

Typst – Hat L^AT_EX abgedankt?



Eine kurze Einführung in Typst

Tristan Pieper tristan.pieper@uni-rostock.de · 1. Juni 2023

Inhaltsverzeichnis

1. Kurzes Kennenlernen	3
2. Probleme von L ^A T _E X.....	4
3. Die Lösung aller Probleme(?)	12
4. Die Web-App	14
5. Grundlegende Formatierung	16
6. Die Typst-Dokumentation...	33
7. Eigene Templates und Skripts	45
8. Typst kann noch mehr!	50
9. Abschluss und Weiteres	54

1. Kurzes Kennenlernen

2. Probleme von L^AT_EX

2.1. Alles begann mit...



Donald E. Knuth¹ (geb. 10. Januar 1938)

2.2. Dann kam...



Leslie Lamport¹ (geb. 7. Februar 1941)

2.3. Die Probleme

1. Riesige Programmgröße
2. Auswahl an Compilern
3. Unverständliche Fehler

2.4. Größe des Programms

```
% du -sch /usr/share/texmf-dist/* | sort -hr
2,5G    insgesamt
1,9G    /usr/share/texmf-dist/fonts
499M    /usr/share/texmf-dist/tex
58M     /usr/share/texmf-dist/scripts
44M     /usr/share/texmf-dist/tex4ht
24M     /usr/share/texmf-dist/bibtex
15M     /usr/share/texmf-dist/metapost
7,5M    /usr/share/texmf-dist/dvips
3,8M    /usr/share/texmf-dist/xindy
3,4M    /usr/share/texmf-dist/ls-R
2,6M    /usr/share/texmf-dist/asymptote
1,7M    /usr/share/texmf-dist/context
516K    /usr/share/texmf-dist/omega
344K    /usr/share/texmf-dist/makeindex
```

Verglichen mit 21MB des Typst-Compilers...

```
% du -sch /usr/bin/typst
21M    /usr/bin/typst
21M    insgesamt
```

2.5. Die Vielfalt

„ \LaTeX “ ist kein Programm, sondern:

- pdfTeX
- LuaTeX
- XeTeX
- MikTeX
- KaTeX
- ...

2.6. Beispiel-Fehlermeldung (Typst)

Typst:

```
$  
+ Dies  
+ Ist  
+ Eine  
+ Liste!
```

```
error: expected dollar sign  
└ test.typ:5:8  
5 | + Liste!  
   ^
```

2.7. Beispiel-Fehlermeldung (L^AT_EX)

L^AT_EX:

```
\documentclass{article}

\begin{document}

\$

\begin{enumerate}
\item Dies
\item Ist
\item Eine
\item Liste!
\end{enumerate}

\end{document}
```

```
Latexmk: This is Latexmk, John Collins, 17 Mar. 2022. Version 4.77,
version: 4.77.
Latexmk: applying rule 'pdflatex'...
Rule 'pdflatex': File changes, etc:
    Changed files, or newly in use since previous run(s):
    /path/Desktop/Projekte/Typst/typst-seminar/.lt/test.tex
    test.tex
Rule 'pdflatex': The following rules & subrules became out-of-date:
    pdflatex
-----
Run number 1 of rule 'pdflatex'
-----
-----
Running 'pdflatex -synctex=1 -interaction=nonstopmode -file-line-
error -recorder "/path/Desktop/Projekte/Typst/typst-seminar/.lt/
test.tex"'
-----
This is pdfTeX, Version 3.141592653-2.6-1.40.24 (TeX Live 2022/Arch
Linux) (preloaded format=pdflatex)
  restricted \write18 enabled.
entering extended mode
(/path/Desktop/Projekte/Typst/typst-seminar/.lt/test.tex
LaTeX2e <2021-11-15> patch level 1
L3 programming layer <2022-04-10> (/usr/share/texmf-dist/tex/latex/
base/article.cls
Document Class: article 2021/10/04 v1.4n Standard LaTeX document
class
(/usr/share/texmf-dist/tex/latex/base/size10.clo)) (/usr/share/
texmf-dist/tex/latex/l3backend/l3backend-pdftex.def) (./test.aux)
/path/Desktop/Projekte/Typst/typst-seminar/.lt/test.tex:5: Missing
$ inserted.
<inserted text>
                $
1.5

[1{/var/lib/texmf/fonts/map/pdftex/updmap/pdftex.map}] (./
test.aux) )
(see the transcript file for additional information)</usr/share/
texmf-dist/fonts/type1/public/amsfonts/cm/cmr10.pfb>
Output written on test.pdf (1 page, 13646 bytes).
SyncTeX written on test.synctex.gz.
Transcript written on test.log.
Latexmk: If appropriate, the -f option can be used to get latexmk
  to try to force complete processing.
Latexmk: Getting log file 'test.log'
Latexmk: Examining 'test.fls'
Latexmk: Examining 'test.log'
Latexmk: Log file says output to 'test.pdf'
Latexmk: Errors, so I did not complete making targets
Collected error summary (may duplicate other messages):
```

