Typst

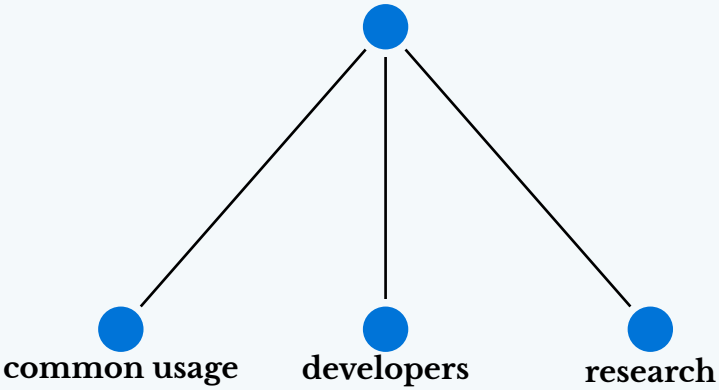
Typesetting made simple

Anush M

OUTLINE

Typesetting

Users for typesetting



Typst

similarities with markdown

Markdown	Typst
`# header1`	`= header1`
`## header2`	`== header2`
`### header3`	`=== header3`
`#### header4`	`==== header4`

similarities with markdown (ii)

Markdown

```
- bullet1
- bullet2
- bullet3
```

```
_emphasise things_
```

```
**bold text**
```

Typst

```
- bullet1
- bullet2
- bullet3
```

```
_emphasise things_
```

```
*bold text*
```

similarities with markdown (iii)

Markdown

````
```python
print("hello")
```
````

```
---
```

Typst

````
```python
print("hello")
```
````

```
---
```

Differences to markdown

Generic differences

Markdown

```
[link name](https://example.com)
```

Typst

```
#link("https://example.com")[link name]
```

```
~strike through~
```

```
#strike("strike through")
```

Typst Syntax

basic logic

Code

```
#let x = 5
#x
```

Output

5

```
#let x = 5
#if x > 3 { "big" } else
{ "small" }
```

big

```
#let arr = (1, 2, 3);
#arr
```

```
(1, 2, 3)
```

basic logic (ii)

Code

```
#let dict = (name:
"Typst", year: 2022)
#dict
```

Output

```
(name: "Typst", year: 2022)
```

```
#let dict = (name:
"Typst", year: 2022)
#dict.keys()
#dict.values()
```

```
("name", "year")
```

```
("Typst", 2022)
```

basic logic (iii)

Code

```
#let add = (a, b) => a + b
#add(2, 3)
```

Output

5

```
#let arr = (1, 2, 3)
#arr.map(x => x * 2)
```

```
(2, 4, 6)
```

```
#(1, 2, 3).map(x =>
str(x)).join(" ")
```

1 2 3

Solving the N-Queens problem in typst

```
#let n-queens(n) = {
  let is-safe(board, row, col) = {
    for r in range(0,row) {
      let c = board.at(r)
      if c == col or calc.abs(r - row) == calc.abs(c - col) {
        return false
      }
    }
    return true
  }

  let solve(board, row) = {
    if row >= n {
      return (board,)
    }
    let solutions = ()
    for col in range(0,n) {
      if is-safe(board, row, col) {
```

Solving the N-Queens problem in typst

```
        solutions += solve(board + (col,), row + 1)
      }
    }
    return solutions
  }

  solve((), 0)
}


#let n = 5

#let solutions = n-queens(n)

#align(center)[
  Found #solutions.len() solutions for a #n x #n board.
]
```

Solving the N-Queens problem in typst

```
[#solutions]
```

Found 10 solutions for a 5 x 5 board.

```
(
(0, 2, 4, 1, 3),
(0, 3, 1, 4, 2),
(1, 3, 0, 2, 4),
(1, 4, 2, 0, 3),
(2, 0, 3, 1, 4),
(2, 4, 1, 3, 0),
(3, 0, 2, 4, 1),
(3, 1, 4, 2, 0),
(4, 1, 3, 0, 2),
(4, 2, 0, 3, 1),
)
```

Math and emojis

Math symbols in typst

Typst

```
$ (a + b)^2
= a^2
+ text(fill: #maroon, 2 a
b)
+ b^2 $
```

Output

$$(a + b)^2 = a^2 + 2ab + b^2$$

Math symbols in typst (ii)

Typst

```
$ f(x, y) :=
 cases(1 "if" (x dot y)/2
<= 0,
 2 "if" x "is even",
 3 "if" x in NN,
 4 "else",) $
```

Output

$$f(x,y) := \begin{cases} 1 \text{ if } \frac{x \cdot y}{2} \leq 0 \\ 2 \text{ if } x \text{ is even} \\ 3 \text{ if } x \in \mathbb{N} \\ 4 \text{ else} \end{cases}$$

Emojis

u

Typst

```
$ sum^10_(🥸=1)
#rect(width: 4mm,
height: 2mm)/🥸
= 🧠 maltese $
```

Output

Why not LaTeX

some advantages

- Typst has more dedicated syntax for complex types
- Typst compiles faster
- Lower learning curve

some disadvantages

- Plotting systems
- Including PDFs as images
- Changing page margins without a pagebreak

Fun with Typst

Miscellaneous

References

You can get the template for this presentation here

Typst Documentation

Overleaf documentation

Typst Universe