

Package: highlighter (via r-universe)

September 3, 2024

Title Code Syntax Highlighting using the 'Prism.js' Library

Version 0.2

Description Code Syntax Highlighting made easy for code snippets or complete files. Whether you're documenting your data analysis or creating interactive 'shiny' apps.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports cli, digest, glue, htmltools, htmlwidgets, rlang, shiny

Suggests knitr, lintr, rmarkdown, spelling, testthat (>= 3.0.0)

VignetteBuilder knitr

URL <https://federiva.github.io/highlighter/>

BugReports <https://github.com/federiva/highlighter/issues>

Config/testthat/edition 3

Repository <https://federiva.r-universe.dev>

RemoteUrl <https://github.com/federiva/highlighter>

RemoteRef HEAD

RemoteSha db06d49ac018ee04deba97a0d2f570322386bbd

Contents

get_available_languages	2
get_available_themes	2
highlight	2
highlighter	3
highlighterOutput	4
highlight_file	5
line_number	5

Index	7
--------------	----------

get_available_languages

Lists the current available Languages

Description

List the available languages that can be used to highlight

Usage

```
get_available_languages()
```

Value

A character vector that contains the programming languages available to highlight.

get_available_themes *Lists the current available themes*

Description

List the available themes that can be used with highlighter

Usage

```
get_available_themes()
```

Value

A character vector with the names of the themes available.

highlight *Line highlight plugin*

Description

Line highlight plugin

Usage

```
highlight(range)
```

Arguments

range A character indicating the range to be used, for example 2-5 will highlight from 2 up to 5. Also you can highlight two or more ranges in the following way 2-5,10-13,19.

Value

A list (named) with the name of the plugin and the range passed by the function

Examples

```
if (interactive()) {
  highlighter::highlighter(
    "print('Hello, world!')\ncat <- \"Aristofanes\"\nstr(some_variable)",
    language = "r",
    plugins = list(
      highlight(
        range = "1-2"
      )
    )
  )
}
```

highlighter

Highlighter

Description

Highlights code

Usage

```
highlighter(
  code,
  language = "r",
  theme = "default",
  plugins = NULL,
  width = "100%",
  height = "auto",
  elementId = NULL
)
```

Arguments

code The code to be highlighted
language The programming language chosen to be highlighted

theme	A character. Indicating which theme will be used
plugins	Optional. A list of plugins to be used
width	Optional. The width to be used by the widget
height	Optional. The height to be used by the widget
elementId	Optional. The DOM element id to be used by the widget

Value

An object of class `highlighter`

See Also

[get_available_languages\(\)](#) for available languages, [get_available_themes\(\)](#) for available themes

Examples

```
# Highlight R code
if (interactive()) {
  highlighter("print('Hello, world!')", language = "r")
}
```

highlighterOutput *Shiny bindings for highlighter*

Description

Output and render functions for using `highlighter` within Shiny applications and interactive Rmd documents.

Usage

```
highlighterOutput(outputId, width = "100%", height = "auto")
```

```
renderHighlighter(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

outputId	output variable to read from
width, height	Must be a valid CSS unit (like '100%', '400px', 'auto') or a number, which will be coerced to a string and have 'px' appended.
expr	An expression that generates a <code>highlighter</code>
env	The environment in which to evaluate <code>expr</code> .
quoted	Is <code>expr</code> a quoted expression (with <code>quote()</code>)? This is useful if you want to save an expression in a variable.

Value

An object of class shiny.tag.list
An object of class shiny.render.function

highlight_file *Highlight Syntax of a File*

Description

Highlights the content of a given file according to the source language, theme and plugins used.

Usage

```
highlight_file(file_path, language = NULL, plugins = NULL, theme = "default")
```

Arguments

file_path	The path to the file to be highlighted
language	The programming language chosen to be highlighted
plugins	Optional. A list of plugins to be used
theme	A character. Indicating which theme will be used

Value

An object of class highlighter

See Also

[get_available_languages\(\)](#) for available languages, [get_available_themes\(\)](#) for available themes

line_number *Line Number plugin*

Description

Line Number plugin

Usage

```
line_number(use_line_number = TRUE, start_from = 1)
```

Arguments

`use_line_number` Logical
`start_from` A numeric indicating where to start to count from

Value

A list (named) with the name of the plugin, the class passed to the render function and the line number in which the line numbers will start from (optional).

Examples

```
if (interactive()) {  
  highlighter::highlighter(  
    "print('Hello, world!')\ncat <- \"Aristofanes\"\nstr(some_variable)",  
    language = "r",  
    plugins = list(  
      line_number(  
        use_line_number = TRUE,  
        start_from = 2  
      )  
    )  
  )  
}
```

Index

`get_available_languages`, 2
`get_available_languages()`, 4, 5
`get_available_themes`, 2
`get_available_themes()`, 4, 5

`highlight`, 2
`highlight_file`, 5
`highlighter`, 3
`highlighterOutput`, 4

`line_number`, 5

`renderHighlighter (highlighterOutput)`, 4