

Guide to Typst

14.01.2025 - v1.1.0 - for typist v0.12.0

Silvan Zahno

silvan.zahno@hevs.ch

HEI-Vs

Contents

1	Introduction	4
2	Installation	5
2.1	With cargo	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	27

Tables

Table 1: Multiple images one caption	13
Table 2: Multiple images one 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)	27
Equation (2)	27
Equation (3)	27
Equation (4)	27
Equation (5)	29
Equation (6)	29
Equation (7)	29
Equation (8)	29
Equation (9)	29
Equation (10)	29
Equation (11)	29
Equation (12)	29
Equation (13)	29
Equation (14)	29

Equation (15)	29
Equation (16)	29
Equation (17)	29
Equation (18)	29
Equation (19)	29
Equation (20)	29
Equation (21)	29
Equation (22)	29
Equation (23)	29

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`
- `glossarium`
- `tablex` (*for legacy tables only*)
- all files and folders in the `/00-templates/` folder such as
 - `icons/*.svg`
 - `scripts/*.bash`
 - `syntax/*.sublime-syntax`
 - `boxes.typ`
 - `constants.typ`
 - `helpers.typ`
 - `i18n.json`
 - `items.typ`
 - `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.

```
# 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 use **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
```

2.4 Windows

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

```
# install chocolatey
# ensure to use a administrative powershell
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'))

# install typst
choco install typst
```

3 | Formatting

3.1 Markup

Name	Example	Raw
Singleline Comment		//
Multiline Comment		/* */
Paragraph break		blankline
Line break		\
bold	bold	*bold*
italic	<i>italic</i>	_italic_
monospaced	<code>monospaced</code>	`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	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

D

C #v(0.2cm) 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")
Types	36pt text huger text	text(36pt, "36pt text") text(huger "huger text")
	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")
	Arial	text(font:"Arial", "Arial")
	Times New Roman	text(font:"Times New Roman", "Times New Roman")
Alignment	start	align(start){start}
	end	align(end){end}
	left	align(left){left}
	center	align(center){center}

Name	Example	Raw
	right	<code>align(right){right}</code>
	top	<code>align(top){top}</code>
	horizon	<code>align(horizon){horizon}</code>
	bottom	<code>align(bottom){bottom}</code>
	center + horizon	<code>align(center + horizon){center + horizon}</code>
Colors	black	<code>#text(fill:black)[black]</code>
	red	<code>#text(fill:red)[red]</code>
	green	<code>#text(fill:green)[green]</code>
	blue	<code>#text(fill:blue)[blue]</code>
	purple	<code>#text(fill:purple)[purple]</code>
	gray-80	<code>#text(fill:gray-80)[gray-80]</code>
	gray-70	<code>#text(fill:gray-70)[gray-70]</code>
	gray-60	<code>#text(fill:gray-60)[gray-60]</code>
	gray-50	<code>#text(fill:gray-50)[gray-50]</code>
	gray-40	<code>#text(fill:gray-40)[gray-40]</code>
	gray-30	<code>#text(fill:gray-30)[gray-30]</code>
	gray-20	<code>#text(fill:gray-20)[gray-20]</code>
	gray-10	<code>#text(fill:gray-10)[gray-10]</code>
	hei-orange	<code>#text(fill:hei-orange)[hei-orange]</code>
	hei-blue	<code>#text(fill:hei-blue)[hei-blue]</code>
	hei-pink	<code>#text(fill:hei-pink)[hei-pink]</code>
	hei-yellow	<code>#text(fill:hei-yellow)[hei-yellow]</code>
	hei-green	<code>#text(fill:hei-green)[hei-green]</code>
	spl-green	<code>#text(fill:spl-green)[spl-green]</code>
	spl-blue	<code>#text(fill:spl-blue)[spl-blue]</code>
	spl-pink	<code>#text(fill:spl-pink)[spl-green]</code>
	color-info	<code>#text(fill:color-info)[color-info]</code>
	color-idea	<code>#text(fill:color-idea)[color-idea]</code>
	color-warning	<code>#text(fill:color-warning)[color-warning]</code>
	color-important	<code>#text(fill:color-important)[color-important]</code>
	color-fire	<code>#text(fill:color-fire)[color-fire]</code>
	color-rocket	<code>#text(fill:color-rocket)[color-rocket]</code>
	color-todo	<code>#text(fill:color-todo)[color-todo]</code>

