Documentation of the Template

Alexandre Bry

20 May 2025

Contents

1. Template

The template functions are defined in template/template.typ. However, to make it easier to switch between a "long" report and a "short" assignment submission, we created two sub-versions template/template-long.typ and template/template-short.typ that use the same functions but with different default values. Therefore, all the functions mentioned below can be imported from any of these files with the same capabilities, except the default values will be better suited for the type of document you are writing.

Here are the functions available in template/template.typ:

- init()
- cover-container()
- generic-container()
- pre-content-container()
- main-content-container()
- post-content-container()
- appendix-container()

init

Main function of the template that defines the overall style. This function is designed to be used in the main scope of the document, and to be called using #show: init.with(…).

Parameters

```
init(
    body: content ,
    title: str content ,
    authors-names: array ,
    authors-data: dictionary ,
    text-font: str array ,
    math-font: str array ,
    raw-font: str array ,
    headings-font: str array none ,
    font-size-multiplier: float ,
    base-font-size: length ,
    page-numbering: str function ,
    _documentation: bool
) -> content
```

body content

The body of the document.

title str or content

The title of the document.

Default: ""

authors-names array

The names of the authors.

Default: ()

authors-data dictionary

The data of the authors. Can contain any number for (key, value) pairs. The key is a string and the value is an array of strings. The key is used as a label for the piece of information and the values correspond to the authors.

Default: (:)

text-font str *or* array

The font to use for the text.

Default: ("Source Serif 4", "Libertinus Serif")

math-font str *or* array

The font to use for the math.

Default: ("STIX Two Math", "New Computer Modern Math")

raw-font str *or* array

The font to use for the raw text.

Default: ("Source Code Pro", "Libertinus Mono")

headings-font str *or* array *or* none

The font to use for the text in titles. If none, the text font will be used.

Default: none

font-size-multiplier float

The font size multiplier to use for the text. This is used to scale the font size of the document.

Default: 1.0

base-font-size length

The base font size to use for the text.

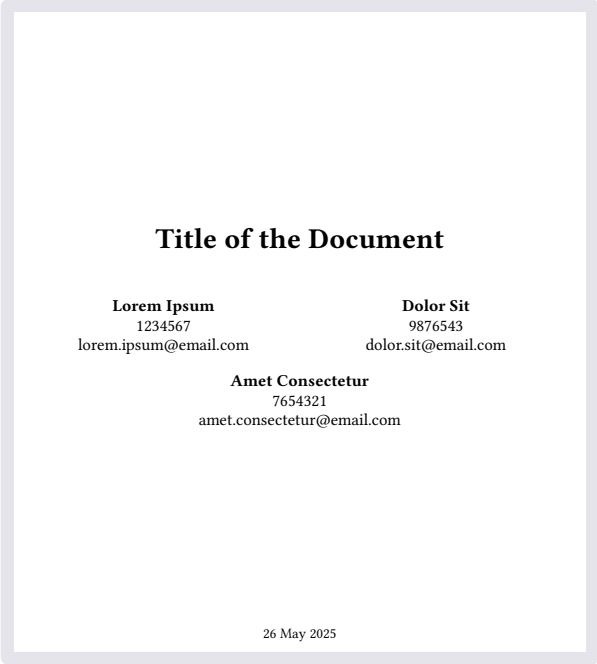Default: 11pt

_documentation `bool`

Special argument for the documentation.

Default: `false`

cover-container

Cover page container. To use in combination with a cover page. Removes the header and footer of the page.

```
#[
  #show: cover-container.with(_documentation: true)
  #cover.cover(
    title: "Title of the Document",
    authors-names: ("Lorem Ipsum", "Dolor Sit", "Amet
Consectetur"),
    authors-data: (
      "Student IDs": ("1234567", "9876543", "7654321"),
      "Email": (
        "lorem.ipsum@email.com",
        "dolor.sit@email.com",
        "amet.consectetur@email.com",
      ),
    ),
    alignment: center,
    date: datetime.today(),
    _documentation: true,
  )
]
```

Title of the Document

Lorem Ipsum
1234567
lorem.ipsum@email.com

Dolor Sit
9876543
dolor.sit@email.com

Amet Consectetur
7654321
amet.consectetur@email.com

20 May 2025

Parameters

```
cover-container(
  body: content ,
  full-page: bool ,
  _documentation: bool
) -> content
```

body `content`

The body of the document.

full-page `bool`

Whether the cover page is a full page or not. If true, the page will be set to have no header or footer.

