

Typst – Warum noch LaTeX lernen?



Eine kurze Einführung in Typst

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

Inhaltsverzeichnis

1. Kurzes Kennenlernen	3
2. Probleme von LaTeX	4
3. Die Lösung aller Probleme(?)	12
4. Die Web-App	14
5. Grundlegende Formatierung	16
6. Was Typst noch so alles kann	19
7. Was noch fehlt	21
8. Wer sollte Typst benutzen?	22
9. Abschluss und Weiteres	23

1. Kurzes Kennenlernen

2. Probleme von LaTeX

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

„LaTeX“ ist kein Programm, sondern:

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

2.6. Beispiel- Fehlernmeldung (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Rule 'pdflatex': The following rules & subrules became out-of-date: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
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
<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: $\forall \epsilon > 0 \exists \delta > 0 \forall x \in D : |y - y_0| < \delta \Rightarrow |f(x) - f(x_0)| < \epsilon$

(L^AT_EX)

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:

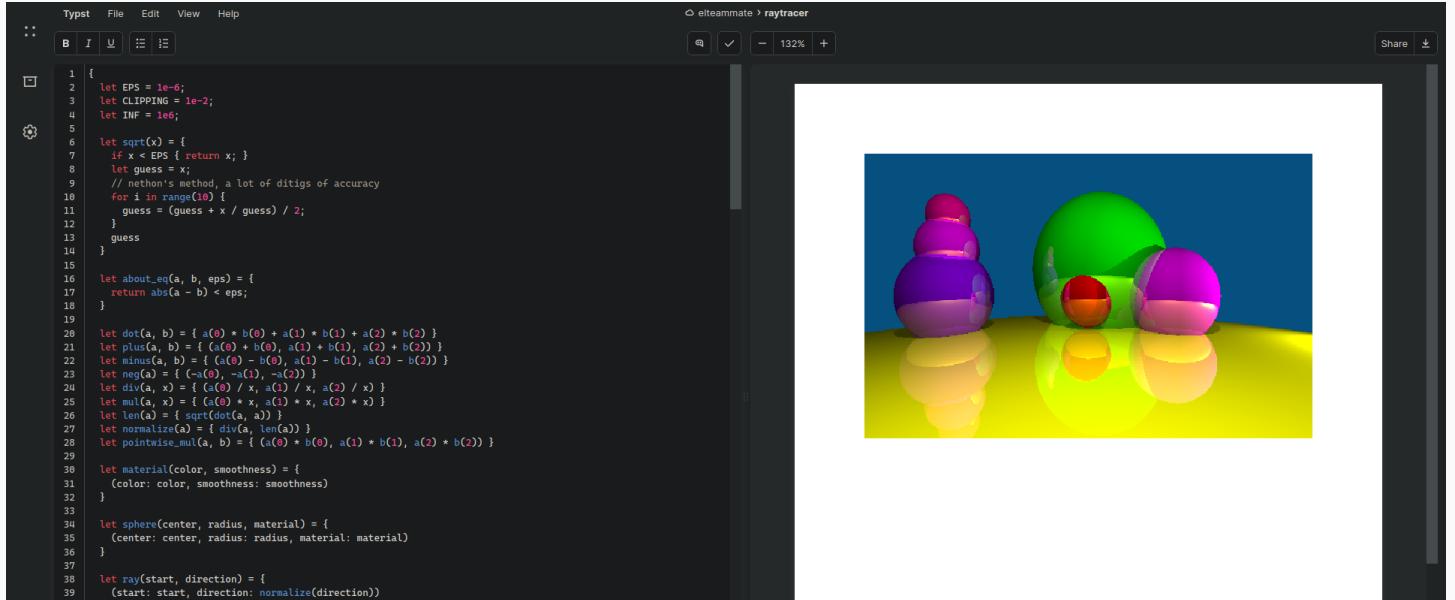
Für alle $\varepsilon > 0$ existiert ein $\delta > 0$, sodass $|f(x) - f(x_0)| < \varepsilon$ für alle $x \in D$ mit $|x - x_0| < \delta$.

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. Was Typst noch so alles kann

