziehbar: **Quellenangaben, Verweise**

Form

- übersichtlich: **Gliederung, Inhaltsverzeichnis**
- ansprechend: **Qualität der Darstellung**
- einheitlich: **Farben, Schriften, Abstände**, ... zentral festgelegt

Aufwand – ihr wollt (vermutlich) ...

- ... Formatierungen nicht häufig anpassen/korrigieren müssen
- ... dass Bilder verschieben das Dokument nicht zerstört
- ... ein leicht zu verwendendes Werkzeug mit Features zur Kollaboration

Die Optionen

Microsoft Word

- + WYSIWYG
- + einfache Bedienung
- + Kollaboration (Office 365)
- Bilder verschieben
- keine „Erweiterbarkeit“
- kein automatisches Syntax-Highlighting
- wegen WYSIWYG: „unsichtbare“ Formatierungen

LaTeX

- + hohe Output-Qualität
- + Programmiersprache: Features anpassbar
- + Kollaboration (Overleaf, git-Versionierung)
- langsam
- schwer zu erlernende Sprache (Makro-basiert)
- schwer verständliche Fehlermeldungen

Typst

- + gute Output-Qualität
- + Programmiersprache mit vertrautem Aufbau
- + schnell
- + Kollaboration (Typst App, git-Versionierung)
- weniger Lernunterlagen
- manche App-Features nur mit Typst Pro
- keine deutsche Rechtschreibprüfung

1 **= Überschrift**

typ

2 Ein Code-Beispiel:

3 ```java

4 private int i; // Syntax Highlighting

5 ```

6

7 **== Auflistungen**

8 - Liste

9 - mit Bullets

10

11 + Liste

12 + geschachtelt

13 + mit Numerierung

Überschrift

Ein Code-Beispiel:

```
private int i; // Syntax Highlighting
```

Auflistungen

- Liste
 - mit Bullets
- 1. Liste
 - 1. geschachtelt
- 2. mit Numerierung

Typst-Beispiel – programmierbar

```
1  #for i in array.range(1, 5) [ typ
2    - Nummer #i
3    #if calc.odd(i) [(wichtig!)]
4  ]
5
6  Blindtext: #lorem(8)
7
8  #table(
9    columns: 3,
10   [Col 1], [Col 2], [Col 3],
11   ..array.range(9).map(i => {
12     let num = i + 1
13     [\\##num]
14   })
15 )
```

- Nummer 1 (wichtig!)
- Nummer 2
- Nummer 3 (wichtig!)
- Nummer 4

Blindtext: Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Col 1	Col 2	Col 3
#1	#2	#3
#4	#5	#6
#7	#8	#9

Typst-Beispiel – Formatierung

```
1  = zu unscheinbar ...
2
3  #show heading: set text(1.4em, red)
4
5  === ... besser
6
7  #set table(stroke: (x, y) => (
8    bottom: if y == 0 { 1pt },
9  ))
10
11 #table(
12   columns: 4,
13   align: right,
14   ..array.range(12).map(i => [#i])
15 )
```

typ

zu unscheinbar ...

... besser

0	1	2	3
4	5	6	7
8	9	10	11

```
1 #set math.equation(numbering: "(1)") typ
2
3 Nehmen wir den Satz
4 des Pythagoras:
5
6 $ a^2 + b^2 = c^2 $ <pythagoras>
7
8 In @pythagoras sind $a$ und $b$ die
9 Katheten und $c$ ist die Hypothenuse.
```

Nehmen wir den Satz des Pythagoras:

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

In Gleichung 1 sind a und b die Katheten und c ist die Hypothenuse.

Typst-Beispiel – LaTeX-Vergleich

How do I create a command, say `\multiply{<first_number>}{<last_number>}`, using LaTeX3, that multiplies all the integers from `<first_number>` to `<last_number>`?

– <https://tex.stackexchange.com/questions/712967/command-for-multiplying-integers>

LaTeX

```
1 \def\xmultiply#1#2{%  
2   \ifnum#1=\numexpr#2\relax\the\numexpr#1\relax\else  
3   \the\numexpr(#2)*\xmultiply{#1}{#2-1}\relax\fi  
4  
5 \xmultiply{4}{9}
```

latex

(die akzeptierte Antwort, die LaTeX3 konform war, hatte 10 statt 3 Zeilen für die Funktion)

Typst-Beispiel – LaTeX-Vergleich

How do I create a command, say `\multiply{<first_number>}{<last_number>}`, using LaTeX3, that multiplies all the integers from `<first_number>` to `<last_number>`?