Default: `true`

_documentation `bool`

Special argument for the documentation.

Default: `false`

generic-container

Generic container for a subset of the document. This function is designed to define the style of the document locally, using a show rule like this:

```
#[
 #show: generic-container.with(
  new-page: true,
  page-numbering: "1"
 )
 #include "../content/content.typ"
]
```

Parameters

```
generic-container(
    body: content ,
    new-page: bool ,
    h1-new-page: bool ,
    reset-page-numbering: bool ,
    page-numbering: none str function ,
    heading-numbering: none str function array ,
    heading-supplement: none str content array
) -> content
```

body `content`

The body of the document.

new-page `bool`

Whether to start a new page at the beginning of the container.

Default: `true`

h1-new-page `bool`

Whether to start a new page before each level 1 heading.

Default: `true`

reset-page-numbering `bool`

Whether to reset the page numbering at the beginning of the container.

Default: `true`

page-numbering `none` or `str` or `function`

The page numbering style to use.

Default: `none`

heading-numbering `none` or `str` or `function` or `array`

The heading numbering style to use. If it is an array, it must have 6 elements (one for each heading level). Each element of the array must be one of the other allowed types.

Default: `none`

heading-supplement `none` or `str` or `content` or `array`

The supplement to use for the headings. If it is an array, it must have 6 elements (one for each heading level). Each element of the array must be one of the other allowed types.

Default: `"Section"`

pre-content-container

Container for the pre-content of the document. Pre-content is defined as the content after the cover page and before the main content (usually containing the preface, acknowledgements, abstract, table of contents, etc.). This function is designed to define the style of the document locally, using a `show` rule, like in *generic-container*.

Parameters

```
pre-content-container(
  body: content ,
  new-page: bool ,
  h1-new-page: bool ,
  reset-page-numbering: bool ,
  page-numbering: none str function ,
  heading-numbering: none str function array ,
  heading-supplement: none str content array
) -> content
```

body `content`

Refer to the _body argument in generic-container_ for more details.

new-page `bool`

Refer to the _new-page argument in generic-container_ for more details.

Default: `true`

h1-new-page `bool`

Refer to the _h1-new-page argument in generic-container_ for more details.

Default: `true`

reset-page-numbering `bool`

Refer to the _reset-page-numbering argument in generic-container_ for more details.

Default: `true`

page-numbering `none` or `str` or `function`

Refer to the _page-numbering argument in generic-container_ for more details.

Default: `"i"`

heading-numbering `none` or `str` or `function` or `array`

Refer to the _heading-numbering argument in generic-container_ for more details.

Default: `none`

heading-supplement `none` or `str` or `content` or `array`

Refer to the _heading-supplement argument in generic-container_ for more details.

Default: `"Section"`

main-content-container

Container for the main content of the document. Main content is defined as the core content, usually containing the chapters, sections, and subsections. This function is designed to define the style of the document locally, using a show rule, like in _generic-container_.

Parameters

```
main-content-container(
    body: content ,
    new-page: bool ,
    h1-new-page: bool ,
    reset-page-numbering: bool ,
    page-numbering: none str function ,
    heading-numbering: none str function array ,
    heading-supplement: none str content array
) -> content
```

body content

Refer to the *body argument in generic-container* for more details.

new-page bool

Refer to the *new-page argument in generic-container* for more details.

Default: true

h1-new-page bool

Refer to the *h1-new-page argument in generic-container* for more details.

Default: true

reset-page-numbering bool

Refer to the *reset-page-numbering argument in generic-container* for more details.

Default: true

page-numbering none or str or function

Refer to the *page-numbering argument in generic-container* for more details.

Default: "1"

heading-numbering none or str or function or array

Refer to the *heading-numbering argument in generic-container* for more details.

Default: ("1.", "1.a.", "1.a.i.", none, none, none)

heading-supplement none or str or content or array

Refer to the *heading-supplement argument in generic-container* for more details.