6.1. Raytracing



The image shows a code editor window with a dark theme. The left pane displays a block of TypeScript code for a raytracer. The right pane shows a 3D rendering of several spheres of different sizes and colors (purple, green, red, yellow) on a reflective 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 (let i = 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) = Math.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,
32     smoothness: smoothness
33   };
34
35   let sphere(center, radius, material) = {
36     center: center,
37     radius: radius,
38     material: material
39   };
39
40   let ray(start, direction) = [
41     start: start,
42     direction: normalize(direction)
43   ];
43
44   let intersect(ray, spheres) = [
45     ray: ray,
46     spheres: spheres
47   ];
47
48   let closestIntersection(ray, spheres) = [
49     ray: ray,
50     spheres: spheres
51   ];
51
52   let intersectSphere(ray, sphere) = [
53     ray: ray,
54     sphere: sphere
55   ];
55
56   let intersectRaySphere(ray, sphere) = [
57     ray: ray,
58     sphere: sphere
59   ];
59
60   let intersectRayRay(ray, ray) = [
61     ray: ray,
62     ray2: ray
63   ];
63
64   let intersectRayPlane(ray, plane) = [
65     ray: ray,
66     plane: plane
67   ];
67
68   let intersectPlanePlane(plane, plane2) = [
69     plane: plane,
70     plane2: plane2
71   ];
71
72   let intersectRayPlane(ray, plane) = [
73     ray: ray,
74     plane: plane
75   ];
75
76   let intersectPlaneRay(plane, ray) = [
77     plane: plane,
78     ray: ray
79   ];
79
80   let intersectRayRay(ray, ray2) = [
81     ray: ray,
82     ray2: ray2
83   ];
83
84   let intersectRaySphere(ray, sphere) = [
85     ray: ray,
86     sphere: sphere
87   ];
87
88   let intersectSphereRay(sphere, ray) = [
89     ray: ray,
90     sphere: sphere
91   ];
91
92   let intersectRayPlane(ray, plane) = [
93     ray: ray,
94     plane: plane
95   ];
95
96   let intersectPlaneRay(plane, ray) = [
97     plane: plane,
98     ray: ray
99   ];
99
100  let intersectRayRay(ray, ray2) = [
101    ray: ray,
102    ray2: ray2
103  ];
103
104  let intersectPlanePlane(plane, plane2) = [
105    plane: plane,
106    plane2: plane2
107  ];
107
108  let intersectPlaneRay(plane, ray) = [
109    ray: ray,
110    plane: plane
111  ];
111
112  let intersectRayPlane(ray, plane) = [
113    ray: ray,
114    plane: plane
115  ];
115
116  let intersectRaySphere(ray, sphere) = [
117    ray: ray,
118    sphere: sphere
119  ];
119
120  let intersectSphereRay(sphere, ray) = [
121    ray: ray,
122    sphere: sphere
123  ];
123
124  let intersectRayRay(ray, ray2) = [
125    ray: ray,
126    ray2: ray2
127  ];
127
128  let intersectPlanePlane(plane, plane2) = [
129    plane: plane,
130    plane2: plane2
131  ];
131
132  let intersectPlaneRay(plane, ray) = [
133    ray: ray,
134    plane: plane
135  ];
135
136  let intersectRayPlane(ray, plane) = [
137    ray: ray,
138    plane: plane
139  ];
139
140  let intersectRaySphere(ray, sphere) = [
141    ray: ray,
142    sphere: sphere
143  ];
143
144  let intersectSphereRay(sphere, ray) = [
145    ray: ray,
146    sphere: sphere
147  ];
147
148  let intersectRayRay(ray, ray2) = [
149    ray: ray,
150    ray2: ray2
151  ];
151
152  let intersectPlanePlane(plane, plane2) = [
153    plane: plane,
154    plane2: plane2
155  ];
155
156  let intersectPlaneRay(plane, ray) = [
157    ray: ray,
158    plane: plane
159  ];
159
160  let intersectRayPlane(ray, plane) = [
161    ray: ray,
162    plane: plane
163  ];
163
164  let intersectRaySphere(ray, sphere) = [
165    ray: ray,
166    sphere: sphere
167  ];
167
168  let intersectSphereRay(sphere, ray) = [
169    ray: ray,
170    sphere: sphere
171  ];
171
172  let intersectRayRay(ray, ray2) = [
173    ray: ray,
174    ray2: ray2
175  ];
175
176  let intersectPlanePlane(plane, plane2) = [
177    plane: plane,
178    plane2: plane2
179  ];
179
180  let intersectPlaneRay(plane, ray) = [
181    ray: ray,
182    plane: plane
183  ];
183
184  let intersectRayPlane(ray, plane) = [
185    ray: ray,
186    plane: plane
187  ];
