Typst — Hat LaTeX ausgedient?

Eine kurze Einführung in Typst

Tristan Pieper tristan.pieper@uni-rostock.de · 2. Juni 2023

Inhaltsverzeichnis

1. Kurzes Kennenlernen

2. Probleme von LaTeX

2.1. Alles begann mit…

Donald E. Knuth[1] (geb. 10. Januar 1938)

2.2. Dann kam...

Leslie Lamport[1] (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:

- pdfLaTeX
- LuaTeX
- XeTeX
- MikTeX
- KaTeX
- …

2.6. Beispiel-Fehlermeldung (Typst)

Typst:

```
$a+b
```

```
error: expected dollar sign
  ┌─ test.typ:1:5
  │
1 │ $a+b
  │     ^
```

2.7. Beispiel-Fehlermeldung (LaTeX)

LaTeX:

```
\documentclass{article}

\begin{document}
    $a+b
\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Rule 'pdflatex': The following rules & subrules became out-of-date: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>
                $
l.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):
```

2.8. Mehr Beispiel-Fehlermeldung

Typst	LaTeX

`#set par(leading: [Hello])`
`                  ^^^^^^^`
`expected integer, found content`
	`\baselineskip=Hello`
	`Missing number, treated as zero.`
	`Illegal unit of measure (pt inserted).`

`#heading()`
`        ^^`
`missing argument: body`
	`\section`
	`Missing \endcsname inserted.`
	`Missing \endcsname inserted.`
	`...`

(aus: Mädje, Laurenz: *„Typst – A Programmable Markup Language for Typesetting.“* Masterarbeit an der TU Berlin, 2022.)

3. Die Lösung aller Probleme(?)

3.1. Typst

Martin Haug
(Webentwicklung)

Laurenz Mädje
(Compilerentwicklung)

https://typst.app/about/, (letzter Zugriff: 26.05.2023, 10:16)

- 2019 Projektstart an TU Berlin
- entstanden aus Frustration über LaTeX

3.2. Das Ziel

„Während bestehende Lösungen langsam, schwer zu bedienen oder einschränkend sind, ist Typst sorgfältig entworfen, um leicht erlernbar, flexibel und schnell zu sein. Dafür haben wir eine komplett eigene Markupsprache und Textsatzengine von Grund auf entwickelt. Dadurch sind wir in allen Bereichen des Schreib- und Textsatzprozesses innovationsfähig.“

https://www.tu.berlin/entrepreneurship/startup-support/unsere-startups/container-profile/startups-2023-typst, (letzter Zugriff: 03.05.2023, 10:13)

3.3. Ein kleiner Vergleich

LaTeX

```
\documentclass{article}

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

Typst

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

Ergebnis

1. Dies
2. Ist
3. Eine
4. 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 Abstatz!
```

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 numerierte Liste
+ Kann sehr schön sein!

1. So geht sie auch!
2. Jawohl!

- Und hier nicht-nummeriert!
- Ganz ohne Nummern!
```

1. Eine numerierte 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!])

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

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

Hallo Welt!

Hallo Welt!

Hallo Welt!

5.5. Content, Strings, Scripts

- zwei Arten von Inhalten in Typst:
  - Content in `[...]`
  - Scripts in `{...}`
  - von Content zu Script mit `#`
- jedes Dokument ist grundlegend Content

```
#strong([Hier ist Content
in Script.])

#{
  strong([Fett!])
  [Auch hier ist Content
in Script!]
}

#{ 3/4 }
```

Hier ist Content.

Fett!Auch hier ist Content in Script!

0.75

5.6. Text[1]

```
*Hallo!* #strong([Hallo!])

_Hallo!_ #emph([Hallo!])

Hallo!#super([Hallo!])

Hallo!#sub([Hallo!])

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

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

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

Hallo! Hallo!

Hallo! Hallo!

Hallo!Hallo!

Hallo!Hallo!

Hallo rot!

Hallo pink!

Hallo pink!

5.7. Ausrichtung

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

Hallo!

Hallo!

Hallo!

Hallo!

Aufgabe 1

Wie realisiert man das Folgende in Typst?

