typst

Compose papers faster.

Martin Haug ▪ Laurenz Mädje

July 20, **2023**

Typst helps students and scientists compose **technical documents faster**

Louis Vignoli, PhD, Industrial quantum scientist

„I am over the moon with Typst and converted as many of my coworkers as possible.“

Christoph

„This is the most pleasant typesetting software that I have ever used by a million miles.“

Prof. Christopher Métrailler, HES-SO Valais

„Typst is a perfect tool for me and my use cases“

Mordrag

„I love how simple Typst seems, while retaining the features one would expect.“

Donald Knuth invented TEX in **1978** to typeset the second volume of his book on „The Art of Computer Programming“

Macro-based typesetting system

Computer Modern: Font family

DVI: Document output format

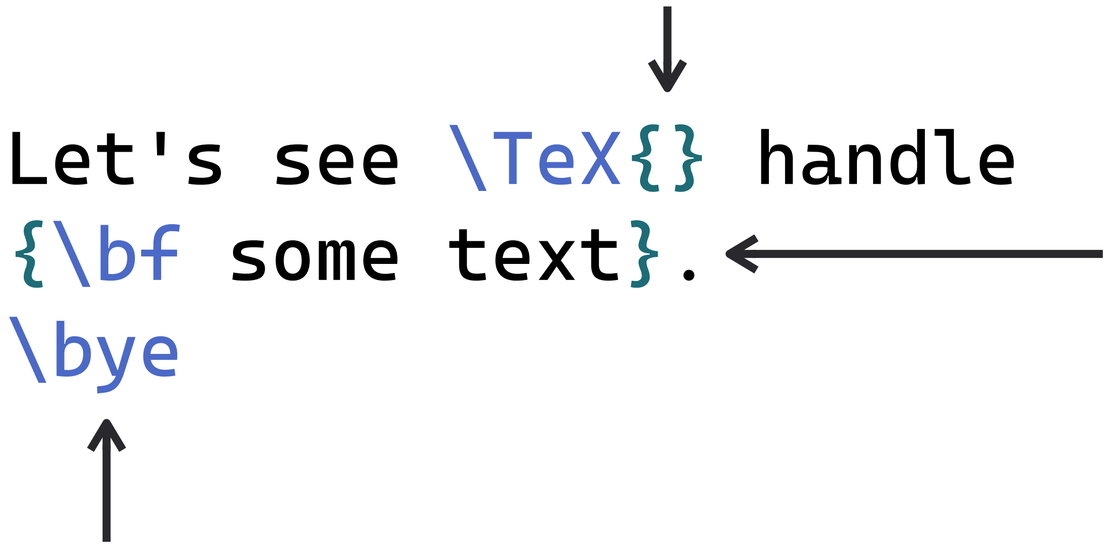
Line-breaking algorithm

METAFONT: Font Technology

WEB: Programming language

TECHNOLOGIES DEVELOPED FOR TEX

Empty group to separate macro and space.

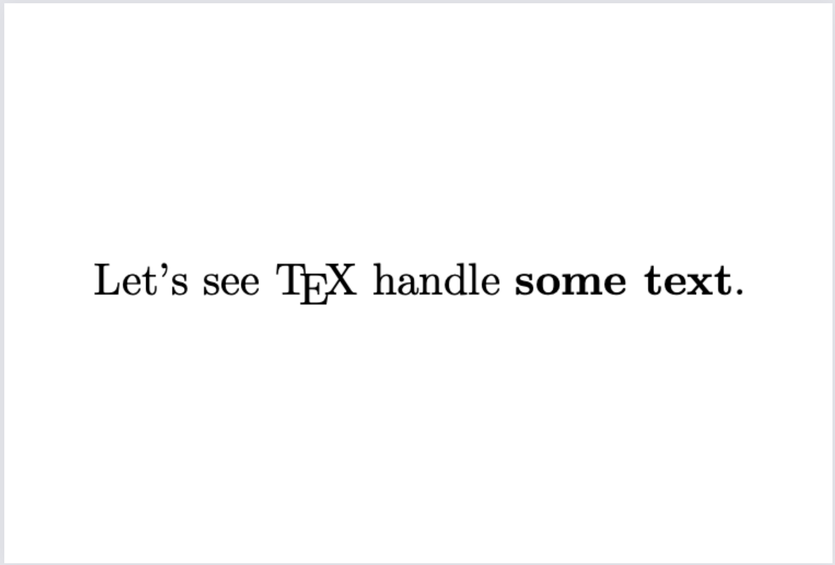


```
Let's see \TeX{} handle
{\bf some text}.
\bye
```