187
188  let intersectRaySphere(ray, sphere) = [
189    ray: ray,
190    sphere: sphere
191  ];
191
192  let intersectSphereRay(sphere, ray) = [
193    ray: ray,
194    sphere: sphere
195  ];
195
196  let intersectRayRay(ray, ray2) = [
197    ray: ray,
198    ray2: ray2
199  ];
199
200  let intersectPlanePlane(plane, plane2) = [
201    plane: plane,
202    plane2: plane2
203  ];
203
204  let intersectPlaneRay(plane, ray) = [
205    ray: ray,
206    plane: plane
207  ];
207
208  let intersectRayPlane(ray, plane) = [
209    ray: ray,
210    plane: plane
211  ];
211
212  let intersectRaySphere(ray, sphere) = [
213    ray: ray,
214    sphere: sphere
215  ];
215
216  let intersectSphereRay(sphere, ray) = [
217    ray: ray,
218    sphere: sphere
219  ];
219
220  let intersectRayRay(ray, ray2) = [
221    ray: ray,
222    ray2: ray2
223  ];
223
224  let intersectPlanePlane(plane, plane2) = [
225    plane: plane,
226    plane2: plane2
227  ];
227
228  let intersectPlaneRay(plane, ray) = [
229    ray: ray,
230    plane: plane
231  ];
231
232  let intersectRayPlane(ray, plane) = [
233    ray: ray,
234    plane: plane
235  ];
235
236  let intersectRaySphere(ray, sphere) = [
237    ray: ray,
238    sphere: sphere
239  ];
239
240  let intersectSphereRay(sphere, ray) = [
241    ray: ray,
242    sphere: sphere
243  ];
243
244  let intersectRayRay(ray, ray2) = [
245    ray: ray,
246    ray2: ray2
247  ];
247
248  let intersectPlanePlane(plane, plane2) = [
249    plane: plane,
250    plane2: plane2
251  ];
251
252  let intersectPlaneRay(plane, ray) = [
253    ray: ray,
254    plane: plane
255  ];
255
256  let intersectRayPlane(ray, plane) = [
257    ray: ray,
258    plane: plane
259  ];
259
260  let intersectRaySphere(ray, sphere) = [
261    ray: ray,
262    sphere: sphere
263  ];
263
264  let intersectSphereRay(sphere, ray) = [
265    ray: ray,
266    sphere: sphere
267  ];
267
268  let intersectRayRay(ray, ray2) = [
269    ray: ray,
270    ray2: ray2
271  ];
271
272  let intersectPlanePlane(plane, plane2) = [
273    plane: plane,
274    plane2: plane2
275  ];
275
276  let intersectPlaneRay(plane, ray) = [
277    ray: ray,
278    plane: plane
279  ];
279
280  let intersectRayPlane(ray, plane) = [
281    ray: ray,
282    plane: plane
283  ];
283
284  let intersectRaySphere(ray, sphere) = [
285    ray: ray,
286    sphere: sphere
287  ];
287
288  let intersectSphereRay(sphere, ray) = [
289    ray: ray,
290    sphere: sphere
291  ];
291
292  let intersectRayRay(ray, ray2) = [
293    ray: ray,
294    ray2: ray2
295  ];
295
296  let intersectPlanePlane(plane, plane2) = [
297    plane: plane,
298    plane2: plane2
299  ];
299
300  let intersectPlaneRay(plane, ray) = [
301    ray: ray,
302    plane: plane
303  ];
303
304  let intersectRayPlane(ray, plane) = [
305    ray: ray,
306    plane: plane
307  ];
307
308  let intersectRaySphere(ray, sphere) = [
309    ray: ray,
310    sphere: sphere
311  ];
311
312  let intersectSphereRay(sphere, ray) = [
313    ray: ray,
314    sphere: sphere
315  ];
315
316  let intersectRayRay(ray, ray2) = [
317    ray: ray,
318    ray2: ray2
319  ];
319
320  let intersectPlanePlane(plane, plane2) = [
321    plane: plane,
322    plane2: plane2
323  ];
323
324  let intersectPlaneRay(plane, ray) = [
325    ray: ray,
326    plane: plane
327  ];
327
328  let intersectRayPlane(ray, plane) = [
329    ray: ray,
330    plane: plane
331  ];
331
332  let intersectRaySphere(ray, sphere) = [
333    ray: ray,
334    sphere: sphere
335  ];
335
336  let intersectSphereRay(sphere, ray) = [
337    ray: ray,
338    sphere: sphere
339  ];
339
340  let intersectRayRay(ray, ray2) = [
341    ray: ray,
342    ray2: ray2
343  ];
343
344  let intersectPlanePlane(plane, plane2) = [
345    plane: plane,
346    plane2: plane2
347  ];
347
348  let intersectPlaneRay(plane, ray) = [
349    ray: ray,
350    plane: plane
351  ];
351
352  let intersectRayPlane(ray, plane) = [
353    ray: ray,
354    plane: plane
355  ];
355
356  let intersectRaySphere(ray, sphere) = [
357    ray: ray,
358    sphere: sphere
359  ];
359