Ein Brief an Lorem

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

Lösung 1

```
#align(center, emph[Ein Brief an Lorem])
#lorem(20)
```

5.8. Abstände #1

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

Vertikaler

Horizontaler Abstand

5.9. Abstände (Vertikal)

```
Listen-Beispiel von Folie 21
mit vertikalen Abständen.

+ Eine numerierte 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 21 mit vertikalen Abständen.

1. Eine numerierte Liste
2. Kann sehr schön sein!

1. So geht sie auch!
2. Jawohl!

- Und hier nicht-nummeriert!
- Ganz ohne Nummern!

5.10. 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

5.11. Bilder

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

5.12. #block() und #box()

```
Das lässt sich mit `#block`
machen: #block(stroke: black,
inset: 0.5em, [ `#block()`
erzeugt einen Zeilenumbruch
und lässt sich über Seiten
brechen. Es hat viele
optionale Argumente.])
So sieht's aus.
```

Das lässt sich mit `#block` machen:

`#block()` erzeugt einen Zeilenumbruch und lässt sich über Seiten brechen. Es hat viele optionale Argumente.

So sieht’s aus.

5.13. #block() und #box()

```
Hingegen: #box(stroke: black, inset: 2pt, [`#box()`]) erzeugt #box(stroke: (bottom: black), inset: 2pt, [keinen]) Zeilenumbruch und lässt sich #box(stroke: (bottom: black), inset: 2pt, [nicht]) über Seiten brechen. Man kann damit aber Dinge zum Beispiel umrahmen. Zum Unterstreichen sollte man aber eher #underline[`#underline`] benutzen.
```

Hingegen: `#box()` erzeugt keinen Zeilenumbruch und lässt sich nicht über Seiten brechen. Man kann damit aber Dinge zum Beispiel umrahmen. Zum Unterstreichen sollte man aber eher `#underline` benutzen.

5.14. Tabellen

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

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

5.15. 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) $
```

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

$$A = \pi r^2$$

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

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

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

Aufgabe 2

Wie realisiert man das Folgende in Typst?

Formel	**Zugeschrieben**
$a^2 + b^2 = c^2$	Pythagoras
$c \leq a + b$	Unbekannt

Lösung 2

```
#table(columns: (auto, auto),
  strong[Formel], strong[Zugeschrieben],
  $a^2 + b^2 = c^2$, [Pythagoras],
  $c <= a + b$, [Unbekannt])
```

5.16. Set-Regeln[1]

```
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 magnam aliquam quaerat.

5.17. Show-Regeln[1]

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

=== 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.
```

5.17.1. Hallo!

5.17.1.1. Welt!

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

Aufgabe 3

Wie realisiert man das Folgende in Typst?

> In **Typst** sind show- und set-Regeln sehr mächtig. Kann man Sie, kann man auch **Typst**. **Typst** soll immer fett gedruckt werden. Der Absatz ist Blocksatz. *Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat.*

Lösung 3

```
#set par(justify: true)
#show "Typst": strong

In Typst sind show- und set-Regeln sehr mächtig. Kann man
Sie, kann man auch Typst. Typst soll immer fett gedruckt
werden. Der Absatz ist Blocksatz. _#lorem(20)_
```

```latex
\documentclass[14pt,a4paper]{extarticle}
\usepackage{bold-extra}
\usepackage{amssymb}
\usepackage[T1]{fontenc}
\usepackage[paperwidth=22cm,paperheight=5.5cm,
    left=0.5cm,right=0.5cm,top=0.5cm,
    bottom=0.5cm]{geometry}

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

\begin{document}
    \noindent\textbf{\textsc{Definition 1.}}
\textit{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:}

    \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}
```

```typst
#set page(margin: 0.5cm, width: 22cm, height: 5.5cm)
#set text(size: 14pt, font: "New Computer Modern")
#set par(justify: true)

*#smallcaps([Definition 1.])* _Sei $D subset.eq RR$
und sei $f: D -> RR$ 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: $forall epsilon > 0 exists delta >
0 forall x in D : |y-y_0| < delta => |f(x)-f(x_0)| <
epsilon$_

#v(1em)
(Typst)
```

Definition 1. *Sei $D \subseteq \mathbb{R}$ und sei $f : D \to \mathbb{R}$ 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 $f : D \to \mathbb{R}$ 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)