3. Die Lösung aller Probleme(?)

3.1. Ein kleiner Vergleich

LaTeX	Typst	Ergebnis
<pre>\documentclass{article} \begin{document} \begin{enumerate} \item Dies \item Ist \item Eine \item Liste! \end{enumerate} \end{document}</pre>	<ul style="list-style-type: none">+ Dies+ Ist+ Eine+ Liste!	<ol style="list-style-type: none">1. Dies2. Ist3. Eine4. Liste!

4. Die Web-App

4.1. Ab ans Werk!

Vorteile:

- alle Dateien online
- verschiedene Projekte erstellbar
- guter online Editor
- als Team gleichzeitig an Dateien arbeiten
- eingebaute Dokumentation

<https://typst.app/>

5. Grundlegende Formatierung

5.1. Überschriften

```
// Hier ein Kommentar,  
// er wird ignoriert!
```

= Überschrift 1!
== Überschrift 2!
== Überschrift 3!
Text

Neuer Abstand!

1. Überschrift 1!

1.1. Überschrift 2!

1.1.1. Überschrift 3!

Text

Neuer Absatz!

5.2. Absätze

= Mein cooler Titel

In Typst beginnt ein neuer Absatz, sobald im Code eine freie Zeile steht.

Leider sind standardmäßig Absätze linksbündig und nicht Blocksatz. Wie man das ändert, lernen wir gleich!

1. Mein cooler Titel

In Typst beginnt ein neuer Absatz, sobald im Code eine freie Zeile steht.

Leider sind standardmäßig Absätze linksbündig und nicht Blocksatz. Wie man das ändert, lernen wir gleich!

5.3. Listen

- + Eine nummerierte Liste
 - + Kann sehr schön sein!
 - 1. So geht sie auch!
 - 2. Jawohl!
 - Und hier nicht-nummeriert!
 - Ganz ohne Nummern!
- 1. Eine nummerierte Liste
 - 2. Kann sehr schön sein!
 - 1. So geht sie auch!
 - 2. Jawohl!
 - Und hier nicht-nummeriert!
 - Ganz ohne Nummern!

5.4. Schriftart

```
#text(font: "Arial", [Hallo Welt!])
```

Hallo Welt!

```
#text(font: "Courier New", [Hallo  
Welt!])
```

Hallo Welt!

```
#text(font: "New Computer Modern",  
[Hallo Welt!])
```

Hallo Welt!

5.5. Text¹

```
*Hello!* #strong[Hello!]
```

Hello! Hello!

```
_Hello!_ #emph[Hello!]
```

Hallo! Hallo!

```
Hello!#super([Hello!])
```

Hello!^{Hello!}

```
Hello!#sub([Hello!])
```

Hello!_{Hallo!}

```
#text(fill: red, [Hello rot!])
```

Hallo rot!

```
#text(fill: rgb("#ff00ff"), [Hello  
pink!])
```

Hallo pink!

```
#text(fill: rgb("#ff00ff"),  
smallcaps(strong(emph[Hello pink!])))
```

HALLO PINK!

5.6. Ausrichtung

```
#align(left, [Hallo!])      Hallo!  
#align(center, [Hallo!])    Hallo!  
#align(right, [Hallo!])     Hallo!
```

5.7. Abstände #1

```
#align(right, [Vertikaler])
#v(2cm)
Horizontaler #h(2cm) Abstand
```

Vertikaler

Horizontaler Abstand

5.8. Abstände (Vertikal)

Listen-Beispiel von Folie 19 mit vertikalen Abständen.

- + Eine nummerierte Liste
 - + Kann sehr schön sein!
- #v(2em)
1. So geht sie auch!
 2. Jawohl!
- #v(2em)
- Und hier nicht-nummeriert!
 - Ganz ohne Nummern!

Listen-Beispiel von Folie 19 mit vertikalen Abständen.

1. Eine nummerierte Liste
 2. Kann sehr schön sein!
-
1. So geht sie auch!
 2. Jawohl!
-
- Und hier nicht-nummeriert!
 - Ganz ohne Nummern!

5.9. Abstände (Horizontal)

- + Gott kommen aufgrund seines vollkommenen Wesens alle vollkommenen Eigenschaften zu.
- + Zu existieren ist vollkommener als nicht zu existieren.
- + Also ist Existenz eine vollkommene Eigenschaft.
- + Also existiert Gott. #h(1fr) QED

1. Gott kommen aufgrund seines vollkommenen Wesens alle vollkommenen Eigenschaften zu.
2. Zu existieren ist vollkommener als nicht zu existieren.
3. Also ist Existenz eine vollkommene Eigenschaft.
4. Also existiert Gott.

