

# Typst – Hat L<sup>A</sup>T<sub>E</sub>X abgedankt?



Eine kurze Einführung in Typst

Tristan Pieper [tristan.pieper@uni-rostock.de](mailto:tristan.pieper@uni-rostock.de) · 1. Juni 2023

# Inhaltsverzeichnis

1. Kurzes Kennenlernen .....	3
2. Probleme von L <sup>A</sup> T <sub>E</sub> X .....	4
3. Die Lösung aller Probleme(?) .....	12
4. Die Web-App .....	14
5. Grundlegende Formatierung .....	16
6. Eigene Templates und Skripts .....	19
7. Was Typst noch so alles kann .....	20
8. Was noch fehlt .....	22
9. Abschluss und Weiteres .....	23

# 1. Kurzes Kennenlernen

## 2. Probleme von L<sup>A</sup>T<sub>E</sub>X

## 2.1. Alles begann mit...



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

## 2.2. Dann kam...



Leslie Lamport (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

„ $\text{\LaTeX}$ “ ist kein Programm, sondern:

- pdf  $\text{\LaTeX}$
- 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- Fehlernmeldung ( LATEX )

LATEX :

\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Rule 'pdflatex': The following rules & subrules became out-of-date: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
  LATEX 2e <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

L <small>AT</small> E <small>X</small>	T <small>yp</small>	st	E <small>rgebnis</small>
<code>\documentclass{article}</code>	+ Dies		1. Dies
<code>\begin{document}</code>	+ Ist		2. Ist
<code>\begin{enumerate}</code>	+ Eine		3. Eine
<code>    \item Dies</code>	+ Liste!		4. Liste!
<code>    \item Ist</code>			
<code>    \item Eine</code>			
<code>    \item Liste!</code>			
<code>\end{enumerate}</code>			
<code>\end{document}</code>			

# 4. Die Web-App

## 4.1. Ab ans Werk!

Vorteile:

- alle Dateien online
- verschiedene Projekte erstellbar
- guter online Editor
- eingebaute Dokumentation

<https://typst.app/>

# 5. Grundlegende Formatierung

```

\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 $D$ \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}

```

```

#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  $|x - x_0| < \delta$ .

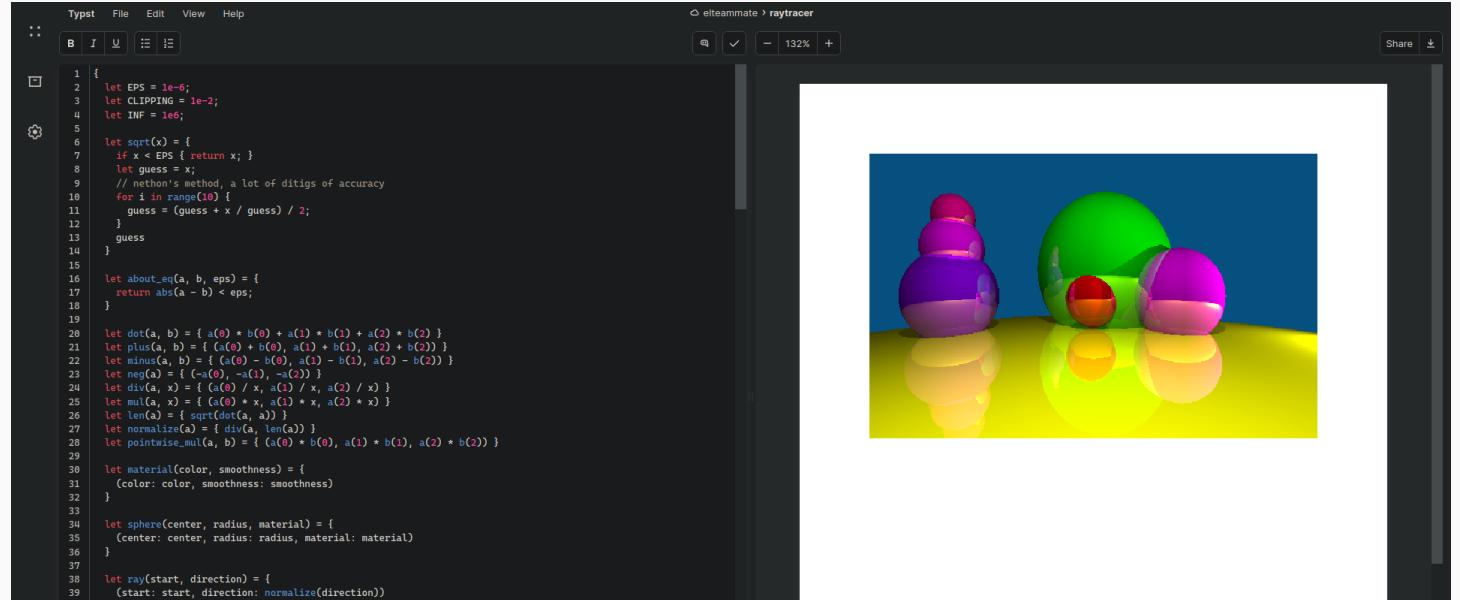
Oder Alternativ:  $\forall \varepsilon > 0 \exists \delta > 0 \forall x \in D : |y - y_0| < \delta \Rightarrow |f(x) - f(x_0)| < \varepsilon$

(Typst)

# 6. Eigene Templates und Skripts

## 7. Was Typst noch so alles kann

# 7.1. Raytracing



The image shows a code editor window with a dark theme. The top bar includes 'Typst' and 'File', followed by standard menu items like 'Edit', 'View', and 'Help'. Below the menu is a toolbar with icons for bold ('B'), italic ('I'), underline ('U'), and other document-related functions. The main area contains a block of Typst code. The code defines various mathematical operations (dot product, addition, subtraction, negation, division, multiplication, length, normalization), materials (color, smoothness), spheres, and a ray tracing function. The rendered output on the right side of the editor shows a 3D scene with several spheres of different sizes and colors (purple, green, red, yellow) resting on a yellow surface against a blue background.

