

# Guide to Typst

03.08.2025 - v1.2.0  
for typist v0.13.1  
for hei-synd-thesis v0.2.2

Silvan Zahno  
[silvan.zahno@hevs.ch](mailto:silvan.zahno@hevs.ch)  
HEI-Vs

## Contents

1	Introduction .....	4
2	Installation .....	5
2.1	With <b>cargo</b> .....	5
2.2	MacOS .....	5
2.3	Linux .....	5
2.4	Windows .....	5
3	Formatting .....	6
3.1	Markup .....	6
3.2	Page Formatting .....	6
3.3	Space .....	6
3.4	Text Formatting .....	6
4	Elements .....	10
4.1	Headings .....	10
4.2	Lists .....	10
4.3	Custom Lists .....	11
4.4	Minitoc .....	11
4.5	Images .....	12
4.5.1	Alignment .....	12
4.5.2	Caption .....	13
4.5.3	Cluster .....	13
4.6	Tables .....	15
4.7	Icon Boxes .....	18
4.8	Color Boxes .....	19
4.8.1	Todo Box .....	20
4.8.2	Option Style .....	20
4.9	Title Box .....	21
4.10	Word/Character Count .....	21
5	References .....	23
5.1	Links .....	23
5.2	Crossreferences .....	23
5.3	External References .....	23

5.4 Glossary .....	24
--------------------	----

# Figures

Figure 1 One image one caption .....	13
Figure 2 Caption left image .....	14
Figure 3 Caption right image .....	14
Figure 4 Caption topleft image .....	14
Figure 5 Caption topright image .....	14
Figure 6 Caption bottomleft image .....	15
Figure 7 Caption bottomright image .....	15
Equation 8 Some proof .....	26

# Tables

Table 1 Multiple images <b>one</b> caption .....	13
Table 2 Multiple images <b>one</b> caption .....	14
Table 3 Table caption .....	15
Table 4 Links .....	23

# Listings

Listing 1 Label inserts .....	23
Listing 2 Rust Code .....	25

# Equations

Equation (1) .....	26
Equation (2) .....	26
Equation (3) .....	26
Equation (4) .....	26
Equation (5) .....	27
Equation (6) .....	27
Equation (7) .....	27
Equation (8) .....	27
Equation (9) .....	28
Equation (10) .....	28
Equation (11) .....	28

Equation (12) .....	28
Equation (13) .....	28
Equation (14) .....	28
Equation (15) .....	28
Equation (16) .....	28
Equation (17) .....	28
Equation (18) .....	28
Equation (19) .....	28
Equation (20) .....	28
Equation (21) .....	28
Equation (22) .....	28
Equation (23) .....	28

# 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:

- `codelst`
- `codelst`
- `glossarium`
- `wordometer`
- `ice-datetime`
- `cheq`
- `muchpdf`
- all files and folders in the `/00-templates/` folder such as
  - ▶ `karnaugh.typ`
  - ▶ `page-*.typ`
  - ▶ `sections.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.

```

1 # install rust and cargo
2 curl https://sh.rustup.rs -sSf | sh
3
4 # install typst
5 cargo install --git https://github.com/typst/typst

```

## 2.2 MacOS

On MacOS you can use **homebrew**

```

1 # install homebrew
2 /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/
install.sh)"
3
4 # install typst
5 brew install typst

```

## 2.3 Linux

In Linux you can use the commonly available package manager

```

1 brew install typst
2 pacman -S typst
3 xbps-install typst
4 sudo apt-get install typst

```

## 2.4 Windows

On Windows you can use **chocolatey**. See: <https://chocolatey.org/install>

```

1 # install chocolatey
2 # ensure to use a administrative powershell
3 Set-ExecutionPolicy Bypass -Scope Process -Force;
[System.Net.ServicePointManager]::SecurityProtocol =
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object
System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps
1'))
4
5 # install typst
6 choco install typst

```

# 3 | Formatting

## 3.1 Markup

Name	Example	Raw
Singleline Comment		//
Multiline Comment		/* */
Paragraph break		<b>blankline</b>
Line break		\
bold	<b>bold</b>	*bold*
italic	<i>italic</i>	_italic_
monospaced	<b>monospaced</b>	`monospaced`
math	$x = 1$	\$x=1\$
lowercase	lower	#lower("LoWeR")
uppercase	UPPER	#upper("UpPeR")
smallcaps	SmallCaps	#smallcaps("SmallCaps")
smartquote	"test"	#smartquote() test#smartquote()
overline	<u>overline</u>	#overline("overline")
underline	<u>underline</u>	#underline("underline")
strike	<del>strike</del>	#strike("strike")
sub	Text <sub>sub</sub>	Text#sub("sub")
super	Text <sup>super</sup>	Text#super("super")
Label		<label>
Reference		@label

## 3.2 Page Formatting

```

1 #pagebreak() // pagebreak
2 #parbreak() // parbreak
3 \ // linebreak

```

## 3.3 Space

A

B

```

1 A #h(5cm) B,

```

C

```

1 C #v(0.2cm) D

```

D

## 3.4 Text Formatting

For the custom textsizes and colors you need to import:

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

Name	Example	Raw
8pt text tiny text		<pre>1 text(8pt, "8pt text") 2 text(tiny "tiny text")</pre>
9pt text smaller text		<pre>1 text(9pt, "9pt text") 2 text(smaller "smaller text")</pre>
10pt text small text		<pre>1 text(10pt, "10pt text") 2 text(small "small text")</pre>
11pt text normal text		<pre>1 text(11pt, "11pt text") 2 text(normal "normal text")</pre>
Sizes		
14pt text large text		<pre>1 text(14pt, "14pt text") 2 text(large "large text")</pre>
16pt text larger text		<pre>1 text(16pt, "16pt text") 2 text(larger "larger text")</pre>
24pt text huge text		<pre>1 text(24pt, "24pt text") 2 text(huge "huge text")</pre>
36pt text huger text		<pre>1 text(36pt, "36pt text") 2 text(huger "huger text")</pre>
Types		
Fira Sans		<pre>1 text(font:"Fira Sans", "Fira Sans")</pre>
Fira Mono		<pre>1 text(font:"Fira Mono", "Fira Mono")</pre>
Source Sans Pro		<pre>1 text(font:"Source Sans Pro", "Source Sans Pro")</pre>
Arial		

Name	Example	Raw
		1 text(font:"Arial", "Arial")
	Times New Roman	1 text(font:"Times New Roman", "Times New Roman")
Alignment	start	
	end	
	left	
	center	
	right	
	top	
	horizon	
	bottom	
	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]

Name	Example	Raw
color-info	color-info	#text(fill:color-info)[color-info]
color-idea	color-idea	#text(fill:color-idea)[color-idea]
color-warning	color-warning	#text(fill:color-warning)[color-warning]
color-important	color-important	#text(fill:color-important)[color-important]
color-fire	color-fire	#text(fill:color-fire)[color-fire]
color-rocket	color-rocket	#text(fill:color-rocket)[color-rocket]
color-todo	color-todo	#text(fill:color-todo)[color-todo]
code-bg	code-bg	#text(fill:code-bg)[code-bg]
code-border	code-border	#text(fill:code-border)[code-border]

# 4 | Elements

## Contents

---

4.1	Headings .....	10
4.2	Lists .....	10
4.3	Custom Lists .....	11
4.4	Minitoc .....	11
4.5	Images .....	12
4.5.1	Alignment .....	12
4.5.2	Caption .....	13
4.5.3	Cluster .....	13
4.6	Tables .....	15
4.7	Icon Boxes .....	18
4.8	Color Boxes .....	19
4.8.1	Todo Box .....	20
4.8.2	Option Style .....	20
4.9	Title Box .....	21
4.10	Word/Character Count .....	21