QED

5.10. Bilder

```
#image(height: 50%, "leslie_lamport.png")
```



5.11. Tabellen

```
#table(  
  columns: (auto, 3cm, auto),  
  [Hallo1!],  
  [2a],  
  [Hallo3!],  
  [Welt1!],  
  [2b],  
  [Welt3!]  
)
```

Hallo1!	2a	Hallo3!
Welt1!	2b	Welt3!

5.12. Mathematik

```
$ sum_(k=0)^n k = 1 + ... + n $  
  
$ A = pi r^2 $  
  
$ "area" = pi dot.op "radius"^2 $  
  
$ cal(A) :=  
  { x in RR | x "is natural" } $  
  
$ frac(a^2, 2) $
```

$$\sum_{k=0}^n k = 1 + \dots + n$$

$$A = \pi r^2$$

$$\text{area} = \pi \cdot \text{radius}^2$$

$$\mathcal{A} := \{x \in \mathbb{R} \mid x \text{ is natural}\}$$

$$\frac{a^2}{2}$$

5.13. Set-Regeln¹

Hier ist noch die Standard-Schriftart! Q

```
#set text(font: "New Computer Modern", fill: blue)
```

Q Ab jetzt ist alles vollkommen in der anderen Schriftart und sogar blau!

```
#set par(first-line-indent: 1.5em, justify: true)
```

Ab jetzt wird jede erste Zeile eines Absatzes eingerückt und Blocksatz!

Wirklich, versprochen!
`#lorem(20)`

Hier ist noch die Standard-Schriftart! Q

Q Ab jetzt ist alles vollkommen in der anderen Schriftart und sogar blau!

Ab jetzt wird jede erste Zeile eines Absatzes eingerückt und Blocksatz!

Wirklich, versprochen! Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliquam quaerat.

5.14. Show-Regeln¹

```
#show heading: set text(navy)
```

==== Hallo!

===== Welt!

```
// aus dem offiziellem  
// Typst-Tutorial  
#show "Project": smallcaps  
#show "badly": "great"
```

We started Project in 2019
and are still working on it.
Project is progressing badly.

5.14.1. Hallo!

5.14.1.1. Welt!

We started PROJECT in
2019 and are still
working on it. PROJECT
is progressing great.

```

\documentclass[14pt,a4paper]{extarticle}
\usepackage{bold-extra}
\usepackage{amssymb}
\usepackage[T1]{fontenc}
\usepackage[left=2cm,right=2cm,top=2cm,bottom=2cm]
{geometry}

\setlength{\parskip}{0.65em}
\setlength{\parindent}{0pt}

\begin{document}
    \noindent\textbf{\textsc{Definition 1.}}\\
\textit{Sei  $D \subsetneq \mathbb{R}$  und sei  $f: D \rightarrow \mathbb{R}$  eine Funktion.  $f$  ist stetig in  $x_0 \in D$  genau dann, wenn die folgende Aussage gilt:}

    \textit{Für alle  $\epsilon > 0$  existiert ein  $\delta > 0$ , sodass  $|f(x) - f(x_0)| < \epsilon$  für alle  $x \in D$  mit  $|x - x_0| < \delta$ .}

    \textit{Oder Alternativ:  $\forall \epsilon > 0 \exists \delta > 0 \forall x \in D : |y - y_0| < \delta \Rightarrow |f(x) - f(x_0)| < \epsilon$ }

    \bigskip
    (\LaTeX)
\end{document}

```

```

#set page(margin: 2cm)
#set text(size: 14pt, font: "New Computer Modern")
#set par(justify: true)

*#smallcaps([Definition 1.])* _Sei  $D \subsetneq \mathbb{R}$  und sei  $f: D \rightarrow \mathbb{R}$  eine Funktion.  $f$  ist stetig in  $x_0 \in D$  genau dann, wenn die folgende Aussage gilt:_

_Für alle  $\epsilon > 0$  existiert ein  $\delta > 0$ , sodass  $|f(x) - f(x_0)| < \epsilon$  für alle  $x \in D$  mit  $|x - x_0| < \delta$ ._

_Oder Alternativ:  $\forall \epsilon > 0 \exists \delta > 0 \forall x \in D : |y - y_0| < \delta \Rightarrow |f(x) - f(x_0)| < \epsilon$ _

#v(1em)
(Typst)

```

DEFINITION 1. Sei $D \subseteq \mathbb{R}$ und sei $f : D \rightarrow \mathbb{R}$ eine Funktion. f ist stetig in $x_0 \in D$ genau dann, wenn die folgende Aussage gilt:

Für alle $\epsilon > 0$ existiert ein $\delta > 0$, sodass $|f(x) - f(x_0)| < \epsilon$ für alle $x \in D$ mit $|x - x_0| < \delta$.

