A GLIMPSE OF

typst

Plan

1. Introduction to Typst
2. In practice: comparison with LaTeX
3. Callisto, a package for literate programming

Typst is a
typesetting
system

The project in numbers (as of April 2026)

Started by 2 students at TU Berlin, currently a team of 5.

Command-line compiler

- Open source release in 2023
- 414 contributors
  - 20 in last month

Community and ecosystem

- 2934 users on the forum
- 1269 packages

Webapp

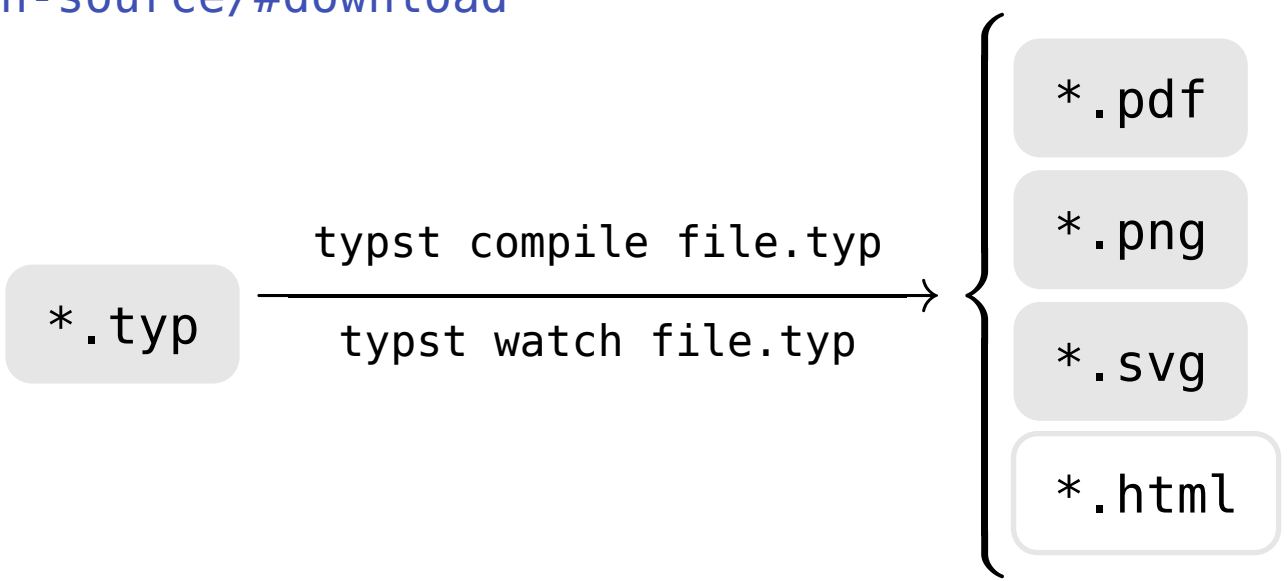
- 200K users
- 600K projects

Funding

- WebApp (Pro plan and On-Premises)
- One engineer funded by Posit (Quarto)
- Initial €225k angel + EU grants

Local installation

typst.app/open-source/#download



Webapp

Try it at `typst.app/play/`

Free plan

✓ Live preview

✓ Real-time collaboration

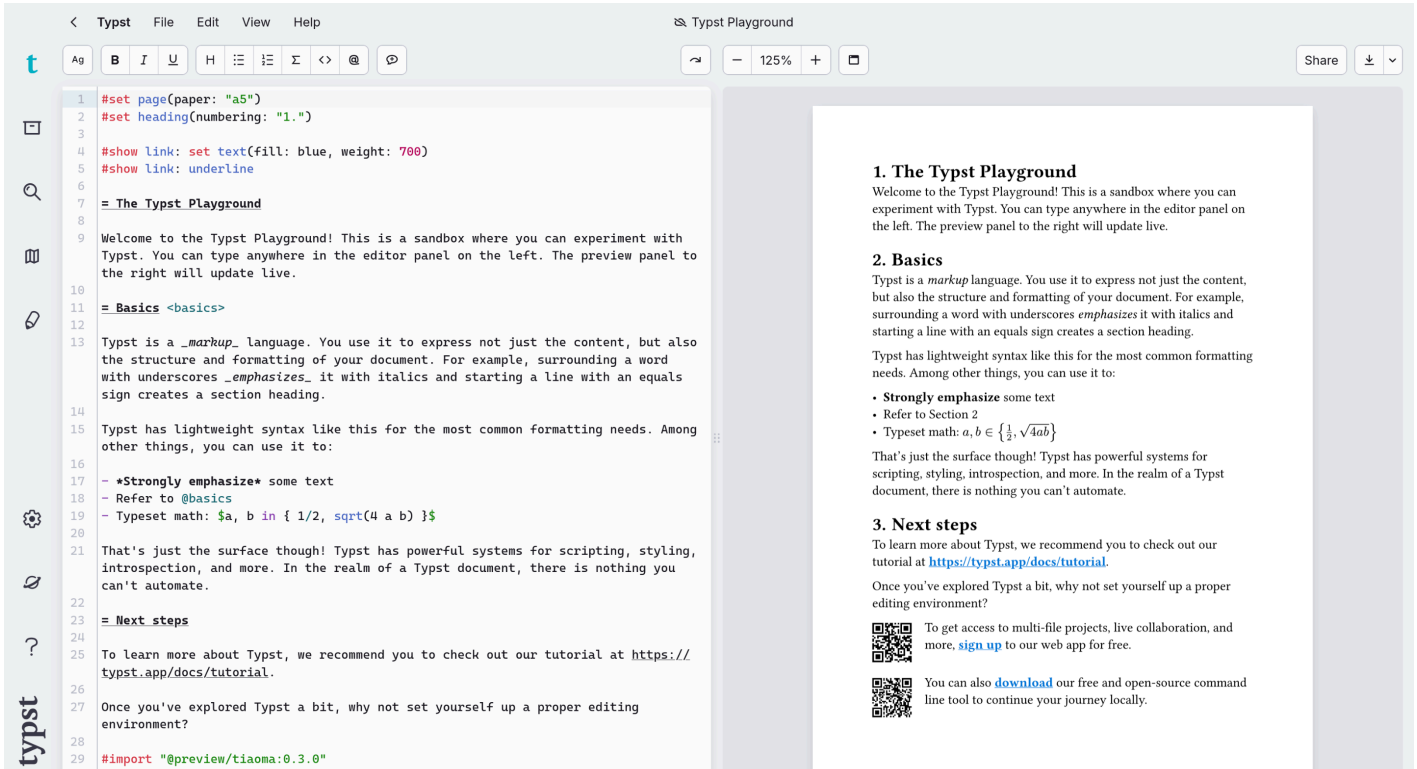
Pro plan

✓ Comments

✓ GitHub/GitLab sync

✓ Zotero/Mendeley sync

✓ Presentation mode (pointer, pen, speaker mode…)



Typst is a
markup
language

3 modes:

markup, *math*, `code`

A source file starts in Markup mode

```
= Introduction
Some things are *important* and
must be _emphasized_.

== First approach <part1>
Some more text.

== Second approach
Something different from @part1.