Name	Example	Raw
code-bg	code-bg	<code>#text(fill:code-bg) [code-bg]</code>
code-border	code-border	<code>#text(fill:code-border) [code-border]</code>

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

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

4.2 Lists

- 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 "/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

```
#item-list(content: "item-list")
#item-checkbadge(content: "item-checkbadge")
#item-circle(content: "item-circle")
#item-square(content: "item-square")
#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 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

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

```
#add-chapter(
    "/02-main/03-elements.typ",
    after: <sec:elem>,
    before: <sec:ref>,
)
// or
#add-chapter(
    after: <sec:elem>,
    before: <sec:ref>,
)[
    Content of the Chapter
]
```

4.5 Images

4.5.1 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)
)
```

4.5.2 Caption



Figure 1 - One image one caption

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

4.5.3 Cluster

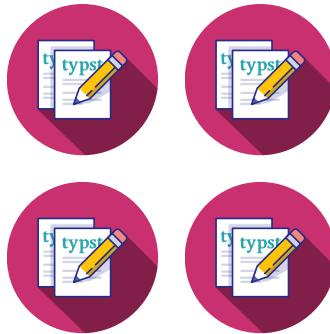
Two images one caption



Table 1 - Multiple images **one** caption

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

Four images one caption

Table 2 - Multiple images **one** caption

```
#figure(
  table(
    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 2 - Caption left image Figure 3 - Caption right image

```
#align(center,
  table(
    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 4 - Caption topleft image



Figure 5 - Caption topright image



Figure 6 - Caption bottomleft image



Figure 7 - Caption bottomright image

```
#align(center,
table(
  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]),
))
```

4.6 Tables

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

```
#import "@preview/tablex:0.0.9" : *
// or
#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

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

```
figure(
  table(
    columns: 3,
    align: center + horizon,
    table.header([], [*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

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

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

```
table(
  columns: 3,
  align: center + horizon,
  table.header([], [*Col1*] , ,
[*Col2*],),
  [*Row1*], table.cell(colspan: 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

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

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

```
table(
  columns: 3,
  align: center + horizon,
  table.header([], [*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

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

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

```
table(
  columns: 3,
  align: center + horizon,
  stroke: (x:none),
  table.header([], [*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

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

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

```
table(
  columns: 3,
  align: center + horizon,
  stroke: (x:none),
  table.hline(y:0, stroke:none), // remove first line
  table.hline(y:3, stroke:none), // remove last line
  table.vline(x:1),
  table.vline(x:2, start:1, end:2,
  stroke: red),
  table.header([], [*Col1*] , [*Col2*],),
  [*Row1*], "cell-0-0", "cell-1-0",
  [*Row2*], "cell-0-1", "cell-1-1",
),
```

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

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

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

```
table.vline(x:4, stroke:0.5pt), table.vline(x:5, stroke:0.5pt),
  table.header([$c$], [$b$], [$a$], [$c b$], [$b a$], [$y$], table.hline(stroke: 1pt)),
  [^0], [^0], [^0], [^0], [^0], [^0], table.hline(stroke: 0.5pt),
  [^0], [^0], [^1], [^0], [^0], [^1], table.hline(stroke: 0.5pt),
  [^0], [^1], [^0], [^0], [^0], [^0], table.hline(stroke: 0.5pt),
  [^0], [^1], [^1], [^0], [^1], [^0], table.hline(stroke: 1pt),
  [^1], [^0], [^0], [^0], [^0], [^0], table.hline(stroke: 0.5pt),
  [^1], [^0], [^1], [^0], [^0], [^1], table.hline(stroke: 0.5pt),
  [^1], [^1], [^0], [^1], [^0], [^1], table.hline(stroke: 0.5pt),
  [^1], [^1], [^1], [^1], [^1], [^1]
```

4.7 Icon Boxes

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

Info Box: #infobox()["infobox"]

Idea Box: #ideabox()["ideabox"]

Warning Box: #warningbox()["warningbox"]

Important Box: #importantbox()["importantbox"]

Fire Box: #firebox()["firebox"]

Rocket Box: #rocketbox()["rocketbox"]



```
#todobox()["todobox"]
```



```
#iconbox(icon: "/04-resources/placeholder.svg", linecolor: hei-blue)["iconbox"]
```

```
#iconbox(linecolor: hei-pink)["iconbox without icon"]
```

4.8 Color Boxes

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

Exercise

Some text

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

Attention

Some text

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

Consider

Some text

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

Information

Some text

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

4.8.1 Todo Box

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

TODO

This is not finished

```
#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.

