□□□□□□□ □□□□□□ □□□□□□□□

□□□

Typst □□□□□□□□□

12-345678 □□ □□

2024 □ 2 □ 1 □ □□

□□

□□□□□□□□□□□□□□. □□□□□□□□□□□□□□. □□□□□□□□□□□□□□. □□□□□□□□□□□□□□. □□□□□□□□□□□□□□. □□□□□□□□□□□□□□. □□□□□□□□□□□□□□. □□□□□□□□□□□□□□. □□□□□□□□□□□□□□. □□□□□□□□□□□□□□.

□□□□□□: □□, □□□□

□□

□□ ..... p. i

□ **1** □ □□ ..... p. 1

1.1 Typst □□□□ ..... p. 1

1.1.1 □□□□□□□□□□ ..... p. 1

□ **2** □ □□□□ ..... p. 2

2.1.1 LATEX □□□□□□□□□ ..... p. 2

□ **3** □ □□ ..... p. 3

3.1 □□□ ..... p. 3

□□□□ ..... p. 5

□ 1 □ □□

Typst □ markdown like □□□□□□□□ pdf, □□□□, □□□□□□□□□□□□□□□□□□□□□□. Rust □□□□□□□□□□, □□□□□□ LaTeX □□ □□□□□□□□□□□□.

1.1 Typst □□□□

□□□□□□□ @ss8843592 or #cite(<ss8843592>) □□□□□□□□

□□□□□□□ [1] or [1] □□□□□□□□

1.1.1 □□□□□□□□□□

□□

```
$ mat(1, 2; 3, 4) $ <eq1>
```

□□□□

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \tag{1.1}$$

□ (1.1) □□□□□□□□□□□.

□□□□□□□

typst

□ 1.1 □□□□

□ 1.1 □□□□□□□□□,

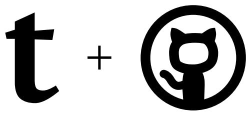
□ 1.1 □□□□ [2]

t	1	2	3
y	0.3s	0.4s	0.8s

□ 1.1 □□□□□□□□.

[illegible] 2 [illegible] [illegible]

LaTeX [illegible], [illegible], [illegible] Typst [2] [illegible] [illegible]([illegible]).



[illegible] 2.1 Typst + git [2]

2.1.1 LaTeX [illegible]

[illegible], LaTeX [illegible]. [illegible]

```
#include path.typ
```

[illegible], [illegible], [illegible], [illegible].
[illegible].
[illegible] [illegible] [illegible]

[illegible] 3 [illegible] [illegible]

Typst [illegible].

3.1 [illegible]

`thmbox` [illegible].

```
#let theorem = thmbox(
  "theorem", //identifier
  "[illegible]",
  base_level: 1
)

#theorem("[illegible]-")[
  Typst [illegible].
] <theorem>
```

[illegible] **3.1** ([illegible]-): Typst [illegible].

```
#let lemma = thmbox(
  "theorem", //identifier
  "[illegible]",
  base_level: 1,
)

#lemma[
  Tex [illegible].
] <lemma>
```

[illegible] **3.2**: Tex [illegible].

[illegible], [illegible] 3.1 , [illegible] 3.2 [illegible] .

[illegible]. [illegible], identifier [illegible], [illegible]. identifier [illegible], [illegible] [illegible].

```
#let definition = thmbox(
  "definition", //identifier
  "[illegible]",
  base_level: 1,
  stroke: black + 1pt
)
#definition("Prime numbers")[
  A natural number is called a _prime number_ if it is greater than $1$ and
  cannot be written as the product of two smaller natural numbers.
] <definition>
```

□□ **3.1**: Typst is a new markup-based typesetting system for the sciences.

□□ 3.1 □□□□□□□□□□□□□□□□□□□□□.

```
#let corollary = thmbox(
  "corollary",
  "Corollary",
  base: "theorem",
)

#corollary[
  If $n$ divides two consecutive natural numbers, then $n = 1$.
] <corollary>
```

Corollary 3.2.1: If n divides two consecutive natural numbers, then $n = 1$.

base □ identifier □□□□□□□ Corollary 3.2.1 □□□□□□□□□□□□□□□□□□□.

```
#let example = thmplain(
  "example",
  "Example"
).with(numbering: none)

#example[
  □□□\$\$ □□□
] <example>
```

□: □□□ $$ □□□

thmplain □□□□□□ plain □□□□□□□.

□□□□

[1] S. Hussain, S. Bai, and S. Khoja, “Content MathML(CMML) conversion using LATEX Math Grammar (LMG)”, in *2019 7th International Conference on Smart Computing & Communications (ICSCC)*, 2019, pp. 1–5. doi: 10.1109/ICSCC.2019.8843592.

[2] L. Mädje, “A Programmable Markup Language for Typesetting”, 2022.