Oder Alternativ: $\forall \epsilon > 0 \exists \delta > 0 \forall x \in D : |y - y_0| < \delta \Rightarrow |f(x) - f(x_0)| < \epsilon$

(LATEX)

DEFINITION 1. Sei $D \subseteq \mathbb{R}$ und sei $f : D \rightarrow \mathbb{R}$ eine Funktion. f ist stetig in $x_0 \in D$ genau dann, wenn die folgende Aussage gilt:

Für alle $\varepsilon > 0$ existiert ein $\delta > 0$, sodass $|f(x) - f(x_0)| < \varepsilon$ für alle $x \in D$ mit $|x - x_0| < \delta$.

Oder Alternativ: $\forall \varepsilon > 0 \exists \delta > 0 \forall x \in D : |y - y_0| < \delta \Rightarrow |f(x) - f(x_0)| < \varepsilon$

(Typst)

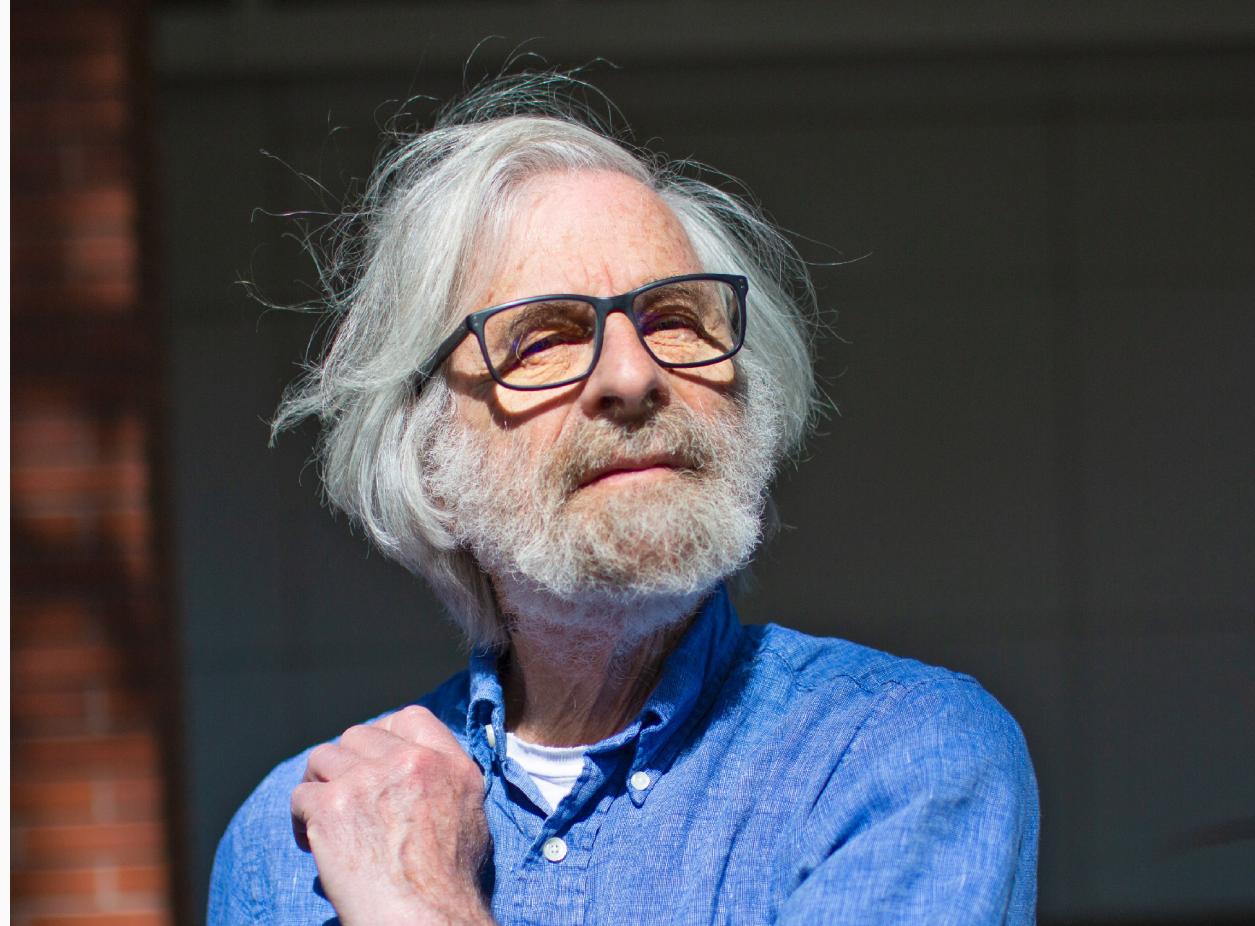
6. Die Typst-Dokumentation...