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

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

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

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

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

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

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

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

4.9 Title Box

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

Title

Subtitle

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

Title

Subtitle

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

Title

```
#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.

```
#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.

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



The character count does not include spaces.

5 | References

5.1 Links

Example	Raw
https://tschinz.github.io	<code>https://tschinz.github.io</code>
https://tschinz.github.io	<code>#link("https://tschinz.github.io")</code>
See https://tschinz.github.io	<code>#link("https://tschinz.github.io") [See https://tschinz.github.io]</code>
silvan.zahno@hevs.ch	<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.

```
=_References <sec:ref>
==_Links <sec:links>
#figure(image("/04-resources/icon.svg", width: 2cm)) <fig:icon>
#figure(table(...), 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	<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-eq1</code>

5.3 External References

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

5.4 Glossary

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

```
#let entry-list = (
(
  key    : "hei",
  short  : "HEI",
  long   : "Haute École d'Ingénierie",
),
(
  key      : "fpga",
  short    : "FPGA",
  plural   : "FPGAs",
  long     : "Field Programmable Gate Array",
  longplural : "Field Programmable Gate Arrays",
  group    : "Technology"
  description : "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'.",
),
)
```

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.

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

Example

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

Raw

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

Haute École d'Ingénierie (HEI)

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

whatever you want

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

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

```
#glsp("fpga") and a second time
#glsp("fpga")
```

Field Programmable Gate Arrays (FPGAs)

```
#glsp("fpga", long: true))
```

[Infotronics \(IT\)](#) 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

inline monospaced string

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

```
`inline monospaced string`
```

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

```
-- Test 2: INPUT sX, pp
opCode <= "INPUT sX, pp      ";
code   <= "00010";
cIn    <= '0';
A      <= "1110000";
B      <= "00001111";
wait for clockPeriod;
assert Y = "00001111"
    report "test 2 INPUT wrong"
    severity note
```

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

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

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

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

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

Listing 2 - Rust Code

A plugin allows to get linenumbers

```
#import "@preview/codelst:2.0.2": sourcecode
```

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

```
#sourcecode()[
    ```\`rust
 fn main() {
```

```
 println!("Hello world!")
}
```

# 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

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

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

Math with caption

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

Equation 8 - Some proof

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

## 7.1 Align

Formula

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

Raw

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

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

```
$
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 symbols can be accessed with .

### 7.2.1 Accents

Sym- bol	Raw	Sym- bol	Raw	Sym- bol	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>
$\ddot{\cdot}$		$\circledcirc$		$\acute{\cdot}$	
$\ddot{x}$	<code>\$diaer(x)\$</code>	$\circledcirc$	<code>\$circle(x)\$</code>	$\acute{x}$	<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>
$\overbrace{xyz}$	<code>\$underbrace(xyz)\$</code>	$\overbrace{xyz}$	<code>\$overbrace(xyz)\$</code>	$\overbrace{xyz}$	<code>\$overbrace(xyz)\$</code>

### 7.2.2 Equals & Operators

Sym- bol	Raw	Sym- bol	Raw	Sym- bol	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>	$\not\geq$	<code>\$gt.eq.not\$</code>	$\leq$	<code>\$lt.eq\$</code>
$\not\leq$	<code>\$lt.eq.not\$</code>	$\approx$	<code>\$approx\$</code>	$\approx$	<code>\$approx.eq\$</code>
$\not\approx$	<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>\$-\$</code>	$-$	<code>\$minus\$</code>
$\mp$	<code>\$minus.plus\$</code>	$\ominus$	<code>\$minus.circle\$</code>		

### 7.2.3 Scripts

Sym- bol	Raw	Sym- bol	Raw	Sym- bol	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