---

### 4.1 Headings

```

1 = Heading 1
2 == Heading 1.1
3 === Heading 1.1.1
4 === Heading 1.1.1.1
5 ...

```

### 4.2 Lists

- First
- Second
- Third

```

1 - First
2 - Second
3 - Third

```

- First
- Second
  - Third

```

1 - First
2 - Second
3 - Third

```

- First
- Second
- Third

```

1 - First
2 - Second
3 - Third

```

- First
- Second
- Third

```

1 list(
2   [First],
3   [Second],
4   [Third],
5 )

```

1. First
  1. Second
2. Third

Text  
4. Fourth  
5. Fifth

```

1 + First
2 + Second
3 + Third
4 Text
5 4. Fourth
6 + Fifth

```

1. First
    - a) Second
  2. Third
- Text  
4. Fourth  
5. Fifth

```

1 + First
2 #set enum(numbering: "a")
3 + Second
4 + Third
5 Text
6 4. Fourth
7 + Fifth

```

### 4.3 Custom Lists

```
1 #import "/00-templates/items.typ": *
```

- item-list
- item-checkbadge
- item-circle
- item-square
- item-checkcircle
- item-checksquare
- item-check
- item-file
- item-folder
- item-xcircle
- item-xsquare
- item-x

```

1 #item-list()[item-list]
2 #item-checkbadge()[item-checkbadge]
3 #item-circle()[item-circle]
4 #item-square()[item-square]
5 #item-checkcircle()[item-checkcircle]
6 #item-checksquare()[item-checksquare]
7 #item-check()[item-check]
8 #item-file()[item-file]
9 #item-folder()[item-folder]
10 #item-xcircle()[item-xcircle]
11 #item-xsquare()[item-xsquare]
12 #item-x()[item-x]

```

### 4.4 Minitoc

The **minitoc** is specific for this template and allows to show a TOC between two labels. The **minitoc** is also used within the **#add-chapter** function if **after** and **before** labels are defined.

## Contents

---

4.1	Headings .....	10
4.2	Lists .....	10
4.3	Custom Lists .....	11
4.4	Minitoc .....	11
4.5	Images .....	12
4.5.1	Alignment .....	12
4.5.2	Caption .....	13
4.5.3	Cluster .....	13
4.6	Tables .....	15
4.7	Icon Boxes .....	18
4.8	Color Boxes .....	19
4.8.1	Todo Box .....	20
4.8.2	Option Style .....	20
4.9	Title Box .....	21
4.10	Word/Character Count .....	21

---

```
1 #minitoc(after: <sec:elem>, before: <sec:ref>)
```

```
1 #add-chapter(
2   "/02-main/03-elements.typ",
3   after: <sec:elem>,
4   before: <sec:ref>,
5 )
6 // or
7 #add-chapter(
8   after: <sec:elem>,
9   before: <sec:ref>,
10 )[
11   Content of the Chapter
12 ]
```

## 4.5 Images

### 4.5.1 Alignment

left



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

center



```

1 #align(center,
2   image("/04-resources/icon.svg",
3     width: 2cm)
4 )

```

right



```

1 #align(right,
2   image("/04-resources/icon.svg",
3     width: 2cm)
4 )

```

#### 4.5.2 Caption



Figure 1 - One image one caption

```

1 #figure(
2   image("04-resources/icon.svg",
3     width: 2cm),
4   caption: [One image one caption]
5 ) <fig:icon>

```

#### 4.5.3 Cluster

Two images one caption

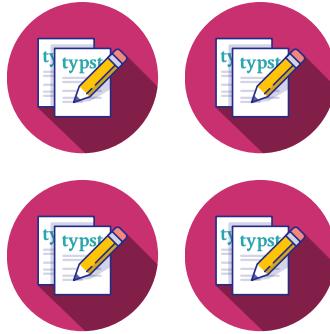
Table 1 - Multiple images **one** caption

```

1 #figure(
2   table(
3     columns: 2,
4     stroke: none,
5     align: center + horizon,
6     image(icon, width: 2cm),image(icon, width: 2cm)
7   ),
8   caption: [Multiple images *one* caption]
9 )

```

Four images one caption

Table 2 - Multiple images **one** caption

```

1 #figure(
2   table(
3     columns: 2,
4     stroke: none,
5     align: center + horizon,
6     image(icon, width: 2cm), image(icon, width: 2cm),
7     image(icon, width: 2cm), image(icon, width: 2cm),
8   ),
9   caption: [Multiple images *one* caption]
10 )

```

Two images two caption



Figure 2 - Caption left image   Figure 3 - Caption right image

```

1 #align(center,
2   table(
3     columns: 2,
4     stroke: none,
5     align: center + horizon,
6     figure(image(icon, width: 2cm), caption: [Caption left image]),
7     figure(image(icon, width: 2cm), caption: [Caption right image]),
8   ))

```

Four images four caption



Figure 4 - Caption topleft image



Figure 5 - Caption topright image



Figure 6 - Caption bottomleft image



Figure 7 - Caption bottomright image

```

1 #align(center,
2   table(
3     columns: 2,
4     stroke: none,
5     align: center + horizon,
6     figure(image(icon, width: 2cm), caption: [Caption topleft image]),
7     figure(image(icon, width: 2cm), caption: [Caption topright image]),
8     figure(image(icon, width: 2cm), caption: [Caption bottomleft image]),
9     figure(image(icon, width: 2cm), caption: [Caption bottomright image]),
10   ))

```

## 4.6 Tables

For new table use the integrated **#table** command for legacy the **tablex** plugin is also imported.

```

1 #import "@preview/tablex:0.0.9" :
2 // or
3 #import "/00-templates/helpers.typ": *

```

Tables with and without caption

	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 3 - Table caption

```

1 table(
2   columns: 3,
3   align: center + horizon,
4   table.header([], [*Col1*] ,
5   [*Col2*],),
6   [*Row1*], "cell-0-0", "cell-1-0",
7   [*Row2*], "cell-0-1", "cell-1-1",
8 )

```

```

1 figure(
2   table(
3     columns: 3,
4     align: center + horizon,
5     table.header([], [*Col1*] ,
6     [*Col2*],),
7     [*Row1*], "cell-0-0", "cell-1-0",
8     [*Row2*], "cell-0-1", "cell-1-1",
9   ),
10   kind: table,
11   caption: [Table Caption]
12 )

```

Tables with cell spans

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

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

```

1 table(
2   columns: 3,
3   align: center + horizon,
4   table.header([], [*Col1*] ,
5   [*Col2*],),
6   [*Row1*], table.cell(rowspan: 2)
7 [cell-0], "cell-1-0",
8   [*Row2*],
9   "cell-1-1",
10 )

```

```

1 table(
2   columns: 3,
3   align: center + horizon,
4   table.header([], [*Col1*] ,
5   [*Col2*],),
6   [*Row1*], table.cell(colspan: 2)
7 [cell-0],
8   [*Row2*], "cell-0-1", "cell-1-1",
9 )

```

## Table Design

	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

```

1 table(
2   columns: 3,
3   align: center + horizon,
4   table.header([], [*Col1*] ,
5   [*Col2*],),
6   [*Row1*], "cell-0-0", "cell-1-0",
7   [*Row2*], "cell-0-1", "cell-1-1",
8 )

```

```

1 table(
2   columns: 3,
3   align: center + horizon,
4   table.header([], [*Col1*] ,
5   [*Col2*],),
6   [*Row1*], "cell-0-0", "cell-1-0",
7   [*Row2*], "cell-0-1", "cell-1-1",
8 )