- <https://typst.apps/docs> als Nachschlagwerk
- Dokumentation ist wichtig!

6.1. Dokumentations-Beispiel: image-Funktion

image kann noch ein bisschen was!

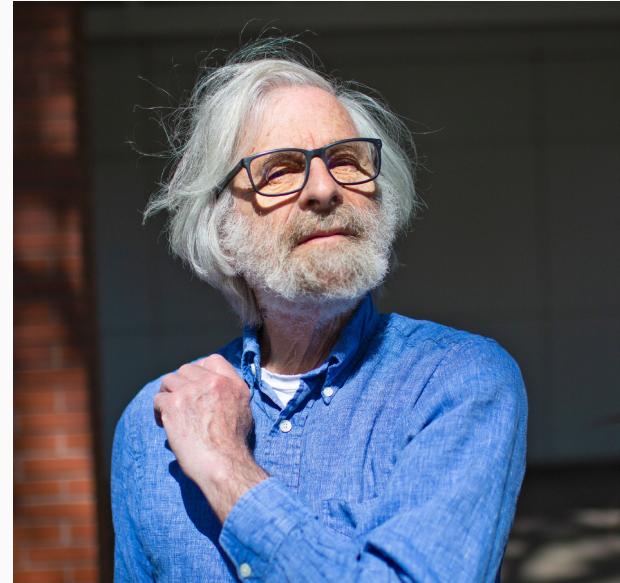
```
#image("leslie_lamport.png")
```



6.2. Dokumentations-Beispiel: image-Funktion

image kann noch ein bisschen was!

```
#image(height: 50%, "leslie_lamport.png")
```



6.3. Dokumentation Beispiel: `image`- Funktion

`image` kann noch ein bisschen was!

```
#image(fit: "stretch", width: 100%, height: 100%,  
"leslie_lamport.png")
```



[Overview](#)[Tutorial](#)[Reference](#)[Changelog](#)[Community](#)

Overview

Welcome to Typst's documentation! Typst is a new markup-based typesetting system for the sciences. It is designed to be an alternative both to advanced tools like LaTeX and simpler tools like Word and Google Docs. Our goal with Typst is to build a typesetting tool that is highly capable *and* a pleasure to use.

This documentation is split into two parts: A beginner-friendly tutorial that introduces Typst through a practical use case and a comprehensive reference that explains all of Typst's concepts and features.

We also invite you to join the community we're building around Typst. Typst is still a very young project, so your feedback is more than valuable.



Tutorial

Step-by-step guide to help you get started.



Reference

Details about all syntax, concepts, types, and functions.

image

☰ > Overview

Results

Image FUNCTION

Formatting CHAPTER

Box FUNCTION

Baseline PARAMETER OF
BOX

Padding FUNCTION

Background PARAMETER
OF PAGE

Table FUNCTION

Path PARAMETER OF IMAGE

Width PARAMETER OF
IMAGEHeight PARAMETER OF
IMAGE[Overview](#)[Tutorial](#)[Reference](#)[Changelog](#)[Community](#)

Overview

Welcome to Typst's documentation! Typst is a new markup-based typesetting system for the sciences. It is designed to be an alternative both to advanced tools like LaTeX and simpler tools like Word and Google Docs. Our goal with Typst is to build a typesetting tool that is highly capable *and* a pleasure to use.

This documentation is split into two parts: A beginner-friendly tutorial that introduces Typst through a practical use case and a comprehensive reference that explains all of Typst's concepts and features.

We also invite you to join the community we're building around Typst. Typst is still a very young project, so your feedback is more than valuable.



Tutorial

Step-by-step guide to help you get started.



Reference

Details about all syntax, concepts, types, and functions.

[Overview](#)[Tutorial](#)[Reference](#)[LANGUAGE](#)[Syntax](#)[Styling](#)[Scripting](#)[Types](#)[CONTENT](#)[Text](#)[Math](#)[Layout](#)[Visualize](#)[Circle](#)[Ellipse](#)[Image](#)[Line](#)[Path](#)[Polygon](#)[Rectangle](#)[Square](#)

image Element

A raster or vector graphic.

Supported formats are PNG, JPEG, GIF and SVG.

Example

```
#figure(  
  image("molecular.jpg", width: 80%),  
  caption: [  
    A step in the molecular testing  
    pipeline of our lab.  
  ],  
)
```



Figure 1: A step in the molecular testing pipeline of our lab.