Symbol	Raw	Symbol	Raw

$\binom{n}{k}$	(5) <code>\$ binom(n, k) \$</code>	$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$	(6) <code>\$ vec(1, 2, delim: "["") \$</code>
$\left\lfloor \frac{1}{2} \right\rfloor$	(7) <code>\$ round(1, 2) \$</code>	$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$	(8) <code>\$ mat(1,2; 3,4) \$</code>
$\begin{pmatrix} 1 & 2 & \dots & 10 \\ 2 & 2 & \dots & 10 \\ \vdots & \vdots & \ddots & \vdots \\ 10 & 10 & \dots & 10 \end{pmatrix}$	(9)	<pre>\$ mat(   1, 2, ..., 10;   2, 2, ..., 10;    dots.v, dots.v,   dots.down,   dots.v;   10, 10, ...   10; ) \$</pre>	$\sum a_k$ (10) <code>\$ sum a_k \$</code>
$\sum_{k=0}^n a_k$	(11) <code>\$ sum_(k=0)^n a_k \$</code>	$\sum_{k=0}^n a_k$	(12) <code>\$ scripts(sum)_(k=0)^n a_k \$</code>
$\sqrt[3]{x}$	(13) <code>\$ root(3, x) \$</code>	$f(x, y) := \begin{cases} 1 & \text{if } \frac{x \cdot y}{2} \leq 0 \\ 2 & \text{if } x \text{ is even} \\ 3 & \text{if } x \in \mathbb{N} \\ 4 & \text{else} \end{cases}$	(14)
<pre>\$ f(x, y) := cases(   1 "if" (x dot y)/2 &lt;= 0,   2 "if" x "is even",   3 "if" x in NN,   4 "else", ) \$</pre>		$\frac{1}{2}$	(15) <code>\$ 1/2 \$</code>
$\frac{1}{2}$	(16) <code>\$ frac(1,2) \$</code>	$\frac{x+1}{x+2}$	(17) <code>\$ (x+1)/(x+2) \$</code>
$\frac{(x+1)}{(x+2)}$	(18) <code>\$ ((x+1))/((x+2)) \$</code>	$\prod$	(19) <code>\$ product \$</code>
$n! = \prod_{k=1}^n k$	(20) <code>\$ product_(k=1)^n k \$</code>	$n! = \prod_{k=1}^n k(21) \quad \$ n! = scripts(product)_(k=1)^n k $$	
$\int$	(22) <code>\$ integral \$</code>	$\int_a^b f(x)$	(23) <code>\$ integral \$</code>