Group with curly braces scopes the \bf-macro

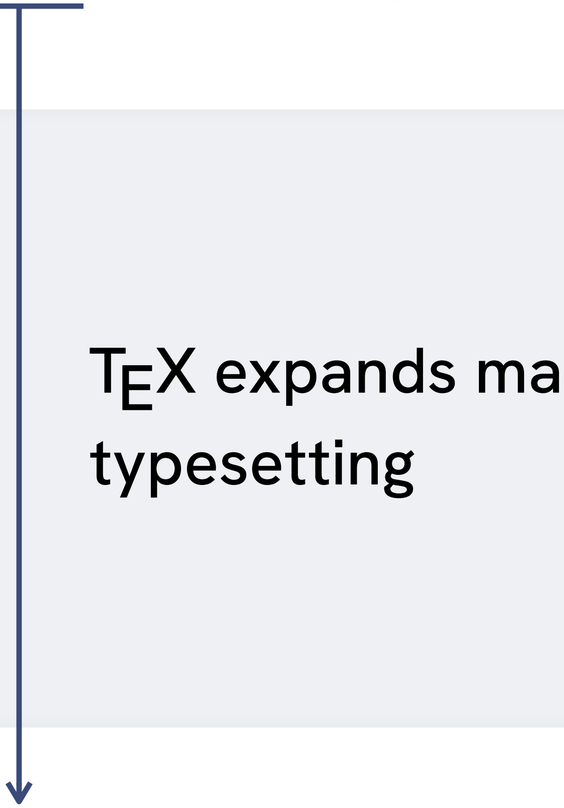
This macro outputs the final page.

```
Let's see \TeX{} handle
{\bf some text}.
\bye
```



```
{\bf writing something}
```

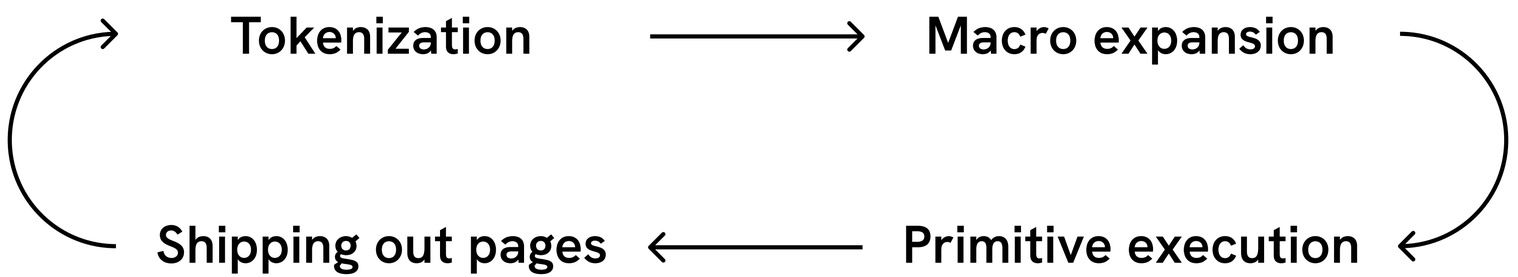
Macro from plain.tex



TEX expands macros during typesetting

```
{\fam\bffam\tenbf writing …}
```

\radical \moveleft \message \lccode \lastbox \pagegoal
athaccent \noalign \omit \mark \leftskip \kern
\multiply \maxdeadcycles \mathop \jobname \meani
\medmuskip \everycr \futurelet \fontdimen \delcode \immediate \clead
ness \leqno \displaywidowpenalty \everyjob \above \errhelp \globa
ders
uage \abovedisplayshortskip \begingroup \crcr \floatingpena
\batchmode \inputlineno \csname \dump \endlinechar
gafter \everypar \holdinginsert
tcode \hfuzz \ifeof \ifcase \ifx \badness \aftergroup
\hfilneg \belowdisplayskip \chardef \firstmark \exhyphe
enout \divide \copy \adjdemerits
r \baselineskip \botmark \hss \prevgraf \expandafter
\abovedisplayshortskip \countdef \deadcycles \dump



TEX reads the input file from top to bottom

`*.aux`

Peeking ahead with helper files and repeated compilation

`\catcode`

Dynamic redefinition of the syntax possible

TEX is very low-level and operates with formatting-driven primitives

„For a document to be easy to read, its visual structure must reflect its logical structure.“

— Leslie Lamport, Inventor of LATEX, 1985

LATEX is a macro collection for TEX that puts structure first

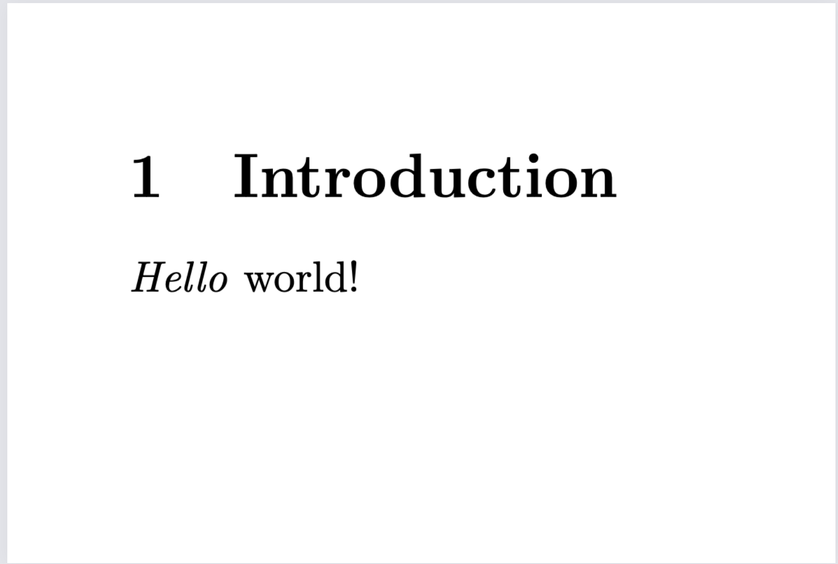
```
\documentclass{article}

\begin{document}

\section{Introduction}
\emph{Hello} world!

\end{document}
```

