Package 'rotulador'

February 2, 2026

Title Useful Functions for Programming and Generating Documents

Version 1.0.0

Description Tools to help developers and producers manipulate R objects and outputs. It includes tools for displaying results and objects, and for formatting them in the correct format.

License MIT + file LICENSE

URL https://github.com/TractorTom/rotulador, https://tractortom.github.io/rotulador/

BugReports https://github.com/TractorTom/rotulador/issues

Depends R (>= 3.6)

Imports checkmate, clipr, knitr, rmarkdown, utils

Suggests htm2txt, pdftools, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Author Tanguy Barthelemy [aut, cre]

Maintainer Tanguy Barthelemy <tangbarth@hotmail.fr>

Repository CRAN

Date/Publication 2026-02-02 10:10:02 UTC

Contents

`create_preamble_tex` *Create preamble .tex for code font*

Description

Create preamble .tex for code font

Usage

```
create_preamble_tex(font_size = 12, monofont_path = get_fira_path())
```

Arguments

`font_size`	a numeric. The font size, only available in pdf format.
`monofont_path`	a string. The path to the font used to render code chunks. It should link to a `.ttf` file. Only available in pdf format.

Value

a vector of characters representing an Rmd file (each element being a line)

Examples

```
create_preamble_tex()
create_preamble_tex(font_size = 18.0)
```

`generate_chunk_header` *Generate R chunk header*

Description

Function to generate R chunk header for rmarkdown rendering in different output

Usage

```
generate_chunk_header(...)
```

Arguments

`...`	The different options for R code chunks.

Details

To get the list of all accepted options, you can call `names(knitr::opts_chunk$get())` and to get the default values you can call `knitr::opts_chunk$get()`.

More information in the function #' opts_chunk or directly https://yihui.org/knitr/options/#chunk-options to see all available options and their descriptions.

Value

a string of length 1

Examples

```
generate_chunk_header()
generate_chunk_header(eval = TRUE, echo = TRUE)
generate_chunk_header(results = "asis")
generate_chunk_header(fig.width = "4px", fig.height = "3px")
```

`generate_rmd_file` *Generate Rmd file*

Description

This function creates the Rmd file which will be rendered in a specific format.

Usage

```
generate_rmd_file(
  content,
  output_format = c("word", "pdf", "html", "word_document", "pdf_document",
    "html_document"),
  code = TRUE,
  ...
)
```

Arguments

`content`	a string. The body of the Rmd file (for example code or text)
`output_format`	a string representing the output format. The values `"pdf"`, `"html"` or `"word"` and their knitr equivalent `"pdf_document"`, `"html_document"` or `"word_document"` are accepted.
`code`	a boolean. Should the `content` string have to be inserted in R chunk or is it just text? Default is TRUE (so the `content` will be inserted in R chunk).
`...`	other arguments passed to R chunk (for example `eval = TRUE`, `echo = FALSE`...)

Details

More information about the argument ... in the documentation of the function `render_code`.

Value

a vector of characters representing an Rmd file (each element being a line)

Examples

```
generate_rmd_file(content = "Bonjour tout le monde",
                  code = FALSE,
                  output_format = "word")
generate_rmd_file(content = "print(AirPassengers)",
                  code = TRUE,
                  output_format = "pdf",
                  eval = TRUE,
                  echo = FALSE)
generate_rmd_file(content = "plot(AirPassengers)",
                  code = TRUE,
                  output_format = "html_document",
                  eval = FALSE,
                  echo = TRUE)
```

---

get_fira_path *The path to the font Fira Code*

Description

This function returns the path to the font Fira Code installed with the package rotulador.

Usage

```
get_fira_path()
```

Details

This function helps the other functions to find the path to the font Fira Code to render documents and use this font by default for code chunks.

Value

a character vector of length 1 representing the path

Examples

```
get_fira_path()
```

get_latex_engine *The latex engine*

Description

This function returns the latex engine available to render .tex file into pdf.

Usage

```
get_latex_engine()
```

Details

If several latex engine are available, the choice will be done in this order:

- xelatex;
- lualatex;
- pdflatex;
- tectonic.

Value

a character vector of length 1 representing the latex engine.

Examples

```
get_latex_engine()
```

get_word_template_path *The path to the word template*

Description

This function returns the path to the word template installed with the package rotulador.

Usage

```
get_word_template_path()
```

Details

This function helps the other functions to find the template of the word document used to render in `.docx` output.

Value

a character vector of length 1 representing the path

Examples

```
get_word_template_path()
```

`render_code` *Generate a file with formatted code*

Description

Format a piece of code to copy it into an email, a pdf, a document, etc.

Usage

```
render_code(
  output_format = c("word", "pdf", "html", "word_document", "pdf_document",
    "html_document"),
  browser = getOption("browser"),
  font_size = 12,
  code = TRUE,
  open = TRUE,
  monofont_path = get_fira_path(),
  word_template_path = get_word_template_path(),
  ...
)
```

Arguments

`output_format`	a string representing the output format. The values "pdf", "html" or "word" and their knitr equivalent "pdf_document", "html_document" or "word_document" are accepted.
`browser`	a string. The path to the browser which will open the generated file format
`font_size`	a numeric. The font size, only available in pdf format.
`code`	a boolean. Should the copied content have to be inserted in R chunk or is it just text? Default is TRUE (so the copied content will be inserted in R chunk).
`open`	a boolean. Default is TRUE meaning that the document will open automatically after being generated.
`monofont_path`	a string. The path to the font used to render code chunks. It should link to a `.ttf` file. Only available in pdf format.
`word_template_path`	a string. The path to the word template file used when rendering with word. By default, the template used is the one included in the package. Only used with word output.
`...`	other arguments passed to R chunk (for example eval = TRUE, echo = FALSE...)

Details

This function allows the user to generate formatted code (for email, document, copy, message, etc.) on the fly.

It accepts mainly word, pdf and html formats, but any format accepted by rmarkdown on the computer.

To use this function, simply copy a piece of code and run `render_code()` with the arguments that interest us. If you want content that is not R code, use the `code` argument to `FALSE`. In pdf format, you can change the font size using the `font_size` argument. Also, you can change the browser that opens the file by default with the `browser` argument.

With the argument `...`, you can specify knitr arguments to be included in the chunk. For example, you can add `eval = TRUE` (if you want the R code to be evaluated (and the result displayed)), `echo = FALSE` (if you don't want to display the code)...

More information in the function `opts_chunk` or directly `https://yihui.org/knitr/options/#chunk-options` to see all available options and their descriptions.

If the `open` argument is set to `FALSE` then the `browser` argument will be ignored.

Value

This function returns invisibly (with `invisible()`) a vector of length two with two element:

- the path of the created rmarkdown (template) document (`.Rmd`);
- the path of the created output (in the format `.pdf`, `.docx` or `.html`).

Examples

```
# Copy a snippet of code
if (clipr::clipr_available()) {
    clipr::write_clip("plot(AirPassengers)", allow_non_interactive = TRUE)
}

render_code(
    output_format = "word",
    echo = TRUE
)

render_code(
    output_format = "html",
    eval = FALSE
)


render_code(
    output = "pdf",
    eval = TRUE,
    font_size = 16
)
```

Index