Guide to Typst

May 25, 2023 - v0.1 - for typist v0.4.0

tschinz
whynotlogic@gmail.com
HEI-Vs

Table of content

Table of figures

Table of tables

Table of listings

Table of equation

1 Introduction

The goal of this document is to have the most common used elements for the markup language `typst` readily available. A detailed documentation can be found on theirs website: https://typst.app/docs It is to note that these are **my** most common used elements. For some elements custom templates are needed:

- tablex
- myref
- all files in the `00-templates/` folder such as
  - `boxes.typ`
  - `constants.typ`
  - `helpers.typ`
  - `items.typ`
  - `metadata.typ`
  - `template-*`

2 Installation

2.1 With `cargo`

If you use already the `rust` programming language then you can use rust to install the latest toolchain.

```
# install rust and cargo
curl https://sh.rustup.rs -sSf | sh

# install typst
cargo install --git https://github.com/typst/typst
```

2.2 MacOS

On MacOS you can sue `homebrew`

```
# install homebrew
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/
install.sh)"

# install typst
brew install typst
```

2.3 Linux

In Linux you can use the commonly available package manager

```
brew install typst
pacman -S typst
xbps-install typst
sudo apt-get install typst
```

3 Formatting

3.1 Markup

Name	Example	Raw
Singleline Comment		`//`
Multiline Comment		`/*  */`
Paragraph break		`blankline`
Line break		`\`
bold	**bold**	`*bold*`
italic	*italic*	`_italic_`
monospaced	`monospaced`	`` `monospaced` ``
math	$x = 1$	`$x=1$`
lowercase	lower	`#lower("LoWeR")`
uppercase	UPPER	`#upper("UpPeR")`
smallcaps	SMALLCAPS	`#SmallCaps("smallcaps")`
smartquote	“test”	`#smartquote()test#smartquote())`
overline	overline	`#overline("overline")`
underline	underline	`#underline("underline")`
strike	~~strike~~	`#strike("strike")`
sub	Text$_{sub}$	`Text#sub("sub")`
super	Text^{super}	`Text#super("super")`
Label		`<label>`
Reference		`@label`

3.2 Page Formatting

```
#pagebreak()   // pagebreak
#parbreak()    // parbreak
\              // linebreak
```

3.3 Space

A

B

```
A #h(5cm) B,
```

C

```
C #v(0.2cm) D
```

D

3.4 Text Formatting

For the custom textsizes and colors you need to import:

```
#import "../01-tail/constants.typ": *
```

Name	Example	Raw
Sizes	8pt text tiny text	`text(8pt, "8pt text")` `text(tiny "tiny text")`
	9pt text smaller text	`text(9pt, "9pt text")` `text(smaller "smaller text")`
	10pt text small text	`text(10pt, "10pt text")` `text(small "small text")`
	11pt text normal text	`text(11pt, "11pt text")` `text(normal "normal text")`
	14pt text large text	`text(14pt, "14pt text")` `text(large "large text")`
	16pt text larger text	`text(16pt, "16pt text")` `text(larger "larger text")`
	24pt text huge text	`text(24pt, "24pt text")` `text(huge "huge text")`
	36pt text huger text	`text(36pt, "36pt text")` `text(huger "huger text")`
Types	Fira Sans	`text(font:"Fira Sans", "Fira Sans")`
	Fira Mono	`text(font:"Fira Mono", "Fira Mono")`
	Source Sans Pro	`text(font:"Source Sans Pro", "Source Sans Pro")`
	New Computer Modern	`text(font:"New Computer Modern", "New Computer Modern")`
	New Computer Modern Sans	`text(font:"New Computer Modern Sans", "New Computer Modern S`

Alignment	start	`align(start){start}`
	end	`align(end){end}`
	left	`align(left){left}`
	center	`align(center){center}`
	right	`align(right){right}`
	top	`align(top){top}`
	horizon	`align(horizon){horizon}`
	bottom	`align(bottom){bottom}`
	center + horizon	`align(center + horizon){center + horizon}`