`\documentclass` determines the kind of document we're writing

Emphasis-macro instead of italic font

LaTeX's custom start and end macros

```latex
\documentclass{article}

\begin{document}

\section{Introduction}
\emph{Hello} world!


\end{document}
```



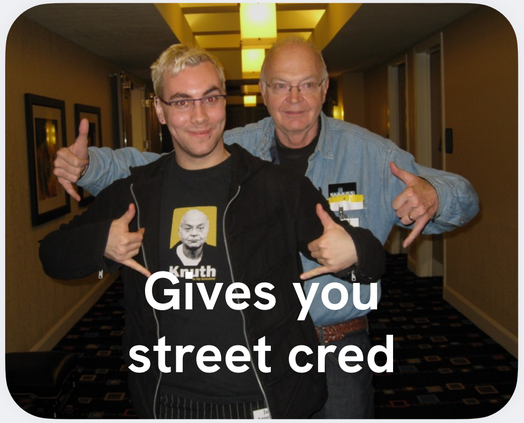
Benefits of LaTeX

Wide acceptance among publishers

Focus on structure instead formatting makes reformatting easy

6,380

LaTeX-packages available on CTAN



Gives you street cred

Runs on IBM 4300 mainframe

245,905

Questions about (LaTeX on tex.stackexchange.com

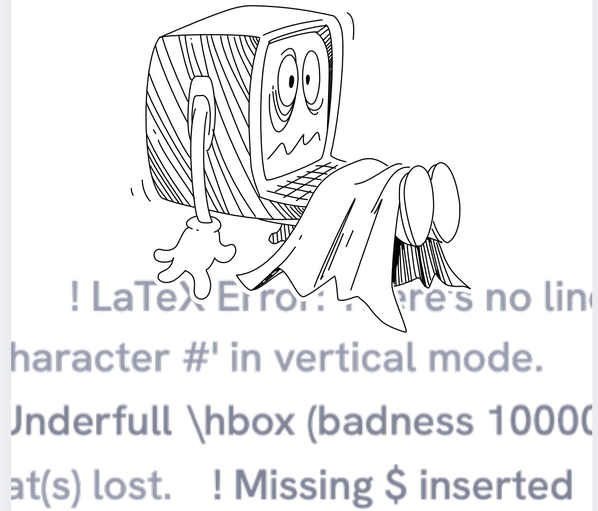
PDF/Print only

Changing the formatting requires expert knowledge and is complex

`graphicx`
`tabularx`
`babel`
`xcolor`
`amsmath`

Many basic tasks require using packages

Downsides of LaTeX



Unhelpful error messages

typst


TU Berlin: Idea
2019
TECHNISCHE UNIVERSITÄT


Development
2020...


Berliner Startup Stipendium
Jan-Jun 2023

Demo

Typst File Edit View Text Structure Layout Code Help — Johanna's Typst › Space Mail

Latin Modern Roman | B I | H | 100% | Share

```
= Introduction
Our concept suggests three ways that A-Mail can be best utilized.

- First is to reduce the probability of the failure of a space mission. This
  problem is known as the Mars problem and suggest problems with human
  communication.

- High round-trip times required for communication between Mars and Earth inhibits
  successful human developments on the planet. In contrast, the delivery speed of
  an A-Mail can be determined through this simple formula:
  $[ v(t) = lim(t -> inf) integrate(t..inf) c * sqrt(t^2) dt ]$

#figure(
  image("a-mail.svg"),
  caption: [
    Visualization of the FTL Earth-Mars communication
    capabilities enabled by Typst.
  ],
)

The foundations of A-Mail promise exciting new ways to predict problems and apply
existing and new best practices to ensure the mail is delivered without any
issues. We call this extension AI-Mail. AI-Mail is a new concept designed and
```

Egon Johanna Nicole

typst

Towards Faster Interstellar Mail Delivery

Johanna Swift Delivery Institute
Egon Stellaris Space Institute
Oliver Liam Mail Institute

May 17th, 2022

Until there is a definitive answer to the mystery of the dead star, please use the old postal system to submit your question and report the location of missing letters to the P.I.

ABSTRACT

Recent advances in space-based document processing have enabled faster mail delivery between different planets of a solar system. Given the time it takes for a message to be transmitted from one planet to the next, its estimated that even a one-way trip to a distant destination could take up to one year. During these periods of interplanetary mail delivery there is a slight possibility of mail being lost in transit. This issue is considered so serious that space management employs P.I. agents to track down and retrieve lost mail. We propose A-Mail, a new anti-matter based approach that can ensure that mail loss occurring during interplanetary transit is unobservable and therefore potentially undetectable. Going even further, we extend A-Mail to predict problems and apply existing and new best practices to ensure the mail is delivered without any issues. We call this extension AI-Mail. Integrating quantum computing could open up even more possible applications, ranging from inter-galactic delivery to mind reading.