Parameters ?

```
image(  
  string,  
  width: auto relative length,
```

ON THIS PAGE

[Summary](#)[Parameters](#)[path](#)[width](#)[height](#)[alt](#)[fit](#)

Parameters ⓘ

```
image(  
    string ,  
    width: auto | relative length ,  
    height: auto | relative length ,  
    alt: none | string ,  
    fit: string ,  
) -> content
```

path string Required Positional ⓘ

width `auto` or `relative length` *Settable* ?

The width of the image.

height `auto` or `relative length` *Settable* ?

The height of the image.

alt `none` or `string` *Settable* ?

A text describing the image.

fit `string` *Settable* ?

How the image should adjust itself to a given area.

- `"cover"` The image should completely cover the area. This is the default.
- `"contain"` The image should be fully contained in the area.
- `"stretch"` The image should be stretched so that it exactly fills the area, even if this means that the image will be distorted.

numbering `string` or `function` *Settable* 

How to number the enumeration. Accepts a [numbering pattern or function](#).

If the numbering pattern contains multiple counting symbols, they apply to nested enums. If given a function, the function receives one argument if `full` is `false` and multiple arguments if `full` is `true`.

› [View example](#)

numbering

string

or function

Settable 

How to number the enumeration. Accepts a [numbering pattern or function](#).

If the numbering pattern contains multiple counting symbols, they apply to nested enums. If given a function, the function receives one argument if `full` is `false` and multiple arguments if `full` is `true`.

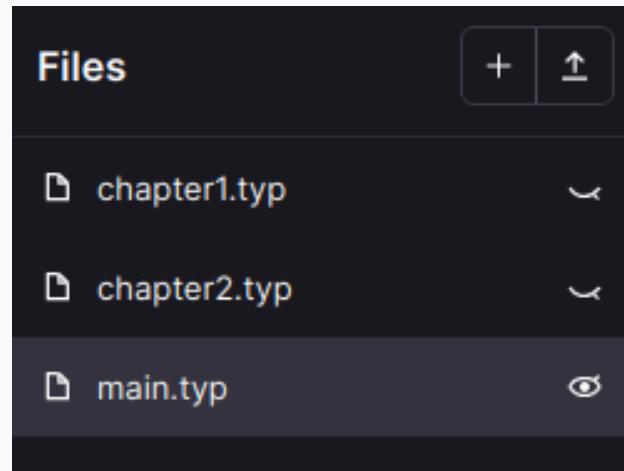
▼ View example

```
#set enum(numbering: "1.a")  
+ Different  
+ Numbering  
  + Nested  
  + Items  
+ Style  
  
#set enum(numbering: n => super[#n])  
+ Superscript  
+ Numbering!
```

- 1) Different
 - 2) Numbering
 - a) Nested
 - b) Items
 - 3) Style
- ¹ Superscript
² Numbering!

7. Eigene Templates und Skripts

7.1. Mehrere Typst-Dateien verwenden



main.typ:

= Mein tolles Buch

Hier, Kapitel!

```
#include "chapter1.typ"
#include "chapter2.typ"
```

chapter1.typ:

== Kapitel 1

```
#lorem(10)
```

chapter2.typ:

== Kapitel 2

```
#lorem(10)
```

7.2. Eigene Funktionen und Variablen #1

```
#let var = 3.14159
#let double(e) = {
    return 2*e
}

$pi$ ist etwa #var!
$tau$ ist etwa #double(var)!

$pi approx var$ \
$tau approx double(var)$
```

π ist etwa 3.14159!
 τ ist etwa 6.28318!

$\pi \approx 3.14159$
 $\tau \approx 6.28318$

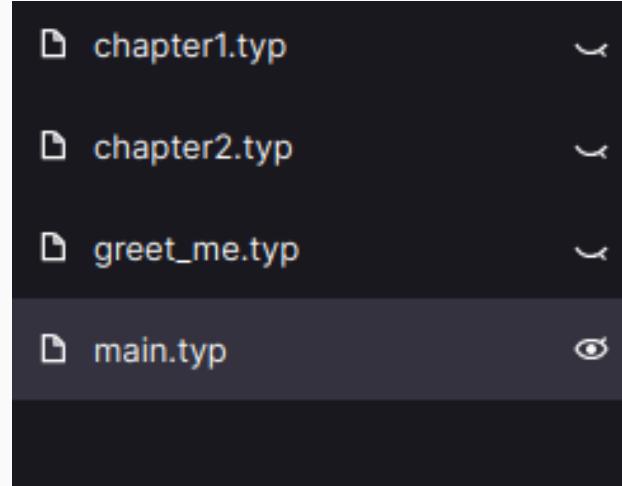
7.3. Eigene Funktionen und Variablen #2

```
#let names = ("Peter", "Petra", "Josef", "Josefa")
#let greet(names) = {
    [Hallo ]
    names.join(last: " und ", ", ")
    [! #names.len() wundervolle Namen!]
}

#greet(names)
```

Hallo Peter, Petra, Josef und Josefa! 4 wundervolle Namen!