```

1. Introduction

Some things are **important** and must be *emphasized*.

1.1. First approach

Some more text.

1.2. Second approach

Something different from Section 1.1.

```
Enumerations:
+ Lemon
+ Banana
```

Enumerations:

1. Lemon
2. Banana

```
+ Lemon

+ Banana
```

1. Lemon

2. Banana

````
A code block:
```py
x = 3
print(x + 5)
```
````

A code block:

```py
x = 3
print(x + 5)
```

Switch to math mode: $ ... $

```
Euler's formula:
$
  e^(i theta)
  = sum_(n=0)^oo (i theta)^n / n!
  = cos(theta) + i sin(theta).
$
```

Euler's formula:

$$e^{i\theta} = \sum_{n=0}^{\infty} \frac{(i\theta)^n}{n!} = \cos(\theta) + i\sin(\theta).$$

```
Using $theta = pi$ the formula
gives $e^(i pi) = -1$.
```

Using $\theta = \pi$ the formula gives $e^{i\pi} = -1$.

```
$
  A inter B = emptyset =>
  A without B = A
$
```

$$A \cap B = \varnothing \Rightarrow A \setminus B = A$$

```
$
  Q = 2 pi E_"start" /
    E_"dissipated"
$
```

$$Q = 2\pi \frac{E_{\text{start}}}{E_{\text{dissipated}}}$$

```
$ W = arrow(F) dot arrow(d) $
```

$$W = \vec{F} \cdot \vec{d}$$

```
$ QQ = { p/q : p in ZZ, q in NN } $
```

$$\mathbb{Q} = \left\{ \frac{p}{q} : p \in \mathbb{Z}, q \in \mathbb{N} \right\}$$

```
$
  vec(3,9,15) =
  mat(1,2,3; 4,5,6; 7,8,9)
  vec(1,1,0)
$
```

$$\begin{pmatrix} 3 \\ 9 \\ 15 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

```
$
  f(x) &= (x+1)^2 &&= x^2 + 2x + 1 \
  g(x) &= 10(x-1) &&= 10x - 10
$
```

$$f(x) = (x+1)^2 \quad = x^2 + 2x + 1$$
$$g(x) = 10(x-1) = 10x - 10$$

Switch to code mode: `#...`

```
Example: bold text

- With *markup*

- With #strong[function call]
```

Example: bold text

- With **markup**
- With **function call**

Here `[]` goes back to markup mode. Compare:

```
- #strong[Some _markup_ value]
- #strong("Some _string_ value")
```

- **Some *markup* value**
- **Some _string_ value**

```
#let n = 3

Powers of 2 up to $n = #n$:

#for i in range(n+1) {
  let value = calc.pow(2, i)
  $ 2^#i = #value $
}
```

Powers of 2 up to $n = 3$:

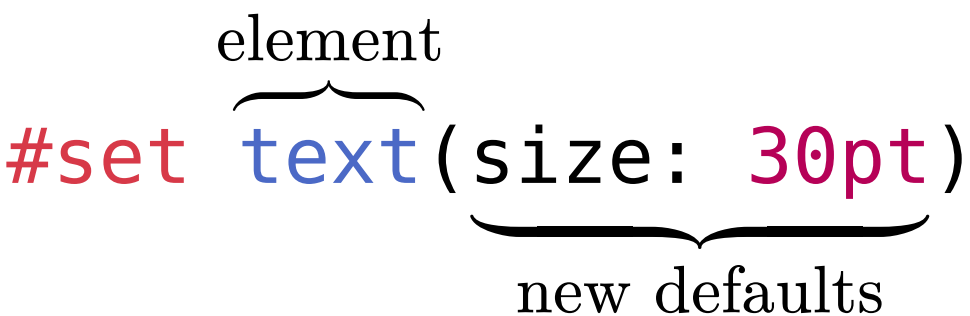
$$2^0 = 1$$

$$2^1 = 2$$

$$2^2 = 4$$

$$2^3 = 8$$

set & show

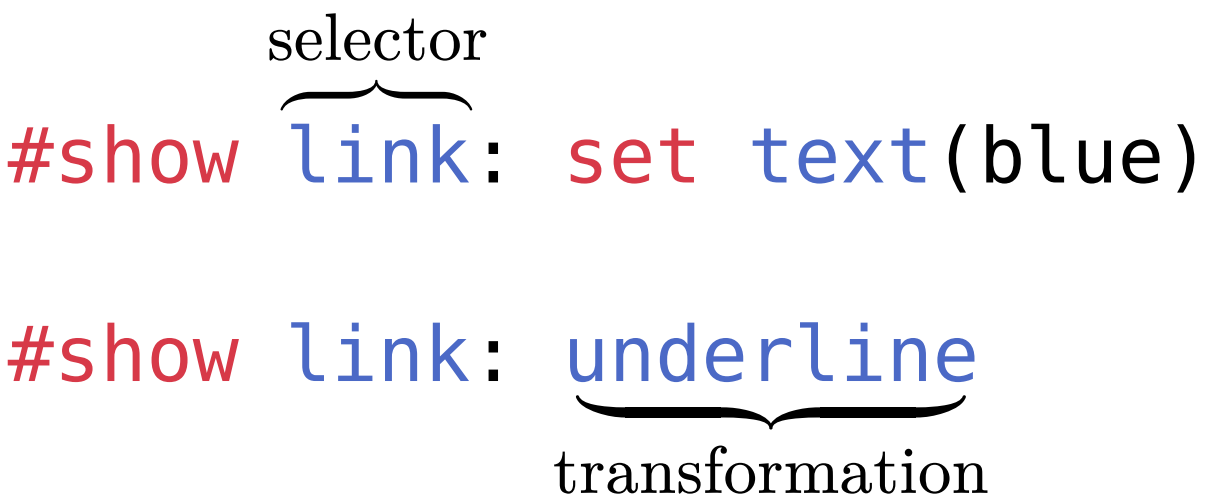

element
#set text(size: 30pt)
new defaults

```
#set text(
  font: "New Computer Modern",
  size: 24pt,
)

This is the familiar LaTeX font.
```