Reference: Johanna Swift, Egon Stellaris, Oliver Liam. *Towards Faster Interstellar Mail Delivery*. https://doi.org/10.7891/1209348510

– High round-trip times required for communication between Mars and Earth inhibits successful human developments on the planet. In contrast, the delivery speed of an A-Mail can be determined through this simple formula:

$$v(t) = \lim_{t \to \infty} \int_t^{\infty} c \cdot \sqrt{t^2}\, dt$$

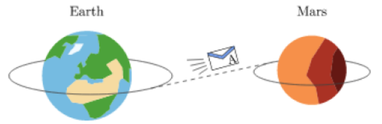
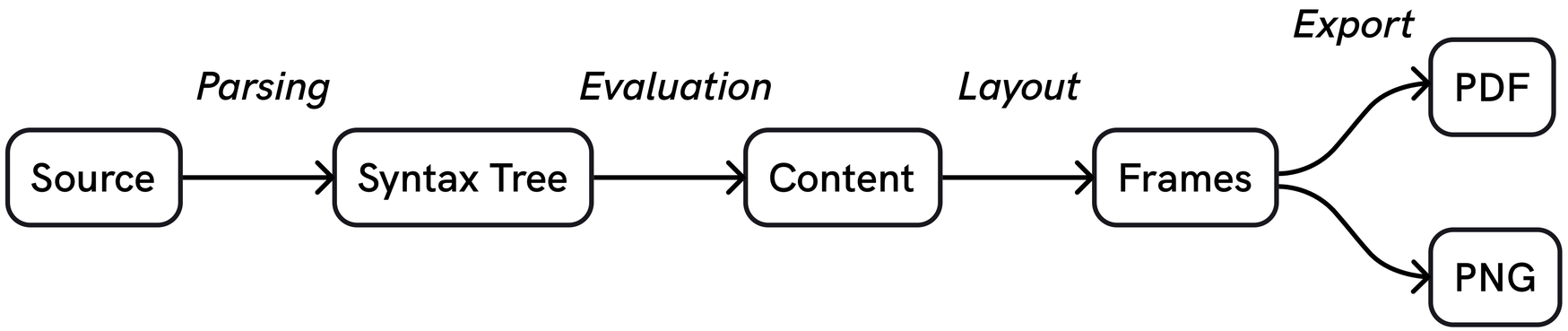


Figure 1: Visualization of the FTL Earth-to-Mars communication capabilities enabled by Typst.

The foundations of A-Mail promise exciting new ways to predict problems and apply existing and new best practices to ensure the mail is delivered

Compiler architecture



Introspection

Query system

Incrementality

Fast recompilation

Recent Developments Within Glaciers

July 20, 2023

Laurenz
laurenz.maedje@typst.app

Martin
martin.haug@typst.app

Outline

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.

1. Introduction

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

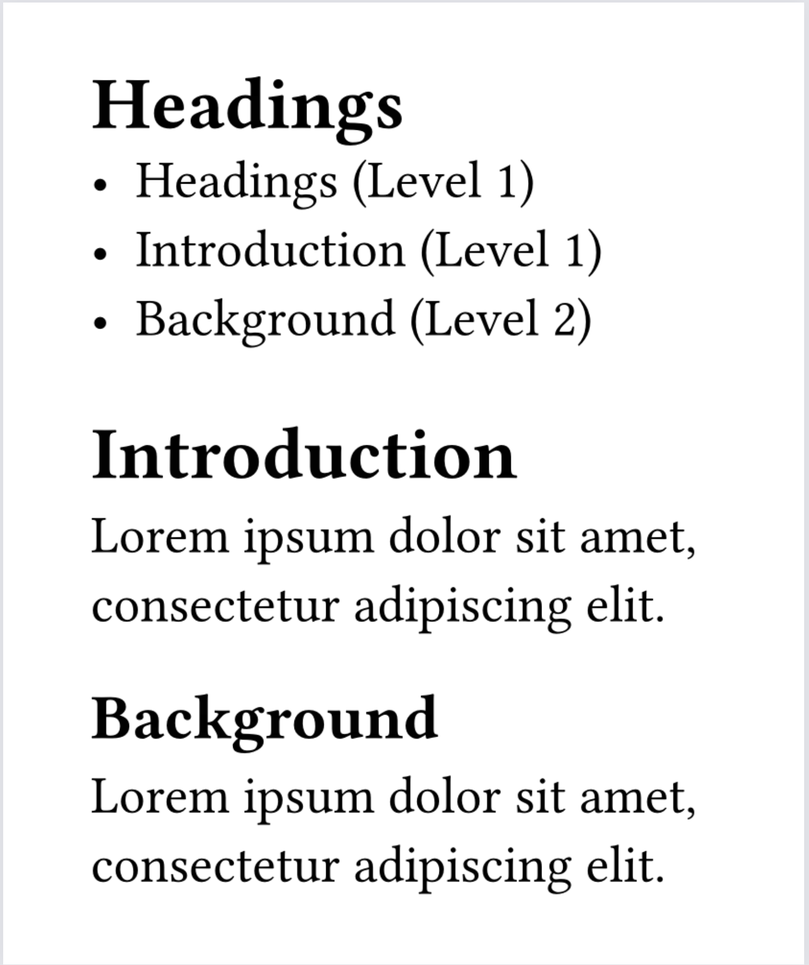
Figure 1: A glacier which might not exist for much longer.

