

Package ‘gitlink’

July 22, 2025

Title Add 'Git' Links to Your Web Based Assets

Version 0.1.3

Description Provides helpers to add 'Git' links to 'shiny' applications, 'rmarkdown' documents, and other 'HTML' based resources. This is most commonly used for 'GitHub' ribbons.

License MIT + file LICENSE

URL <https://github.com/colearendt/gitlink>

BugReports <https://github.com/colearendt/gitlink/issues>

Depends R (>= 3.1)

Imports htmltools (>= 0.3.6), rlang (>= 0.3.1)

Suggests covr (>= 3.0.1), spelling (>= 2.0), testthat (>= 2.0.0.9000), rmarkdown, fs, rprojroot, shiny

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Author Cole Arendt [aut, cre],
RStudio [cph, fnd]

Maintainer Cole Arendt <cole@rstudio.com>

Repository CRAN

Date/Publication 2019-07-23 12:20:02 UTC

Contents

ribbon_css	2
ribbon_img	3
Index	5

 ribbon_css

Build a GitHub Ribbon with CSS

Description

A customizable CSS ribbon with a link to "Fork me on GitHub." CSS passed to the function will override the default CSS to give you the ultimate flexibility in customizing the ribbon to fit your needs.

Usage

```
ribbon_css(link, position = c("left", "right"), color = "white",
  font_color = "black", border_color = "white",
  text = "Fork me on GitHub", ..., fade = TRUE, link_css = list(),
  parent_css = list(), hover_css = list())
```

Arguments

link	The URL that the ribbon will link to
position	The position to place the ribbon in (either "left" or "right"). Default: right
color	Any css-valid color specification for the background of the ribbon
font_color	Any css-valid color specification for the text of the ribbon
border_color	Any css-valid color specification for the border of the ribbon
text	The text to show on the ribbon
...	key=value CSS passed along to the ribbon div
fade	boolean. Whether or not the default opacity should be set to < 1, but transition on hover. Default TRUE
link_css	A list of key=value CSS passed along to the link text
parent_css	A list of key=value CSS passed along to the parent div of the ribbon
hover_css	A list of key=value CSS passed along to the .ribbon:hover CSS

Details

The benefits of using CSS are:

- any color you can dream up
- any font / font color you want
- any text you want
- customizable size / location / etc.
- do not have to worry about DPI and image resolution
- link is only clickable on the banner itself
- fancy fade / hover CSS out of the box

Value

HTML that can be injected into any output

Examples

```
ribbon_css("https://github.com/colearendt/gitlink")
ribbon_css("https://github.com/colearendt/gitlink", position = "left", color = "#e4e4e4" )
ribbon_css("https://github.com/colearendt/gitlink", position = "left", color = "#eafffc")

# make default opacity 1
ribbon_css("https://github.com/colearendt/gitlink", fade = FALSE)

# customize the hover css
ribbon_css("https://github.com/colearendt/gitlink", hover_css = list("opacity" = "0.9"))

url <- "https://github.com/colearendt/gitlink"

# this one is particularly ugly, but proves a point
ribbon_css(url, parent_css = list("background-color" = "red"))

# modify the location for a flexdashboard
ribbon_css(url, parent_css = list(top = "50px", "z-index" = "10"))
```

 ribbon_img

Build a GitHub Ribbon from an Image

Description

Publicly available ribbon images began the "GitHub ribbon craze." This is a helper function that client-side retrieves these images. As a result, only certain colors are available. [ribbon_css](#) is usually preferred for many reasons.

Usage

```
ribbon_img(link, position = "right", color = "white")

get_color_lookup()
```

Arguments

link	The URL that the ribbon will link to
position	The position to place the ribbon in (either "left" or "right")
color	The color for the ribbon. One of <code>get_color_lookup()</code>

Details

NOTE that this creates a dependency on these image files, which are hosted publicly by GitHub on AWS.

Value

HTML that can be injected into any HTML output

See Also

[ribbon_css](#)

Index

`get_color_lookup(ribbon_img)`, [3](#)

`ribbon_css`, [2](#), [3](#), [4](#)

`ribbon_img`, [3](#)