Colors	black	`#text(fill:black)[black]`
	red	`#text(fill:red)[red]`
	green	`#text(fill:green)[green]`
	blue	`#text(fill:blue)[blue]`
	purple	`#text(fill:purple)[purple]`
	gray-80	`#text(fill:gray-80)[gray-80]`
	gray-70	`#text(fill:gray-70)[gray-70]`
	gray-60	`#text(fill:gray-60)[gray-60]`
	gray-50	`#text(fill:gray-50)[gray-50]`
	gray-40	`#text(fill:gray-40)[gray-40]`
	gray-30	`#text(fill:gray-30)[gray-30]`
	gray-20	`#text(fill:gray-20)[gray-20]`
	gray-10	`#text(fill:gray-10)[gray-10]`
	hei-orange	`#text(fill:hei-orange)[hei-orange]`
	hei-blue	`#text(fill:hei-blue)[hei-blue]`
	hei-pink	`#text(fill:hei-pink)[hei-pink]`
	hei-yellow	`#text(fill:hei-yellow)[hei-yellow]`
	hei-green	`#text(fill:hei-green)[hei-green]`
	spl-green	`#text(fill:spl-green)[spl-green]`
	spl-blue	`#text(fill:spl-blue)[spl-blue]`
	spl-pink	`#text(fill:spl-pink)[spl-green]`
	color-info	`#text(fill:color-info)[color-info]`
	color-idea	`#text(fill:color-idea)[color-idea]`
	color-warning	`#text(fill:color-warning)[color-warning]`
	color-important	`#text(fill:color-important)[color-important]`
	color-fire	`#text(fill:color-fire)[color-fire]`
	color-rocket	`#text(fill:color-rocket)[color-rocket]`
	color-todo	`#text(fill:color-todo)[color-todo]`
	code-bg	`#text(fill:code-bg)[code-bg]`
	code-border	`#text(fill:code-border)[code-border]`

4 Elements

4.1 Headings

```
= Heading 1
== Heading 1.1
=== Heading 1.1.1
==== Heading 1.1.1.1
...
```

4.2 Lists

- First
- Second
- Third

```
- First
- Second
- Third
```

- First
  - Second
    - Third

```
- First
  - Second
    - Third
```

- First
- Second
- Third

```
- First
- Second
- Third
```

- First
- Second
- Third

```
list(
  [First],
  [Second],
  [Third],
)
```

1. First
   1. Second
2. Third

Text

4. Fourth
5. Fifth

```
+ First
  + Second
+ Third
Text
4. Fourth
+ Fifth
```

1. First
   a) Second
2. Third

Text

4. Fourth
5. Fifth

```
+ First
  #set enum(numbering: "a)")
  + Second
+ Third
Text
4. Fourth
+ Fifth
```

4.3 Custom Lists

```
#import "../01-head/items.typ": *
```

- item-list
- item-checkbadge
- item-checkcircle
- item-checksquare
- ✓ item-check
- item-file
- item-folder
- item-xcircle
- item-xsquare
- × item-x

```
#item-list(content:"item-list")
#item-checkbadge(content:"item-checkbadge")
#item-checkcircle(content:"item-checkcircle")
#item-checksquare(content:"item-checksquare")
#item-check(content:"item-check")
#item-file(content:"item-file")
#item-folder(content:"item-folder")
#item-xcircle(content:"item-xcircle")
#item-xsquare(content:"item-xsquare")
#item-x(content:"item-x")
```

4.4 Images

4.4.1 Alignment

left

```
#image("../04-resources/icon.svg",
  width: 2cm)
```

center

```
#align(center,
  image("../04-resources/icon.svg",
    width: 2cm)
)
```

right

```
#align(right,
  image("../04-resources/icon.svg",
    width: 2cm)
)
```

4.4.2 Caption

Figure 1: ZNotes Icon

```
#figure(
  image("../04-resources/icon.svg",
    width: 2cm),
  caption: [ZNotes Icon]
) <fig-icon>
```

4.4.3 Cluster

Two images one caption

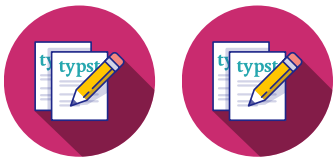

Figure 2: Multiple images **one** caption

```
#figure(
  tablex(
    columns: 2,
    stroke: none,
    align: center + horizon,
    image(icon, width: 2cm),image(icon, width: 2cm)
  ),
  caption: [Multiple images *one* caption]
)
```