### 7.2.5 Alphabeth

#### Symbol

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

#### 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$

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

$AA BB CC DD EE FF GG HH II JJ KK LL MM
NN OO PP QQ RR SS TT UU VV WW XX YY ZZ$

ΑΒΓΔΕΖΗΘΙΚΑΜΝΞΟΠΡΣΤΥΦΧΨΩ
ΑΒCΔEFGHIJKLMNOPQRSTUVWXYZ

```

### 7.2.6 Logical

Symbol	Raw	Symbol	Raw	Symbol	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- bol	Raw	Sym- bol	Raw	Sym- bol	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\$	$\rightarrowtail$	\$arrow.long\$	$\rightarrowbar$	\$arrow.bar\$
$\rightarrowtail$	\$arrow.bar.long\$	$\Rightarrow$	\$arrow.double\$	$\Longrightarrow$	\$arrow.double.long\$
$\Rrightarrow$	\$arrow.double.bar\$	$\Rrightarrowtail$	\$arrow.double.bar.long\$	$\Rrightarrowtail$	\$arrow.quad\$
$\Rightarrowtail$	\$arrow.stroked\$	$\rightarrowtail$	\$arrow.filled\$	$\dashrightarrow$	\$arrow.dashed\$
$\looparrowright$	\$arrow.curve\$	$\rightsquigarrow$	\$arrow.squiggly\$	$\looparrowrighttail$	\$arrow.loop\$
Arrows left					
$\leftarrow$	\$arrow.l\$	$\leftarrowtail$	\$arrow.l.long\$	$\leftarrowbar$	\$arrow.l.bar\$
$\leftarrowtail$	\$arrow.l.bar.long\$	$\Leftarrowtail$	\$arrow.l.double\$	$\Leftarrowtail$	\$arrow.l.double.long\$

$\Leftarrow$	<code>\$arrow.l.double.bar\$</code>	$\Leftarrow\Rightarrow$	<code>\$arrow.l.double.bar.long\$</code>	<code>\$arrow.l.quad\$</code>	
$\Leftarrow$	<code>\$arrow.l.stroked\$</code>	$\Leftarrow$	<code>\$arrow.l.filled\$</code>	$\Leftarrow$	<code>\$arrow.l.dashed\$</code>
$\Leftarrow$	<code>\$arrow.l.curve\$</code>	$\Leftarrow\curvearrowright$	<code>\$arrow.l.squiggly\$</code>	$\Leftarrow\looparrowright$	<code>\$arrow.l.loop\$</code>
<b>Double Arrows Left Right</b>					
$\Leftarrow$	<code>\$arrow.l.r\$</code>	$\Leftarrow\Rightarrow$	<code>\$arrow.l.r.not\$</code>	$\Leftarrow\Rightarrow$	<code>\$arrow.l.r.long\$</code>
$\Leftarrow\Rightarrow$	<code>\$arrow.l.r.double\$</code>	$\Leftarrow\Rightarrow$	<code>\$arrow.l.r.double.long\$</code>	$\Leftarrow\Rightarrow$	<code>\$arrow.l.r.double.not\$</code>
$\Leftarrow\Rightarrow$	<code>\$arrow.l.r.stroked\$</code>	$\Leftarrow\Rightarrow$	<code>\$arrow.l.r.filled\$</code>	$\Leftarrow\Rightarrow$	<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>
$\Uparrow\Uparrow$	<code>\$arrow.t.triple\$</code>	$\Uparrow\Uparrow$	<code>\$arrow.t.quad\$</code>	$\Uparrow\Uparrow$	<code>\$arrow.t.stroked\$</code>
$\Uparrow$	<code>\$arrow.t.filled\$</code>	$\Uparrow$	<code>\$arrow.t.dashed\$</code>	$\Uparrow$	<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>
$\Downarrow\Downarrow$	<code>\$arrow.b.triple\$</code>	$\Downarrow\Downarrow$	<code>\$arrow.b.quad\$</code>	$\Downarrow\Downarrow$	<code>\$arrow.b.stroked\$</code>
$\Downarrow$	<code>\$arrow.b.filled\$</code>	$\Downarrow$	<code>\$arrow.b.dashed\$</code>	$\Downarrow$	<code>\$arrow.b.curve\$</code>
<b>Double Arrows Top Bottom</b>					
$\Updownarrow$	<code>\$arrow.t.b\$</code>	$\Updownarrow$	<code>\$arrow.t.b.double\$</code>	$\Updownarrow$	<code>\$arrow.t.b.stroked\$</code>
$\Updownarrow$	<code>\$arrow.t.b.filled\$</code>				
<b>Arrows Diagonal Top Right</b>					
$\nearrow$	<code>\$arrow.tr\$</code>	$\nearrow\swarrow$	<code>\$arrow.tr.double\$</code>	$\nearrow$	<code>\$arrow.tr.stroked\$</code>
$\nearrow\swarrow$	<code>\$arrow.tr.filled\$</code>		$\nearrow$	<code>\$arrow.tr.hook\$</code>	
<b>Arrows Diagonal Bottom Right</b>					
$\searrow$	<code>\$arrow.br\$</code>	$\searrow\swarrow$	<code>\$arrow.br.double\$</code>	$\searrow$	<code>\$arrow.br.stroked\$</code>
$\searrow\swarrow$	<code>\$arrow.br.filled\$</code>		$\searrow$	<code>\$arrow.br.hook\$</code>	
<b>Arrows Diagonal Bottom Left</b>					
$\swarrow$	<code>\$arrow.bl\$</code>	$\swarrow\searrow$	<code>\$arrow.bl.double\$</code>	$\swarrow$	<code>\$arrow.bl.stroked\$</code>
$\swarrow\searrow$	<code>\$arrow.bl.filled\$</code>		$\swarrow$	<code>\$arrow.bl.hook\$</code>	
<b>Arrows Diagonal Top Left</b>					
$\nwarrow$	<code>\$arrow.tl\$</code>	$\nwarrow\swarrow$	<code>\$arrow.tl.double\$</code>	$\nwarrow$	<code>\$arrow.tl.stroked\$</code>
