

# Typst Template\*

多鹿 智哉†

The other Author‡

2024 年 4 月 7 日

## Abstract

Typst の機能の個人的なメモです。数ある機能のうちのほんの少しを紹介しています。Typst 0.11.0 を前提にしています。

## 1 Typst について

- TeX/LaTeX に代わる (かもしれない) 組版システム
- 特徴
  - TeX/LaTeX よりコンパイルが早い (リアルタイム)
  - 直観的な構文
  - カスタマイズの容易さ
  - エラーメッセージのわかりやすさ
- 詳しく知りたい方は公式ドキュメント(<https://typst.app/docs/reference>)を読むこと。

## 2 使用方法/インストール方法

- 公式アプリ(<https://typst.app>)が使いやすいのでそちらを使ってみましょう。
- ローカルにインストールするには

<https://github.com/typst/typst#installation> をご覧ください

- mac ならターミナルで以下のように入力する。

```
brew install typst
```

- Windows でもいろいろ方法があります。
  - 例えば「アプリインストーラー」がインストールされていれば Powershell/コマンドプロンプトから以下のように入力します。

```
winget install --id Typst.Typst
```

- エディタはいろいろありますが VS code では Typst LSP および Typst Preview の二つの拡張機能をインストールします。

---

\*This template assumes version 0.11.0

†日本大学経済学部, [tajika.tomoya@nihon-u.ac.jp](mailto:tajika.tomoya@nihon-u.ac.jp)

‡Institutions and mail address

- Typst のバージョンのアップデートには、インストール時と同じことをするか、v0.8.0 以降であれば powershell/ コマンドプロンプトやターミナルから

```
typst update
```

でアップデートされます。

## 3 簡単な使い方

### 3.1 はじめに

初めにいろいろ文字を入力してみましょう。Typst では書いた文字がそのまま反映されます。

#### 3.1.1 見出し

見出しは次のように設定します。

```
= 見出しレベル 1  
== 見出しレベル 2
```

見出しのレベルは **=** を重ねると下がります。

Typst では特殊な機能を使うために **#** を用います。**#** のついた命令は **関数** といいます。例えば見出しは次のようにも書くことができます。

```
#heading("見出し")
```

見出しのレベルを設定するには次のようにします。

```
#heading("見出し", level: 2)
```

また、次のようにすれば見出しに番号をつけることができます。

```
#heading("見出し", numbering: "1.1")
```

#### 3.1.2 箇条書き

箇条書きは次のようにします。

```
- 箇条書き項目 1  
- 箇条書き項目 2
```

これによって次のように出力されます。

- 箇条書き項目 1

---

<sup>1</sup>Windows 10/11 であれば既にインストールされているか、Microsoft Store にあります。「アプリインストーラー」で検索してみてください。

- 箇条書き項目 2

箇条書きのレベルを下げるにはインデントを下げる (Tab キーで空白を開ける) だけで十分です。

```
- 箇条書き項目 1
  - 箇条書き項目 2
```

また、番号付きの箇条書きをするには次のように入力します。

```
+ 箇条書き項目 1
+ 箇条書き項目 2
```

これによって次のよう出力されます。

1. 箇条書き項目 1
2. 箇条書き項目 2

箇条書きの記号や番号の設定については Section 4.1 をご覧ください。

### 3.1.3 脚注

脚注は `#footnote[脚注の中身]` と入力します<sup>2</sup>。

### 3.1.4 改行

改行には `\` を用います。

### 3.1.5 段落の左/右/中央揃え

段落を右揃えにしたり中央揃えにするには `#align` を用います。

```
#align(center, "中央に配置")
```

と入力すれば次のように配置されます。

中央に配置

### 3.1.6 文章の強調

斜体は `_italic_`、あるいは `#emph[italic]` とします。そうすれば *italic* と表示されます。

太字は `*strong*`、あるいは `#strong[strong]` とします。そうすれば **strong** と表示されます。

---

<sup>2</sup>このように脚注ができます。

### 3.1.7 下線を引く

下線を引くには `#underline[下線を引く]` とします。そうすれば 下線を引く と出力されます。

### 3.1.8 文字色/ハイライト

文字色を変更するには `#text(fill:red)[赤い文字]` とします。そうすると **赤い文字** と出力できます。 **ハイライトで強調** するには `#highlight[ハイライトで強調]` と入力します。