```

	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

```

1 table(
2   columns: 3,
3   align: center + horizon,
4   table.vline(x:0, stroke: none),
5   table.vline(x:1 , stroke: blue),
6   table.vline(x:2),
7   table.header([], [*Col1*] ,
8   [*Col2*], table.hline(stroke: red)),
9   [*Row1*], "cell-0-0", "cell-1-0",
10  table.hline(),
11   [*Row2*], "cell-0-1", "cell-1-1",
12 )

```

```

1 table(
2   columns: 3,
3   align: center + horizon,
4   stroke: (x:none),
5   table.header([], [*Col1*] ,
6   [*Col2*],),
7   [*Row1*], "cell-0-0", "cell-1-0",
8   [*Row2*], "cell-0-1", "cell-1-1",
9 )

```

	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

```

1 table(
2   columns: 3,
3   align: center + horizon,
4   stroke: (y:none),
5   table.header([], [*Col1*] ,
6   [*Col2*],),
7   [*Row1*], "cell-0-0", "cell-1-0",
8   [*Row2*], "cell-0-1", "cell-1-1",
8 )

```

```

1 table(
2   columns: 3,
3   align: center + horizon,
4   stroke: (x:none),
5   table.hline(y:0, stroke:none), //
6   remove first line
7   table.hline(y:3, stroke:none), //
8   remove last line
9   table.vline(x:1),
10  table.vline(x:2, start:1, end:2,
11  stroke: red),
12  table.header([], [*Col1*] ,
13  [*Col2*],),
14  [*Row1*], "cell-0-0", "cell-1-0",
15  [*Row2*], "cell-0-1", "cell-1-1",
16 )

```

```

1 #table(
2   columns: 3,
3   align: center + horizon,
4   table.vline(x:0, stroke: none), table.vline(x:1 , stroke: blue), table.vline(x:2),
5   table.header([], [*Col1*] , [*Col2*], table.hline(stroke: red)),
6   [*Row1*], "cell-0-0", "cell-1-0", table.hline(),
7   [*Row2*], "cell-0-1", "cell-1-1",
8 )

```

c	b	a	cb	ba	y
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	0	0
0	1	1	0	1	0
1	0	0	0	0	0
1	0	1	0	0	1
1	1	0	1	0	1
1	1	1	1	1	1

```

1 #table(
2   columns: 6,
3   stroke: none,
4   align: center+ horizon,
5   table.vline(x:1, stroke:0.5pt), table.vline(x:2, stroke:0.5pt), table.vline(x:3),
6   table.vline(x:4, stroke:0.5pt), table.vline(x:5, stroke:0.5pt),

```

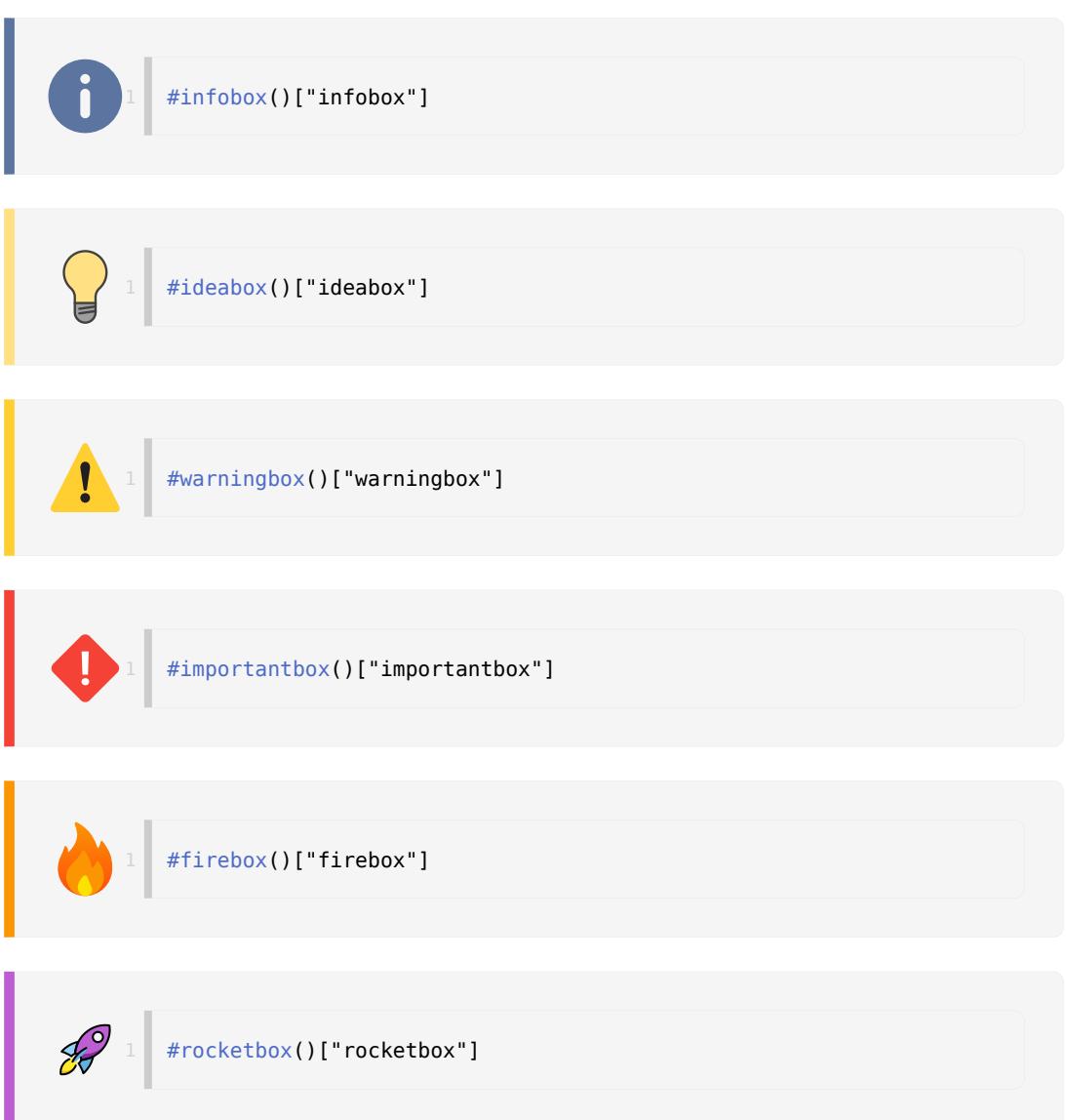
```

6   table.header([$c$], [$b$], [$a$], [$c b$], [$b a$], [$y$], table.hline(stroke:
7     1pt)),
8     [``0`], [``0`], [``0`], [``0`], [``0`], table.hline(stroke: 0.5pt),
9     [``0`], [``0`], [``1`], [``0`], [``0`], [``1`], table.hline(stroke: 0.5pt),
10    [``0`], [``1`], [``0`], [``0`], [``0`], [``0`], table.hline(stroke: 0.5pt),
11    [``0`], [``1`], [``1`], [``0`], [``1`], [``1`], [``0`], table.hline(stroke: 1pt),
12    [``1`], [``0`], [``0`], [``0`], [``0`], [``0`], [``0`], table.hline(stroke: 0.5pt),
13    [``1`], [``1`], [``0`], [``1`], [``0`], [``1`], [``1`], table.hline(stroke: 0.5pt),
14    [``1`], [``1`], [``1`], [``1`], [``1`], [``1`], [``1`],
15  )