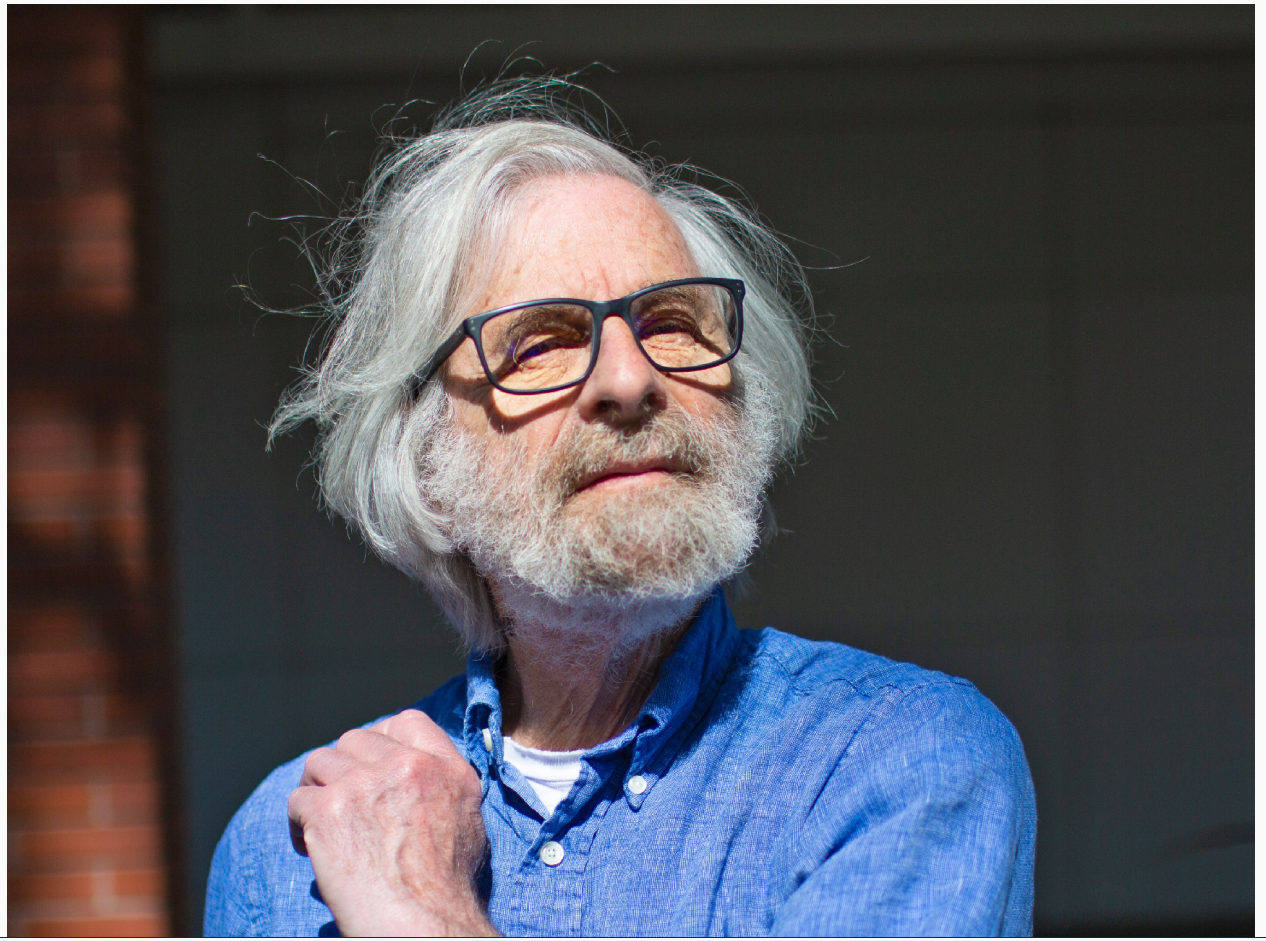
6. Die Typst-Dokumentation…

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

6.1. Doku-Beispiel: `image`-Funktion

`image` kann noch ein bisschen was!