## 3.2 数式

Typst で数式は `$x^2$` などと入力します。これによって  $x^2$  と出力できます。基本的に `$` で囲んだ領域は数式モードになります。

- いくつか関数などもあります。以下が例です.<sup>3</sup>

- ▶  $x_2$ :  `$x_2$`
- ▶  $\sin(x)$ :  `$\sin(x)$`
- ▶  $\lim_{x \rightarrow \infty} x$ :  `$\lim_{x \rightarrow \infty} x$`
- ▶  $\sum_{x=1}^n x^2$ :  `$\sum_{x=1}^n x^2$`
- ▶  $\frac{1}{2}$ :  `$\frac{1}{2}$`
- ▶  $\frac{\partial f}{\partial x}$ :  `$\frac{\partial f}{\partial x}$`
- ▶  $\frac{df}{dx}$ :  `$\frac{df}{dx}$`
- ▶  $\int_a^b f(x) dx$ :  `$\int_a^b f(x) dx$`
- ▶  $\left[\frac{1}{2}, 1\right]$ :  `$\left[\frac{1}{2}, 1\right]$`
- ▶  $\left[\frac{1}{2}, 1\right)$ :  `$\left[\frac{1}{2}, 1\right)$`
- ▶  $\mathbb{R}$ :  `$\mathbb{R}$`

- 記号のリストについては以下を参照してください。

<https://typst.app/docs/reference/symbols/sym/>

- ▶ こちらは手書きで記号を探すアプリです。

<https://detypify.quarticcat.com/>

また、`$ f(x)$`のように`$`の後にスペースを入れるか、あるいは改行するとディスプレイ数式ができます。例えば次のような入力をしてみましょう。

```
$
f(x)&=integral_0^x y dif y \
&= x^2/2
$
```

すると次のように出力されます。

$$\begin{aligned} f(x) &= \int_0^x y \, dy \\ &= \frac{x^2}{2} \end{aligned}$$

ここで`\`は改行、`&`は位置を揃えることを意味します。

またカッコの大きさは自動で調整されます。ただし`||`は自動調整されません。なので絶対値やノルムは`$abs(a)$`、`$norm(x/2)$`と入力します。そうすると $|a^2|$ 、 $\left\|\frac{x}{2}\right\|$ と出力されます。

長々しい数式を繰り返し使うときは`#let`をつかって省略してみましょう。例えば以下のようにしてみてください。

```
#let Bnm = $sum_(i=0)^n binom(n,i) x^i (1-x)^(n-i)$
$Bnm$
```

こうすることで`Bnm`を新しい数学記号として定義できるのです。この文字列は既存のものとかぶっていなければ何でも良いです。（出力がおかしい場合には既存のものとかぶっている可能性があるので少し変更してみましょう）

数式について詳しくは <https://typst.app/docs/reference/math/> をご覧ください。

## 3.3 図表

図表を

### 3.3.1 図

図には`#image`を使います。画像を取り込むときは`#image(ファイル名.拡張子)`とします。詳しくは以下のページを参照してください

---

<sup>3</sup>LaTeXのように`\`をつける必要はありません。

### 3.3.2 表

表は以下のように書きます。

```
table(columns:3,[1],[2],[3],[4])
```

1	2	3
4		

このように `columns` で列の数を指定すれば、あとは配列を書き連ねると表を組版してくれます。より細かい設定は以下をご覧ください

<https://typst.app/docs/reference/model/table/>

### 3.4 相互参照/引用

引用したい図、数式や見出しにラベルをつけます。ラベルの名称は何でも良いです。ラベルをつけられるのは、`figure`, `heading`, `math.equation`, `footnote` のみです

```
= 引用したい節<label>  
@label で引用できます。
```

- 図表を参照するには以下のように `figure` で囲む必要があります。 `figure` の後ろにラベルをつけます。

```
#figure(  
  #table([],[],[])  
  caption: "my first table"  
)<table>
```