```

## 4.7 Icon Boxes

```
1 #import "/00-templates/boxes.typ": *
```





## 4.8 Color Boxes

```
1 #import "/00-templates/boxes.typ": *
```

### Exercise

Some text

```
1 #colorbox(title: "Exercise", color: hei-blue)[Some text]
```

### Attention

Some text

```
1 #colorbox(title: "Attention", color: hei-pink)[Some text]
```

### Consider

Some text

```
1 #slanted-colorbox(title: "Consider", color: hei-green)[Some text]
```

## Information

Some text

```
1 #slanted-colorbox(title: "Information", color: hei-orange)[Some text]
```

### 4.8.1 Todo Box

```
1 #import "/00-templates/boxes.typ": *
```

**TODO**

This is not finished

```
1 #todo("This is not finished")
```

### 4.8.2 Option Style

The option style allows to underlight a text depending on the type or state of the document. Within the `/01-settings/metadata.typ` the `option.type` can be set to `draft` or `final` or other types.

```
1 #import "/00-templates/boxes.typ": *
```

*"This text has an option style and is shown in the case the type is draft"*

```
1 #option-style(type: "draft")["This text has an option style and is shown in the case  
the type is draft"]
```

```
1 #option-style(type: "final")["This text has an option style and is shown in the case  
the type is final"]
```

```
1 #option-style(type: "minimal")["This text has an option style and is shown in the  
case the type is minimal"]
```

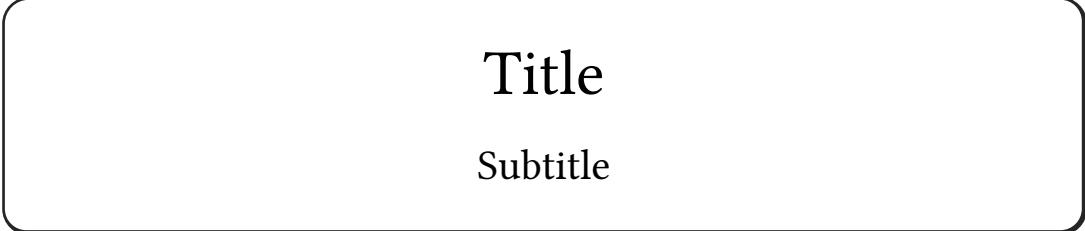
```
1 #option-style(type: "full")["This text has an option style and is shown in the case  
the type is full"]
```

```
1 #option-style(type: "student")["This text has an option style and is shown in the  
case the type is student"]
```

```
1 #option-style(type:"solution")["This text has an option style and is shown in the  
case the type is solution"]
```

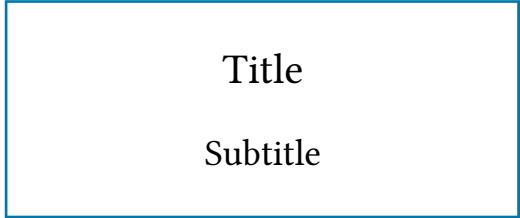
## 4.9 Title Box

```
1 #import "/00-templates/sections.typ": *
```



Title  
Subtitle

```
1 #titlebox(title: [Title], subtitle: [Subtitle])
```



Title  
Subtitle

```
1 #titlebox(width: 50%, radius: 0pt, border: 1pt, linecolor: hei-blue, titlesize:  
larger, subtitlesize: large, title: [Title], subtitle: [Subtitle])
```



Title

```
1 #titlebox(linecolor: hei-green, titlesize: larger, subtitlesize: large, title:  
[Title])
```

## 4.10 Word/Character Count

With the plugin wordometer it is possible to count the words and characters of a text.

```
1 #import "@preview/wordometer:0.1.4": word-count
```

*The following section contains 50 Words and 295 Characters without counting this text.*

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliquam quaerat voluptatem. Ut enim aequo doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

```
1 #word-count(total => [
2   #[The following section contains #total.words Words and #total.characters
3   Characters without counting this text.]<no-wc>
4   #lorem(50)
5 ], exclude: <no-wc>)
```



The character count does not include spaces.

# 5 | References

## 5.1 Links

Example	Raw
<a href="https://tschinz.github.io">https://tschinz.github.io</a>	<code>https://tschinz.github.io</code>
<a href="https://tschinz.github.io">https://tschinz.github.io</a>	<code>#link("https://tschinz.github.io")</code>
See <a href="https://tschinz.github.io">https://tschinz.github.io</a>	<code>#link("https://tschinz.github.io") [See https://tschinz.github.io]</code>
<a href="mailto:silvan.zahno@hevs.ch">silvan.zahno@hevs.ch</a>	<code>#link("mailto:silvan.zahno@hevs.com") [silvan.zahno@hevs.ch]</code>
	<code>#link("https://tschinz.github.io") [#image(icon, width:0.5cm)]</code>

Table 4 - Links

## 5.2 Crossreferences

In the document the following references were added.

```

1 = References <sec:ref>
2 == Links <sec:links>
3 #figure(image("/04-resources/icon.svg", width: 2cm)) <fig:icon>
4 #figure(table(...), kind:table) <tab:links>
5 #figure(align(left, raw(...))) <code-ref>
6 $ sum_(k=1)^n k = (n(n+1)) / 2 $ <math-eql> #ref(<math-eql>)

```

Listing 1 - Label inserts

They can be references as follows:

Type	Example	Raw
Section	Section 5	<code>@sec:ref</code>
Subsection	Section 5.1	<code>@sec:links</code>
Figure	Figure 1	<code>@fig:icon</code>
Table	Table 4	<code>@tab:links</code>
Code	Listing 1	<code>@code-ref</code>
Equation	Equation 1	<code>@math-eql</code>

## 5.3 External References

Example	Raw
[1]	<code>#cite(label("zahnoDynamicProjectPlanning2023"))</code>
[1, p.7ff]	<code>#cite(&lt;zahnoDynamicProjectPlanning2023&gt;, supplement:[p.7ff])</code>
[1]	<code>@zahnoDynamicProjectPlanning2023</code>

## 5.4 Glossary

The glossary entries need to be defined in `/03-tail/glossary.typ`.

```

1 #let entry-list = (
2   (
3     key    : "hei",
4     short  : "HEI",
5     long   : "Haute École d'Ingénierie",
6   ),
7   (
8     key      : "fpga",
9     short    : "FPGA",
10    plural   : "FPGAs",
11    long     : "Field Programmable Gate Array",
12    longplural: "Field Programmable Gate Arrays",
13    group    : "Technology"
14    description : "A field-programmable gate array (FPGA) is an integrated circuit
15    designed to be configured by a customer or a designer after manufacturing – hence
16    the term 'field-programmable'.",
17  ),
18)

```

For the glossary functions the “import” of `/00-templates/helpers.typ` is needed. Underneath it uses the glossarium plugin. It can also be used for acronyms.

```
1 #import "/00-templates/helpers.typ": *
```

### Example

Haute École d'Ingénierie (HEI) and a second time HEI

### Raw

```
1 #gls("hei") and a second time
#gls("hei")
```

Haute École d'Ingénierie (HEI)

```
1 #gls("hei", long:true)
```

whatever you want

```
1 #gls("hei", display: "whatever you
want"))
```

Field Programmable Gate Arrays (FPGAs) and a second time FPGAs

```
1 #glspl("fpga") and a second time
#glspl("fpga")
```

Field Programmable Gate Arrays (FPGAs)

```
1 #glspl("fpga", long: true))
```

[Infotonics \(IT\)](#) is a specialization of [Systems Engineering \(SYND\)](#) which is part of the [HEI](#). The second time a glossary entry is used the short form will be used: see [IT](#) and [SYND](#). To get the long form back use [Systems Engineering \(SYND\)](#).