Default: ("Chapter", "Section", "Subsection", none, none, none)

post-content-container

Container for the post-content of the document. Post-content is defined as the content after the main content and before the appendix (usually containing the bibliography, index, etc.). This function is designed to define the style of the document locally, using a show rule, like in *generic-container*.

Parameters

```
post-content-container(
    body: content ,
    new-page: bool ,
    h1-new-page: bool ,
    reset-page-numbering: bool ,
    page-numbering: none  str  function ,
    heading-numbering: none  str  function  array ,
    heading-supplement: none  str  content  array
) -> content
```

body content

Refer to the *body argument in generic-container* for more details.

new-page bool

Refer to the *new-page argument in generic-container* for more details.

Default: true

h1-new-page bool

Refer to the *h1-new-page argument in generic-container* for more details.

Default: true

reset-page-numbering bool

Refer to the *reset-page-numbering argument in generic-container* for more details.

Default: false

page-numbering none or str or function

Refer to the *page-numbering argument in generic-container* for more details.

Default: "1"

heading-numbering none or str or function or array

Refer to the *heading-numbering argument in generic-container* for more details.

Default: none

heading-supplement `none` or `str` or `content` or `array`

Refer to the *heading-supplement argument in generic-container* for more details.

Default: `"Section"`

appendix-container

Container for the appendix of the document. Appendix is defined as a separate part at the end of the document that gives extra information. This function is designed to define the style of the document locally, using a `show` rule, like in *generic-container*.

Parameters

```
appendix-container(
    body: content ,
    new-page: bool ,
    h1-new-page: bool ,
    reset-page-numbering: bool ,
    page-numbering: none str function ,
    heading-numbering: none str function array ,
    heading-supplement: none str content array
) -> content
```

body `content`

Refer to the *body argument in generic-container* for more details.

new-page `bool`

Refer to the *new-page argument in generic-container* for more details.

Default: `true`

h1-new-page `bool`

Refer to the *h1-new-page argument in generic-container* for more details.

Default: `true`

reset-page-numbering `bool`

Refer to the *reset-page-numbering argument in generic-container* for more details.

Default: `true`

page-numbering   `none` or `str` or `function`

Refer to the *page-numbering argument in generic-container* for more details.

Default: numbly.numbly("A-{1:1}")

heading-numbering   `none` or `str` or `function` or `array`

Refer to the *heading-numbering argument in generic-container* for more details.

Default: numbly.numbly("Appendix {1:A}", "{1:A}.{2:1}.", "{1:A}.{2:1}.{3:a}.", none, none, none)

heading-supplement   `none` or `str` or `content` or `array`

Refer to the *heading-supplement argument in generic-container* for more details.

Default: (none, "Appendix", "Appendix", none, none, none)

2. Cover Page

A simple cover page is defined in template/cover.typ. It is a simple function that takes the title, authors, and date as arguments, as well as some optional arguments to customize the layout.

Here are the functions available in template/cover.typ:

- cover()

cover

Cover page template. Supports full page and non-full page formats. There are two versions of the cover page depending on the desired alignment.

```
#[
  #show: cover-container.with(_documentation: true)
  #for alignment in (left, center, right) {
    cover.cover(
      title: "Title of the Document",
      authors-names: ("Lorem Ipsum", "Dolor Sit", "Amet
Consectetur"),
      authors-data: (
        "Student IDs": ("1234567", "9876543", "7654321"),
        "Email": (
          "lorem.ipsum@email.com",
          "dolor.sit@email.com",
          "amet.consectetur@email.com",
        ),
      ),
      full-page: false,
      alignment: alignment,
      date: datetime.today(),
      _documentation: true,
    )
    v(2cm, weak: true)
  }
]
```

Title of the Document

20 May 2025

Name	Student IDs	Email
Lorem Ipsum	1234567	lorem.ipsum@email.com
Dolor Sit	9876543	dolor.sit@email.com
Amet Consectetur	7654321	amet.consectetur@email.com

Title of the Document

20 May 2025

Lorem Ipsum	**Dolor Sit**
1234567	9876543
lorem.ipsum@email.com	dolor.sit@email.com

Amet Consectetur
7654321
amet.consectetur@email.com

Title of the Document

20 May 2025

Name	Student IDs	Email
Lorem Ipsum	1234567	lorem.ipsum@email.com
Dolor Sit	9876543	dolor.sit@email.com
Amet Consectetur	7654321	amet.consectetur@email.com

Parameters

```
cover(
  title: str content ,
  authors-names: str array ,
  authors-data: dictionary ,
  date: datetime ,
  full-page: bool ,
  alignment: alignment ,
  _documentation: bool
) -> content
```

title `str` or `content`

The title of the document.