Four images one caption

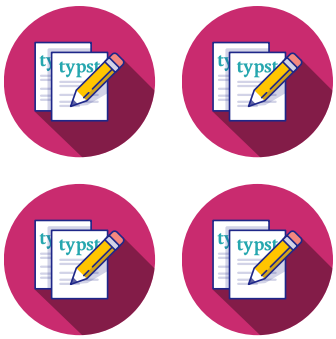

Figure 3: Multiple images **one** caption

```
#figure(
  tablex(
    columns: 2,
    stroke: none,
    align: center + horizon,
    image(icon, width: 2cm), image(icon, width: 2cm),
    image(icon, width: 2cm), image(icon, width: 2cm),
  ),
```

```
  caption: [Multiple images *one* caption]
)
```

Two images two caption

Figure 4: Caption left image

Figure 5: Caption right image

```
#align(center,
  tablex(
    columns: 2,
    stroke: none,
    align: center + horizon,
   figure(image(icon, width: 2cm), caption: [Caption left image]), figure(image(icon,
width: 2cm), caption: [Caption right image]),
))
```

Four images four caption

Figure 6: Caption topleft image

Figure 7: Caption topright image

Figure 8: Caption bottomleft image

Figure 9: Caption bottomright image

```
#align(center,
  tablex(
    columns: 2,
    stroke: none,
    align: center + horizon,
    figure(image(icon, width: 2cm), caption: [Caption topleft image]),
    figure(image(icon, width: 2cm), caption: [Caption topright image]),
    figure(image(icon, width: 2cm), caption: [Caption bottomleft image]),
    figure(image(icon, width: 2cm), caption: [Caption bottomright image]),
))
```

4.5 Tables

For all `#tablex` command the appropriate module nedds to be imported

```
#import "../01-head/tablex.typ": *
```

Tables with and without caption

	Col1	Col2
Row1	cell-0-0	cell-1-0
Row2	cell-0-1	cell-1-1

```
tablex(
  columns: 3,
  align: center + horizon,
  []      , [*Col1*]  , [*Col2*],
  [*Row1*], "cell-0-0", "cell-1-0",
  [*Row2*], "cell-0-1", "cell-1-1",
)
```

	Col1	Col2
Row1	cell-0-0	cell-1-0
Row2	cell-0-1	cell-1-1

Table 1: Table caption

```
figure(
  tablex(
    columns: 3,
    align: center + horizon,
    []      , [*Col1*]  , [*Col2*],
    [*Row1*], "cell-0-0", "cell-1-0",
    [*Row2*], "cell-0-1", "cell-1-1",
  ),
  kind: table,
  caption: [Table Caption]
)
```

Tables with cell spans

	Col1	Col2
Row1	cell-0	cell-1-0
Row2		cell-1-1

```
tablex(
  columns: 3,
  align: center + horizon,
  []      , [*Col1*]  , [*Col2*],
     [*Row1*],  rowspanx(2)[cell-0],
"cell-1-0",
  [*Row2*],               "cell-1-1",
)
```

	Col1	Col2
Row1	cell-0	
Row2	cell-0-1	cell-1-1

```
tablex(
  columns: 3,
  align: center + horizon,
  []      , [*Col1*]  , [*Col2*],
  [*Row1*], colspanx(2)[cell-0],
  [*Row2*], "cell-0-1", "cell-1-1",
)
```

Table Design

	Col1	Col2
Row1	cell-0-0	cell-1-0
Row2	cell-0-1	cell-1-1

```
tablex(
  columns: 3,
```

	Col1	Col2
Row1	cell-0-0	cell-1-0
Row2	cell-0-1	cell-1-1

```
tablex(
  columns: 3,
```

```
    auto-vlines: false,
    align: center + horizon,
    []      , [*Col1*]  , [*Col2*],
    [*Row1*], "cell-0-0", "cell-1-0",
    [*Row2*], "cell-0-1", "cell-1-1",
  )
```

```
    auto-hlines: false,
    align: center + horizon,
    []      , [*Col1*]  , [*Col2*],
    [*Row1*], "cell-0-0", "cell-1-0",
    [*Row2*], "cell-0-1", "cell-1-1",
  )
```