360  let intersectSphereRay(sphere, ray) = [
361    ray: ray,
362    sphere: sphere
363  ];
363
364  let intersectRayRay(ray, ray2) = [
365    ray: ray,
366    ray2: ray2
367  ];
367
368  let intersectPlanePlane(plane, plane2) = [
369    plane: plane,
370    plane2: plane2
371  ];
371
372  let intersectPlaneRay(plane, ray) = [
373    ray: ray,
374    plane: plane
375  ];
375
376  let intersectRayPlane(ray, plane) = [
377    ray: ray,
378    plane: plane
379  ];
379
380  let intersectRaySphere(ray, sphere) = [
381    ray: ray,
382    sphere: sphere
383  ];
383
384  let intersectSphereRay(sphere, ray) = [
385    ray: ray,
386    sphere: sphere
387  ];
387
388  let intersectRayRay(ray, ray2) = [
389    ray: ray,
390    ray2: ray2
391  ];
391
392  let intersectPlanePlane(plane, plane2) = [
393    plane: plane,
394    plane2: plane2
395  ];
395
396  let intersectPlaneRay(plane, ray) = [
397    ray: ray,
398    plane: plane
399  ];
399
400  let intersectRayPlane(ray, plane) = [
401    ray: ray,
402    plane: plane
403  ];
403
404  let intersectRaySphere(ray, sphere) = [
405    ray: ray,
406    sphere: sphere
407  ];
407
408  let intersectSphereRay(sphere, ray) = [
409    ray: ray,
410    sphere: sphere
411  ];
411
412  let intersectRayRay(ray, ray2) = [
413    ray: ray,
414    ray2: ray2
415  ];
415
416  let intersectPlanePlane(plane, plane2) = [
417    plane: plane,
418    plane2: plane2
419  ];
419
420  let intersectPlaneRay(plane, ray) = [
421    ray: ray,
422    plane: plane
423  ];
423
424  let intersectRayPlane(ray, plane) = [
425    ray: ray,
426    plane: plane
427  ];
427
428  let intersectRaySphere(ray, sphere) = [
429    ray: ray,
430    sphere: sphere
431  ];
431
432  let intersectSphereRay(sphere, ray) = [
433    ray: ray,
434    sphere: sphere
435  ];
435
436  let intersectRayRay(ray, ray2) = [
437    ray: ray,
438    ray2: ray2
439  ];
439
440  let intersectPlanePlane(plane, plane2) = [
441    plane: plane,
442    plane2: plane2
443  ];
443
444  let intersectPlaneRay(plane, ray) = [
445    ray: ray,
446    plane: plane
447  ];
447
448  let intersectRayPlane(ray, plane) = [
449    ray: ray,
450    plane: plane
451  ];
451
452  let intersectRaySphere(ray, sphere) = [
453    ray: ray,
454    sphere: sphere
455  ];
455
456  let intersectSphereRay(sphere, ray) = [
457    ray: ray,
458    sphere: sphere
459  ];
459
460  let intersectRayRay(ray, ray2) = [
461    ray: ray,
462    ray2: ray2
463  ];
463
464  let intersectPlanePlane(plane, plane2) = [
465    plane: plane,
466    plane2: plane2
467  ];
467
468  let intersectPlaneRay(plane, ray) = [
469    ray: ray,
470    plane: plane
471  ];
471
472  let intersectRayPlane(ray, plane) = [
473    ray: ray,
474    plane: plane
475  ];
475
476  let intersectRaySphere(ray, sphere) = [
477    ray: ray,
478    sphere: sphere
479  ];
479
480  let intersectSphereRay(sphere, ray) = [
481    ray: ray,
482    sphere: sphere
483  ];
483
484  let intersectRayRay(ray, ray2) = [
485    ray: ray,
486    ray2: ray2
487  ];
487
488  let intersectPlanePlane(plane, plane2) = [
489    plane: plane,
490    plane2: plane2
491  ];
491
492  let intersectPlaneRay(plane, ray) = [
493    ray: ray,
494    plane: plane
495  ];
495
496  let intersectRayPlane(ray, plane) = [
497    ray: ray,
498    plane: plane
499  ];
499
500  let intersectRaySphere(ray, sphere) = [
501    ray: ray,
502    sphere: sphere
503  ];
503
504  let intersectSphereRay(sphere, ray) = [
505    ray: ray,
506    sphere: sphere
507  ];
507
508  let intersectRayRay(ray, ray2) = [
509    ray: ray,
510    ray2: ray2
511  ];
511
512  let intersectPlanePlane(plane, plane2) = [
513    plane: plane,
514    plane2: plane2
515  ];
515
516  let intersectPlaneRay(plane, ray) = [
517    ray: ray,
518    plane: plane
519  ];
519
520  let intersectRayPlane(ray, plane) = [
521    ray: ray,
522    plane: plane
523  ];
523
524  let intersectRaySphere(ray, sphere) = [
525    ray: ray,
526    sphere: sphere
527  ];
527
528  let intersectSphereRay(sphere, ray) = [
529    ray: ray,
530    sphere: sphere
531  ];
531
532  let intersectRayRay(ray, ray2) = [
533    ray: ray,