2. Related Work

In Figure 1, we can clearly observe: 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

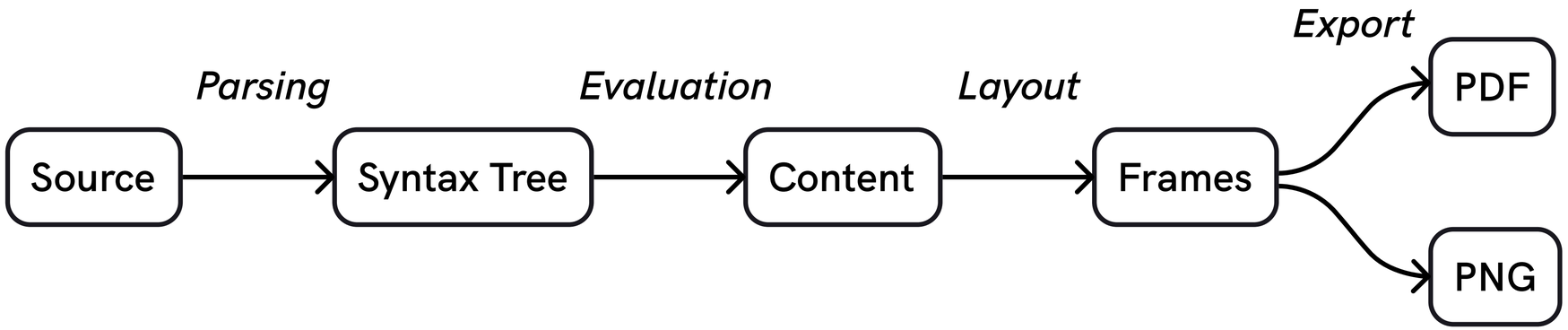
Introspection

```
= Headings
#locate(loc => {
  let headings = query(heading, loc)
  for elem in headings [
    - #elem.body (Level #elem.level)
  ]
})

= Introduction
#lorem(8)

== Background
#lorem(8)
```