- ▶ 詳しくは以下を参照してください

<https://typst.app/docs/reference/model/figure/>

- 文献の引用には BibTeX が使えます。まず `bib` ファイルを作成し<sup>4</sup>、参考文献を出力したい場所に以下のように入力した上で参照キーを `@` を使って引用すれば引用文献が参考文献リストに現れます。詳しくはこちらを参照してください

<https://typst.app/docs/reference/model/bibliography/>

### 3.5 パッケージ

パッケージを利用することで Typst の機能を拡張できます。パッケージの一覧はこちらです。 <https://typst.app/universe>

パッケージの一例を挙げれば次のようなものがあります。

---

<sup>4</sup>例えば <https://mathlandscape.com/latex-bib/> を参照のこと

**Physica** 数式関数の拡張. より容易に数式を書くことができます.

**Lemmify** 定理環境

**CeTZ** 図の描画 (LaTeX で言う TikZ のようなもの)

**Showybox** カスタマイズ可能なボックス (LaTeX で言う tcolorbox)

## 4 カスタマイズする

### 4.1 set-show rule

ここでは関数のデフォルトを変更することを考えます。例えば、フォントを変更したいときには次のようにします。

```
#set text(font:"フォントの名前")
```

このように **#set** を使うことでデフォルトを変更できます。他にも、箇条書きの記号を変更するときは次のようにします。

```
#set list(marker: ([•], [--]))  
- Top-level  
  - Nested  
  - Items  
- Items
```

これによって記号を次のように変更できます。

- Top-level
  - Nested
  - Items
- Items

Typst の関数にはカスタマイズできる項目が多数あり、それを **#set** を使うことで変更しています。

これに対して、**#show** 関数を使うと、関数を丸ごとカスタマイズすることができます。例えば以下のようにすれば強調関数をそのまま書き換えることができます。

```
#show strong: it => {text(fill:red)[[#it.body]]}  
*強調*
```

set-rule や show-rule について詳しくは以下を見てください。

<https://typst.app/docs/reference/styling/>

## 4.2 自作関数/変数

`#let` 命令を使うことで、自作の関数を作ることができます。例えば以下のようにします。

```
#let pd(num,denom) = {$(diff num)/(diff denom)$}
$pd(f,x)$
```

そうすると次のように出力されます。  $\frac{\partial f}{\partial x}$

デフォルトを設定することもできます。例えば次のようにしてみましょう。

```
#let pd(num,denom,mode:none) = {
  if mode == "D" {
    $D_(denom) num$
  }
  else {
    $(diff num)/(diff denom)$
  }
}
```

すると、デフォルトで `$pd(f,x)$` を入力すれば  $\frac{\partial f}{\partial x}$  が、`$pd(f,x,mode:"D")$` とすれば  $D_x f$  と出力されます。

なお、physica パッケージを用いれば、偏微分などの記号は設定されています。

## 5 このテンプレートに含まれる機能

このファイル自体がテンプレートになっています。必要に応じて、書き換えてみてください。

### 5.1 数式の自動参照

以下の関数を使うと、数式は、参照されたときのみ数式番号をつけます。

```
#aeq("math_eq")[$
integral_(-oo)^oo (1/root(,2 pi))e^(-x^2/2) dif x = 1
$]
```

$$\int_{-\infty}^{\infty} \left( \frac{1}{\sqrt{2\pi}} \right) e^{-\frac{x^2}{2}} dx = 1 \quad (1)$$

(1) は参照されています。



$$\int_{-\infty}^{\infty} \left( \frac{1}{\sqrt{2\pi}} \right) e^{-\frac{x^2}{2}} dx = 1$$

参照されない数式には番号はつきません

## 5.2 定理環境

以下の定理が使えます. `project` の `lang` のところを `en` か `jp` にすれば, 変更できます. 辞書の中身は `libs/translation.typ` にあるので, そこを編集すれば, 変えられます.

**Theorem 1.** 定理の中身

**Proposition 1** (名称). 命題の中身

**Proof of Proposition 1.** 証明を書く ■

**Assumption 1.**

**Example 1.**

**Fact 1.**

**Corollary 1.**

**Lemma 1.**