	Col1	Col2
Row1	cell-0-0	cell-1-0
Row2	cell-0-1	cell-1-1

```
#tablex(
columns: 3,
auto-lines: false,
align: center + horizon,
(), vlinex(stroke: blue) , vlinex(), (),
[]      , [*Col1*]  , [*Col2*], hlinex(stroke: red),
[*Row1*], "cell-0-0", "cell-1-0", hlinex(),
[*Row2*], "cell-0-1", "cell-1-1",
)
```

4.6 Boxes

```
#import "../01-head/boxes.typ": *
```

```
#infobox(content:"infobox")
```

```
#ideabox(content:"ideabox")
```

```
#warningbox(content:"warningbox")
```

```
#importantbox(content:"importantbox")
```

```
#firebox(content:"firebox")
```

```
#rocketbox(content:"rocketbox")
```

```
#todobox(content:"todobox")
```

```
#box(content:"todobox", icon:"../04-resources/
placeholder.svg", linecolor: hei-pink)
```

5 References

5.1 Links

Example	Raw
https://example.com	`https://example.com`
https://example.com	`#link("https://example.com")`
See example.com	`#link("https://example.com")[See example.com]`
whynotlogic@gmail.com	`#link("mailto:whynotlogic@gmail.com")[whynotlogic\@gmail.com]`
	`#link("https://tschinz.github.io/znotes")[#image(icon, width:0.5cm)]`

Table 2: Links

5.2 Crossreferences

In the document the following references were added.

```
= References <sec-ref>
== Links <sec-links>
#figure(image("../04-resources/icon.svg", width: 2cm)) <fig-icon>
#figure(tablex(...), kind:table) <tab-links>
#figure(align(left, raw(...)) <code-ref>
$ sum_(k=1)^n k = (n(n+1)) / 2 $ <math-eq1> #ref(<math-eq1>)
```

Listing 1: Label inserts

They can be references as follows:

Type	Example	Raw
Section	Section 5	`@sec-ref`
Subsection	Section 5.1	`@sec-links`
Table	Table 2	`@tab-links`
Code	Listing 1	`@code-ref`

5.3 External References

Example	Raw
[1]	`#cite("stateoftheArt")`
[1]	`@stateoftheArt`

5.4 Glossary

The glossary entries need to be defined in `03-tail/glossary.typ`. For the glossary functions the "import" of `01-head/helpers.typ` is needed.

```
#import "../01-head/helpers.typ": *
#import "../03-tail/glossary.typ": *
```

Example	Raw
Scrum	`#gls-scrumm.name`
Scrum is an agile process framework for managing complex knowledge work, with an initial emphasis on software development, although it has been used in other fields and is slowly starting to be explored for other complex work, research and advanced technologies.	`#gls-scrumm.description`

5.5 Acronym

The acronym entries need to be defined in `03-tail/glossary.typ`. For the acronym functions the "import" of `01-head/helpers.typ` is needed.

```
#import "../01-head/helpers.typ": *
#import "../03-tail/glossary.typ": *
```

Example	Raw
AR	`#acr-ar.abr`
AR	`#acrshort(acr-ar.abbr)`
Augmented Reality	`#acr-ar.long`
Augmented Reality)	`#acrlong(acr-ar)`
Augmented Reality (AR)	`#acr-ar.long (#acr-ar.abbr)`
Augmented Reality (AR)	`#acrfull(acr-ar)`

6 Code

```
inline monospaced string
```

```
`inline monospaced string`
```

```
fn main() {println!("Hello world!")}
```

```
raw(lang:"rust",
  "fn main() {println!(\"Hello world!
\")"
)
```

```rust
fn main() {
  println!("Hello world!")
}
```

```
``\`rust
fn main() {
  println!("Hello world!")
}
``\`
```

```rust
fn main() {
  println!("Hello world!")
}
```

Listing 2: Rust Code

```
#figure(
  align(left,
    ``\`rust
      fn main() {
        println!("Hello world!")
      }
    ``\`
  ),
  caption: [Rust Code],
)
```

7 Math Equations

Inline math

Let a and b, and c be the side of a right-angled triangle.

```
Let $a$ and $b$, and $c$ be the side of
a right-angled triangle.
```

$\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$

```
$sum_(k=1)^n k = (n(n+1)) / 2$,
```

Fullline math

$$a^2 + b^2 = c^2 \tag{1}$$

```
$ a^2 + b^2 = c^2 $ <math-eq1>
```

Math with caption

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2} \tag{2}$$

Figure 10: Some proof

```
#figure(
  $ sum_(k=1)^n k = (n(n+1)) / 2 $,
  caption: [Some proof]
)
```

