Typst Cheat Sheet

powered by Typst, modified from Touying

by Darstib

2025-06-03

Outline

Outline

2.6.6 Quote 23

1. Syntax

1.1 Modes

fdsaf

New mode	Syntax	Example
Code	Prefix the code with `#`	Number: `#(1 + 2)`
Math	Surround equation with `$..$`	`$-x$` is the opposite of `$x$`
Markup	Surround markup with `[..]`	`#let name = [*Typst!*]`

1.2 Markup

Typst is a markup language. This means that you can use simple syntax to accomplish common layout tasks.

- symbol
  1. **strong**
  2. *emphasis*
  3. `raw text`
  4. `<label>` and `@reference`
  5. Section 1.2[1]
- function
  1. **strong**
  2. *emphasis*
  3. `fn main();`
  4. Link to raw usage
  5. $\overline{\text{overline}}$
  6. $\underline{\text{underline}}$
  7. highlight
  8. ~~strike~~
  9. [1]

[1] this is a footnote

1.3 Math

A in line function: $a^2 + b^2 = c^2$

A block function:

$$e^{i\pi} + 1 = 0$$

more complex:

$$\mathrm{M} := \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{pmatrix}$$

More symbol

1.4 Code

Data type: `#type()`

- Length: 1em, 2pt, 3mm
- Angle: 1deg, 2rad, 3grad
- Fraction: 1fr
- Ratio: 50%
- Array: (1, 2, 3)
- Dictionary: (a: 1, b: 2, c: “hi”)
- `[content](link)`
- Content block: `[content]`
- Code block: {let x =1; x+1}

Usage:

- Field access: field.name
- Function call: function()
- Arg spreading: function(..args)
- Method call: field.method()
- Unnamed func: `#show "once?": it => [#it #it]"`
- Named: `#let add(a,b)=a+b`
- Let blinding: `#let a = 1`
- `#set` and `#show`
- `#context`

2. Get deeper

2.1 Scripting

2.1.1 Field

```
#let it = [== subtitle]
#let dict = (greet: "Hello")
#it.fields()
#dict.greet
#emoji.face
```

```
(depth: 2, body: [subtitle])
```

Hello

😀

🌀

2.1 Scripting

2.1.2 Method

Just like in python

```
#let demo_str = "Hello, typst!"
#demo_str
#demo_str.len()
#str.len(demo_str)
```

```
Hello, typst!
13
13
```

2.1 Scripting

2.1.3 Flow

Loop

```
#let n = 2
#while n < 10 {
    n = (n * 2) - 1
    (n,)
}

#let s = "Hello, typst!"
#for char in s [
    (#char,)
]
```

Just like in python

Condition

```
#if 1 < 2 [
  This is shown
] else [
  This is not.
]
```

2.2 Styling

Words before #set looks like.

```
#set text(font: "New Computer Modern") // set font
```

Words after #set looks like.

```
#show table.cell.where(y: 0): strong
#show link: set text(rgb("#347c67"))
```

2.3 Selector

2.3.1 constructor

- **func element**: `heading` `figure`
- **special field**: `self.where(..any)`
- **regex**: `^[a-z]+`
- **location**: `#here().page()`
- **`<lable>`**

```
#show table.cell.where(y: 0): strong
```

2.3.2 definition

- **self.or**(selector)
- **self.and**(selector)
- **self.before**(selector)
- **self.after**(selector)

2.4 Function

2.4.1 define

Using `#let` blinding to define a function, with a code block as the body.

Warning:
This is a warning message.

```
#let alert(body, fill: red,
inset: 8pt, radius: 4pt) = {
  set text(white)
  set align(center)
  rect(
    fill: fill,
    inset: inset,
    radius: radius,
    [*Warning:\ #body*],
  )
}
```

2.4 Function

2.4.2 Import

Functions can be imported from one file (module) into another using import. For example, assume that we have defined the alert function from the previous example in a file called foo.typ. We can import it into another file by writing `import 'foo.typ': alert`.

2.5 Figure

2.5.1 image

Figure 1: Kamisato Ayaka

The Figure 1 is a demo picture.

2.5 Figure

2.5.2 table

Table 1: Timing results

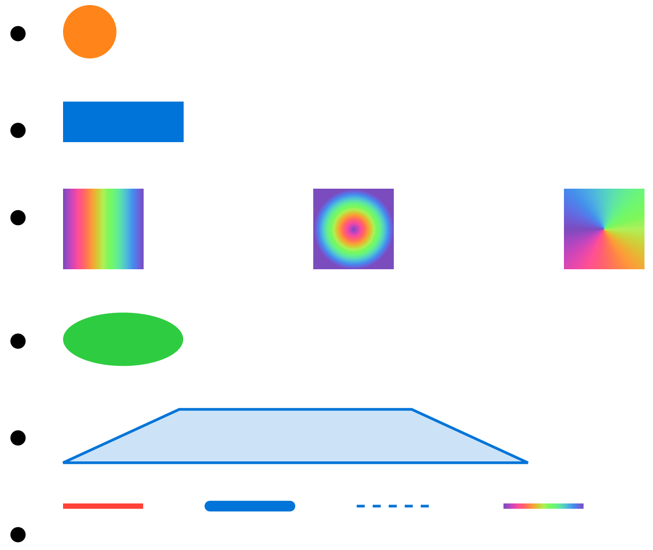
	Volume	**Parameters**
function1	$\pi h \frac{D^2 - d^2}{4}$	h: height D: outer radius d: inner radius
function2	$\frac{\sqrt{2}}{12} a^3$	a: edge length

The Table 1 is a demo table.

2.6 Other

2.6.1 basic graph

```
- #circle(radius: 10pt, fill: orange)
- #rect(height: 20pt, fill: blue)
- #square with gradient
- #ellipse(height: 20pt, fill: green)
- #polygon
- #line
```



2.6 Other

2.6.2 Sinks & Spreading

We can specify an argument sink which collects all excess arguments as `..args` and just spread it with `..args`

ArtosFlow

Written by Jane, Joe and Jake 9

```
#let format(title, ..authors)
= {
  let by = authors.pos()
    .join(", ", last: " and
")
  [*#title* \ _Written by
#by;_]
}
#format("ArtosFlow", "Jane",
"Joe", "Jake")
#let arr = (1, 3, 5, 7, 9)
#calc.min(..arr)
```

2.6 Other

2.6.3 Regex

```
// Works with string methods.
#"a,b;c".split(regex("[,;]"))
```

```
("a", "b", "c")
```

```
// Works with show rules.
#show regex("\d+"): set
text(red)

The numbers 1 to 10.
```

The numbers 1 to 10.

2.6 Other

2.6.4 Box & block

The `#box` is a simple box, which inline just like this. While the `#block` is a block element, which will be displayed as a “separate paragraphs”, just like:

this

This is a rectangle, which is more convenient to use.

2.6 Other

2.6.5 Spacing

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.

2.6 Other

2.6.6 Quote

> … ἔοικα γοῦν τούτου γε σμικρῷ τινι αὐτῷ τούτῳ σοφώτερος εἶναι, ὅτι ἃ μὴ οἶδα οὐδὲ οἴομαι εἰδέναι.
>
> — Plato

2.6 Other

2.6.6 Quote

… ἔοικα γοῦν τούτου γε σμικρῷ τινι αὐτῷ τούτῳ σοφώτερος εἶναι, ὅτι ἃ μὴ οἶδα οὐδὲ οἴομαι εἰδέναι.

— Plato

… 因此，在我看来，在这件小事上，我至少比这个人稍微聪明一点，因为我不知道的事情，我也不认为自己知道。

— 柏拉图