

# 1 Typst Syntax und Requirements

Die [Typst-Dokumentation](#) ist die wichtigste Anlaufstelle für Wissen rund um Typst.

Die [Syntax-Referenz](#) bietet eine Übersicht über die grundlegende Syntax und die wichtigsten Funktionen.

Bei komplexeren Problemsstellungen ist auch der Typst Discord-Server empfehlenswert. Der aktuelle Link befindet sich in der Kopfzeile der [Typst-Website](#).

## 1.1 Schriftarten

Damit das Dokument korrekt kompilieren kann, ist es notwendig, dass folgende Schriftarten am System vorhanden sind:

- **Cambria:** Verwendet für *Text* und *Überschriften*.
- **Source Code Pro:** Verwendet für *Quellcode*.

Unter Windows sollten die Schriftarten schon vorhanden sein. Unter Linux können diese mit einem Package Manager heruntergeladen werden:

- **Arch Linux:** `yay -S ttf-vista-fonts adobe-source-code-pro-fonts`
- **Ubuntu:** `sudo apt install ttf-mscorefonts-installer` # Source Code Pro exkludiert.



Unter Ubuntu ist es am leichtesten sich *Source Code Pro* auf Google-Fonts herunterzuladen: <https://fonts.google.com/specimen/Source+Code+Pro>

Unter Arch Linux kann, falls yay nicht installiert ist, kann das Package manuell installiert werden:

```
1 git clone https://aur.archlinux.org/ttf-vista-fonts.git
2 makepkg -i -D ./ttf-vista-fonts
3 sudo pacman -S adobe-source-code-pro-fonts
```

## 2 Templatespezifische-Funktionen

**Angeben des Kapitelautors:** Die Autorenangabe muss **vor** der Überschrift erfolgen.

```
1 #htl3r.author("Viktor Kreuzer")
2 = Templatespezifische-Funktionen
```

Output: Im Footer wird der Autor gelistet.

**Breite von Figures festlegen:**

```
1 #htl3r.fspace(
2   total-width: 50%,
3   // hier können mehrere Figures nebeneinandergestellt werden!
4   figure(image("assets/16x9.png"), caption: [Bild mit 16:9]),
5   figure(image("assets/1x1.png"), caption: [Bild mit 1:1]),
6 )
```

Output:

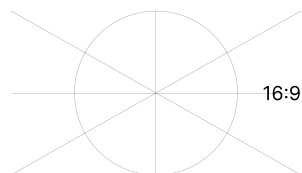


Abbildung 2.1: Bild mit 16:9

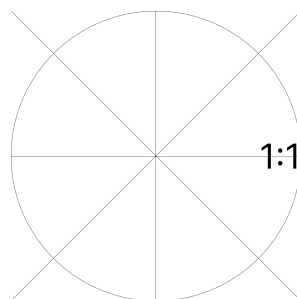


Abbildung 2.2: Bild mit 1:1

### Codeblock als Figure:

```
#htl3r.code(description: [Bashskript], caption: [Codeblock Beispiel])[
  ``bash
  #!/bin/bash
  echo test123
  ``
]
```

Output:

Bashskript

```
1 #!/bin/bash
2 echo test123
```

Quellcode 2.1: Codeblock Beispiel

### Codeblock aus einem File importiert:

```
1 #htl3r.code-file(
2   lang: "bash",
3   filename: [Test],
4   text: read("assets/code-example.sh"),
5 )
```

Output:

Test

```
1 #!/bin/bash
2 echo "Hello, World!"
```

### Todo-Eintrag:

```
1 #htl3r.todo[SDO stolz machen.]  
2 #htl3r.info[Information hier einfügen.]  
3 #htl3r.warn[Achtung!]
```

Output:



SDO stolz machen.



Information hier einfügen.



Achtung!

### Inline-Todo-Eintrag:

```
1 #lorem(10) #htl3r.inline-todo[Das geht so nicht!] #lorem(20)
```

Output:

▶ Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do. **Das geht so nicht!** Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua quaerat.

## Breadcrumbs

Können zur Darstellung eines Pfades oder einer Menüstruktur verwendet werden.

```
1 #lorem(10)
2 #htl3r.breadcrumbs(("Das", "ist", "eine", "mögliche", "Menüstruktur"))
3 #lorem(10)
```

Output:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do. Das ist eine mögliche Menüstruktur Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do.

## Abkürzungen:

Folgende Arten der Abkürzungen werden unterstützt:

```
1 Kurzform Singular: #htl3r.short[da] \
2 Kurzform Plural: #htl3r.shortpl[da] \
3 Langform Singular: #htl3r.long[da] \
4 Langform Plural: #htl3r.longpl[da] \
5 Vollform Singular: #htl3r.full[da] \
6 Vollform Plural: #htl3r.fullpl[da] \
```

Output:

Kurzform Singular: DA<sup>[1]</sup>

Kurzform Plural: DA

Langform Singular: Diplomarbeit

Langform Plural: Diplomarbeiten

Vollform Singular: Diplomarbeit (DA)

Vollform Plural: Diplomarbeiten (DA)

<sup>[1]</sup>Diplomarbeit: Abschlussarbeit einer BHS

## Zitieren:

Die Literatur kann entweder in einer BibLaTeX .bib Datei oder in einer Datei im [Hayagriva-Format](#) hinterlegt werden.

Bei aufeinanderfolgendem Zitieren der gleichen Quelle wird diese mit „ebd.“ (ebenda) abgekürzt.



```
1 @htl3r-website[comp] \
2 @htl3r-website[#htl3r.comp[S. 12]]
```

Output:

```
(vgl. HTL Rennweg, 2024)
(vgl. ebd., S. 12)
```

## Beispiel:

```
refs.yml
1 htl3r-website:
2   author: HTL Rennweg
3   title: Schulwebsite
4   type: Web
5   url: https://www.htlrennweg.at/
6   date: 2024-12-19
```

### 1. direktes Zitat

```
1 // Direktes Zitat = normaler Typst Syntax
2 @htl3r-website[]
```

Output:

```
(HTL Rennweg, 2024)
```

## 2. direktes Zitat mit Supplement

```
1 @htl3r-website[, S. 12]
```

Output:

(HTL Rennweg, 2024, S. 12)

## 3. indirektes Zitat

```
1 // Merkhilfe: comp ... comparison  
2 @htl3r-website[comp]
```

Output:

(vgl. HTL Rennweg, 2024)



Der String „comp“ ist im Template hardcoded und kann damit **NICHT** als Supplement eingesetzt werden!

## 4. indirektes Zitat mit Supplement

```
1 // Merkhilfe: comp ... comparison  
2 @htl3r-website[#htl3r.comp[S. 12]]
```

Output:

(vgl. HTL Rennweg, 2024, S. 12)

# Abkürzungsverzeichnis

**DA:** Diplomarbeit  
S.: 5

*Glossar (S. 9)*



---

# Glossar

**Diplomarbeit:** Abschlussarbeit einer BHS