7.1 Align

Formula

$$\begin{aligned} a_1 &= b_1 + c_1 = z_1 \\ a_2 &= b_2 + c_2 - d_2 + e_2 = z_1 \end{aligned} \tag{3}$$

Raw

```
$
a_1 = b_1 + c_1 = z_+ \
a_2 = b_2 + c_2 - d_2 + e_2 = z_1
$
```

$$\begin{aligned} a_1 &= b_1 + c_1 &&= z_1 \\ a_2 &= b_2 + c_2 - d_2 + e_2 &&= z_1 \end{aligned} \tag{4}$$

```
$
a_1 &= b_1 + c_1 &= z_1 \
a_2 & = b_2 + c_2 - d_2 + e_2 &= z_1
$
```

7.2 Symbols

This is an incomplete list for all symbols goto here
Outside of the $$ math environment the symboils can be accessed with .

7.2.1 Accents

Symbol	Raw	Symbol	Raw	Symbol	Raw
$\grave{x}$	`$grave(x)$`	$\acute{x}$	`$acute(x)$`	$\hat{x}$	`$hat(x)$`
$\tilde{x}$	`$tilde(x)$`	$\breve{x}$	`$breve(x)$`	$\dot{x}$	`$dot(x)$`
$\ddot{x}$	`$dot.double(x)$`	$\dddot{x}$	`$dot.triple(x)$`	$\ddddot{x}$	`$dot.quad(x)$`
$\ddot{x}$	`$diaer(x)$`	$\mathring{x}$	`$circle(x)$`	$\acute{\acute{x}}$	`$acute.double(x)$`
$\check{x}$	`$caron(x)$`	$\vec{x}$	`$arrow(x)$`	$\overleftarrow{x}$	`$arrow.l(x)$`
$\cancel{x}$	`$cancel(x)$`	$\bar{x}$	`$macron(x)$`	$\overline{xyz}$	`$overline(xyz)$`
$\underline{xyz}$	`$overline(xyz)$`	$\underbrace{xyz}$	`$underbrace(xyz)$`	$\overbrace{xyz}$	`$overbrace(xyz)$`
$\underbracket{xyz}$	`$underbracket(xyz)$`	$\overbracket{xyz}$	`$overbracket(xyz)$`		

7.2.2 Equals & Operators

Symbol	Raw	Symbol	Raw	Symbol	Raw
$=$	`$=$`	$=$	`$eq$`	$\neq$	`$eq.not$`
$\neq$	`$!=$`	$\equiv$	`$ident$`	$\not\equiv$	`$ident.strict$`
$\simeq$	`$tilde.eq$`	$\not\simeq$	`$tilde.eq.not$`	$-$	`$eq.small$`
$\geq$	`$gt.eq$`	$\ngeq$	`$gt.eq.not$`	$\leq$	`$lt.eq$`
$\nleq$	`$lt.eq.not$`	$\approx$	`$approx$`	$\approxeq$	`$approx.eq$`
$\not\approx$	`$approx.not$`	$:$	`$colon$`	$:=$	`$colon.eq$`
$=:$	`$eq.colon$`	$::=$	`$colon.double.eq$`	$+$	`$+$`
$+$	`$plus$`	$+$	`$plus.small$`	$\pm$	`$plus.minus$`
$\oplus$	`$plus.circle$`	$-$	`$-$`	$-$	`$minus$`
$\mp$	`$minus.plus$`	$\ominus$	`$minus.circle$`		

7.2.3 Scripts

Symbol	Raw	Symbol	Raw	Symbol	Raw
x_1	`$x_1$`	x_{12}	`$x_(12)$`	x_1	`$scripts(x)_1$`
x_1	`$x_1$`	x_{12}	`$x_(12)$`	x_1	`$scripts(x)_1$`
x_1^2	`$x_1^2$`	x_{12}^{34}	`$x_(12)^(34)$`	x_1^2	`$scripts(x)_1^2$`
$\overset{2}{x}_1$	`$x_1^2$`	x_{12}^{34}	`$x_(12)^(34)$`	x_1^2	`$scripts(x)_1^2$`