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



6.2. Doku-Beispiel: `image`-Funktion

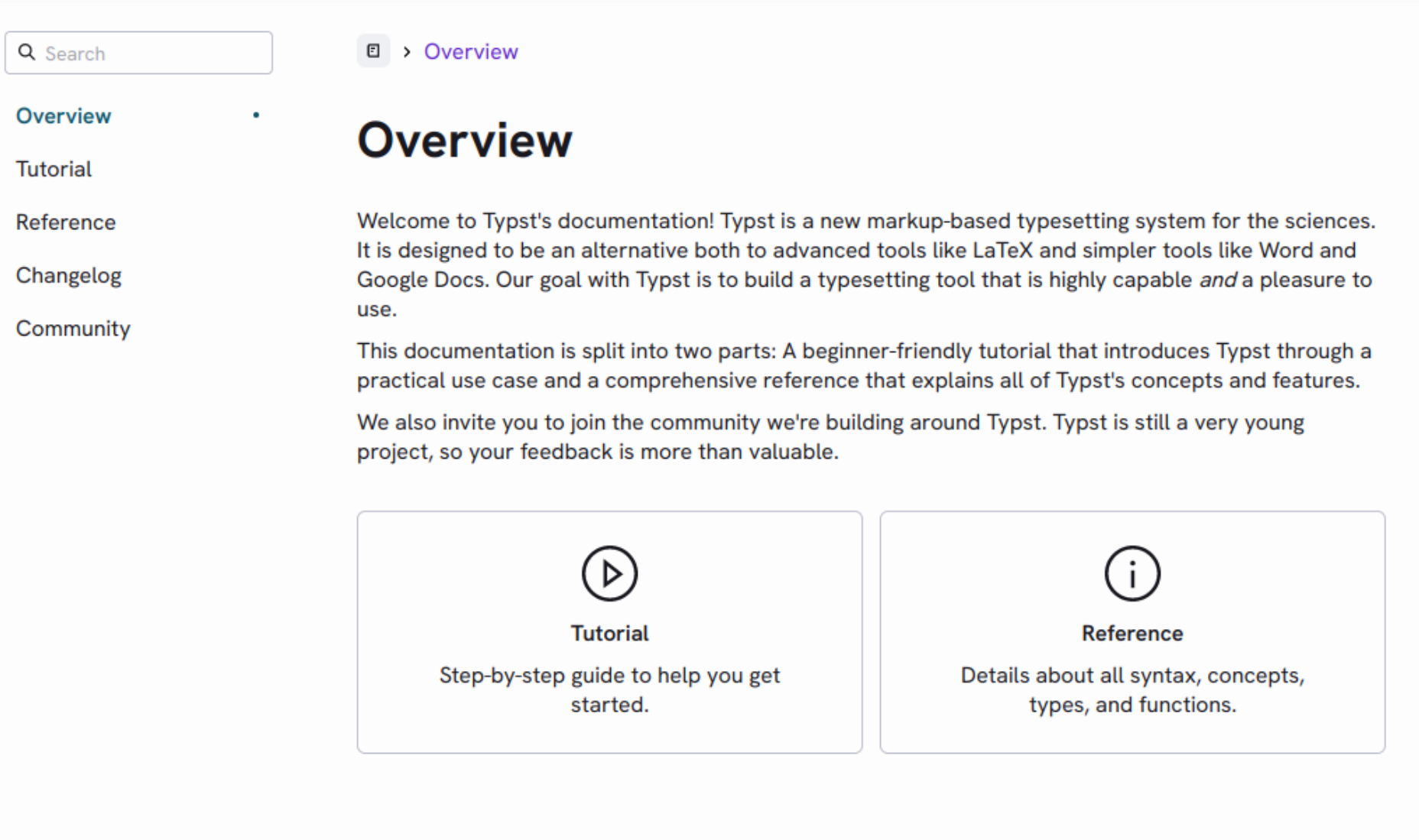
`image` kann noch ein bisschen was!

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

6.3. Doku-Beispiel: `image`-Funktion

`image` kann noch ein bisschen was!

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


Search
Overview
Tutorial
Reference
Changelog
Community
Overview
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.