This is the familiar LaTeX font.

```
#set enum(numbering: "a1)")
+ Element A
  + Element A1
  + Element A2
+ Element B
```

a) Element A
   1) Element A1
   2) Element A2
b) Element B


selector
#show link: set text(blue)
#show link: underline
transformation

```
#show link: set text(red)
#show link: underline
Go to https://www.example.com.
```

Go to https://www.example.com.

```
// Anonymous transformation function
#show link: it => [🔗] + it
Go to https://www.example.com.
```

Go to 🔗https://www.example.com.

```
#set text(font: "Pennstander")

#show math.equation: set text(
  font: "Pennstander Math",
)

The definite integral of $f$
on $[a, b]$ is defined as

$
  integral_a^b = lim_(n->oo)
    (b-a)/n sum_(i=1)^n f(xi_i)
$
```

The definite integral of f on $[a, b]$ is defined as

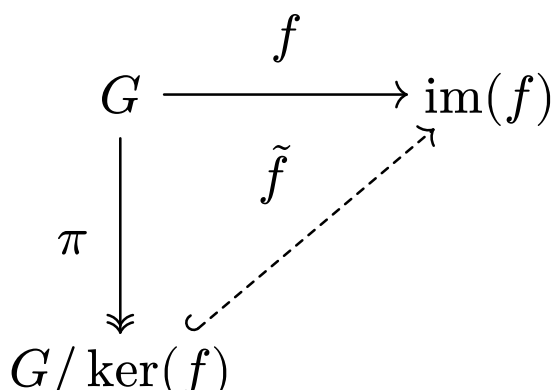
$$\int_a^b = \lim_{n \to \infty} \frac{b-a}{n} \sum_{i=1}^{n} f(\xi_i)$$

Packages

from the **Typst Universe**

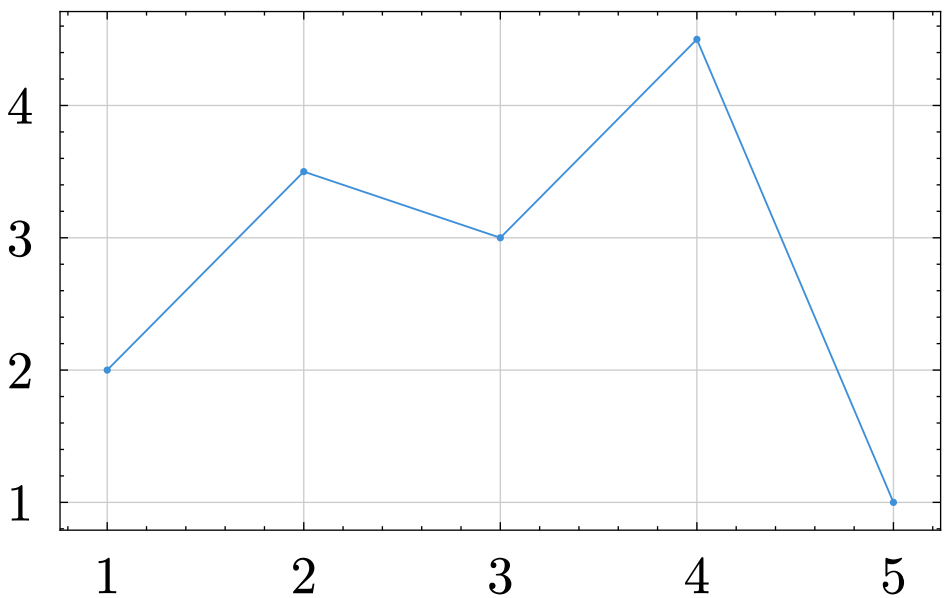
Diagrams with fletcher

```
#import "@preview/fletcher:0.5.8": *

#diagram(cell-size: 15mm, $
  G edge(f, ->) edge("d", pi, ->>) & im(f) \
  G slash ker(f) edge("ur", tilde(f), "hook-->")
$)
```



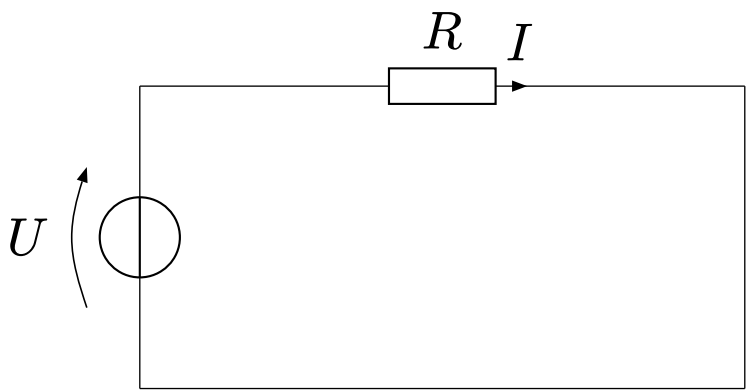
Plotting from CSV with Lilaq

```
#import "@preview/lilaq:0.6.0"

#let data = lilaq.load-txt(
  read("data.csv"), header: true)

#lilaq.diagram(
  lilaq.plot(data.t, data.U),
  width: 100% - 1cm,
  height: 60%,
)
```



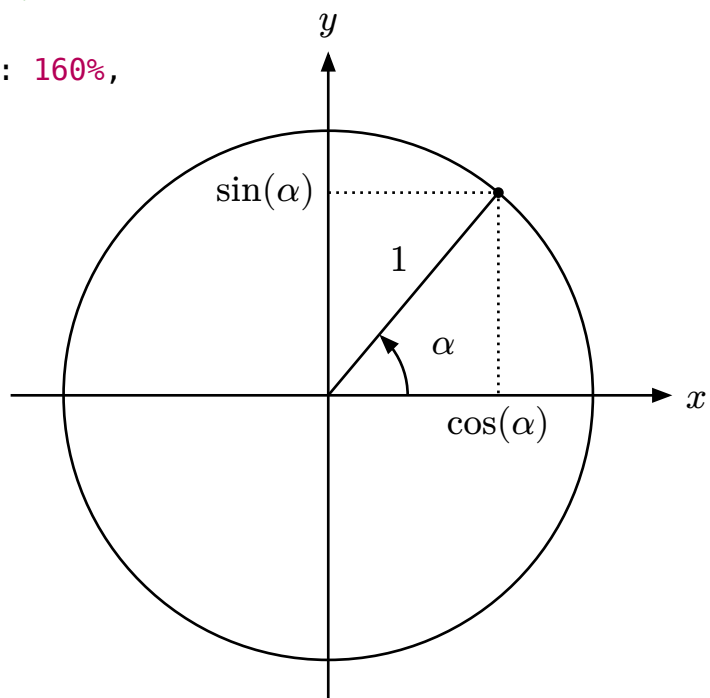
Electronic circuits with Zap

```
#import "@preview/zap:0.5.0"

#zap.circuit({
  import zap: *

  vsource("v1", (0, 0), (0, 4),
    u: $U$)
  resistor("r1", "v1.out", (8, 4),
    label: $R$, i: $I$)
  wire((8, 4), (8, 0))
  wire((8, 0), (0, 0))
})
```