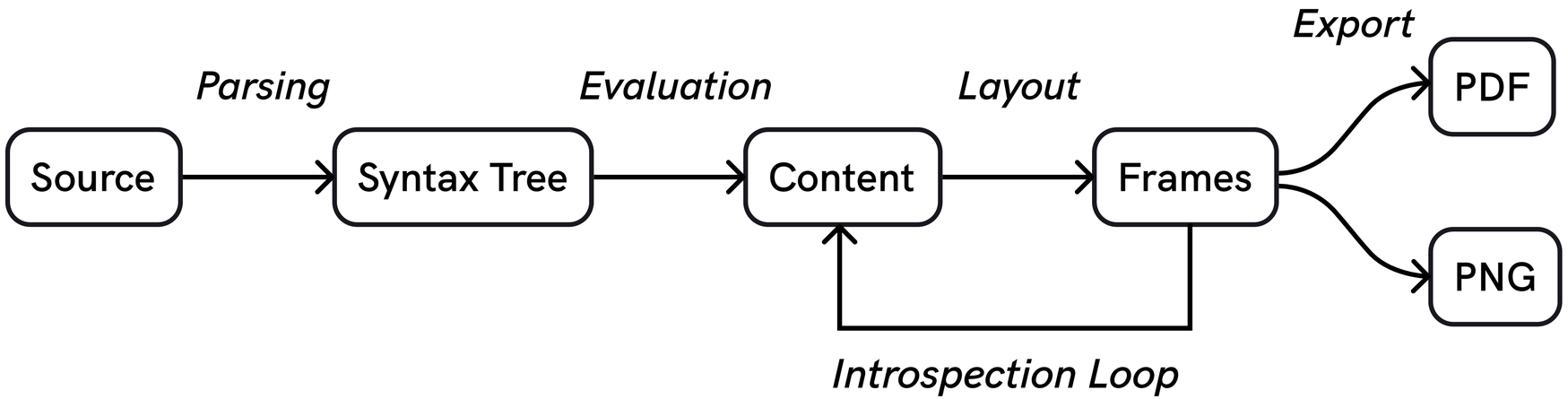
Introspection

Compiler architecture



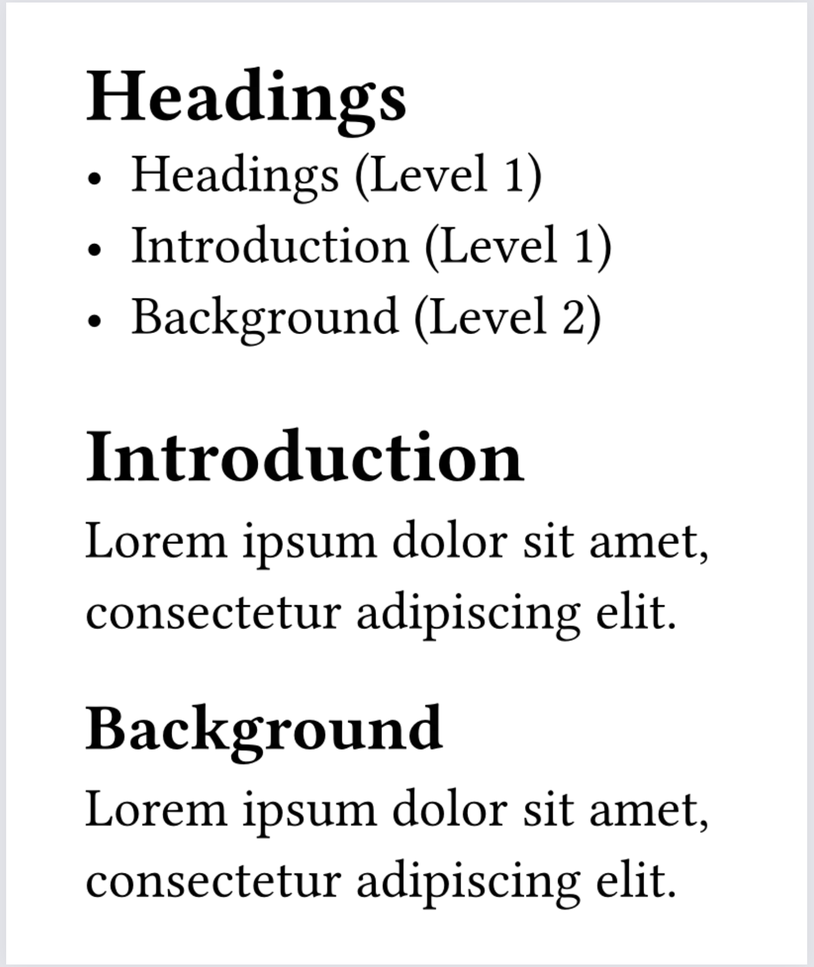
Introspection

Compiler architecture



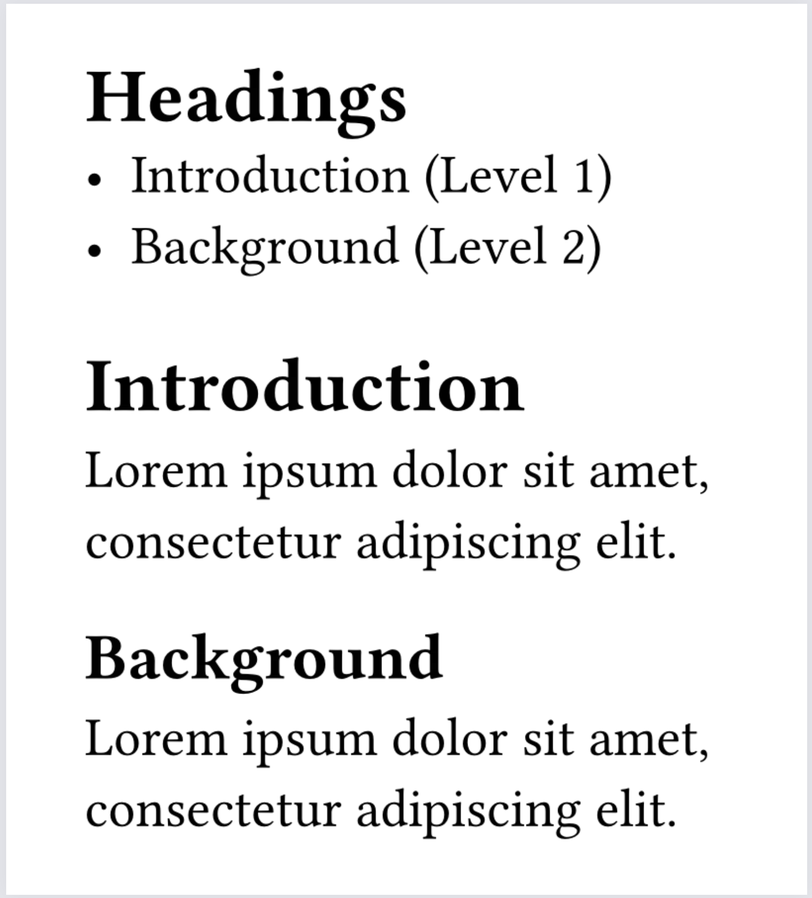
Introspection

```
= Headings
#locate(loc => {
  let headings = query(heading, loc)
  for elem in headings [
    - #elem.body (Level #elem.level)
  ]
})

= Introduction
#lorem(8)

== Background
#lorem(8)
```



Introspection

```
= Headings
#locate(loc => {
  let headings = query(
    selector(heading).after(loc),
    loc,
  )
  for elem in headings [
    - #elem.body (Level #elem.level)
  ]
})

= Introduction
#lorem(8)

== Background
#lorem(8)
```



Headings

- Introduction (Level 1)
- Background (Level 2)

Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Background

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Introspection

```
#locate(loc => {
  let headings = query(heading, loc)
  [= Heading] * (headings.len() + 1)
})
```