# 6 | Code

There are two ways to include code in Typst: inline and block. For the block code you can use **codly** or **codebst**. **codly** retakes the traditional ````lang```` syntax while **codebst** uses a function taking the traditional syntax `#sourcecode[```lang```]`. Typst also provides the **raw** function to include code in the document. The **raw** function can be used for inline and block code.

## inline monospaced string

```
1 `inline monospaced string`
```

```
1 raw(lang:"rust",
2     "fn main() {println!(\"Hello world!\")}
3     )
```

```
1-- Test 2: INPUT sX, pp
2opCode <= "INPUT sX, pp      ";
3code   <= "00010";
4cIn    <= '0';
5A      <= "11110000";
6B      <= "00001111";
```

vhdl

```
1 raw(block:true, lang:"vhdl",
2     read("code-example.vhdl"))
3 )
```

```
1fn main() {
2 println!("Hello world!")
3 }
```

Rust

```
1 ````rust
2 fn main() {
3     println!("Hello world!")
4 }
5 ````
```

```
1 fn main() {
2     println!("Hello world!")
3 }
```

```
1 sourcecode[````rust
2 fn main() {
3     println!("Hello world!")
4 }
5 ````]
```

```
1fn main() {
2 println!("Hello world!")
3 }
```

Rust

```
1 #figure(
2     align(left,
3 ````rust
4         fn main() {
5             println!("Hello world!")
6         }
7     ````)
8     ,
9     caption: [Rust Code],
10 )
11 ````
```

Listing 2 - 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.

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

$$1 \quad \$\text{sum\_}(k=1)^n k = (n(n+1)) / 2$,$$

Fullline math

$$a^2 + b^2 = c^2 \quad (1)$$

$$1 \quad \$ a^2 + b^2 = c^2 \$ \text{ <math-eq1>}$$

Math with caption

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

Equation 8 - Some proof

```
1 #figure(
2   $ sum_(k=1)^n k = (n(n+1)) / 2 $,
3   caption: [Some proof]
4 )
```

## 7.1 Align

Formula

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

Raw

$$(3) \quad \begin{aligned} 1 &\$ \\ 2 &a_1 = b_1 + c_1 = z_1 \\ 3 &a_2 = b_2 + c_2 - d_2 + e_2 = z_1 \\ 4 &\$ \end{aligned}$$

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

$$(4) \quad \begin{aligned} 1 &\$ \\ 2 &a_1 &= b_1 + c_1 &= z_1 \\ 3 &a_2 &= b_2 + c_2 - d_2 + e_2 &= z_1 \\ 4 &\$ \end{aligned}$$

## 7.2 Symbols

This is an incomplete list for all symbols goto [here](#)

Outside of the `$$` math environment the symbols can be accessed with .

### 7.2.1 Accents

Sym	Raw	Sym	Raw	Sym	Raw
$\grave{x}$	<code>\$grave(x)\$</code>	$\acute{x}$	<code>\$acute(x)\$</code>	$\hat{x}$	<code>\$hat(x)\$</code>
$\tilde{x}$	<code>\$tilde(x)\$</code>	$\breve{x}$	<code>\$breve(x)\$</code>	$\dot{x}$	<code>\$dot(x)\$</code>
$\ddot{x}$	<code>\$dot.double(x)\$</code>	$\ddot{\cdot}$	<code>\$dot.triple(x)\$</code>	$\ddot{x}$	<code>\$dot.quad(x)\$</code>
$\ddot{x}$	<code>\$diaer(x)\$</code>	$\circledcirc{x}$	<code>\$circle(x)\$</code>	$\ddot{x}$	<code>\$acute.double(x)\$</code>
$\check{x}$	<code>\$caron(x)\$</code>	$\vec{x}$	<code>\$arrow(x)\$</code>	$\bar{x}$	<code>\$arrow.l(x)\$</code>
$\cancel{x}$	<code>\$cancel(x)\$</code>	$\bar{x}$	<code>\$macron(x)\$</code>	$\overline{xyz}$	<code>\$overline(xyz)\$</code>
$\underline{xyz}$	<code>\$overline(xyz)\$</code>	$\overbrace{xyz}$	<code>\$underbrace(xyz)\$</code>	$\overbrace{xyz}$	<code>\$overbrace(xyz)\$</code>
$\overline{xyz}$	<code>\$underbrace(xyz)\$</code>	$\overbrace{\overbrace{xyz}}$	<code>\$overbrace(xyz)\$</code>	$\overbrace{\overbrace{xyz}}$	<code>\$overbrace(xyz)\$</code>

### 7.2.2 Equals & Operators

Sym	Raw	Sym	Raw	Sym	Raw
$=$	<code>==\$</code>	$=$	<code>\$eq\$</code>	$\neq$	<code>\$eq.not\$</code>
$\neq$	<code>!=\$</code>	$\equiv$	<code>\$equiv\$</code>	$\not\equiv$	<code>\$equiv.not\$</code>
$\simeq$	<code>\$tilde.eq\$</code>	$\not\simeq$	<code>\$tilde.eq.not\$</code>	$\approx$	<code>\$eq.small\$</code>
$\geq$	<code>\$gt.eq\$</code>	$\ngeq$	<code>\$gt.eq.not\$</code>	$\leq$	<code>\$lt.eq\$</code>
$\nleq$	<code>\$lt.eq.not\$</code>	$\approx$	<code>\$approx\$</code>	$\approx$	<code>\$approx.eq\$</code>
$\approx$	<code>\$approx.not\$</code>	$:$	<code>\$colon\$</code>	$::=$	<code>\$colon.eq\$</code>
$=:$	<code>\$eq.colon\$</code>	$::=$	<code>\$colon.double.eq\$</code>	$+$	<code>+\$</code>
$+$	<code>\$plus\$</code>	$+$	<code>\$plus.small\$</code>	$\pm$	<code>\$plus_MINUS\$</code>
$\oplus$	<code>\$plus.circle\$</code>	$-$	<code>\$minus\$</code>	$-$	<code>\$minus\$</code>
$\mp$	<code>\$minus.plus\$</code>	$\ominus$	<code>\$minus.circle\$</code>		

### 7.2.3 Scripts

Sym	Raw	Sym	Raw	Sym	Raw
$x_1$	<code>\$x_1\$</code>	$x_{12}$	<code>\$x_(12)\$</code>	$x_1$	<code>\$scripts(x)_1\$</code>
$x_1$	<code>\$x_1\$</code>	$x_{12}$	<code>\$x_(12)\$</code>	$x_1$	<code>\$scripts(x)_1\$</code>
$x_1^2$	<code>\$x_1^2\$</code>	$x_{12}^{34}$	<code>\$x_(12)^{(34)}\$</code>	$x_1^2$	<code>\$scripts(x)_1^2\$</code>
$x_1^2$	<code>\$x_1^2\$</code>	$x_{12}^{34}$	<code>\$x_(12)^{(34)}\$</code>	$x_1^2$	<code>\$scripts(x)_1^2\$</code>

### 7.2.4 Special Elements

Sym	Raw	Sym	Raw
$\binom{n}{k}$	(5) <code>\$ binom(n, k) \$</code>	$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$	(6) <code>\$ vec(1, 2, delim: "[" ) \$</code>
$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$	(7) <code>\$ round(1, 2) \$</code>	$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$	(8) <code>\$ mat(1,2; 3,4) \$</code>

```

1 $ mat(
2   1, 2, ..., 10;
3   2, 2, ..., 10;
4   dots.v,
5   dots.down,
6   dots.v;
7   10, 10, ...
8   10;
9 ) $