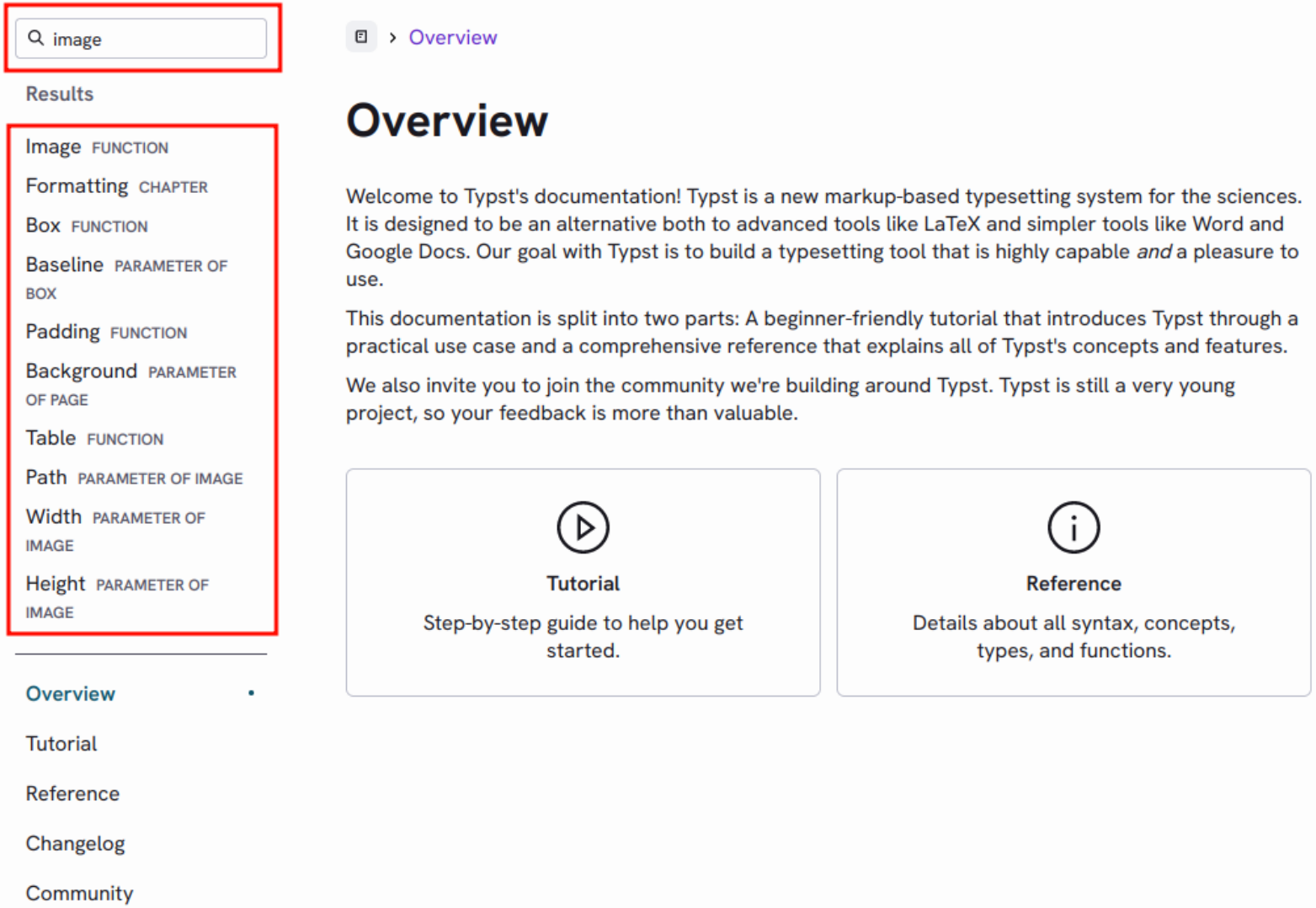


image
Results
Image FUNCTION
Formatting CHAPTER
Box FUNCTION
Baseline PARAMETER OF BOX
Padding FUNCTION
Background PARAMETER OF PAGE
Table FUNCTION
Path PARAMETER OF IMAGE
Width PARAMETER OF IMAGE
Height PARAMETER OF IMAGE
Overview
Tutorial
Reference
Changelog
Community
Overview
Overview
Welcome to Typst's 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.