Heading

Heading

Heading

Heading

Heading


1

1

Incrementality

alpha.calc

```
1 + 2 + beta.calc
```

beta.calc

```
gamma.calc + 4
```

gamma.calc

```
8 + 1
```

calc.rs

```rust
fn evaluate(script: &str, files: &Files) -> i32 {
    script
        .split('+')
        .map(str::trim)
        .map(|item| match item.parse::<i32>() {
            Ok(num) => num,
            Err(_) => evaluate(&files.get(item), files),
        })
        .sum()
}
```

Incrementality

alpha.calc

```
1 + 2 + beta.calc
```

beta.calc

```
gamma.calc + 4
```

gamma.calc

```
8 + 1
```

calc.rs

```rust
#[comemo::memoize]
fn evaluate(script: &str, files: Tracked<Files>) -> i32 {
    script
        .split('+')
        .map(str::trim)
        .map(|item| match item.parse::<i32>() {
            Ok(num) => num,
            Err(_) => evaluate(&files.get(item), files),
        })
        .sum()
}
```

Incrementality

alpha.calc

```
1 + 2 + beta.calc
```

beta.calc

```
gamma.calc + 4
```

gamma.calc

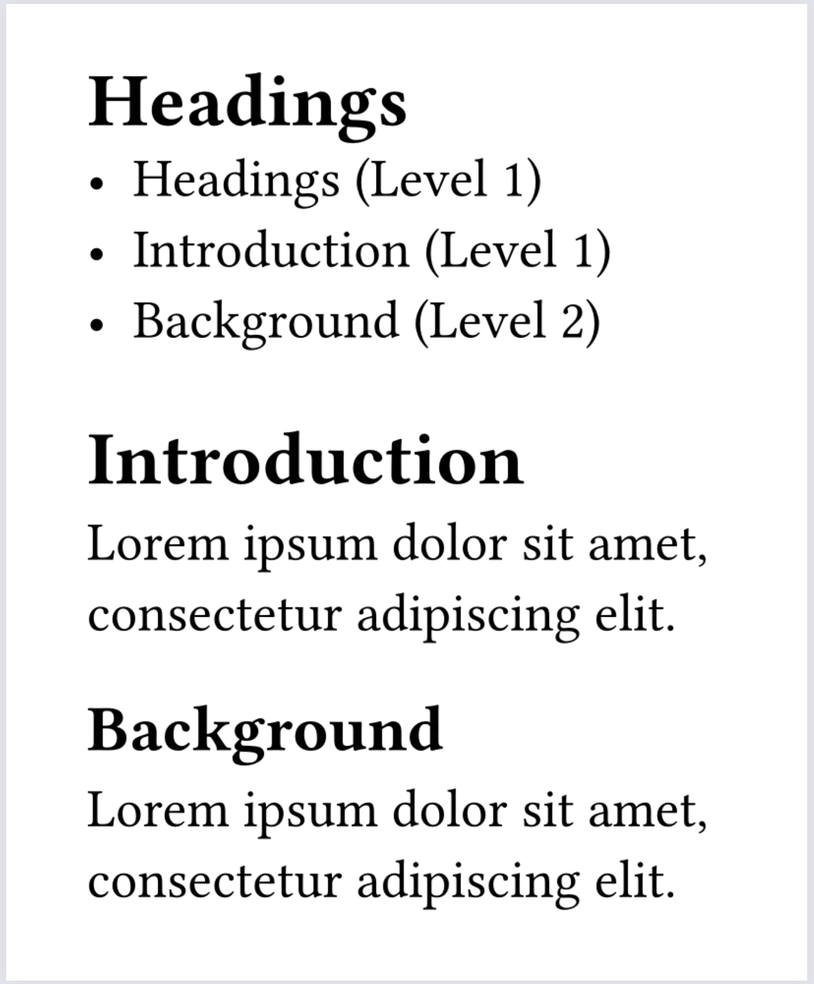
```
8 + 1
```

calc.rs

```rust
#[comemo::memoize]
fn evaluate(script: &str, files: Tracked<Files>) -> i32 {
    script
        .split('+')
        .map(str::trim)
        .map(|item| match item.parse::<i32>() {
            Ok(num) => num,
            Err(_) => evaluate(&files.get(item), files),
        })
        .sum()
}

#[comemo::track]
impl Files {
    fn get(&self, path: &str) -> String {
        /* load and cache file */
    }
}
```