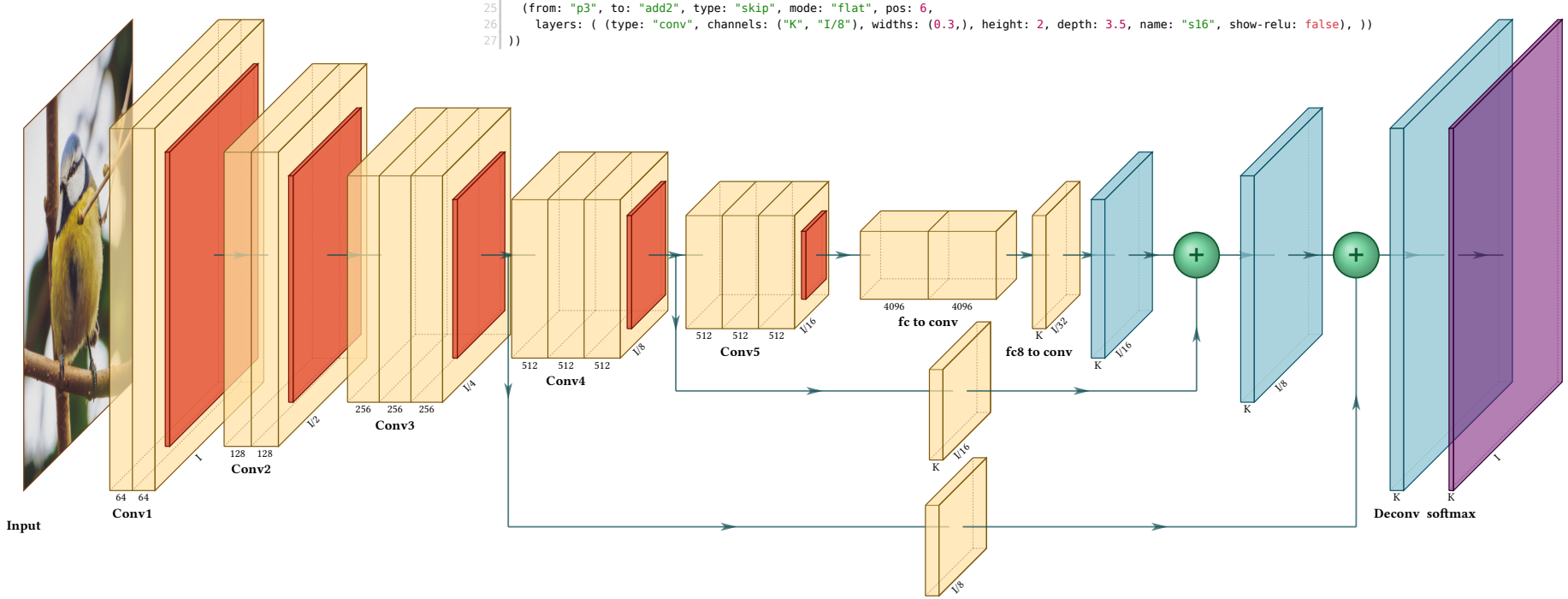
General-purpose drawing with CeTZ

```
#import "@preview/cetz:0.4.2": canvas, draw, angle
#canvas(length: 3.5cm, {
  import draw: *
  let my-alpha = 50deg
  let arrow = (end: ">", fill: black)
  let node = content.with(padding: 5pt)
  let P = (my-alpha, 1)
  let Px = (P, "|-", (0, 0))
  let Py = (P, "-|", (0, 0))
  line((-1.2, 0), (1.3, 0), mark: arrow)
  node((), anchor: "west", $x$)
  line((0, -1.2), (0, 1.3), mark: arrow)
  node((), anchor: "south", $y$)
  circle((0, 0), radius: 1)
  circle(P, radius: 1.5pt, fill: black)
  line((0, 0), P)
  line(Py, P, Px, stroke: (dash: "dotted"))
  node(Px, anchor: "north", $cos(alpha)$)
  node(Py, anchor: "east", $sin(alpha)$)
  node((my-alpha, 0.55), anchor: "south-east", $1$)
  angle.angle(
    (0, 0), (1, 0), P,
    label: $alpha$,
    radius: 0.3,
    label-radius: 160%,
    mark: arrow)
})
```



Neural network architectures with neural-netz

```
#import "@preview/neural-netz:0.3.0": draw-network
#draw-network((
  (type: "conv", channels: ("256", "256", "256", "I/4"), widths: (0.7, 0.7, 0.7), height: 5, depth: 5, label: "Conv3", name: "c3"),
  (type: "pool", height: 3.5, depth: 3.5, name: "p3"),
  (type: "conv", channels: ("512", "512", "512", "I/8"), widths: (0.8, 0.8, 0.8), height: 3.5, depth: 3.5, label: "Conv4", name: "c4"),
  (type: "pool", height: 2.5, depth: 2.5, name: "p4"),
  (type: "conv", channels: ("512", "512", "512", "I/16"), widths: (0.8, 0.8, 0.8), height: 2.5, depth: 2.5, label: "Conv5", name: "c5"),
  (type: "pool", height: 1.5, depth: 1.5, name: "p5"),
  (type: "conv", channels: ("4096", "4096"), widths: (1.5, 1.5), height: 1.5, depth: 1.5, label: "fc to conv", name: "fc"),
  (type: "conv", channels: ("K", "I/32"), widths: (0.3,), height: 2.5, depth: 2.5, label: "fc8 to conv", name: "s32", offset: 0.8, show-relu: false),
  (type: "deconv", channels: ("K", "I/16"), height: 3.5, depth: 3.5, name: "up1", offset: 1),
  (type: "sum", radius: 0.5, symbol: "+", name: "add1", offset: 1),
  (type: "deconv", height: 5, depth: 5, channels: ("K", "I/8"), name: "up2", offset: 0.5),
  (type: "sum", radius: 0.5, symbol: "+", name: "add2", offset: 1),
  (type: "deconv", height: 8, depth: 8, channels: ("K",), label: "Deconv", name: "up3", offset: 0.5),
  (type: "convsoftmax", height: 8, depth: 8, channels: ("K", "I"), label: "softmax", offset: 1),
), connections: (
  (from: "p4", to: "add1", type: "skip", mode: "flat", pos: 3,
    layers: ( (type: "conv", channels: ("K", "I/16"), widths: (0.3,), height: 2, depth: 3.5, name: "s16", show-relu: false), ) ),
  (from: "p3", to: "add2", type: "skip", mode: "flat", pos: 6,
    layers: ( (type: "conv", channels: ("K", "I/8"), widths: (0.3,), height: 2, depth: 3.5, name: "s16", show-relu: false), ))
))
```