```

$$\sum a_k \quad (10) \quad \$ \text{ sum } a_k \$$$

$$\sum_{k=0}^n a_k \quad (11) \quad \$ \text{ sum_(k=0)^n } a_k \$$$

$$\sum_{k=0}^n a_k \quad (12) \quad \$ \text{ scripts(sum)__(k=0)^n } a_k \$$$

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

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

```

1 $ f(x, y) := cases(
2   1 "if" (x dot y)/2 <= 0,
3   2 "if" x "is even",
4   3 "if" x in NN,
5   4 "else",
6 ) $

```

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

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

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

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

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

$$n! = \prod_{k=1}^n k \quad (20) \quad \$ \text{ product_(k=1)^n } k \$$$

$$n! = \prod_{k=1}^n k \quad \$ n! = \text{ scripts(product)__(k=1)^n } k \$$$

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

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

## 7.2.5 Alphabeth

### Sym

$\alpha \beta \gamma \delta \varepsilon \zeta \eta \theta \iota \kappa \lambda \mu \nu \xi \circ \pi \rho \sigma \tau \upsilon \varphi \chi \psi \omega$

ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ

### Raw

```
$alpha beta gamma delta epsilon zeta
eta theta iota kappa lambda mu nu xi
omicron pi rho sigma tau upsilon phi
chi psi omega$
```

```
$Alpha Beta Gamma Delta Epsilon Zeta
Eta Theta Iota Kappa Lambda Mu Nu Xi
Omicron Pi Rho Sigma Tau Upsilon Phi
Chi Psi Omega$
```

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

Sym	Raw	Sym	Raw	Sym	Raw
$\wedge$	\$and\$	$\wedge$	\$and.big\$	$\&$	\$amp\$
$\vee$	\$or\$	$ $	\$bar.v\$	$*$	\$ast.op\$
$*$	\$ast.basic\$	$*$	\$ast.low\$	$\oplus$	\$plus.circle\$
$\oplus$	\$plus.circle.big\$				

### 7.2.7 Operators

Sym	Raw	Sym	Raw	Sym	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\$	$\rightarrow$ \$arrow.long\$	$\rightarrowtail$ \$arrow.bar\$
$\rightarrowtail$ \$arrow.bar.long\$	$\Rightarrow$ \$arrow.double\$	$\Longrightarrow$ \$arrow.double.long\$
$\Rightarrowtail$ \$arrow.double.bar\$	$\Longrightarrowtail$ \$arrow.double.bar.long\$	$\Rightarrowtail$ \$arrow.quad\$
$\Rightarrowtail$ \$arrow.stroked\$	$\rightarrowtail$ \$arrow.filled\$	$\dashrightarrowtail$ \$arrow.dashed\$
$\looparrowright$ \$arrow.curve\$	$\rightsquigarrowtail$ \$arrow.squiggly\$	$\looparrowrighttail$ \$arrow.loop\$
Arrows left		
$\leftarrow$ \$arrow.l\$	$\leftarrow$ \$arrow.l.long\$	$\leftarrowtail$ \$arrow.l.bar\$
$\leftarrowtail$ \$arrow.l.bar.long\$	$\Leftarrowtail$ \$arrow.l.double\$	$\Leftarrowtail$ \$arrow.l.double.long\$
$\Leftarrowtail$ \$arrow.l.double.bar\$	$\Leftarrowtail$ \$arrow.l.double.bar.long\$	$\Leftarrowtail$ \$arrow.l.quad\$
$\Leftarrowtail$ \$arrow.l.stroked\$	$\leftarrowtail$ \$arrow.l.filled\$	$\Leftarrowtail$ \$arrow.l.dashed\$
$\looparrowleft$ \$arrow.l.curve\$	$\rightsquigleftarrowtail$ \$arrow.l.squiggly\$	$\looparrowlefttail$ \$arrow.l.loop\$
Double Arrows Left Right		
$\leftrightarrow$ \$arrow.l.r\$	$\leftrightarrowtail$ \$arrow.l.r.not\$	$\leftrightarrowtail$ \$arrow.l.r.long\$

$\Leftrightarrow$	<code>\$arrow.l.r.double\$</code>	$\Leftrightarrow$	<code>\$arrow.l.r.double.long\$</code>	$\Leftrightarrow$	<code>\$arrow.l.r.double.not\$</code>
$\Leftrightarrow$	<code>\$arrow.l.r.stroked\$</code>	$\Leftrightarrow$	<code>\$arrow.l.r.filled\$</code>	$\Leftrightarrow$	<code>\$arrow.l.r.wave\$</code>
<b>Arrows Top</b>					
$\uparrow$	<code>\$arrow.t\$</code>	$\uparrow$	<code>\$arrow.t.bar\$</code>	$\uparrow$	<code>\$arrow.t.double\$</code>
$\nwarrow$	<code>\$arrow.t.triple\$</code>	$\nwarrow$	<code>\$arrow.t.quad\$</code>	$\uparrow$	<code>\$arrow.t.stroked\$</code>
$\uparrow$	<code>\$arrow.t.filled\$</code>	$\uparrow$	<code>\$arrow.t.dashed\$</code>	$\rightarrow$	<code>\$arrow.t.curve\$</code>
<b>Arrows Bottom</b>					
$\downarrow$	<code>\$arrow.b\$</code>	$\downarrow$	<code>\$arrow.b.bar\$</code>	$\downarrow$	<code>\$arrow.b.double\$</code>
$\ddownarrow$	<code>\$arrow.b.triple\$</code>	$\ddownarrow$	<code>\$arrow.b.quad\$</code>	$\downarrow$	<code>\$arrow.b.stroked\$</code>
$\downarrow$	<code>\$arrow.b.filled\$</code>	$\downarrow$	<code>\$arrow.b.dashed\$</code>	$\searrow$	<code>\$arrow.b.curve\$</code>
<b>Double Arrows Top Bottom</b>					
$\Downarrow$	<code>\$arrow.t.b\$</code>	$\Downarrow$	<code>\$arrow.t.b.double\$</code>	$\Downarrow$	<code>\$arrow.t.b.stroked\$</code>
$\Downarrow$	<code>\$arrow.t.b.filled\$</code>				
<b>Arrows Diagonal Top Right</b>					
$\nearrow$	<code>\$arrow.tr\$</code>	$\nearrow$	<code>\$arrow.tr.double\$</code>	$\nearrow$	<code>\$arrow.tr.stroked\$</code>
$\nearrow$	<code>\$arrow.tr.filled\$</code>	$\nearrow$	<code>\$arrow.tr.hook\$</code>		
<b>Arrows Diagonal Bottom Right</b>					
$\searrow$	<code>\$arrow.br\$</code>	$\searrow$	<code>\$arrow.br.double\$</code>	$\searrow$	<code>\$arrow.br.stroked\$</code>
$\searrow$	<code>\$arrow.br.filled\$</code>	$\searrow$	<code>\$arrow.br.hook\$</code>		
<b>Arrows Diagonal Bottom Left</b>					
$\swarrow$	<code>\$arrow.bl\$</code>	$\swarrow$	<code>\$arrow.bl.double\$</code>	$\swarrow$	<code>\$arrow.bl.stroked\$</code>
$\swarrow$	<code>\$arrow.bl.filled\$</code>	$\swarrow$	<code>\$arrow.bl.hook\$</code>		
<b>Arrows Diagonal Top Left</b>					
$\nwarrow$	<code>\$arrow.tl\$</code>	$\nwarrow$	<code>\$arrow.tl.double\$</code>	$\nwarrow$	<code>\$arrow.tl.stroked\$</code>
$\nwarrow$	<code>\$arrow.tl.filled\$</code>	$\nwarrow$	<code>\$arrow.tl.hook\$</code>		
<b>Double Arrows Diagonal</b>					
$\nwarrow$	<code>\$arrow.tl.br\$</code>	$\nwarrow$	<code>\$arrow.tr.bl\$</code>		
<b>Other Arrows</b>					
$\circlearrowleft$	<code>\$arrow.cw\$</code>	$\curvearrowleft$	<code>\$arrow.cw.half\$</code>	$\circlearrowright$	<code>\$arrow.ccw\$</code>
$\curvearrowleft$	<code>\$arrow.ccw.half\$</code>				