Search

Overview

Tutorial

Reference

LANGUAGE

Syntax

Styling

Scripting

Types

CONTENT

Text

Math

Layout

Visualize

Circle

Ellipse

Image

Line

Path

Polygon

Rectangle

Square

Reference › Visualize › Image

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.
  ],
)
```

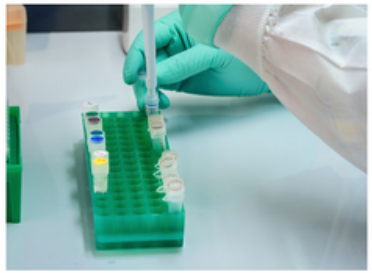


Figure 1: A step in the molecular testing pipeline of our lab.

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*

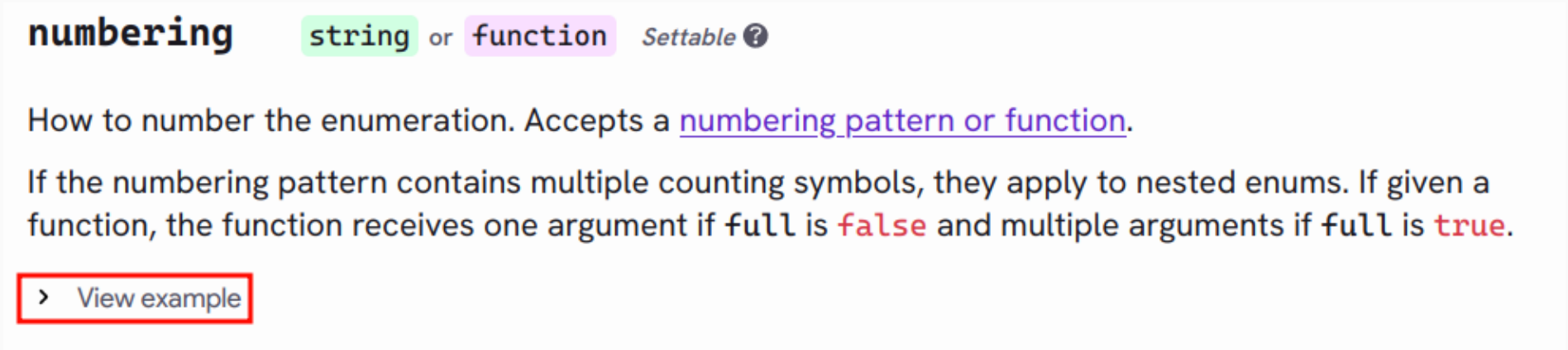
A text describing the image.

`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.

Beispiel bei `#enum()`:



Beispiel bei `#enum()`:

`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!

Aufgabe 4

Welcher Parameter kann durch eine `set`-Regel wie verändert werden, damit man eine nummerierte Listen (`#enum()`) wie folgt formatieren kann? Die Dokumentation hilft!

I. Element 1
II. Element 2
III. Element 3

Lösung 4

```
#set enum(numbering: "I.")
+ Element 1
+ Element 2
+ Element 3
```

Aufgabe 5

Welcher Parameter kann durch eine `set`-Regel wie verändert werden, damit man eine nicht-nummerierte Listen wie folgt formatieren kann? Die Dokumentation hilft!

> - Element 1
> - Element 2
> - Element 3

Lösung 5

```
#set list(marker: ">")
- Element 1
- Element 2
- Element 3
```

7. Eigene Templates und Skripts

7.1. 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!

$\pi \approx 3.14159$

$\tau \approx 6.28318$

7.2. Eigene Funktionen und Variablen #2

Eine Einschränkung: wir arbeiten mit *Pure Functions*.

Folgendes geht nicht:

```
#let var = 2
#let change_var(new_v) = {
  var = new_v
  return var
}

#change_var(100)
```

Fehler:

```
#let change_var(new_v) = {
  var = new_v
  ^^^
variables from outside the
function are read-only and
cannot be modified
```

7.3. Eigene Funktionen und Variablen #3

```
#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!

Aufgabe 6

Es soll eine Funktion erstellt werden, die zwei Zahlen addiert und das Ergebnis rot und **fett** formatiert. Etwa so:

```
#add(20, 40)
```

60

Lösung 6

Viele Möglichkeiten:

```
#let add(a, b) = strong(text(red, [#(a+b)]))
#add(20, 40)
```

oder mit `#str()` Zahlen zu Strings machen:

```
#let add(a, b) = strong(text(red, str(a+b)))
#add(20, 40)
```

oder mit Klammern:

```
#let add(a, b) = {
  let result = a+b
  strong(text(red, str(result)))
}

#add(20, 40)
```

8. Typst kann noch mehr!

8.1. Figuren und Referenzen[1]

```
@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. Bibliographie[1]

works.bib:

```
@article{netwok,
  title={At-scale impact of the {Net Wok}: A culinarically 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]