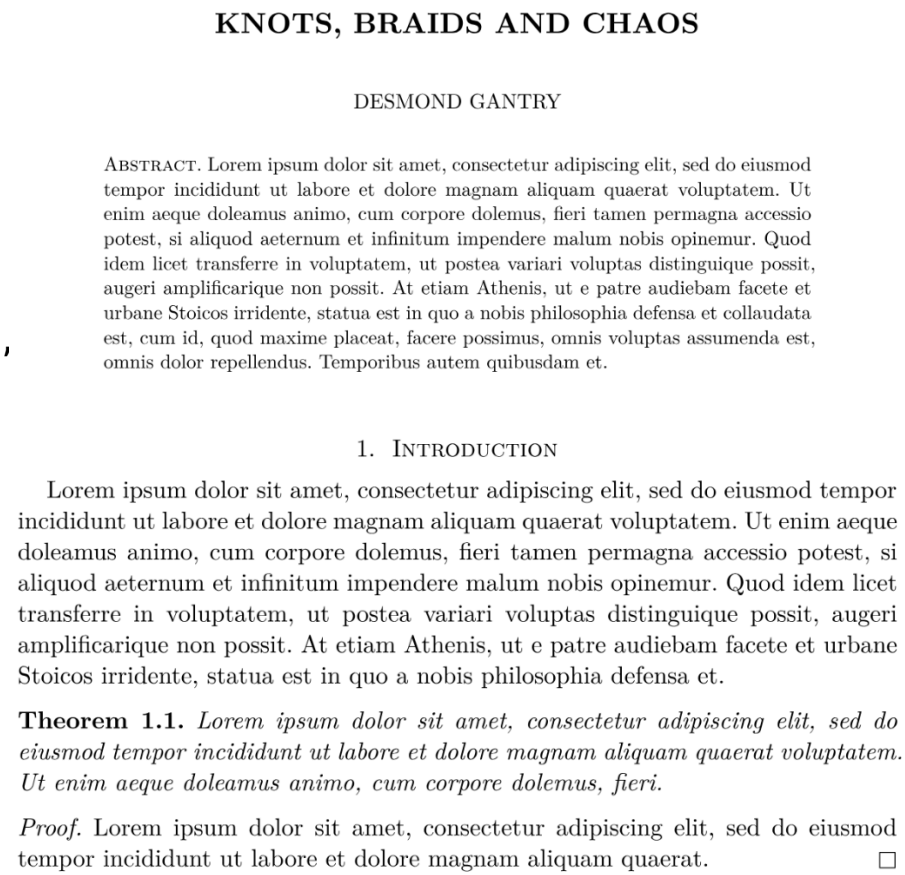
Templates

select all content below

```
#show: template
```

transformation

```
#import "@preview/unequivocal-ams:0.1.2": *
#show: ams-article.with(
  title: [Knots, Braids and Chaos],
  authors: (
    (
      name: "Desmond Gantry",
      department: [Department of Mathematics],
      organization: [University of Rummidge],
      location: [Rummidge RU15 2TT, United Kingdom],
      email: "d.gantry@rummidge.ac.uk",
      url: "www.rummidge.ac.uk/~dp01"
    ),
  ),
  abstract: lorem(100),
)

= Introduction
#lorem(80)

#theorem(lorem(30))
#proof(lorem(20))
```



KNOTS, BRAIDS AND CHAOS

DESMOND GANTRY

Abstract. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et.

1. Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et.

Theorem 1.1. *Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri.*

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

In practice

Comparison with LaTeX

Installation

Typst

- 1 file
- 50MB
- Packages downloaded as necessary

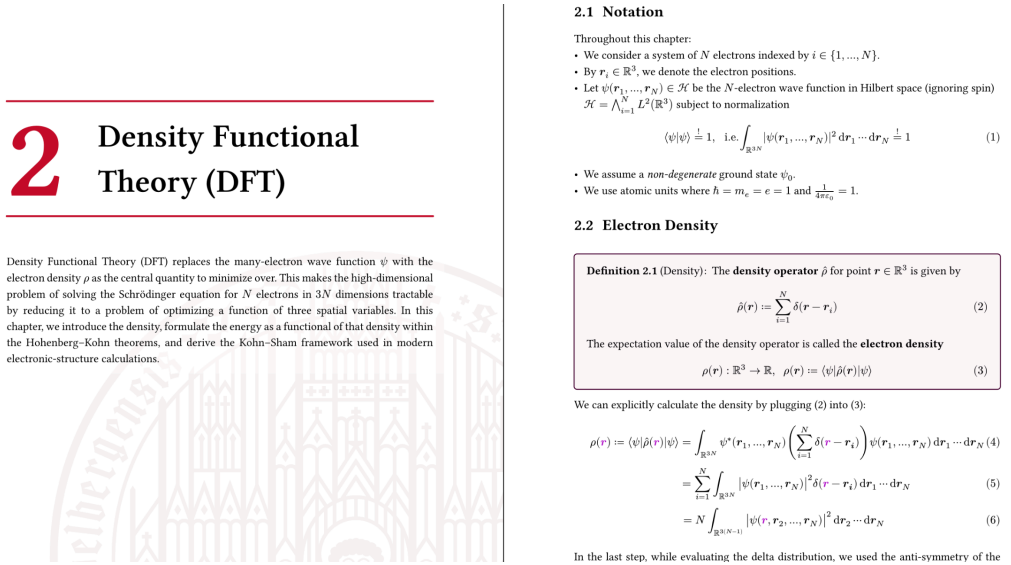
TeX Live

- 200K files
- 7GB
- Packages preinstalled

Output quality

You know LaTeX.