Incrementality

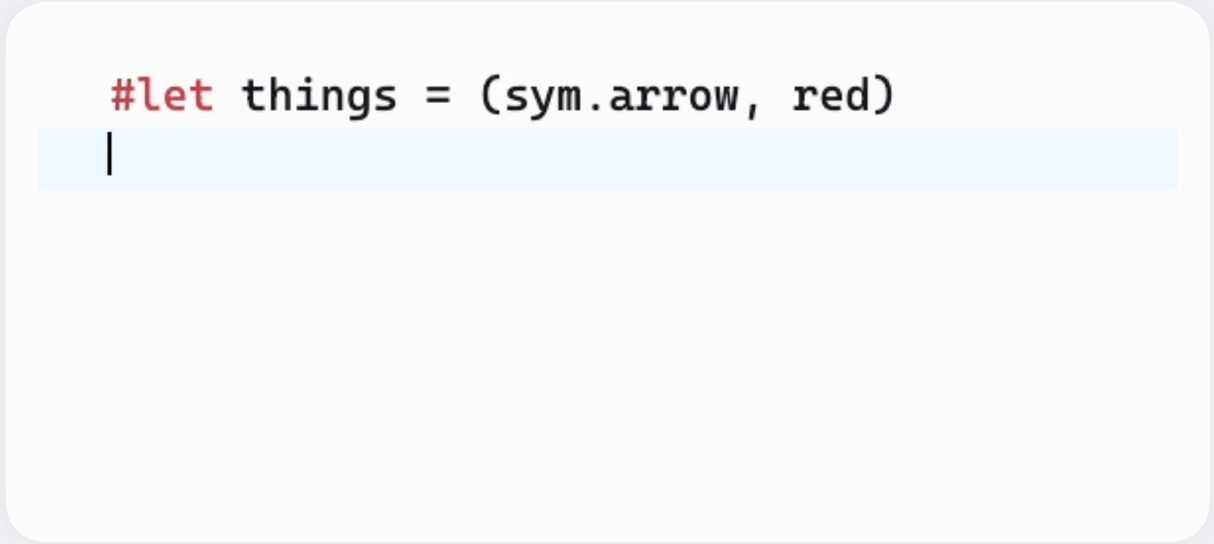
```
= Headings
#locate(loc => {
  let headings = query(heading, loc)
  for elem in headings [
    - #elem.body (Level #elem.level)
  ]
})

= Introduction
#lorem(8)

== Background
#lorem(8)
```



Incrementality

eval/mod.rs

```rust
#[comemo::memoize]
fn eval(
    world: Tracked<dyn World + '_>,
    route: Tracked<Route>,
    tracer: TrackedMut<Tracer>,
    source: &Source,
) -> SourceResult<Module> {
    ...
}
```



Autocomplete

```
#let things = (sym.arrow, red)
```

Autocomplete

Hover tooltips

```
#for val in range(15, step: 4) {
  str(val) + " and "
}
```

```
#let things = (sym.arrow, red)
#things
```

1. Mark relevant expressions
2. Fully compile document and save values of relevant expressions
3. Autocompletion based on observed values

Relevant expressions

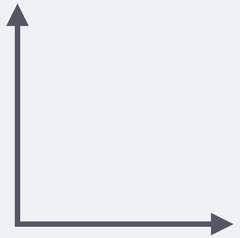
```
#for val in range(15, step: 4) {
  str(val) + " and "
}
```

Web app

In Browser

UI
TypeScript
React
Sass

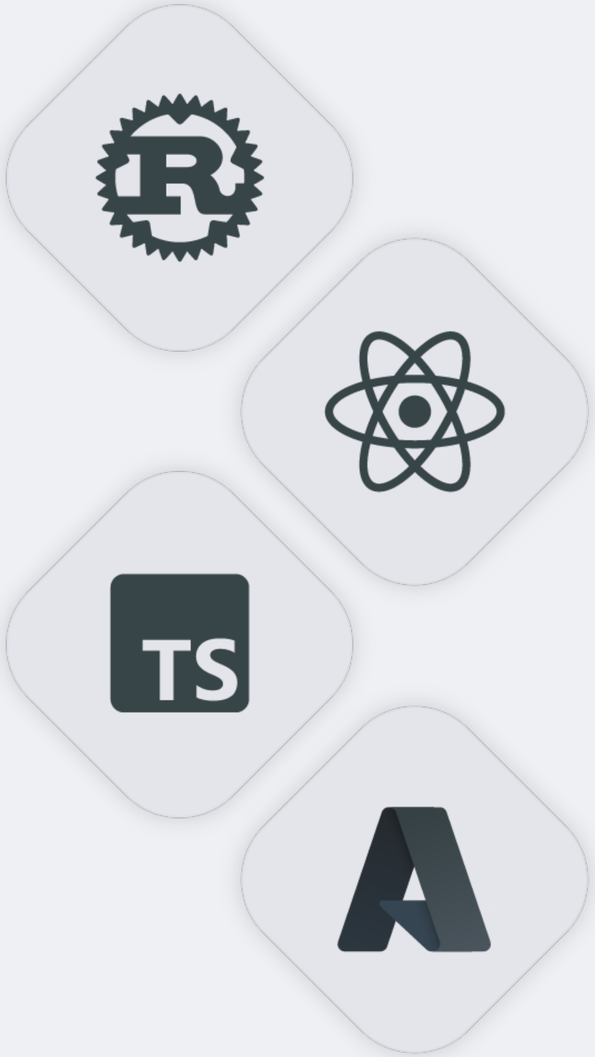
Compiler
WebAssembly
Web Worker



API
TypeScript
Serverless
PostgreSQL

Collaboration
TypeScript
Docker

Microsoft Azure



Local compiler

```
$ typst watch hello.typ
watching hello.typ
writing to hello.pdf

[18:35:33] compiled successfully
```

```
$ typst watch hello.typ
watching hello.typ
writing to hello.pdf

[18:35:33] compiled with errors

error: expected length, found color
   ┌─ /hello.typ:1:16
   │
1  │ #set text(size: red)
   │                 ^^^
```