Rubber Article - Template Documentation

v.0.4.1

Contents

Example Usage

```typst
#import "@preview/rubber-article:0.4.1": *
#show: article.with()
#maketitle(
  title: "The Title of the Paper",
  authors: ("Authors Name",),
  date: datetime.today().display("[day].[month].[year]"),
)
```

Styling functions

These functions are used to style certain elements of the document. They are usually called with a `#show` statement. They do not output any content, but rather modify the appearance of the document.

- appendix()
- article()

appendix

Function to format the Appendix. This function is intended to be used after the document has been styled with the `article` function.

Example usage:

```typst
#show: article.with()
// A lot of content goes here...

#show: appendix.with(
  title: "Appendix",
  title-align: center,
)
```

Parameters

```
appendix(
  numbering: none str function,
  title: none str content,
  title-align: alignment,
  title-size: length,
  numbering-start: int,
  content
) -> content
```

numbering `none` or `str` or `function`

The numbering of the Appendix

Default: `"A.1"`

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

The title of the Appendix

Default: `none`

title-align `alignment`

The alignment of the title

Default: `center`

title-size `length`

The size of the title

Default: `none`

numbering-start `int`

Startting the appendex after this number

Default: `0`

article

A Template recreating the look of the classic Article Class.

Example usage:

```typst
#show: article.with(
  lang: "de",
)
```

This will format the document with the specified options. For more styling options, explore the following parameters.

Parameters

```
article(
  lang: str,
  eq-numbering: str none,
  eq-chapterwise: bool,
  text-size: length,
  page-numbering: none str function,
  page-numbering-align: alignment,
  heading-numbering: none str function,
  margins: auto relative dictionary,
  enum-indent: length,
  list-indent: length,
  header-display: bool,
  alternating-header: bool,
  first-page-header: int float,
  header-title: str content,
  justify: bool,
  fig-caption-width: relative,
  header-line-stroke: length,
  content: content
) -> content
```

lang `str`

Set the language of the document.

Default: `"de"`

eq-numbering `str` or `none`

Set the equation numbering style.

Default: `none`

eq-chapterwise `bool`

Chapterwise numbering of equations.

Default: `false`

text-size `length`

Set the text size. Headings are adjusted automatically.

Default: `10pt`

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

Set the page numbering style.

Default: `"1"`

page-numbering-align `alignment`

Set the page numbering alignment.

Default: `center`

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

Set the heading numbering style.

Default: `"1.1"`

margins `auto` or `relative` or `dictionary`

Set the margins of the document.

Default: `(left: 25mm, right: 25mm, top: 30mm, bottom: 30mm)`

enum-indent `length`

Set the Enum indentation.

Default: `1.5em`

list-indent `length`

Set the List indentation.

Default: `1.5em`

header-display `bool`

Set if the default header should be used.

Default: `false`

alternating-header `bool`

Set if the default header should be alternating.

Default: `true`

first-page-header `int` or `float`

Set the first page of the header.

Default: `1`

header-title `str` or `content`

Set the Header Titel

Default: `none`

justify `bool`

Set if document should be justified.

Default: `true`

fig-caption-width `relative`

Set the width of the figure captions.

Default: `75%`

header-line-stroke `length`

Set the width of the headerline.

Default: `.65pt`

Constructor functions

These functions are used to create certain elements of the document. They can be called with certain arguments and output some content.

- ctable()
- fig-outline()
- maketitle()
- shortcap()
- tab-outline()

ctable

This function will display a custom table. The table uses the `pillar` package under the hood to interact with the table in a similar manner as in Latex. This means, that the columns and vertical lines can be defined with a string. Furthermore, the table automatically adds 3 horizontal lines.

Example usage:

```typst
#ctable(
  cols:"l|cr",
  [A], [B], [C],
 ..range(1,16).map(str),
)
```

Parameters

```
ctable(
  ..data,
  cols: string,
  stroke: length,
  middle-stroke: length,
  vertical-stroke: length,
  header-rows: int
) -> content
```

cols `string`

A string that defines the columns and vertical lines of the table.

Default: `"ccc"`

stroke `length`

The linesytle of the table, especially the top and bottom horizontal lines.

Default: `.75pt`

middle-stroke `length`

The linesytle of the middle horizontal line.

Default: `.6pt`

vertical-stroke `length`

The linesytle of the vertical lines.

Default: `.6pt`

header-rows `int`

The number of header rows.

Default: `1`

fig-outline

This function is a wrapper of the `outline` function and allows for an easy way to create the table of figures.

Example usage:

```typst
#fig-outline()
```

Parameters

```
fig-outline(
  title: string content,
  target: label selector location function,
  ..args
) -> content
```

title `string` or `content`

The title of the table of figures.

Default: `"List of Figures"`

target `label` or `selector` or `location` or `function`

The Target of the table of figures.

Default: `figure.where(kind: image)`

..args

Additional optional arguments to the outline function.

maketitle

This function will display the frontmatter of the document. This includes the title, authors, and date.

Example usage:

```typst
#maketitle(
  title: "The Title of the Paper",
  authors: ("Authors Name",),
  date: "2025-01-01",
)
```

Parameters

```
maketitle(
  title: string content,
  authors: array,
  date: string content datetime,
  metadata: bool
) -> content
```

title `string` or `content`

The title of the document.

Default: `""`

authors `array`

The authors of the document.

Default: `()`

date `string` or `content` or `datetime`

The date of the document.

Default: `none`

metadata `bool`

Use title and author information for the document metadata.

Default: `true`

shortcap

This function will help you to provide a long caption to a figure, but a short caption to the outline.

Example usage:

```typst
#figure(
  rect(),
  caption: shortcap("Short caption", "Long caption"),
)
```

Parameters

```
shortcap(
  short: string content,
  long: string content
) -> content
```

short `string` or `content`

The short caption of the figure.

long `string` or `content`

The long caption of the figure.

tab-outline

This function is a wrapper of the `outline` function and allows for an easy way to create the list of tables.

Example usage:

```typst
#tab-outline()
```

Parameters

```
tab-outline(
  title: string content,
  target: label selector location function,
  ..args
) -> content
```

title `string` or `content`

The title of the table of figures.

Default: `"List of Tables"`

target `label` or `selector` or `location` or `function`

The Target of the table of figures.

Default: `figure.where(kind: table)`

..args

Additional optional arguments to the outline function.