7.2.4 Special Elements

Symbol		Raw	Symbol		Raw
$\binom{n}{k}$	(5)	`$ binom(n, k) $`	$\begin{bmatrix}1\\2\end{bmatrix}$	(6)	`$ vec(1, 2, delim: "[") $`
$\left\lceil \frac{1}{2} \right\rceil$	(7)	`$ round(1, 2) $`	$\begin{pmatrix}1 & 2\\3 & 4\end{pmatrix}$	(8)	`$ mat(1,2; 3,4) $`

$$\begin{pmatrix} 1 & 2 & \dots & 10 \\ 2 & 2 & \dots & 10 \\ \vdots & \vdots & \ddots & \vdots \\ 10 & 10 & \dots & 10 \end{pmatrix} \quad (9)$$

```
$ mat(
  1, 2, ..., 10;
  2, 2, ..., 10;
        dots.v,
dots.v, dots.down,
dots.v;
   10,  10,  ...,
10;
) $
```

$$\sum a_k \quad (10) \quad \texttt{\$ sum a\_k \$}$$

$$\sum_{k=0}^{n} a_k \quad (11) \quad \texttt{\$ sum\_(k=0)\^n a\_k \$}$$

$$\sum_{k=0}^{n} a_k \quad (12) \quad \texttt{\$ scripts(sum)\_(k=0)\^n a\_k \$}$$

$$\sqrt[3]{x} \quad (13) \quad \texttt{\$ root(3, x) \$}$$

$$f(x,y) := \begin{cases} 1 \text{ if } \frac{x \cdot y}{2} \leq 0 \\ 2 \text{ if } x \text{ is even} \\ 3 \text{ if } x \in \mathbb{N} \\ 4 \text{ else} \end{cases} \quad (14)$$

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

$$\frac{1}{2} \quad (15) \quad \texttt{\$ 1/2 \$}$$

$$\frac{1}{2} \quad (16) \quad \texttt{\$ frac(1,2) \$}$$

$$\frac{x+1}{x+2} \quad (17) \quad \texttt{\$ (x+1)/(x+2) \$}$$

$$\frac{(x+1)}{(x+2)} \quad (18) \quad \texttt{\$ ((x+1))/((x+2)) \$}$$

$$\prod \quad (19) \quad \texttt{\$ product \$}$$

$$n! = \prod_{k=1}^{n} k \quad (20) \quad \texttt{\$ n! = product\_(k=1)\^n k \$}$$

$$n! = \prod_{k=1}^{n} k \quad (21) \quad \texttt{\$ n! = scripts(product)\_(k=1)\^n k \$}$$

$$\int \quad (22) \quad \texttt{\$ integral \$}$$

$$\int_a^b f(x) \quad (23) \quad \texttt{\$ integral \$}$$

7.2.5 Alphabeth

Symbol	Raw
$\alpha\beta\gamma\delta\varepsilon\zeta\eta\theta\iota\kappa\lambda\mu\nu\xi o\pi\rho\sigma\tau\upsilon\varphi\chi\psi\omega$	`$alpha beta gamma delta epsilon zeta eta theta iota kappa lambda mu nu xi omicron pi rho sigma tau upsilon phi chi psi omega$`
$\mathrm{AB\Gamma\Delta EZH\Theta IK\Lambda MN\Xi O\Pi P\Sigma T\Upsilon\Phi X\Psi\Omega}$	`$Alpha Beta Gamma Delta Epsilon Zeta Eta Theta Iota Kappa Lambda Mu Nu Xi Omicron Pi Rho Sigma Tau Upsilon Phi Chi Psi Omega$`
$\mathbb{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$	`$AA BB CC DD EE FF GG HH II JJ KK LL MM NN OO PP QQ RR SS TT UU VV WW XX YY ZZ$`

7.2.6 Logical

Symbol	Raw	Symbol	Raw	Symbol	Raw
$\wedge$	`$and$`	$\bigwedge$	`$and.big$`	$\&$	`$amp$`
$\vee$	`$or$`	$\|$	`$bar.v$`	$*$	`$ast.op$`
$*$	`$ast.basic$`	$*$	`$ast.low$`	$\oplus$	`$plus.circle$`
$\bigoplus$	`$plus.circle.big$`				