534    ray2: ray2
535  ];
535
536  let intersectPlanePlane(plane, plane2) = [
537    plane: plane,
538    plane2: plane2
539  ];
539
540  let intersectPlaneRay(plane, ray) = [
541    ray: ray,
542    plane: plane
543  ];
543
544  let intersectRayPlane(ray, plane) = [
545    ray: ray,
546    plane: plane
547  ];
547
548  let intersectRaySphere(ray, sphere) = [
549    ray: ray,
550    sphere: sphere
551  ];
551
552  let intersectSphereRay(sphere, ray) = [
553    ray: ray,
554    sphere: sphere
555  ];
555
556  let intersectRayRay(ray, ray2) = [
557    ray: ray,
558    ray2: ray2
559  ];
559
560  let intersectPlanePlane(plane, plane2) = [
561    plane: plane,
562    plane2: plane2
563  ];
563
564  let intersectPlaneRay(plane, ray) = [
565    ray: ray,
566    plane: plane
567  ];
567
568  let intersectRayPlane(ray, plane) = [
569    ray: ray,
570    plane: plane
571  ];
571
572  let intersectRaySphere(ray, sphere) = [
573    ray: ray,
574    sphere: sphere
575  ];
575
576  let intersectSphereRay(sphere, ray) = [
577    ray: ray,
578    sphere: sphere
579  ];
579
580  let intersectRayRay(ray, ray2) = [
581    ray: ray,
582    ray2: ray2
583  ];
583
584  let intersectPlanePlane(plane, plane2) = [
585    plane: plane,
586    plane2: plane2
587  ];
587
588  let intersectPlaneRay(plane, ray) = [
589    ray: ray,
590    plane: plane
591  ];
591
592  let intersectRayPlane(ray, plane) = [
593    ray: ray,
594    plane: plane
595  ];
595
596  let intersectRaySphere(ray, sphere) = [
597    ray: ray,
598    sphere: sphere
599  ];
599
600  let intersectSphereRay(sphere, ray) = [
601    ray: ray,
602    sphere: sphere
603  ];
603
604  let intersectRayRay(ray, ray2) = [
605    ray: ray,
606    ray2: ray2
607  ];
607
608  let intersectPlanePlane(plane, plane2) = [
609    plane: plane,
610    plane2: plane2
611  ];
611
612  let intersectPlaneRay(plane, ray) = [
613    ray: ray,
614    plane: plane
615  ];
615
616  let intersectRayPlane(ray, plane) = [
617    ray: ray,
618    plane: plane
619  ];
619
620  let intersectRaySphere(ray, sphere) = [
621    ray: ray,
622    sphere: sphere
623  ];
623
624  let intersectSphereRay(sphere, ray) = [
625    ray: ray,
626    sphere: sphere
627  ];
627
628  let intersectRayRay(ray, ray2) = [
629    ray: ray,
630    ray2: ray2
631  ];
631
632  let intersectPlanePlane(plane, plane2) = [
633    plane: plane,
634    plane2: plane2
635  ];
635
636  let intersectPlaneRay(plane, ray) = [
637    ray: ray,
638    plane: plane
639  ];
639
640  let intersectRayPlane(ray, plane) = [
641    ray: ray,
642    plane: plane
643  ];
643
644  let intersectRaySphere(ray, sphere) = [
645    ray: ray,
646    sphere: sphere
647  ];
647
648  let intersectSphereRay(sphere, ray) = [
649    ray: ray,
650    sphere: sphere
651  ];
651
652  let intersectRayRay(ray, ray2) = [
653    ray: ray,
654    ray2: ray2
655  ];
655
656  let intersectPlanePlane(plane, plane2) = [
657    plane: plane,
658    plane2: plane2
659  ];
659
660  let intersectPlaneRay(plane, ray) = [
661    ray: ray,
662    plane: plane
663  ];
663
664  let intersectRayPlane(ray, plane) = [
665    ray: ray,
666    plane: plane
667  ];
667
668  let intersectRaySphere(ray, sphere) = [
669    ray: ray,
670    sphere: sphere
671  ];
671
672  let intersectSphereRay(sphere, ray) = [
673    ray: ray,
674    sphere: sphere
675  ];
675
676  let intersectRayRay(ray, ray2) = [
677    ray: ray,
678    ray2: ray2
679  ];
679
680  let intersectPlanePlane(plane, plane2) = [
681    plane: plane,
682    plane2: plane2
683  ];
683
684  let intersectPlaneRay(plane, ray) = [
685    ray: ray,
686    plane: plane
687  ];
687
688  let intersectRayPlane(ray, plane) = [
689    ray: ray,
690    plane: plane
691  ];
691
692  let intersectRaySphere(ray, sphere) = [
693    ray: ray,
694    sphere: sphere
695  ];
695
696  let intersectSphereRay(sphere, ray) = [
697    ray: ray,
698    sphere: sphere
699  ];
699
700  let intersectRayRay(ray, ray2) = [
701    ray: ray,
702    ray2: ray2
703  ];
703
704  let intersectPlanePlane(plane, plane2) = [
705    plane: plane,
706    plane2: plane2
