

Package ‘tippy’

May 8, 2026

Title Add Tooltips to 'R markdown' Documents or 'Shiny' Apps

Version 0.1.0

Description 'Htmlwidget' of 'Tippyjs' to add tooltips to 'Shiny' apps and 'R markdown' documents.

Depends R (>= 3.4.0)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports htmlwidgets, htmltools, shiny, jsonlite

RoxygenNote 7.1.1

URL <https://tippy.john-coene.com/>

BugReports <https://github.com/JohnCoene/tippy/issues>

NeedsCompilation no

Author John Coene [aut, cre]

Maintainer John Coene <jcoenep@gmail.com>

Repository CRAN

Date/Publication 2021-01-11 21:10:02 UTC

Contents

tippy-shiny	2
tippy_funcs	2
use_tippy_funcs	3
Index	5

 tippy-shiny

Shiny bindings for tippy

Description

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

Usage

```
tippyOutput(outputId, width = "100%", height = "400px")
```

```
renderTippy(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 tippy
env	The environment in which to evaluate expr.
quoted	Is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable.

 tippy_funcs

Create a tooltip

Description

Add tooltips to your document.

Usage

```
tippy(text, tooltip, ..., elementId = NULL)
```

```
tippy_this(elementId, tooltip, ...)
```

```
with_tippy(element, tooltip, ...)
```

Arguments

text	Element text.
tooltip	Element tooltip.
...	Any other options from the official documentation .
elementId	string id as a valid CSS element id.
element	An object of class shiny.tag.

See Also

[official documentation](#)

Examples

```
tippy("Hover me!", tooltip = "Hi, I'm the tooltip!")
tippy("Hover me!", tooltip = "Hi, I'm the tooltip!", placement = "right",
      theme = "light")
tippy("Hover me!", tooltip = "Hi, I'm the tooltip!", animation = "scale",
      duration = 1000, placement = "bottom")
tippy("Click me!", tooltip = "Hi, I'm the tooltip!", trigger = "click",
      theme = "light")

# use tooltip on other elements.
if(interactive()){
  library(shiny)

  shinyApp(
    ui = fluidPage(
      with_tippy(textInput("input", "input with tooltip"), "Input text", placement = "right")
    ),
    server = function(input, output) {}
  )
}
```

use_tippy_funcs

Use tippy

Description

Dynamically use tippy.

Usage

```
use_tippy()

call_tippy(target, ...)

tippy_class(class, ...)
```

Arguments

target	Target element.
...	Any option from the official documentation .
class	Class of elements to apply tooltip to.

Functions

- `use_tippy` Includes `tippy.js` in header.
- `call_tippy` call `tippy` on specific target element(s), place after elements to be targeted.

See Also

[official documentation](#)

Examples

```
if(interactive()){
  library(shiny)

  shinyApp(
    ui = fluidPage(
      p("Some text", class = "tooltip"),
      p("Some text", class = "tooltip"),
      p("Some text", class = "tooltip"),
      p("Some text", class = "tooltip"),
      p("Some text", class = "tooltip"),
      p("Some text", class = "tooltip"),
      tippy_class("tooltip", content = "Hi!") # all elements with class
    ),
    server = function(input, output) {}
  )
}
```

Index

`call_tippy (use_tippy_funcs)`, 3

`renderTippy (tippy-shiny)`, 2

`tippy (tippy_funcs)`, 2

`tippy-shiny`, 2

`tippy_class (use_tippy_funcs)`, 3

`tippy_funcs`, 2

`tippy_this (tippy_funcs)`, 2

`tippyOutput (tippy-shiny)`, 2

`use_tippy (use_tippy_funcs)`, 3

`use_tippy_funcs`, 3

`with_tippy (tippy_funcs)`, 2