### 7.2.9 Angles

Sym	Raw	Sym	Raw	Sym	Raw
$\angle$	<code>\$angle\$</code>	$\triangleright$	<code>\$angle.rev\$</code>	$\prec$	<code>\$angle.acute\$</code>
$\angle$	<code>\$angle.acute\$</code>	$\triangleleft$	<code>\$angle.arc\$</code>	$\succ$	<code>\$angle.arc.rev\$</code>
$\langle$	<code>\$angle.l\$</code>	$\rangle$	<code>\$angle.r\$</code>	$\langle\!\langle$	<code>\$angle.l.double\$</code>
$\rangle\!\rangle$	<code>\$angle.r.double\$</code>	$\llcorner$	<code>\$angle.right\$</code>	$\curlyeqsucc$	<code>\$angle.right.rev\$</code>

$\triangleleft$	<code>\$angle.right.arc\$</code>	$\triangleleft$	<code>\$angle.right.dot\$</code>	$\triangleleft$	<code>\$angle.right.sq\$</code>
$\triangleleft$	<code>\$angle.spheric\$</code>	$\triangleright$	<code>\$angle.spheric.rev\$</code>	$\triangleright$	<code>\$angle.spheric.top\$</code>

## 7.2.10 Cool Symbols

Sym	Raw	Sym	Raw	Sym	Raw
$\text{@}$	<code>\$at\$</code>	$\%$	<code>\$co\$</code>	$\textcircled{C}$	<code>\$copyright\$</code>
$\textcircled{P}$	<code>\$copyright.sound\$</code>	$^{\circ}C$	<code>\$degree C\$</code>	$\textcircled{E}$	<code>\$euro\$</code>
$\text{\$}$	<code>\$dollar\$</code>	$\text{\textsterling}$	<code>\$pound\$</code>	$\text{\textwronski}$	<code>\$won\$</code>
$\text{\textyen}$	<code>\$yen\$</code>	$\text{\texteuro}$	<code>\$bitcoin\$</code>	${}^{\circ}F$	<code>\$degree F\$</code>
$!$	<code>\$excl\$</code>	$\text{\textexcl}$	<code>\$excl.inv\$</code>	$!!$	<code>\$excl.double\$</code>
$!?$	<code>\$excl.quest\$</code>	$\text{\textarrowleft}$	<code>\$arrow.zigzag\$</code>	$\text{\textcircledast}$	<code>\$ast.circle\$</code>
$^{**}$	<code>\$ast.triple\$</code>	$\chi$	<code>\$chi\$</code>	$\text{\textcircled{?}}$	<code>\$floral\$</code>
$\text{\textmaltese}$	<code>\$maltese\$</code>	$\text{\textpilcrow}$	<code>\$pilcrow\$</code>	$h$	<code>\$planck\$</code>
$\text{\textclubsuit}$	<code>\$suit.club\$</code>	$\text{\textdiamondsuit}$	<code>\$suit.diamond\$</code>	$\text{\textheartsuit}$	<code>\$suit.heart\$</code>
$\text{\textspadesuit}$	<code>\$suit.spade\$</code>	$\text{\texttriangleleft}$	<code>\$triangle.stroked.nested2\$</code>		

## 7.2.11 Style

Sym	Raw	Sym	Raw
$ABC123$	<code>\$sans(A B C 1 2 3)\$</code>	$\mathfrak{ABC}123$	<code>\$frak(A B C 1 2 3)\$</code>
$ABC123$	<code>\$mono(A B C 1 2 3)\$</code>	$\mathbb{ABC}123$	<code>\$bb(A B C 1 2 3)\$</code>
$\mathcal{ABC}123$	<code>\$cal(A B C 1 2 3)\$</code>		

Sym	Raw
$\sum_{i \in \mathbb{N}} 1 + i$	

```
1 #show math.equation: set text(font: "Cambria Math")
2 $sum_(i in NN) 1 + i$,
```

# 8 | Emoji Symbols

This is an incomplete list for all emoji goto [here](#)

If the emoji module is imported the `#emoji` can be removed

```
1 #import emoji: *
```

Sym Raw	Sym Raw	Sym Raw
Smileys		
😊 #emoji.face	👶 #emoji.baby	👽 #emoji.alien
👽 #emoji.alien.monster	👽 #emoji.alien.monster	👁 #emoji.eye
👀 #emoji.eyes	😁 #emoji.face.grin	😡 #emoji.face.angry
😡 #emoji.face.angry.red	Ө #emoji.face.anguish	😯 #emoji.face.astonish
🤕 #emoji.face.bandage	😁 #emoji.face.beam	😌 #emoji.face.concern
😎 #emoji.face.cool	🤗 #emoji.face.cover	😔 #emoji.face.down
😓 #emoji.face.down.sweat	🤤 #emoji.face.drool	🤯 #emoji.face.explode
🙄 #emoji.face.eyeroll	😊 #emoji.face.friendly	😱 #emoji.face.fear
😅 #emoji.face.fear.sweat	🤒 #emoji.face.fever	😳 #emoji.face.flush
光环 #emoji.face.halo	😊 #emoji.face.happy	😍 #emoji.face.heart
😘 #emoji.face.hearts	🥵 #emoji.face.heat	😘 #emoji.face.kiss
😘 #emoji.face.kiss.heart	😘 #emoji.face.kiss.blush	😂 #emoji.face.joy
🤥 #emoji.face.lie	😷 #emoji.face.mask	😐 #emoji.face.meh
🥰 #emoji.face.melt	😐 #emoji.face.neutral	🥳 #emoji.face.party
😜 #emoji.face.peek	😉 #emoji.face.plead	😌 #emoji.face.relief
🤮 #emoji.face.vomit	🙁 #emoji.face.unhappy	😮 #emoji.face.wow
😪 #emoji.face.yawn	😉 #emoji.face.wink	🤐 #emoji.face.zip
👉 #emoji.finger.l	👆 #emoji.finger.t	👉 #emoji.finger.b
👆 #emoji.finger.m	👊 #emoji.finger.front	👉 #emoji.fingers.cross
👉 #emoji.fingers.pinch	👉 #emoji.fingers.snap	👉 #emoji.fist.front
👉 #emoji.fist.r	👉 #emoji.fist.l	👉 #emoji.fist.raised
👓 #emoji.glasses	🕶 #emoji.glasses.sun	✋ #emoji.hand.raised
👋 #emoji.hand.r	👋 #emoji.hand.l	👋 #emoji.hand.t

**Sym Raw**

👉	#emoji.hand.b
🕒	#emoji.hand.love
👉	#emoji.hand.pinch
✍	#emoji.hand.write
Ձ	#emoji.hands.heart

**Sym Raw**

👌	#emoji.hand.ok
✋	#emoji.hand.part
🤘	#emoji.hand.rock
🙏	#emoji.hands.folded
🤝	#emoji.hands.shake

**Sym Raw**

👍	#emoji.hand.call
✌	#emoji.hand.peace
👋	#emoji.hand.wave
👏	#emoji.hands.clap
🖐	#emoji.hand.splay