707  ];
707
708  let intersectPlaneRay(plane, ray) = [
709    ray: ray,
710    plane: plane
711  ];
711
712  let intersectRayPlane(ray, plane) = [
713    ray: ray,
714    plane: plane
715  ];
715
716  let intersectRaySphere(ray, sphere) = [
717    ray: ray,
718    sphere: sphere
719  ];
719
720  let intersectSphereRay(sphere, ray) = [
721    ray: ray,
722    sphere: sphere
723  ];
723
724  let intersectRayRay(ray, ray2) = [
725    ray: ray,
726    ray2: ray2
727  ];
727
728  let intersectPlanePlane(plane, plane2) = [
729    plane: plane,
730    plane2: plane2
731  ];
731
732  let intersectPlaneRay(plane, ray) = [
733    ray: ray,
734    plane: plane
735  ];
735
736  let intersectRayPlane(ray, plane) = [
737    ray: ray,
738    plane: plane
739  ];
739
740  let intersectRaySphere(ray, sphere) = [
741    ray: ray,
742    sphere: sphere
743  ];
743
744  let intersectSphereRay(sphere, ray) = [
745    ray: ray,
746    sphere: sphere
747  ];
747
748  let intersectRayRay(ray, ray2) = [
749    ray: ray,
750    ray2: ray2
751  ];
751
752  let intersectPlanePlane(plane, plane2) = [
753    plane: plane,
754    plane2: plane2
755  ];
755
756  let intersectPlaneRay(plane, ray) = [
757    ray: ray,
758    plane: plane
759  ];
759
760  let intersectRayPlane(ray, plane) = [
761    ray: ray,
762    plane: plane
763  ];
763
764  let intersectRaySphere(ray, sphere) = [
765    ray: ray,
766    sphere: sphere
767  ];
767
768  let intersectSphereRay(sphere, ray) = [
769    ray: ray,
770    sphere: sphere
771  ];
771
772  let intersectRayRay(ray, ray2) = [
773    ray: ray,
774    ray2: ray2
775  ];
775
776  let intersectPlanePlane(plane, plane2) = [
777    plane: plane,
778    plane2: plane2
779  ];
779
780  let intersectPlaneRay(plane, ray) = [
781    ray: ray,
782    plane: plane
783  ];
783
784  let intersectRayPlane(ray, plane) = [
785    ray: ray,
786    plane: plane
787  ];
787
788  let intersectRaySphere(ray, sphere) = [
789    ray: ray,
790    sphere: sphere
791  ];
791
792  let intersectSphereRay(sphere, ray) = [
793    ray: ray,
794    sphere: sphere
795  ];
795
796  let intersectRayRay(ray, ray2) = [
797    ray: ray,
798    ray2: ray2
799  ];
799
800  let intersectPlanePlane(plane, plane2) = [
801    plane: plane,
802    plane2: plane2
803  ];
803
804  let intersectPlaneRay(plane, ray) = [
805    ray: ray,
806    plane: plane
807  ];
807
808  let intersectRayPlane(ray, plane) = [
809    ray: ray,
810    plane: plane
811  ];
811
812  let intersectRaySphere(ray, sphere) = [
813    ray: ray,
814    sphere: sphere
815  ];
815
816  let intersectSphereRay(sphere, ray) = [
817    ray: ray,
818    sphere: sphere
819  ];
819
820  let intersectRayRay(ray, ray2) = [
821    ray: ray,
822    ray2: ray2
823  ];
823
824  let intersectPlanePlane(plane, plane2) = [
825    plane: plane,
826    plane2: plane2
827  ];
827
828  let intersectPlaneRay(plane, ray) = [
829    ray: ray,
830    plane: plane
831  ];
831
832  let intersectRayPlane(ray, plane) = [
833    ray: ray,
834    plane: plane
835  ];
835
836  let intersectRaySphere(ray, sphere) = [
837    ray: ray,
838    sphere: sphere
839  ];
839
840  let intersectSphereRay(sphere, ray) = [
841    ray: ray,
842    sphere: sphere
843  ];
843
844  let intersectRayRay(ray, ray2) = [
845    ray: ray,
846    ray2: ray2
847  ];
847
848  let intersectPlanePlane(plane, plane2) = [
849    plane: plane,
850    plane2: plane2
851  ];
851
852  let intersectPlaneRay(plane, ray) = [
853    ray: ray,
854    plane: plane
855  ];
855
856  let intersectRayPlane(ray, plane) = [
857    ray: ray,
858    plane: plane
859  ];
859
860  let intersectRaySphere(ray, sphere) = [
861    ray: ray,
862    sphere: sphere
863  ];
863
864  let intersectSphereRay(sphere, ray) = [
865    ray: ray,
866    sphere: sphere
867  ];
867
868  let intersectRayRay(ray, ray2) = [
869    ray: ray,
870    ray2: ray2
871  ];
871
872  let intersectPlanePlane(plane, plane2) = [
873    plane: plane,
874    plane2: plane2
875  ];
875
876  let intersectPlaneRay(plane, ray) = [
877    ray: ray,
878    plane: plane
879  ];
879