7.4. Funktionen auslagern



`greet_me.typ` definiert:

- `greet(names)`
- `double(n)`

```
#import "greet_me.typ": *
#import "greet_me.typ": greet
```

8. Typst kann noch mehr!

8.1. Figuren und Referenzen¹

@glacier shows a glacier. Glaciers are complex systems.

```
#figure(  
  image("glacier.jpg", height: 80%),  
  caption: [A curious figure.],  
) <glacier>
```

Figure 1 shows a glacier. Glaciers are complex systems.



Figure 1: A curious figure.

8.2. Bibliographie¹

works.bib:

```
@article{netwok,
  title={At-scale impact of the {Net Wok}: A culinarily holistic investigation of distributed dumplings},
  author={Astley, Rick and Morris, Linda},
  journal={Armenian Journal of Proceedings},
  volume={61},
  pages={192--219},
  year={2020},
  publisher={Automattic Inc.}

}

@article{arrgh,
  title={The Pirate Organization},
  author={Leeson, Peter T.},
}
```

This was already noted by pirates long ago. @arrgh

Multiple sources say ...
`#cite("arrgh", "netwok")`.

`#bibliography("works.bib")`

This was already noted by pirates long ago. [1]
Multiple sources say ... [1, 2].

Bibliography

- [1] P. T. Leeson, “The pirate organization.”
- [2] R. Astley, and L. Morris, “At-scale impact of the Net Wok: a culinarily holistic investigation of distributed dumplings,” *Armenian J. Proc.*, vol. 61, pp. 192–219, 2020.