**Drinks****Sym Raw**

🍺	#emoji.beer
☕	#emoji.coffee
🥂	#emoji.glass.clink
🍵	#emoji.teacup

**Sym Raw**

🍻	#emoji.beer.clink
🥤	#emoji.cup.straw
🥛	#emoji.glass.milk
🍹	#emoji.mate

**Sym Raw**

🍹	#emoji.cocktail.tropical
🥃	#emoji.glass.tumbler
潽	#emoji.teapot
🥣	#emoji.soup

**Animals****Sym Raw**

🐜	#emoji.ant
🐞	#emoji.beetle
🐞	#emoji.beetle.lady
🐂	#emoji.bison
🦋	#emoji.butterfly
🐱	#emoji.cat
🐿	#emoji.chipmunk
🐊	#emoji.crocodile
🐦	#emoji.dodo
🦆	#emoji.duck
🧚	#emoji.fairy
🦩	#emoji.flamingo
🦒	#emoji.giraffe

**Sym Raw**

hog	#emoji.badger
🐝	#emoji.bee
鼫	#emoji.beaver
🐗	#emoji.boar
🐪	#emoji.camel
🐓	#emoji.chicken
蜚	#emoji.cockroach
🦕	#emoji.dino.pod
🐕	#emoji.dog
🥚	#emoji.egg
🦚	#emoji.feather
🦗	#emoji.fly
🦔	#emoji.hedgehog

**Sym Raw**

🦇	#emoji.bat
🐞	#emoji.beetle
🐻	#emoji.bear
🐛	#emoji.bug
🐪	#emoji.camel.dromedar
🐔	#emoji.chicken.male
🦗	#emoji.cricket
🦖	#emoji.dino.rex
🐬	#emoji.dolphin
🐘	#emoji.elephant
🐟	#emoji.fish
🐐	#emoji.goat
ippo	#emoji.hippo

**Flowers****Sym Raw**

⚜	#emoji.fleur
🌸	#emoji.flower.pink

**Sym Raw**

🌺	#emoji.flower.hibiscus
🌹	#emoji.flower.rose

**Sym Raw**

%;"	#emoji.flower.lotus
☀️	#emoji.flower.sun

<b>Sym</b>	<b>Raw</b>	<b>Sym</b>	<b>Raw</b>	<b>Sym</b>	<b>Raw</b>
	#emoji.flower.wilted		#emoji.flower.yellow		#emoji.plant
<b>Food</b>					
<b>Sym</b>	<b>Raw</b>	<b>Sym</b>	<b>Raw</b>	<b>Sym</b>	<b>Raw</b>
	#emoji.avocado		#emoji.aubergine		#emoji.banana
	#emoji.broccoli		#emoji.burger		#emoji.cake.slice
	#emoji.carrot		#emoji.chocolate		#emoji.corn
	#emoji.cucumber		#emoji.fries		#emoji.honey
<b>Objects</b>					
<b>Sym</b>	<b>Raw</b>	<b>Sym</b>	<b>Raw</b>	<b>Sym</b>	<b>Raw</b>
	#emoji.bin		#emoji.bomb		#emoji.bone
	#emoji.books		#emoji.book		#emoji.book.open
	#emoji.bell		#emoji.bell.not		#emoji.briefcase
	#emoji.broom		#emoji.brush		#emoji.bubbles
	#emoji.cabinet.file		#emoji.chair		#emoji.coffin
	#emoji.compass		#emoji.comet		#emoji.hook
<b>Sport</b>					
<b>Sym</b>	<b>Raw</b>	<b>Sym</b>	<b>Raw</b>	<b>Sym</b>	<b>Raw</b>
	#emoji.baseball		#emoji.billiards		#emoji.boxing
	#emoji.chess		#emoji.climbing		#emoji.football
<b>Vehicles</b>					
<b>Sym</b>	<b>Raw</b>	<b>Sym</b>	<b>Raw</b>	<b>Sym</b>	<b>Raw</b>
	#emoji.airplane		#emoji.bike		#emoji.boat
	#emoji.boat.sail		#emoji.boat.speed		#emoji.cablecar
	#emoji.car		#emoji.car.front		#emoji.car.racing
	#emoji.ship		#emoji.motorcycle		#emoji.wheelchair.motor
<b>Building</b>					
<b>Sym</b>	<b>Raw</b>	<b>Sym</b>	<b>Raw</b>	<b>Sym</b>	<b>Raw</b>
	#emoji.castle.eu		#emoji.castle.jp		#emoji.circus
	#emoji.factory		#emoji.hospital		#emoji.office
	#emoji.house		#emoji.museum		#emoji.crane
<b>Weather</b>					
<b>Sym</b>	<b>Raw</b>	<b>Sym</b>	<b>Raw</b>	<b>Sym</b>	<b>Raw</b>

**Sym Raw**
 `#emoji.cloud`
 `#emoji.cloud.snow`
 `#emoji.cloud.sun.hidden`
 `#emoji.drop`
**Sym Raw**
 `#emoji.cloud.dust`
 `#emoji.cloud.storm`
 `#emoji.cloud.sun.rain`
 `#emoji.drops`
**Sym Raw**
 `#emoji.cloud.rain`
 `#emoji.cloud.sun`
 `#emoji.cloud.thunder`
 `#emoji.sun`
**Special****Sym Raw**
 `#emoji.checkmark.heavy`
 `#emoji.circle.white`
 `#emoji.crossmark`
 `#emoji.explosion`
 `#emoji.flag.white`
 `#emoji.flag.red`
 `#emoji.globe.eu.af`
 `#emoji.notes`
**Sym Raw**
 `#emoji.checkmark.box`
 `#emoji.circle.stroked`
 `#emoji.crossmark.box`
 `#emoji.fire`
 `#emoji.flag.black`
 `#emoji.globe.am`
 `#emoji.globe.meridian`
 `#emoji.notes.tripple`
**Sym Raw**
 `#emoji.circle.black`
 `#emoji.copyright`
 `#emoji.excl`
 `#emoji.firecracker`
 `#emoji.flag.goal`
 `#emoji.globe.as.au`
 `#emoji.hash`
 `#emoji.ast`
**Technology****Sym Raw**
 `#emoji.battery`
 `#emoji.brightness.low`
 `#emoji.calendar`
 `#emoji.cassette`
 `#emoji.clamp`
 `#emoji.clock.alarm`
 `#emoji.controller`
 `#emoji.email`
 `#emoji.folder`
 `#emoji.hammer`
 `#emoji.laptop`
**Sym Raw**
 `#emoji.battery.low`
 `#emoji.bubble.speech.l`
 `#emoji.camera`
 `#emoji.clip`
 `#emoji.clock.one`
 `#emoji.computer`
 `#emoji.crab`
 `#emoji.floppy`
 `#emoji.folder.open`
 `#emoji.hammer.wrench`
**Sym Raw**
 `#emoji.brightness.high`
 `#emoji.bubble.speech.r`
 `#emoji.camera.movie`
 `#emoji.clipboard`
 `#emoji.clock.two`
 `#emoji.computermouse`
 `#emoji.disc.cd`
 `#emoji.flashlight`
 `#emoji.gear`
 `#emoji.headphone`

# Glossary

**FPGA – Field Programmable Gate Array:** A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing – hence the term 'field-programmable'. [24](#), [24](#), [24](#)

**HEI – Haute École d'Ingénierie** [24](#), [24](#), [24](#), [24](#), [24](#)

**IT – Infotronics** [24](#), [24](#)

**SYND – Systems Engineering** [24](#), [24](#), [24](#)

# Bibliography

- [1] S. Zahno *et al.*, “Dynamic Project Planning with Digital Twin,” *Frontiers in Manufacturing Technology*, vol. 3, May 2023, doi: [10.3389/fmtec.2023.1009633](https://doi.org/10.3389/fmtec.2023.1009633).