$\nwarrow\swarrow$	<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>	$\curvearrowright$	<code>\$arrow.cw.half\$</code>	$\circlearrowright$	<code>\$arrow.ccw\$</code>
$\curvearrowleft$	<code>\$arrow.ccw.half\$</code>				

### 7.2.9 Angles

Sym- bol	Raw	Sym- bol	Raw	Sym- bol	Raw
$\angle$	<code>\$angle\$</code>	$\triangleright$	<code>\$angle.rev\$</code>	$\triangleleft$	<code>\$angle.acute\$</code>
$\angle.$	<code>\$angle.acute\$</code>	$\triangleleft.$	<code>\$angle.arc\$</code>	$\triangleleft\triangleleft$	<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>	$\llcorner\llcorner$	<code>\$angle.right.rev\$</code>
$\triangleright$	<code>\$angle.right.arc\$</code>	$\llcorner\llcorner$	<code>\$angle.right.dot\$</code>	$\llcorner\llcorner\llcorner$	<code>\$angle.right.sq\$</code>
$\triangleleft$	<code>\$angle.spheric\$</code>	$\triangleright\triangleright$	<code>\$angle.spheric.rev\$</code>	$\triangleright\triangleright\triangleright$	<code>\$angle.spheric.top\$</code>

### 7.2.10 Cool Symbols

Sym- bol	Raw	Sym- bol	Raw	Sym- bol	Raw
$@$	<code>\$at\$</code>	$\%$	<code>\$co\$</code>	$\circledcirc$	<code>\$copyright\$</code>
$\circledcirc$	<code>\$copyright.sound\$</code>	$^{\circ}\mathrm{C}$	<code>\$degree.c\$</code>	$\text{\texteuro}$	<code>\$euro\$</code>
$\$$	<code>\$dollar\$</code>	$\pounds$	<code>\$pound\$</code>	$\text{\textwon}$	<code>\$won\$</code>
$\text{\textyen}$	<code>\$yen\$</code>	$\text{\texteuro}$	<code>\$bitcoin\$</code>	$^{\circ}\mathrm{F}$	<code>\$degree.f\$</code>
$!$	<code>\$excl\$</code>	$\text{\textexclam}$	<code>\$excl.inv\$</code>	$!!$	<code>\$excl.double\$</code>
$!?$	<code>\$excl.quest\$</code>	$\text{\textarrowleft}$	<code>\$arrow.zigzag\$</code>	$\circledast$	<code>\$ast.circle\$</code>
$*$	<code>\$ast.triple\$</code>	$\chi$	<code>\$chi\$</code>	$\text{\textcircledast}$	<code>\$floral\$</code>
$\text{\textmaltese}$	<code>\$maltese\$</code>	$\text{\textpilcrow}$	<code>\$pilcrow\$</code>	$h$	<code>\$planck\$</code>
$\clubsuit$	<code>\$suit.club\$</code>	$\text{\textdiamond}$	<code>\$suit.diamond\$</code>	$\heartsuit$	<code>\$suit.heart\$</code>
$\spadesuit$	<code>\$suit.spade\$</code>	$\text{\texttriangle}$	<code>\$triangle.stroked.nested2\$</code>		

### 7.2.11 Style

Symbol	Raw	Symbol	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>		

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

# 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	Sym Raw
<b>Smileys</b>		
😊 #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	😢 #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
😇 #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
騙 #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	Ｚ #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.dromedary
🐓	#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

**Sym Raw**
 `#emoji.flower.wilted`
**Sym Raw**
 `#emoji.flower.yellow`
**Sym Raw**
 `#emoji.plant`
**Food****Sym Raw**
 `#emoji.avocado`
**Sym Raw**
 `#emoji.aubergine`
**Sym Raw**
 `#emoji.banana`
 `#emoji.broccoli`
 `#emoji.burger`
 `#emoji.cake.slice`
 `#emoji.carrot`
 `#emoji.chocolate`
 `#emoji.corn`
 `#emoji.cucumber`
 `#emoji.fries`
 `#emoji.honey`
**Objects****Sym Raw**
 `#emoji.bin`
**Sym Raw**
 `#emoji.bomb`
**Sym Raw**
 `#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`
**Sport****Sym Raw**
 `#emoji.baseball`
**Sym Raw**
 `#emoji.billiards`
**Sym Raw**
 `#emoji.boxing`
 `#emoji.chess`
 `#emoji.climbing`
 `#emoji.football`
**Vehicles****Sym Raw**
 `#emoji.airplane`
**Sym Raw**
 `#emoji.bike`
**Sym Raw**
 `#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`
**Building****Sym Raw**
 `#emoji.castle.eu`
**Sym Raw**
 `#emoji.castle.jp`
**Sym Raw**
 `#emoji.circus`
 `#emoji.factory`
 `#emoji.hospital`
 `#emoji.office`
 `#emoji.house`
 `#emoji.museum`
 `#emoji.crane`
**Weather****Sym Raw****Sym Raw****Sym Raw**

**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](#)

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

**IT – Infotronics** [24](#)

**SYND – Systems Engineering** [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).