Default: `""`

authors-names `str` or `array`

The names of the authors.

Default: `()`

authors-data `dictionary`

The data of the authors. Can contain any number for (key, value) pairs. The key is a string and the value is an array of strings. The key is used as a label for the piece of information and the values correspond to the authors.

Default: `(:)`

date `datetime`

The date to be displayed on the cover page. If not provided, the current date is used.

Default: `datetime.today()`

full-page `bool`

Whether to use the full page format or not. This only changes the vertical layout of the content.

Default: `true`

alignment `alignment`

The alignment of the content on the page. Can be left, center, right, start, or end.

Default: `left`

_documentation `bool`

Special argument for the documentation.

Default: `false`

3. Styled Blocks

other-tools/styled-blocks.typ contains a set of functions to add colourful blocks of text to the document. The intended use is to highlight tasks or information useful to the authors in writing the document. Then, the blocks can be hidden or removed when the document is rendered, using the render argument like this at the beginning of the file:

#let block-todo = block-todo.with(render: false)

Here are the functions available in other-tools/styled-blocks.typ:

- styled-block()
- block-todo()
- block-discussion()

styled-block

Create a styled block with a title and content.

#styled-block()[Here is the title][Here is the content that is longer.]

> **Here is the title**
>
> Here is the content that is longer.

Parameters

```
styled-block(
    title: str content none ,
    content: str content none ,
    block-type: str content none ,
    fill-title: color ,
    fill-content: color ,
    inset: length dictionary ,
    radius: length ,
    render: bool
) -> content
```

title   str or content or none

The title of the block.

content   str or content or none

The content of the block.

block-type   str or content or none

The type of the block (e.g., "TODO", "Discussion"), displayed below the title.

Default: none

fill-title   `color`

The colour of the title background.

Default: `rgb(180, 180, 255)`

fill-content   `color`

The colour of the content background.

Default: `rgb(210, 210, 255)`

inset   `length` or `dictionary`

The inset of the title and content blocks.

Default: (x: `6pt`, y: `8pt`)

radius   `length`

The radius of the title and content blocks.

Default: `8pt`

render   `bool`

Whether to render the block.

Default: `true`

block-todo

Create a styled block for TODOs.

```
#block-todo()[We have something to do][Don't forget to do it!]
```

TODO: We have something to do

Don't forget to do it!

Parameters

```
block-todo(
  title: str content none,
  content: str content none,
  render: bool
) -> content
```

content   `str` or `content` or `none`

The content of the block.

render   `bool`

Whether to render the block.

Default: `true`

block-discussion

Create a styled block for discussions.

```
#block-discussion()[Add more explanations?][- Author A:
Maybe it would be better...
- Author B: Yes, but I don't know...
- Author A: What about...]
```

Discussion: Add more explanations?

- Author A: Maybe it would be better...
- Author B: Yes, but I don't know...
- Author A: What about...

Parameters

```
block-discussion(
  title: str content none,
  content: str content none,
  render: bool
) -> content
```

title   `str` `content` `none`

The title of the block.

content   `str` or `content` or `none`

The content of the block.

render   `bool`

Whether to render the block.

Default: `true`

15

4. Custom Style

other-tools/custom-style.typ contains a function to add custom styles to the document. These styles are not directly included in the template because they are not really standard.

Here are the functions available in other-tools/custom-style.typ:

- custom-style()

custom-style

Define a custom style for figures, tables, and raw blocks. This function allows you to customize the appearance of figures, tables, and raw blocks in your document.

Parameters

```
custom-style(
  body: content ,
  custom-figure-captions: bool ,
  custom-tables: bool ,
  raw-background: color
) -> body
```

body `content`

The body of the document

custom-figure-captions `bool`

Whether to use custom figure captions.

Default: true

custom-tables `bool`

Whether to use custom tables.

Default: true

raw-background `color`

The background colour for raw blocks. Set to none to disable the custom styling for raw blocks.

Default: luma(220, 60%)