880  let intersectRayPlane(ray, plane) = [
881    ray: ray,
882    plane: plane
883  ];
883
884  let intersectRaySphere(ray, sphere) = [
885    ray: ray,
886    sphere: sphere
887  ];
887
888  let intersectSphereRay(sphere, ray) = [
889    ray: ray,
890    sphere: sphere
891  ];
891
892  let intersectRayRay(ray, ray2) = [
893    ray: ray,
894    ray2: ray2
895  ];
895
896  let intersectPlanePlane(plane, plane2) = [
897    plane: plane,
898    plane2: plane2
899  ];
899
900  let intersectPlaneRay(plane, ray) = [
901    ray: ray,
902    plane: plane
903  ];
903
904  let intersectRayPlane(ray, plane) = [
905    ray: ray,
906    plane: plane
907  ];
907
908  let intersectRaySphere(ray, sphere) = [
909    ray: ray,
910    sphere: sphere
911  ];
911
912  let intersectSphereRay(sphere, ray) = [
913    ray: ray,
914    sphere: sphere
915  ];
915
916  let intersectRayRay(ray, ray2) = [
917    ray: ray,
918    ray2: ray2
919  ];
919
920  let intersectPlanePlane(plane, plane2) = [
921    plane: plane,
922    plane2: plane2
923  ];
923
924  let intersectPlaneRay(plane, ray) = [
925    ray: ray,
926    plane: plane
927  ];
927
928  let intersectRayPlane(ray, plane) = [
929    ray: ray,
930    plane: plane
931  ];
931
932  let intersectRaySphere(ray, sphere) = [
933    ray: ray,
934    sphere: sphere
935  ];
935
936  let intersectSphereRay(sphere, ray) = [
937    ray: ray,
938    sphere: sphere
939  ];
939
940  let intersectRayRay(ray, ray2) = [
941    ray: ray,
942    ray2: ray2
943  ];
943
944  let intersectPlanePlane(plane, plane2) = [
945    plane: plane,
946    plane2: plane2
947  ];
947
948  let intersectPlaneRay(plane, ray) = [
949    ray: ray,
950    plane: plane
951  ];
951
952  let intersectRayPlane(ray, plane) = [
953    ray: ray,
954    plane: plane
955  ];
955
956  let intersectRaySphere(ray, sphere) = [
957    ray: ray,
958    sphere: sphere
959  ];
959
960  let intersectSphereRay(sphere, ray) = [
961    ray: ray,
962    sphere: sphere
963  ];
963
964  let intersectRayRay(ray, ray2) = [
965    ray: ray,
966    ray2: ray2
967  ];
967
968  let intersectPlanePlane(plane, plane2) = [
969    plane: plane,
970    plane2: plane2
971  ];
971
972  let intersectPlaneRay(plane, ray) = [
973    ray: ray,
974    plane: plane
975  ];
975
976  let intersectRayPlane(ray, plane) = [
977    ray: ray,
978    plane: plane
979  ];
979
980  let intersectRaySphere(ray, sphere) = [
981    ray: ray,
982    sphere: sphere
983  ];
983
984  let intersectSphereRay(sphere, ray) = [
985    ray: ray,
986    sphere: sphere
987  ];
987
988  let intersectRayRay(ray, ray2) = [
989    ray: ray,
990    ray2: ray2
991  ];
991
992  let intersectPlanePlane(plane, plane2) = [
993    plane: plane,
994    plane2: plane2
995  ];
995
996  let intersectPlaneRay(plane, ray) = [
997    ray: ray,
998    plane: plane
999  ];
999
1000 let intersectRayPlane(ray, plane) = [
1001   ray: ray,
1002   plane: plane
1003 ];
1003
1004 let intersectRaySphere(ray, sphere) = [
1005   ray: ray,
1006   sphere: sphere
1007 ];
1007
1008 let intersectSphereRay(sphere, ray) = [
1009   ray: ray,
1010   sphere: sphere
1011 ];
1011
1012 let intersectRayRay(ray, ray2) = [
1013   ray: ray,
1014   ray2: ray2
1015 ];
1015
1016 let intersectPlanePlane(plane, plane2) = [
1017   plane: plane,
1018   plane2: plane2
1019 ];
1019
1020 let intersectPlaneRay(plane, ray) = [
1021   ray: ray,
1022   plane: plane
1023 ];
1023
1024 let intersectRayPlane(ray, plane) = [
1025   ray: ray,
1026   plane: plane
1027 ];
1027
1028 let intersectRaySphere(ray, sphere) = [
1029   ray: ray,
1030   sphere: sphere
1031 ];
1031
1032 let intersectSphereRay(sphere, ray) = [
1033   ray: ray,
1034   sphere: sphere
1035 ];
1035
1036 let intersectRayRay(ray, ray2) = [
1037   ray: ray,
1038   ray2: ray2
1039 ];
1039
1040 let intersectPlanePlane(plane, plane2) = [
1041   plane: plane,
1042   plane2: plane2
1043 ];
1043
1044 let intersectPlaneRay(plane, ray) = [
1045   ray: ray,
1046   plane: plane
1047 ];
1047
1048 let intersectRayPlane(ray, plane) = [
1049   ray: ray,
1050   plane: plane
1051 ];
1051
1052 let intersectRaySphere(ray, sphere) = [