– <https://tex.stackexchange.com/questions/712967/command-for-multiplying-integers>

LaTeX

```
1 \def\xmultiply#1#2{%  
2   \ifnum#1=\numexpr#2\relax\the\numexpr#1\relax\else  
3   \the\numexpr(#2)*\xmultiply{#1}{#2-1}\relax\fi  
4  
5 \xmultiply{4}{9}
```

latex

(die akzeptierte Antwort, die LaTeX3 konform war, hatte 10 statt 3 Zeilen für die Funktion)

Typst

```
1 #let xmultiply(a, b) = array.range(a, b+1).product()  
2  
3 #xmultiply(4, 9)
```

typ

Typst App

- <https://typst.app>
 - ähnliche Bedienung wie z.B. Overleaf (aber schneller)

Lokale Installation

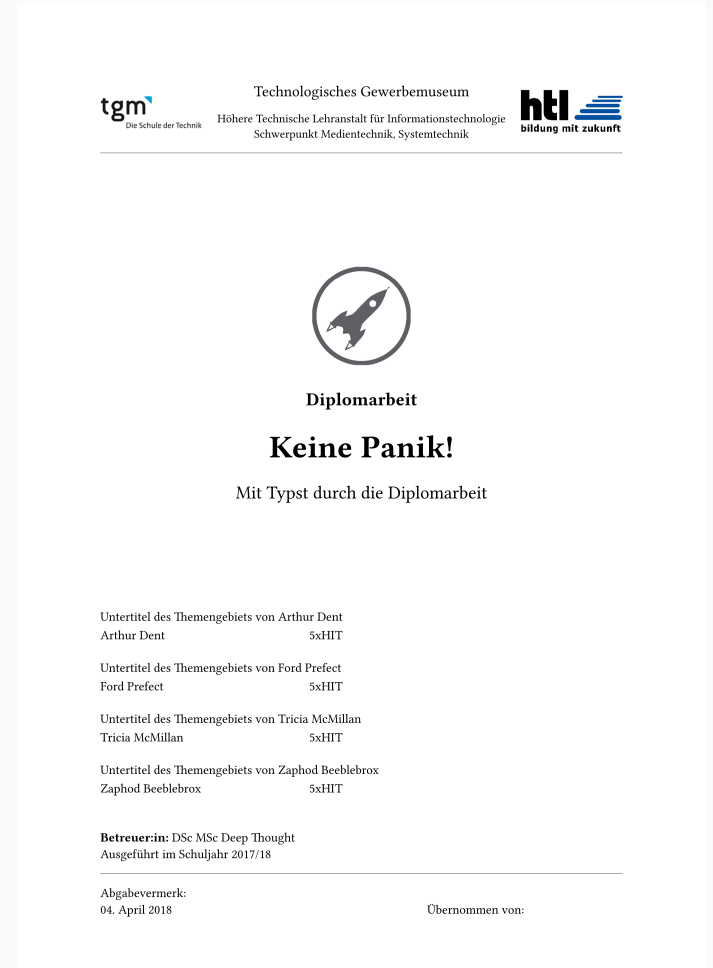
- Compiler: <https://github.com/typst/typst>
 - `typst init` – Projekt initialisieren
 - `typst compile` – pdf/svg/png exportieren
 - `typst watch` – bei Änderungen automatisch kompilieren
- Editor-Plugin: Tinymist – <https://myriad-dreamin.github.io/tinymist/>
 - für VS Code, NeoVim, ...
 - Syntax Highlighting, Live Preview, Export, ...

- der existierenden LaTeX-Vorlage nachempfunden
- kann mit wenigen Klicks/Befehlen verwendet werden
- enthält als „Vorwort“ Anleitungen zu Literatur, Verweisen, Abbildungen, ...

Links

- <https://typst.app/universe/package/tgm-hit-thesis>
- <https://github.com/TGM-HIT/typst-diploma-thesis>

jetzt: Demo!



- Formatierung
- Quellen hinzufügen, verwenden
- Abbildungen/Gleichungen/Codebeispiele einfügen, verwenden
- Daten einlesen
- Wichtige/nützliche Pakete:
CeTZ, Fletcher, Timeliney, Subpar, Physica, MiTeX, (Touying)
- weitere Unterlagen:
 - Dokumentation: <https://typst.app/docs/>
 - Bibliotheken: <https://typst.app/universe/>
 - Discord: <https://discord.gg/2uDybryKPe>