```
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         // newton'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, 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(spheres, ray) = {
43        let (start, direction) = ray;
44        let closest = infinity;
45        let closestSphere = null;
46
47        for sphere in spheres {
48            let (center, radius) = sphere;
49            let distance = distanceFromRayToSphereCenter(ray, center);
50
51            if distance <= radius {
52                if distance < closest {
53                    closest = distance;
54                    closestSphere = sphere;
55                }
56            }
57        }
58
59        if closestSphere != null {
60            return closestSphere;
61        } else {
62            return null;
63        }
64    }
65
66    let colorAtRay(ray) = {
67        let sphere = intersect(spheres, ray);
68
69        if sphere == null {
70            return black;
71        }
72
73        let (center, radius, material) = sphere;
74
75        let normalizedDirection = normalize(ray.direction);
76
77        let distance = distanceFromRayToSphereCenter(ray, center);
78
79        let color = material.color;
80
81        if distance < radius {
82            let normalizedDistance = (distance - radius) / (radius * 2);
83            color = mix(material.color, white, normalizedDistance);
84        }
85
86        color
87    }
88
89    let renderImage(spheres, width, height) = {
90        let image = new Image();
91
92        for y in range(height) {
93            for x in range(width) {
94                let ray = rayFromPixel((x, y), image);
95
96                let color = colorAtRay(ray);
97
98                image.setPixel((x, y), color);
99            }
100        }
101
102        image
103    }
104
105    let main() = {
106        let spheres = [
107            sphere((0, 0, 0), 1, material((1, 0, 0), 0.5)),
108            sphere((1, 0, 0), 0.5, material((0, 1, 0), 0.5)),
109            sphere((0, 1, 0), 0.5, material((0, 0, 1), 0.5)),
110            sphere((0, 0, 1), 0.5, material((1, 0, 1), 0.5)),
111            sphere((0.5, 0.5, 0.5), 0.2, material((1, 1, 1), 1.0))
112        ];
113
114        let image = renderImage(spheres, 1000, 1000);
115
116        image
117    }
118
119    main()
120}
```

Voll funktionsfähiger Raytracer für 3D-Rendering.<sup>1</sup>

## 8. Was noch fehlt

# 9. Abschluss und Weiteres

## 9.1. Wer sollte Typst benutzen?

## 9.2. Erwartete Neuerungen<sup>1</sup>

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

## 9.3. Weiteres

Typst Dokumentation:

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

Offizielles Typst-Tutorial:

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

Code für diese Präsentation und weitere Beispiele:

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