8.3. Raytracing

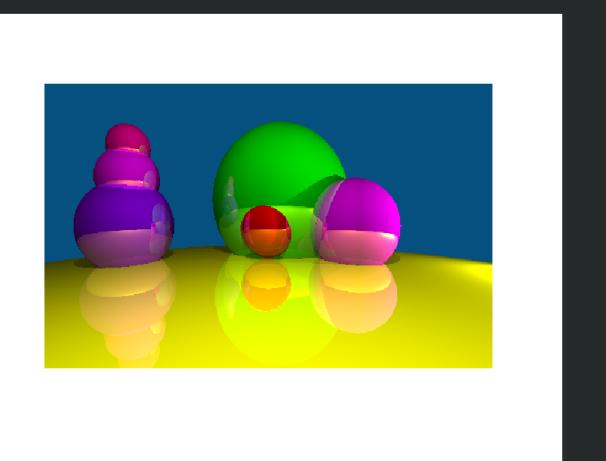
Type File Edit View Help

eliteamate > raytracer

B I U

Share

```
1 {
2     let EPS = 1e-6;
3     let CLIPPING = 1e-2;
4     let INF = 1e6;
5
6     let sqrt(x) = {
7         if x < EPS { return x; }
8         let guess = x;
9         // nethon's method, a lot of digits of accuracy
10        for i in range(10) {
11            guess = (guess + x / guess) / 2;
12        }
13        guess
14    }
15
16    let about_eq(a, b, eps) = {
17        return abs(a - b) < eps;
18    }
19
20    let dot(a, b) = { a(0) * b(0) + a(1) * b(1) + a(2) * b(2) };
21    let plus(a, b) = { (a(0) + b(0), a(1) + b(1), a(2) + b(2)) };
22    let minus(a, b) = { (a(0) - b(0), a(1) - b(1), a(2) - b(2)) };
23    let neg(a) = { (-a(0), -a(1), -a(2)) };
24    let div(a, x) = { (a(0) / x, a(1) / x, a(2) / x) };
25    let mul(a, x) = { (a(0) * x, a(1) * x, a(2) * x) };
26    let len(a) = { sqrt(dot(a, a)) };
27    let normalize(a) = { div(a, len(a)) };
28    let pointwise.mul(a, b) = { (a(0) * b(0), a(1) * b(1), a(2) * b(2)) };
29
30    let material(color, smoothness) = {
31        (color: color, smoothness: smoothness)
32    }
33
34    let sphere(center, radius, material) = {
35        (center: center, radius: radius, material: material)
36    }
37
38    let ray(start, direction) = {
39        (start: start, direction: normalize(direction))
40    }
41
42    let intersect(sphere, ray) = {
43        let ray_start = ray.start;
44        let ray_dir = ray.direction;
45        let center = sphere.center;
46        let radius = sphere.radius;
47
48        let a = ray_dir.x * ray_dir.x + ray_dir.y * ray_dir.y + ray_dir.z * ray_dir.z;
49        let b = 2 * (ray_dir.x * (center.x - ray_start.x) + ray_dir.y * (center.y - ray_start.y) + ray_dir.z * (center.z - ray_start.z));
50        let c = center.x * center.x + center.y * center.y + center.z * center.z - radius * radius;
51
52        let discriminant = b * b - 4 * a * c;
53        if discriminant < 0 { return null; }
54
55        let t = (-b - sqrt(discriminant)) / (2 * a);
56        if t < 0 || t > CLIPPING { return null; }
57
58        let hit_pos = ray_start + ray_dir * t;
59        let normal = (hit_pos - center) / radius;
60
61        let color = material.color;
62        let smoothness = material.smoothness;
63
64        return { pos: hit_pos, normal: normal, color: color, smoothness: smoothness };
65    }
66
67    let render(spheres, camera) = {
68        let width = camera.width;
69        let height = camera.height;
70
71        let image = new Image();
72        let pixels = new Uint8Array(width * height * 3);
73
74        for (let y = 0; y < height; y++) {
75            for (let x = 0; x < width; x++) {
76                let ray = ray((x / width) * camera.aspect, (y / height));
77                let intersection = intersect(spheres, ray);
78
79                if (intersection) {
80                    let color = intersection.color;
81                    let smoothness = intersection.smoothness;
82
83                    let reflectance = 0.5;
84
85                    for (let i = 0; i < 10; i++) {
86                        let ray = ray(intersection.pos, intersection.normal);
87                        let intersection = intersect(spheres, ray);
88
89                        if (intersection) {
90                            let color = intersection.color;
91                            let smoothness = intersection.smoothness;
92
93                            reflectance *= smoothness;
94
95                            if (reflectance < 0.01) break;
96                        }
97                    }
98
99                    let final_color = color;
100                   let final_smoothness = smoothness;
101
102                   if (reflectance > 0.01) {
103                       let reflectance_color = reflectance * color;
104
105                       final_color = reflectance_color;
106                       final_smoothness = smoothness;
107
108                       for (let i = 0; i < 10; i++) {
109                           let ray = ray(intersection.pos, intersection.normal);
110                           let intersection = intersect(spheres, ray);
111
112                           if (intersection) {
113                               let color = intersection.color;
114                               let smoothness = intersection.smoothness;
115
116                               reflectance *= smoothness;
117
118                               if (reflectance < 0.01) break;
119                           }
120                       }
121
122                   }
123
124                   pixels[x * 3 + y * width * 3] = final_color.r;
125                   pixels[x * 3 + y * width * 3 + 1] = final_color.g;
126                   pixels[x * 3 + y * width * 3 + 2] = final_color.b;
127
128               }
129           }
130       }
131
132       image.data = pixels;
133
134       return image;
135   }
136
137   let main() = {
138       let spheres = [
139           sphere({x: 0, y: 0, z: 0}, 1, material({color: "#ff0000", smoothness: 100})),
140           sphere({x: 0, y: 0, z: 0}, 0.5, material({color: "#00ff00", smoothness: 100})),
141           sphere({x: 0, y: 0, z: 0}, 0.2, material({color: "#0000ff", smoothness: 100})),
142           sphere({x: 0.5, y: 0, z: 0}, 0.1, material({color: "#00ffff", smoothness: 100})),
143           sphere({x: 0.2, y: 0.2, z: 0.1}, 0.05, material({color: "#ff00ff", smoothness: 100}))
144       ];
145
146       let camera = camera();
147
148       let image = render(spheres, camera);
149
150       image.style.display = "block";
151       document.body.appendChild(image);
152   }
153
154   main();
155 }
```



Voll funktionsfähiger Raytracer für 3D-Rendering.¹

9. Abschluss und Weiteres

9.1. Was noch fehlt¹

- Fußnoten
- Paketmanager (kurzfristige Alternative: GitHub)
- StackOverflow (kurzfristige Alternative: Discord)

9.2. Erwartete Neuerungen¹

- Fußnoten (und die komplette Überarbeitung der Layout-Engine)
- Paketmanager
- Verbesserung des Mathe-Layouts
- ...

9.3. Wer sollte Typst (nicht) benutzen?

Pros:

- ✓ **steile** Lernkurve
- ✓ sehr dynamisch
- ✓ aktive Community
- ✓ schnelle Kompilierzeit
- ✓ verständliche Fehlermeldungen

Cons:

- viele Programmierer-Ansätze
- komplexes Layouting & Fußnoten schwer umsetzbar
- Pure Functions können schwer sein (States, Counter, ...)

9.4. Weiteres

Übrigens: Diese gesamte Präsentation wurde alleine in Typst erstellt.

Typst Dokumentation:

- <https://typst.app/docs/>

Offizielles Typst-Tutorial:

- <https://typst.app/docs/tutorial>

Offizieller Typst-Discord:

- <https://discord.gg/2uDybryKPe>

Code für diese Präsentation und weitere Beispiele:

- <https://github.com/survari/typst-seminar>