

Rubber Article - Template Documentation

v.0.5.1

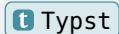
Contents

Styling functions	2
appendix	2
article	3
Constructor functions	8
abstract	8
ctable	9
fig-outline	10
maketitle	11
shortcap	12
tab-outline	12
Utility functions	14
balance	14
vspace	14

This template is a replication/implementation of the classic `article` L^AT_EX class in `typst`. It is designed to be used for writing articles, papers, and other documents. It is a good starting point for people transitioning from L^AT_EX to `typst` or students starting with academic writing.

Example Usage

```
1 #import "@preview/rubber-article:0.5.1": *
2 #show: article
3 #maketitle(
4     title: "The Title of the Paper",
5     authors: ("Authors Name",),
6     date: datetime.today().display("[day].[month].[year]"),
7 )
```

 Typst

It shall be noted here, that this is not an exact copy, but rather a very close reinterpretation, since `typst` is quite different to work with. Additionally some features, that have not been present in the original `article` class, have been added to this template. For example the `header-line` has been added, which in L^AT_EX would have needed the package `fancyhdr`. Also worth a mention is that the default paper size in Typst is the A4 format, while in the original the US Letter format is used.

In order to recreate the original `article` class, use the following settings.

```
1 #show: article.with(
2     page-margins: 1.75in,
3     page-paper: "us-letter",
4 )
```

 Typst

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:

```
1 #show: article
2 // A lot of content goes here...
3
4 #show: appendix.with(
5   title: "Appendix",
6   title-align: center,
7 )
```

 Typst

Parameters

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

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 `none` or `str` or `function`

The numbering of the Appendix

Default: `"A.1"`

numbering-start `int`

Starttting the appendix after this number

Default: `0`

article

A Template recreating the look of the classic Article Class.

Example usage:

```
1 #show: article.with(  
2   lang: "de",  
3 )
```

 Typst

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

Parameters

```
article(  
    cols: none int,  
    eq-chapterwise: bool,  
    eq-numbering: str none,  
    fig-caption-width: relative,  
    header-alternating: bool,  
    header-display: bool,  
    header-first-page: int float,  
    header-line-stroke: length,  
    header-title: str content,  
    heading-numbering: none str function,  
    lang: str,  
    list-bullet-indent: length,  
    list-numbered-indent: length,  
    outline-link-color: color,  
    page-margins: auto relative dictionary,  
    page-numbering-align: alignment,  
    page-numbering: none str function,  
    page-paper: str,  
    par-first-line-indent: length,  
    par-justify: bool,  
    par-spacing: length,  
    text-font: str,  
    text-size: length,  
    content: content  
) -> content
```

cols none or int

Set the number of columns, that the document should have.

Default: none

eq-chapterwise bool

Chapterwise numbering of equations.

Default: false

eq-numbering str or none

Set the equation numbering style.

Default: none

fig-caption-width relative

Set the width of the figure captions.

Default: 75%

header-alternating bool

Set if the default header should be alternating.

Default: `true`

header-display bool

Set if the default header should be used.

Default: `false`

header-first-page int or float

Set the first page of the header.

Default: `1`

header-line-stroke length

Set the width of the headerline.

Default: `.65pt`

header-title str or content

Set the Header Title

Default: `none`

heading-numbering none or str or function

Set the heading numbering style.

Default: `"1.1"`

lang str

Set the language of the document.

Default: `"de"`

list-bullet-indent length

Set the List indentation.

Default: `1.5em`

list-numbered-indent `length`

Set the Enum indentation.

Default: `1.5em`

outline-link-color `color`

Set the color of the Links in the outline

Default: black

page-margins `auto` or `relative` or `dictionary`

Set the margins of the document.

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

page-numbering-align `alignment`

Set the page numbering alignment.

Default: center

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

Set the page numbering style.

Default: `"1"`

page-paper `str`

Set the paper size.

Default: `"a4"`

par-first-line-indent `length`

Set the indentation of the first line of paragraphs.

Default: `1.8em`

par-justify `bool`

Set if document should be justified.

Default: `true`

par-spacing length

Set the spacing between paragraphs.

Default: `0.55em`

text-font str

Set the text font.

Default: `"New Computer Modern"`

text-size length

Set the text size. Headings are adjusted automatically.

Default: `10pt`

Constructor functions

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

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

abstract

This function will display an abstract section with a title and content.

Parameters

```
abstract(  
    title: string | content,  
    alignment: alignment,  
    outlined: bool,  
    numbering: numbering,  
    width: length,  
    content: string | content  
)
```

title string or content

The title of the abstract.

Default: "Abstract"

alignment alignment

The alignment of the abstract.

Default: left

outlined bool

If the heading should be outlined.

Default: true

numbering numbering

The numbering of the heading.

Default: none

width `length`

The width of the abstract block.

Default: `auto`

content `string` or `content`

The content of the abstract.

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:

```
1 #ctable(t Typst
2   cols:"l|cr",
3   [A], [B], [C],
4   ..range(1,16).map(str),
5 )
```

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 linesyle of the table, especially the top and bottom horizontal lines.

Default: `.75pt`

middle-stroke `length`

The linesyle of the middle horizontal line.

Default: `.6pt`

vertical-stroke `length`

The linesyle 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:

```
1 #fig-outline()
```

 Typst

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:

```
1 #maketitle(t Typst
2   title: "The Title of the Paper",
3   authors: ("Authors Name",),
4   date: "2025-01-01",
5 )
```

Parameters

```
maketitle(
  title: string content,
  authors: array,
  date: string content datetime,
  metadata: bool,
  spacing: dict
) -> 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. This does not work when `article` sets columns before!!!

Default: false

spacing dict

The vertical spacing above and below the titleblock.

Default: (above: 60pt, below: 20pt)

shortcap

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

Example usage:

```
1 #figure(  
2     rect(),  
3     caption: shortcap("Short caption", "Long caption"),  
4 )
```

 Typst

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:

```
1 #tab-outline()
```

 Typst

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.

Utility functions

These functions can be used to perform certain tasks in the document. These functions will help you style certain elements of the document, where otherwise complicated functions would be needed.

- `balance()`

Variables

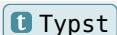
- `vspace`

balance

Balance the content of columns. Have a multicolumn layout with almost equal height columns. Credits go to: <https://github.com/typst/typst/issues/466>

Example usage:

```
1 #balance(columns(n)[#lorem(80)])
```



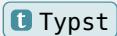
Parameters

`balance(content) -> content`

vspace space

A vertical space, which is weakly enforced. This is useful to add space between paragraphs if the default spacing is not sufficient and the same space should be used throughout the document. By using this function instead of regular `v(xem)` you can ensure the same distance throughout the document.

```
1 #lorem(50) // Some paragraph
2 #vspace // Add some space
3 #lorem(50) // Next paragraph
```



If the space is not as large as you want it to be, you can set the value in the beginning of the document with `#let vspace = v(1.5em)`