7.2.7 Operators

Symbol	Raw	Symbol	Raw	Symbol	Raw
$\sin x$	`$sin x$`	$\cos x$	`$cos x$`	$\tan x$	`$tan x$`
$\arcsin x$	`$arcsin x$`	$\arccos x$	`$arccos x$`	$\arctan x$	`$arctan x$`
$\sinh x$	`$sinh x$`	$\cosh x$	`$cosh x$`	$\tanh x$	`$tanh x$`
$\arg x$	`$arg x$`	$\csc x$	`$csc x$`	$\deg x$	`$deg x$`
$\det x$	`$det x$`	$\dim x$	`$dim x$`	$\exp x$	`$exp x$`
$\mod x$	`$mod x$`	$\inf x$	`$inf x$`	$\log x$	`$log x$`
$\lim x$	`$lim x$`	$\liminf x$	`$liminf x$`	$\limsup x$	`$limsup x$`
$\min x$	`$min x$`	$\max x$	`$max x$`	$\sup x$	`$sup x$`

7.2.8 Arrows

SymRaw	SymRaw	SymRaw
	Arrows right	
$\rightarrow$ `$arrow$`	$\longrightarrow$ `$arrow.long$`	$\mapsto$ `$arrow.bar$`
$\longmapsto$ `$arrow.bar.long$`	$\Rightarrow$ `$arrow.double$`	$\Longrightarrow$ `$arrow.double.long$`
$\Mapsto$ `$arrow.double.bar$`	$\Longmapsto$ `$arrow.double.bar.long$`	$\Rrightarrow$ `$arrow.quad$`
$\Rightarrow$ `$arrow.stroked$`	$\blacktriangleright$ `$arrow.filled$`	$\dashrightarrow$ `$arrow.dashed$`
$\hookrightarrow$ `$arrow.curve$`	$\rightsquigarrow$ `$arrow.squiggly$`	$\looparrowright$ `$arrow.loop$`
	Arrows left	
$\leftarrow$ `$arrow.l$`	$\longleftarrow$ `$arrow.l.long$`	$\mapsfrom$ `$arrow.l.bar$`
$\longmapsfrom$ `$arrow.l.bar.long$`	$\Leftarrow$ `$arrow.l.double$`	$\Longleftarrow$ `$arrow.l.double.long$`
$\Mapsfrom$ `$arrow.l.double.bar$`	$\Longmapsfrom$ `$arrow.l.double.bar.long$`	$\Lleftarrow$ `$arrow.l.quad$`
$\Leftarrow$ `$arrow.l.stroked$`	$\blacktriangleleft$ `$arrow.l.filled$`	$\dashleftarrow$ `$arrow.l.dashed$`
$\hookleftarrow$ `$arrow.l.curve$`	$\leftsquigarrow$ `$arrow.l.squiggly$`	$\looparrowleft$ `$arrow.l.loop$`
	Double Arrows Left Right	
$\leftrightarrow$ `$arrow.l.r$`	$\nleftrightarrow$ `$arrow.l.r.not$`	$\longleftrightarrow$ `$arrow.l.r.long$`
$\Leftrightarrow$ `$arrow.l.r.double$`	$\Longleftrightarrow$ `$arrow.l.r.double.long$`	$\nLeftrightarrow$ `$arrow.l.r.double.not$`
$\Lleftarrow$ `$arrow.l.r.stroked$`	$\blacktriangleleftright$ `$arrow.l.r.filled$`	$\leftrightsquigarrow$ `$arrow.l.r.wave$`

Arrows Top

↑	$arrow.t$	↥	$arrow.t.bar$	⇑	$arrow.t.double$
⟰	$arrow.t.triple$	⤊	$arrow.t.quad$	⇧	$arrow.t.stroked$
⬆	$arrow.t.filled$	⇡	$arrow.t.dashed$	↷	$arrow.t.curve$

Arrows Bottom

↓	$arrow.b$	↧	$arrow.b.bar$	⇓	$arrow.b.double$
⟱	$arrow.b.triple$	⤋	$arrow.b.quad$	⇩	$arrow.b.stroked$
⬇	$arrow.b.filled$	⇣	$arrow.b.dashed$	⤳	$arrow.b.curve$