Typst sample from a bachelor thesis:



2 Density Functional Theory (DFT)

Density Functional Theory (DFT) replaces the many-electron wave function ψ with the electron density ρ as the central quantity to minimize over. This makes the high-dimensional problem of solving the Schrödinger equation for N electrons in $3N$ dimensions tractable by reducing it to a problem of optimizing a function of three spatial variables. In this chapter, we introduce the density, formulate the energy as a functional of that density within the Hohenberg–Kohn theorems, and derive the Kohn–Sham framework used in modern electronic-structure calculations.

2.1 Notation

Throughout this chapter:

- We consider a system of N electrons indexed by $i \in \{1, \dots, N\}$.
- By $\boldsymbol{r}_i \in \mathbb{R}^3$, we denote the electron positions.
- Let $\psi(\boldsymbol{r}_1, \dots, \boldsymbol{r}_N) \in \mathcal{H}$ be the N-electron wave function in Hilbert space (ignoring spin) $\mathcal{H} = \bigwedge_{i=1}^{N} L^2(\mathbb{R}^3)$ subject to normalization

$$\langle \psi | \psi \rangle \overset{!}{=} 1, \quad \text{i.e.} \int_{\mathbb{R}^{3N}} |\psi(\boldsymbol{r}_1, \dots, \boldsymbol{r}_N)|^2 \, \mathrm{d}\boldsymbol{r}_1 \cdots \mathrm{d}\boldsymbol{r}_N \overset{!}{=} 1 \tag{1}$$

- We assume a *non-degenerate* ground state ψ_0.
- We use atomic units where $\hbar = m_e = e = 1$ and $\frac{1}{4\pi\varepsilon_0} = 1$.

2.2 Electron Density

Definition 2.1 (Density): The **density operator** $\hat{\rho}$ for point $\boldsymbol{r} \in \mathbb{R}^3$ is given by

$$\hat{\rho}(\boldsymbol{r}) \coloneqq \sum_{i=1}^{N} \delta(\boldsymbol{r} - \boldsymbol{r}_i) \tag{2}$$

The expectation value of the density operator is called the **electron density**

$$\rho(\boldsymbol{r}) : \mathbb{R}^3 \to \mathbb{R}, \quad \rho(\boldsymbol{r}) \coloneqq \langle \psi | \hat{\rho}(\boldsymbol{r}) | \psi \rangle \tag{3}$$

We can explicitly calculate the density by plugging (2) into (3):

$$\rho(\boldsymbol{r}) \coloneqq \langle \psi | \hat{\rho}(\boldsymbol{r}) | \psi \rangle = \int_{\mathbb{R}^{3N}} \psi^*(\boldsymbol{r}_1, \dots, \boldsymbol{r}_N) \left(\sum_{i=1}^{N} \delta(\boldsymbol{r} - \boldsymbol{r}_i) \right) \psi(\boldsymbol{r}_1, \dots, \boldsymbol{r}_N) \, \mathrm{d}\boldsymbol{r}_1 \cdots \mathrm{d}\boldsymbol{r}_N \tag{4}$$

$$= \sum_{i=1}^{N} \int_{\mathbb{R}^{3N}} |\psi(\boldsymbol{r}_1, \dots, \boldsymbol{r}_N)|^2 \delta(\boldsymbol{r} - \boldsymbol{r}_i) \, \mathrm{d}\boldsymbol{r}_1 \cdots \mathrm{d}\boldsymbol{r}_N \tag{5}$$

$$= N \int_{\mathbb{R}^{3(N-1)}} |\psi(\boldsymbol{r}, \boldsymbol{r}_2, \dots, \boldsymbol{r}_N)|^2 \, \mathrm{d}\boldsymbol{r}_2 \cdots \mathrm{d}\boldsymbol{r}_N \tag{6}$$

In the last step, while evaluating the delta distribution, we used the anti-symmetry of the

Paragraph building and microtypography

Typst

- ✓ Knuth-Plass line breaking
- ✓ Ligatures
- ✓ Protrusions (basic)
- ✓ Character-level justification

LaTeX

- ✓ Knuth-Plass line breaking
- ✓ Ligatures
- ✓ Protrusions
- ✓ Font expansion
- ✓ Manual kerning adjustments (I never used it…)

Reproducibility (does it still compile?)

Typst

- Not stable (still in 0.x phase)
- Package use is versioned:

```
#import "@preview/zap:0.5.0"
```

- Easy to co-install older version

LaTeX

- Core: extremely stable

  Packages: not stable
- No versioning of packages:

```
\usepackage{circuitikz}
```

- Difficult to install older version

Case in point: compiling a very nice LaTeX presentation from 2011

1. `! LaTeX Error: File `dtklogos.sty' not found.`

   In TeX Live 2016, the package was renamed to `dtk-logos.sty`

2. `Package `texshade', Version 1.28 of 2024/01/10`

   ```
   ! LaTeX Error: Command \charge already defined.
                  Or name \end... illegal, see p.192 of the manual.
   ```

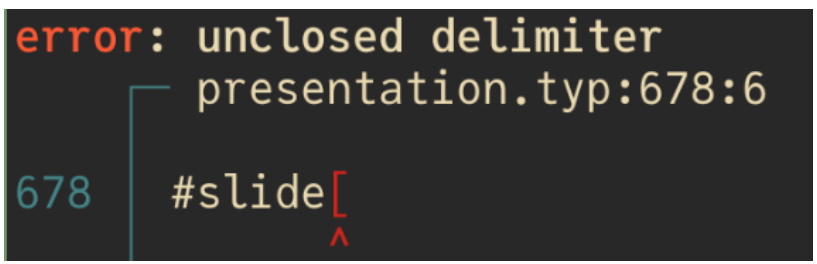
   No clue from Google, I comment out `\usepackage{texshade}` for now.

3. ```
   ! Undefined control sequence.
   <argument> \MakeIndex
   ```

   I comment out all uses of `\MakeIndex`.
   ```

4.
```
[18] (./talk.vrb
! Missing number, treated as zero.
<to be read again>
                   \kern
l.10 \documentclass[
                    a4paper,11pt]{article}
```

   Apparently a breaking change in `listings`, the fix is to replace

```
postbreak=\mbox{{\smaller\color{gray}$\hookrightarrow$}}
```

   with

```
\newsavebox\postbreakbox
\savebox\postbreakbox{\raisebox{0ex}[0ex][0ex]{\smaller\color{gray}$
\hookrightarrow$}}
...
postbreak=\usebox\postbreakbox
```

5.
```
! Undefined control sequence.
<recently read> \MikTeX
```

   Probably a breaking change in `dtk-logos`. I remove all occurrences of `\MikTeX`.