1053   ray: ray,
1054   sphere: sphere
1055 ];
1055
1056 let intersectSphereRay(sphere, ray) = [
1057   ray: ray,
1058   sphere: sphere
1059 ];
1059
1060 let intersectRayRay(ray, ray2) = [
1061   ray: ray,
1062   ray2: ray2
1063 ];
1063
1064 let intersectPlanePlane(plane, plane2) = [
1065   plane: plane,
1066   plane2: plane2
1067 ];
1067
1068 let intersectPlaneRay(plane, ray) = [
1069   ray: ray,
1070   plane: plane
1071 ];
1071
1072 let intersectRayPlane(ray, plane) = [
1073   ray: ray,
1074   plane: plane
1075 ];
1075
1076 let intersectRaySphere(ray, sphere) = [
1077   ray: ray,
1078   sphere: sphere
1079 ];
1079
1080 let intersectSphereRay(sphere, ray) = [
1081   ray: ray,
1082   sphere: sphere
1083 ];
1083
1084 let intersectRayRay(ray, ray2) = [
1085   ray: ray,
1086   ray2: ray2
1087 ];
1087
1088 let intersectPlanePlane(plane, plane2) = [
1089   plane: plane,
1090   plane2: plane2
1091 ];
1091
1092 let intersectPlaneRay(plane, ray) = [
1093   ray: ray,
1094   plane: plane
1095 ];
1095
1096 let intersectRayPlane(ray, plane) = [
1097   ray: ray,
1098   plane: plane
1099 ];
1099
1100 let intersectRaySphere(ray, sphere) = [
1101   ray: ray,
1102   sphere: sphere
1103 ];
1103
1104 let intersectSphereRay(sphere, ray) = [
1105   ray: ray,
1106   sphere: sphere
1107 ];
1107
1108 let intersectRayRay(ray, ray2) = [
1109   ray: ray,
1110   ray2: ray2
1111 ];
1111
1112 let intersectPlanePlane(plane, plane2) = [
1113   plane: plane,
1114   plane2: plane2
1115 ];
1115
1116 let intersectPlaneRay(plane, ray) = [
1117   ray: ray,
1118   plane: plane
1119 ];
1119
1120 let intersectRayPlane(ray, plane) = [
1121   ray: ray,
1122   plane: plane
1123 ];
1123
1124 let intersectRaySphere(ray, sphere) = [
1125   ray: ray,
1126   sphere: sphere
1127 ];
1127
1128 let intersectSphereRay(sphere, ray) = [
1129   ray: ray,
1130   sphere: sphere
1131 ];
1131
1132 let intersectRayRay(ray, ray2) = [
1133   ray: ray,
1134   ray2: ray2
1135 ];
1135
1136 let intersectPlanePlane(plane, plane2) = [
1137   plane: plane,
1138   plane2: plane2
1139 ];
1139
1140 let intersectPlaneRay(plane, ray) = [
1141   ray: ray,
1142   plane: plane
1143 ];
1143
1144 let intersectRayPlane(ray, plane) = [
1145   ray: ray,
1146   plane: plane
1147 ];
1147
1148 let intersectRaySphere(ray, sphere) = [
1149   ray: ray,
1150   sphere: sphere
1151 ];
1151
1152 let intersectSphereRay(sphere, ray) = [
1153   ray: ray,
1154   sphere: sphere
1155 ];
1155
1156 let intersectRayRay(ray, ray2) = [
1157   ray: ray,
1158   ray2: ray2
1159 ];
1159
1160 let intersectPlanePlane(plane, plane2) = [
1161   plane: plane,
1162   plane2: plane2
1163 ];
1163
1164 let intersectPlaneRay(plane, ray) = [
1165   ray: ray,
1166   plane: plane
1167 ];
1167
1168 let intersectRayPlane(ray, plane) = [
1169   ray: ray,
1170   plane: plane
1171 ];
1171
1172 let intersectRaySphere(ray, sphere) = [
1173   ray: ray,
1174   sphere: sphere
117
```

7. Was noch fehlt

8. Wer sollte Typst benutzen?

9. Abschluss und Weiteres

A fancy dynamic slide without a title.

A fancy dynamic slide without a title. This appears later!¹

Focus!

9.1. Take home message

Read the book!

Try it out!

Create themes!

*Lorem ipsum dolor sit amet, consectetur adipiscing
 elit, sed do eiusmod tempor incididunt ut labore et
 dolore magnam aliquam quaerat voluptatem. Ut
enim aequa doleamus animo, cum corpore dolemus,
 fieri tamen permagna accessio potest, si aliquod
 aeternum et infinitum impendere.*

9.2. On the right!

*Lorem ipsum dolor sit amet,
consectetur adipiscing elit, sed do
 eiusmod tempor incididunt ut
labore et dolore magna aliqua
quaerat voluptatem. Ut enim aequa
 doleamus animo, cum corpore
dolemus, fieri tamen permagna
accessio potest, si aliquod aeternum
 et infinitum impendere.*

Lorem ipsum dolor sit amet,
consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut
labore et dolore magna aliquam
quaerat voluptatem. Ut enim aequa
doleamus animo, cum corpore
dolemus, fieri tamen permagna
accessio potest, si aliquod aeternum
et infinitum impendere.

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>