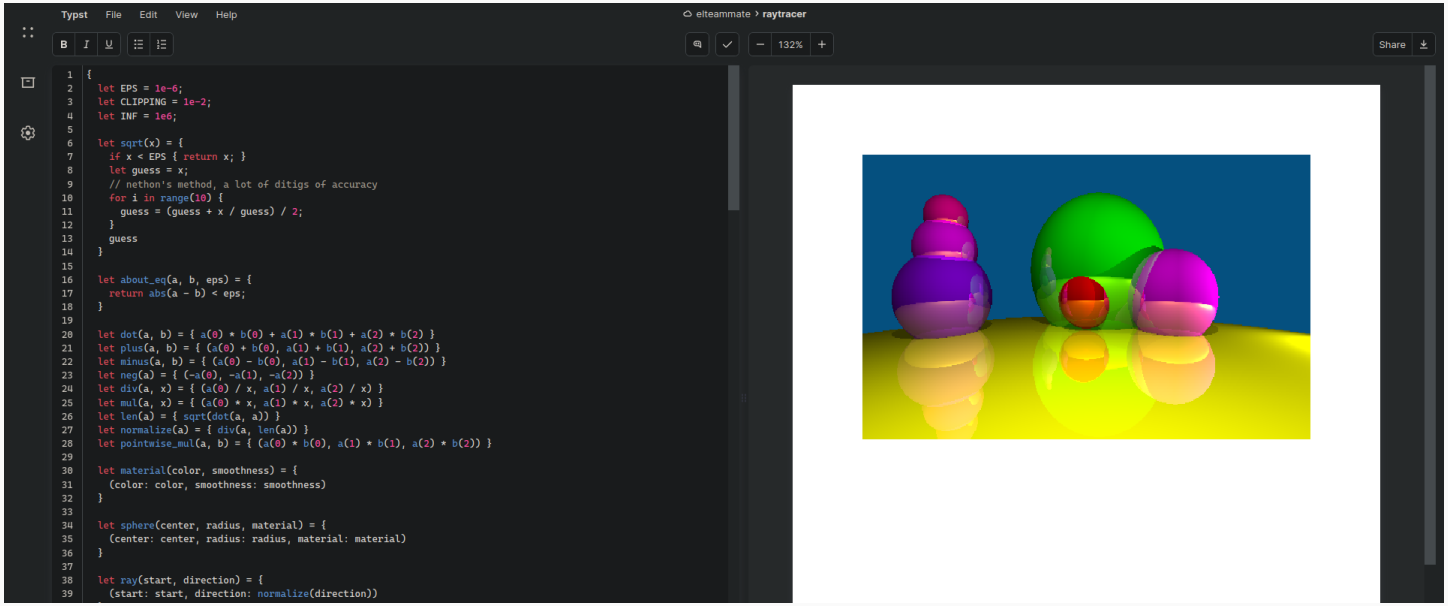
Multiple sources say ... [1, 2].

Bibliography

[1] P. T. Leeson, "The pirate organization."

[2] R. Astley, and L. Morris, "At-scale impact of the Net Wok: a culinarically holistic investigation of distributed dumplings," *Armenian J. Proc.*, vol. 61, pp. 192–219, 2020.

8.3. Raytracing



Voll funktionsfähiger Raytracer für 3D-Rendering.[1]

9. Abschluss und Weiteres

9.1. Was noch fehlt[1]

- ~~Fußnoten~~ (am 20.05.2023 mit v0.4.0 hinzugefügt)
- Paketmanager (kurzfristige Alternative: GitHub)
- StackOverflow (kurzfristige Alternative: Discord)

9.2. Erwartete Neuerungen[1]

- die komplette Überarbeitung der Layout-Engine
- Paketmanager
- Verbesserung des Mathe-Layouts
- HTML Output
- …

9.3. Wer sollte Typst (nicht) benutzen?

Pros:

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

Cons:

- × viele Programmierer-Ansätze
- × komplexes Layouting (keine Floating Figures)
- × Pure Functions können schwer sein (States, Counter, …)
- × bisher keine Unterstützung durch große Journals
- × kein zentrales Paketmanagement

(Stand: 25.05.2023)

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 Beipsiele: https://github.com/survari/typst-seminar
- Masterarbeit von Laurenz Mädje über Typst: https://www.user.tu-berlin.de/laurmaedje/programmable-markup-language-for-typesetting.pdf
- Lambdas, States und Counter: https://typst.app/project/rpnqiqoQNfxXjHQmpJ81nF