6.
```
! Undefined control sequence.
<argument> \tikz [remember picture,overlay]\node
                                                [single arrow,fill=DarkSeaG...
l.47 ...1em] at (current page.center) {pdflatex};}

?
```

   Probably a change in the interaction between TikZ overlays and Beamer uncover.
   I comment out.

7. ```
   ! Package bytefield Error: Macros \wordgroupr, \wordgroupl, \endwordgroupr,
   (bytefield)                and \endwordgroupl no longer exist.

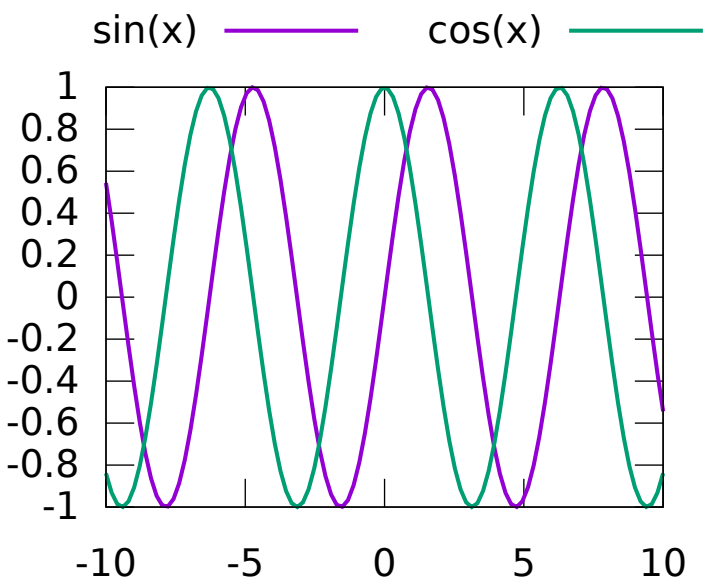
   See the bytefield package documentation for explanation.
   ```

   A breaking change in `bytefield`. Adapting the code.

8. ```
   ! Missing } inserted.
   <inserted text>
                   }
   l.189 \end{frame}

   ?
   ```

   No idea even which file is the problem… I give up.

Error messages

Typst



LaTeX

```
! Missing } inserted.
<inserted text>
                }
l.189 \end{frame}

?
```

Security

Compiling that LaTeX presentation required `-shell-escape` !

Typst

- No execution of external programs
- No reading outside project directory

LaTeX

- Can run external programs when compiling with `-shell-escape`
- Allows reading any file

LaTeX can read *any file*...

```
\documentclass{article}
\usepackage{verbatim}
\usepackage{xcolor}
\begin{document}
% Include public key as invisible text in PDF
% (would work the same with private key!)
\textcolor{white}{\verbatiminput{\string~/.ssh/id_ed25519.pub}}
\end{document}

```

Typst: no `-shell-escape` but WebAssembly plugins!

Example: the `neoplot` package **embeds Gnuplot**

````
#import "@preview/neoplot:0.0.4" as gp
#show raw.where(lang: "gnuplot"): gp.exec

```gnuplot
set termoption font "DejaVu Sans,28"
set key outside top center horizontal
plot sin(x) lw 3, cos(x) lw 3
```
````



- Fairly secure
- Zero installation required
- Works on Windows, macOS, Linux

Syntax

```
#slide[Installation][
  #set text(1.1em)
  #grid(
    columns: (50%, 50%),
    column-gutter: 1em,
    [
      *Typst*
      - 1 file
      - 50MB
    ],
    ...
  )
]
```

```
\begin{frame}{Installation}
\fontsize{11pt}{13pt}\selectfont
\begin{columns}
    \begin{column}{0.48\textwidth}
        \textbf{Typst}
        \begin{itemize}
            \item 1 file
            \item 50MB
        \end{itemize}
    \end{column}
    ...
\end{columns}
\end{frame}
```

Typst

```
$ QQ = { p/q : p in ZZ, q in NN } $
```

$$\mathbb{Q} = \left\{ \frac{p}{q} : p \in \mathbb{Z}, q \in \mathbb{N} \right\}$$

LaTeX

```
\documentclass{article}
\usepackage{amssymb}
\begin{document}
\[ \mathbb{Q} = \left\{ \frac{p}{q} : p \in \mathbb{Z}, q \in \mathbb{N} \right\} \]
\end{document}
```

$$\mathbb{Q} = \left\{ \frac{p}{q} : p \in \mathbb{Z}, q \in \mathbb{N} \right\}$$

Performance

Typst

- Incremental compiler

LaTeX

- Multipass compiler

With my linear algebra lecture notes (140 pages):

Typst

- 6s
- 1 output file
- on change: **instant preview**

LaTeX

- 17s
- 21 output files
- on change: full recompile (6s if lucky)

Programming

Let's print all prime numbers under 50, comma-separated:

$$2,\ 3,\ 5,\ 7,\ 11,\ 13,\ 17,\ 19,\ 23,\ 29,\ 31,\ 37,\ 41,\ 43,\ 47$$

Compare implementations for the **sieve of Eratosthenes** in

1. Plain LaTeX
2. LaTeX3
3. LuaLaTeX
4. Typst

Plain LaTeX

```
\documentclass{article}
\makeatletter
\newcommand{\@crossout}[3]{%
  \ifnum#1>#3\relax
  \else
    \expandafter\gdef\csname
        isprime@\number#1\endcsname{F}%
    \expandafter\@crossout\expandafter
        {\number\numexpr#1+#2\relax}{#2}{#3}%
  \fi
}
\newcommand{\sieve}[1]{%
  \count1=2
  \loop
    \expandafter\gdef\csname
        isprime@\the\count1\endcsname{T}%
    \ifnum\count1<#1 \advance\count1 by 1 \repeat
  \count1=2
  \loop
    \expandafter\if T\csname
        isprime@\the\count1\endcsname

\edef\tempstart{\number\numexpr\count1*\count1\relax}%
      \edef\tempstep{\the\count1}%
      \expandafter\@crossout\expandafter
          {\tempstart}{\tempstep}{#1}%
    \fi
    \ifnum\numexpr\count1*\count1\relax<#1
      \advance\count1 by 1
  \repeat
  \count1=2
  \def\prime@sep{}%
  \loop
    \expandafter\if T\csname
        isprime@\the\count1\endcsname
      \prime@sep\the\count1 %
      \gdef\prime@sep{, }%
    \fi
    \ifnum\count1<#1 \advance\count1 by 1 \repeat
}
\makeatother
\begin{document}
\sieve{50}
\end{document}
```