Double Arrows Top Bottom

↕	$arrow.t.b$	⇕	$arrow.t.b.double$	⇳	$arrow.t.b.stroked$
⬍	$arrow.t.b.filled$				

Arrows Diagonal Top Right

↗	$arrow.tr$	⇗	$arrow.tr.double$	⇗	$arrow.tr.stroked$
⬈	$arrow.tr.filled$	⤤	$arrow.tr.hook$		

Arrows Diagonal Bottom Right

↘	$arrow.br$	⇘	$arrow.br.double$	⇘	$arrow.br.stroked$
⬊	$arrow.br.filled$	⤥	$arrow.br.hook$		

Arrows Diagonal Bottom Left

↙	$arrow.bl$	⇙	$arrow.bl.double$	⇙	$arrow.bl.stroked$
⬋	$arrow.bl.filled$	⤦	$arrow.bl.hook$		

Arrows Diagonal Top Left

↖	$arrow.tl$	⇖	$arrow.tl.double$	⇖	$arrow.tl.stroked$
⬉	$arrow.tl.filled$	⤣	$arrow.tl.hook$		

Double Arrows Diagonal

⤡	$arrow.tl.br$	⤢	$arrow.tr.bl$

Other Arrows

↻	$arrow.cw$	↷	$arrow.cw.half$	↺	$arrow.ccw$
↶	$arrow.ccw.half$				

7.2.9 Angles

Symbol	Raw	Symbol	Raw	Symbol	Raw
∠	$angle$	⦡	$angle.rev$	⦟	$angle.acute$
∡	$angle.acute$	⦛	$angle.arc$	⦢	$angle.arc.rev$
⟨	$angle.l$	⟩	$angle.r$	⟪	$angle.l.double$
⟫	$angle.r.double$	∟	$angle.right$	⦝	$angle.right.rev$
⦜	$angle.right.arc$	⊾	$angle.right.dot$	⦞	$angle.right.sq$
∢	$angle.spheric$	∢	$angle.spheric.rev$	∢	$angle.spheric.top$

7.2.10 Cool Symbols

Symbol	Raw	Symbol	Raw	Symbol	Raw
@	`$at$`	℅	`$co$`	©	`$copyright$`
℗	`$copyright.sound$`	℃	`$degree.c$`	€	`$euro$`
$	`$dollar$`	£	`$pound$`	₩	`$won$`
¥	`$yen$`	฿	`$bitcoin$`	℉	`$degree.f$`
!	`$excl$`	¡	`$excl.inv$`	‼	`$excl.double$`
⁉	`$excl.quest$`	↯	`$arrow.zigzag$`	⊛	`$ast.circle$`
⁂	`$ast.triple$`	χ	`$chi$`	⍰	`$floral$`
✠	`$maltese$`	¶	`$pilcrow$`	h	`$planck$`
♣	`$suit.club$`	♦	`$suit.diamond$`	♥	`$suit.heart$`
♠	`$suit.spade$`	🜲	`$triangle.stroked.nested2$`		

7.2.11 Style

Symbol	Raw	Symbol	Raw
$ABC123$	`$sans(A B C 1 2 3)$`	$\mathfrak{ABC123}$	`$frak(A B C 1 2 3)$`
$\mathtt{ABC123}$	`$mono(A B C 1 2 3)$`	$\mathbb{ABC123}$	`$bb(A B C 1 2 3)$`
$\mathcal{ABC123}$	`$cal(A B C 1 2 3)$`		

Symbol	Raw
$\Sigma_{i \in \mathbb{N}}\ 1 + i$	`#show math.equation: set text(font: "Fira Math")` `$sum_(i in NN) 1 + i$,`

8 Emoji Symbols

This is an incomplete list for all emoji goto here
If the emoji module is imported the `#emoji` can be removed

```
#import emoji: *
```

Sym	Raw	Sym	Raw
	`#emoji.face`		

```
#bibliography("../03-tail/bibliography.bib", style:"apa")
#bibliography("../03-tail/bibliography.bib", style:"chicago-author-date")
#bibliography("../03-tail/bibliography.bib", style:"chicago-notes")
#bibliography("../03-tail/bibliography.bib", style:"ieee")
#bibliography("../03-tail/bibliography.bib", style:"mla")
```

Bibliography

[1] P. Fettke, "State-of-the-Art des State-of-the-Art," *Wirtschaftsinformatik*, pp. 257–266, 2006, doi: 10.1007/s11576-006-0057-3.