LaTeX3

```
\documentclass{article}
\ExplSyntaxOn
\intarray_new:Nn \isprime {10001}
\seq_new:N \l_sieve_results_seq
\NewDocumentCommand{\sieve}{m}{
  \seq_clear:N \l_sieve_results_seq
  \int_step_inline:nnnn{2}{1}{#1}{
    \intarray_gset:Nnn\isprime{##1}{1}
  }
  \int_step_inline:nnnn{2}{1}{
      \fp_to_int:n{floor(sqrt(#1))}}{
    \int_compare:nNnT{
        \intarray_item:Nn\isprime{##1}}={1}{
      \int_step_variable:nnnNn{##1*##1}{##1}{#1}
        \l_tmp_int{
        \intarray_gset:Nnn\isprime{\l_tmp_int}{0}
      }
    }
  }
  \int_step_inline:nnnn{2}{1}{#1}{
    \int_compare:nNnT{
        \intarray_item:Nn\isprime{##1}}={1}{
      \seq_put_right:Nn\l_sieve_results_seq{##1}
    }
  }
  \seq_use:Nn\l_sieve_results_seq{,~}
}
\ExplSyntaxOff

\begin{document}
\sieve{50}
\end{document}
```

LuaLateX

```
\documentclass{article}
\usepackage{luacode}
\begin{luacode*}
  function sieve_lua(n)
    local is_prime = {}
    for i = 2, n do is_prime[i] = true end
    for i = 2, math.sqrt(n) do
      if is_prime[i] then
        for j = i * i, n, i do
          is_prime[j] = false
        end
      end
    end
    local results = {}
    for i = 2, n do
      if is_prime[i] then
        table.insert(results, i)
      end
    end
    tex.print(table.concat(results, ", "))
  end
\end{luacode*}
```

```
\newcommand{\sieve}[1]{\directlua{sieve_lua(#1)}}

\begin{document}
\sieve{50}
\end{document}
```

Typst

```
#let sieve(n) = {
  let is-prime = (false, false) + (true,)*(n - 1)
  for i in range(2, int(calc.sqrt(n)) + 1) {
    if is-prime.at(i) {
      for j in range(i * i, n + 1, step: i) {
        is-prime.at(j) = false
      }
    }
  }
  is-prime.enumerate()
    .filter(it => it.at(1))
    .map(it => str(it.at(0)))
    .join(", ")
}

#sieve(50)
```

Another Typst example: building a point table on the exam's first page

```
#context {
  let n = query(heading).len()
  table(
    inset: 1em,
    align: center,
    columns: (auto,) + (1.5cm,) * n + (auto,),
    [Exercise], ..range(n).map(i => str(i + 1)), [Total],
    [Points],
  ),
}
= Ex 1
= Ex 2
= Ex 3
```

Exercise	1	2	3	Total
Points				

Callisto

a package for literate programming

Render notebooks with style

```
#import "@preview/callisto:0.3.0"

#set text(font: "DejaVu Sans")
#set heading(numbering: "I.")

#callisto.render(
  nb: path("notebook.ipynb"),
  (0, 1), // first two cells
)
```

I. Example notebook

This is a Markdown cell. It can include math like $x^2 = 1$ or

$$\int_a^b f(x)\,dx = \lim_{n\to\infty} \frac{b-a}{n} \sum_{i=1}^{n} f(x_i^*).$$

In [1]:
```
# This is a code cell
2 + 5
```

Out[1]:
```
7
```

Include notebooks as chapters/sections

```
#import "@preview/callisto:0.3.0"
#set heading(numbering: "1.")

= Introduction
Some text.

#callisto.render(
  nb: path("notebook.ipynb"),
  theme: "neat",
  cmarker: (h1-level: 2), // subsection
  (0, 1), // first two cells
)
```

1. Introduction

Some text.

1.1. Example notebook

This is a Markdown cell. It can include math like $x^2 = 1$ or

$$\int_a^b f(x)\,dx = \lim_{n\to\infty} \frac{b-a}{n} \sum_{i=1}^{n} f(x_i^*).$$

```
# This is a code cell
2 + 5
```

```
7
```

Extract cell code and outputs

```
#import "@preview/callisto:0.3.0"
#let (source, output) = callisto.config(
  nb: path("notebook.ipynb"),
)

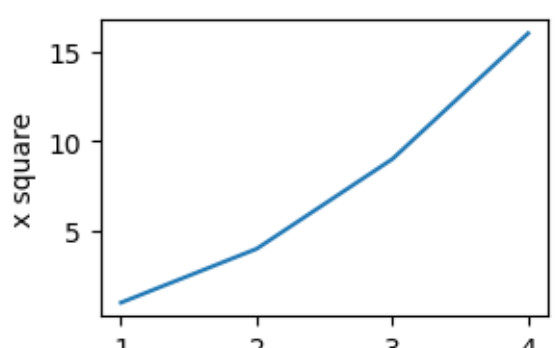
The source of the "plot1" cell:
#source("plot1", keep-cell-header: true)

And its output:
#output("plot1")
```

The source of the “plot1” cell:

```
#| label: plot1
import matplotlib
import matplotlib.pyplot as plt
matplotlib.rcParams['figure.figsize'] = (3, 2)
plt.plot([1, 2, 3, 4], [1, 4, 9, 16])
plt.ylabel('x square')
plt.show()
```

And its output:



Execute code written in Typst source

```typst
#import "@preview/callisto:0.3.0"
#let (execute, stage-notebook) = callisto.config(
  nb: path("export.ipynb"),
  kernel: "python3",
)
#show raw.where(lang: "py-x"): it => {
  set text(1em/0.8)
  execute(it)
}
#stage-notebook()

Let's expand $(a+b)^2$ :

```py-x
import sympy as sp
a, b = sp.symbols('a b')
sp.expand((a+b)**2)
```
```

Let's expand $(a+b)^2$:

In [1]:
```
import sympy as sp
a, b = sp.symbols('a b')
sp.expand((a+b)**2)
```

Out[1]: $a^2 + 2ab + b^2$

This last example requires running commands for **export** and **execution**.

Recommendation: use a Makefile/justfile:

```
default: export execute

export:
    typst eval --input callisto-export=true --in presentation.typ \
      'query(<notebook>).first().value' > export.ipynb

execute:
    jupyter-nbconvert --to notebook --execute --inplace export.ipynb

watch:
    watchexec -w . -f '**/*.typ' just export execute
```

Questions?

Colophon

Slides made in plain Typst (no package needed). Find the source at

github.com/sijow/typst-presentation/

Design and some content adapted from

github.com/ZaninDavide/typst-intro/