Package 'swipeR'

December 12, 2022

Type Package

Title Carousels using the 'JavaScript' Library 'Swiper'

Version 0.1.0

Description Create carousels using the 'JavaScript' library 'Swiper' and the package 'htmlwidgets'. The carousels can be displayed in the 'RStudio' viewer pane, in 'Shiny' applications and in 'R markdown' documents.

License GPL-3

URL https://github.com/stla/swipeR

BugReports https://github.com/stla/swipeR/issues

Imports htmltools, htmlwidgets

Suggests ggplot2, ggthemes, shiny, shinyWidgets

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Stéphane Laurent [aut, cre],
Vladimir Kharlampidi [cph] ('swiper' library)

Maintainer Stéphane Laurent <laurent_step@outlook.fr>

Repository CRAN

Date/Publication 2022-12-12 12:10:02 UTC

R topics documented:

swipeR *HTML widget displaying a carousel*

Description

Create a HTML widget displaying a carousel.

Usage

```
swipeR(
  wrapper,
  width = "100%",
  height = "100%",
  navigationColor = "white",
  paginationColor = "white",
  bulletsSize = "8px",
  id = NULL,
  direction = "horizontal",
  effect = "slide",
 cubeEffect = list(shadow = TRUE, slidesShadow = TRUE, shadowOffset = 20, shadowScale =
    0.94),
  initialSlide = 1,
  zoom = FALSE,
  loop = FALSE,
  rewind = FALSE,
  slidesPerView = 1,
  spaceBetween = 30,
  speed = 300,
  scrollbar = FALSE,
  autoplay = FALSE,
  thumbs = FALSE,
  thumbsPerView = 2,
  thumbsHeight = "60px",
  on = NULL,
  elementId = NULL
)
```

Arguments

wrapper	HTML div element created with swipeRwrapper
width, height	dimensions
navigationColor	color for the navigation arrows
paginationColor	color for the pagination bullets
bulletsSize	size of the pagination bullets

id	a HTML id for the carousel
direction	direction of the slide show, "horizontal" or "vertical"
effect	transition effect, can be "slide", "fade", "cube", "coverflow", "flip", or "cards"
cubeEffect	list of settings for the cube when effect="cube"
initialSlide	index of the first slide to be shown
zoom	Boolean, whether to enable the zoom on slide's double tap; all zoomable slides must be wrapped in a div with swiper-zoom-container class
loop	Boolean, whether to enable the continuous loop mode
rewind	Boolean; if TRUE, clicking "next" navigation button when on last slide will slide back to the first slide, and clicking "prev" navigation button when on first slide will style forward to the last slide
slidesPerView	number of slides per view
spaceBetween	distance between slides in pixels
speed	transition speed in milliseconds
scrollbar	Boolean, whether to enable a scrollbar for navigation
autoplay	Boolean, whether to autoplay the slide show
thumbs	Boolean, whether to display thumbs of the slides
thumbsPerView	number of thumbs per view
thumbsHeight	height of the thumbs carousel
on	named list of event listeners
elementId	a HTML id for the container

Value

A htmlwidget object.

Examples

```
library(swipeR)
library(htmltools)

wrapper <- swipeRwrapper(
  tags$img(src = "https://swiperjs.com/demos/images/nature-1.jpg"),
  tags$img(src = "https://swiperjs.com/demos/images/nature-2.jpg"),
  tags$img(src = "https://swiperjs.com/demos/images/nature-3.jpg"),
  tags$img(src = "https://swiperjs.com/demos/images/nature-4.jpg"),
  tags$img(src = "https://swiperjs.com/demos/images/nature-5.jpg"),
  tags$img(src = "https://swiperjs.com/demos/images/nature-6.jpg"),
  tags$img(src = "https://swiperjs.com/demos/images/nature-7.jpg"),
  tags$img(src = "https://swiperjs.com/demos/images/nature-8.jpg")
)

swipeR(
  wrapper, height = "400px", width = "70%", thumbs = TRUE,
```

```
  on = list(reachEnd = htmlwidgets::JS("function() {alert('the end');}"))
)

# Shiny example ####
library(swipeR)
library(shiny)
library(ggplot2)

wrapper <- swipeRwrapper(
  div(
    plotOutput("ggplot1", width = "70%", height = "400px"),
    align = "center"
  ),
  div(
    plotOutput("ggplot2", width = "70%", height = "400px"),
    align = "center"
  ),
  div(
    plotOutput("ggplot3", width = "70%", height = "400px"),
    align = "center"
  ),
  div(
    plotOutput("ggplot4", width = "70%", height = "400px"),
    align = "center"
  )
)

ui <- fluidPage(
  tags$head(
    tags$style(HTML(
      ".shiny-plot-output {border: 2px solid royalblue;}"
    ))
  ),
  br(),
  fluidRow(
    column(
      12,
      swipeR(
        wrapper, height = "450px", width = "80%", effect = "cube", speed = 2000,
        navigationColor = "black", rewind = TRUE, id = "CAROUSEL"
      )
    ),
    column(
      12,
      br(), br(), br(),
    ),
    column(
      3, align = "center",
      actionButton(
        "btn1", "Scatter plot", class = "btn-primary",
        onclick = "document.getElementById('CAROUSEL').swiper.slideTo(0);"
      )
    ),
```

```
      column(
        3, align = "center",
        actionButton(
          "btn2", "Line chart", class = "btn-primary",
          onclick = "document.getElementById('CAROUSEL').swiper.slideTo(1);"
        )
      ),
      column(
        3, align = "center",
        actionButton(
          "btn3", "Bar chart", class = "btn-primary",
          onclick = "document.getElementById('CAROUSEL').swiper.slideTo(2);"
        )
      ),
      column(
        3, align = "center",
        actionButton(
          "btn4", "Boxplots", class = "btn-primary",
          onclick = "document.getElementById('CAROUSEL').swiper.slideTo(3);"
        )
      )
    )
  )

  server <- function(input, output, session) {
    output[["ggplot1"]] <- renderPlot({
      ggplot(mtcars, aes(wt, mpg)) + geom_point() +
        theme(panel.border = element_rect(fill = NA, color = "firebrick"))
    })
    output[["ggplot2"]] <- renderPlot({
      ggplot(economics, aes(date, unemploy)) + geom_line()
    })
    output[["ggplot3"]] <- renderPlot({
      ggplot(mpg, aes(class)) + geom_bar()
    })
    output[["ggplot4"]] <- renderPlot({
      ggplot(mpg, aes(class, hwy)) + geom_boxplot()
    })
  }

  if(interactive()) shinyApp(ui, server)


  # other Shiny example ####
  library(swipeR)
  library(shiny)
  library(shinyWidgets)
  library(ggplot2)
  library(ggthemes)

  wrapper <- swipeRwrapper(
    div(
      fluidRow(
```

```
      column(
        6,
        awesomeRadio(
          "theme", "Choose a theme",
          c(
            "Calc",
            "Clean",
            "Economist",
            "Excel",
            "FiveThirtyEight",
            "Foundation",
            "Google Docs",
            "Highcharts",
            "Pander",
            "Solarized",
            "Stata",
            "Wall Street"
          )
        )
      ),
      column(
        6,
        tags$p("The Shiny slider does not work here..."),
        tags$label("Base font size"),
        tags$input(
          type = "range", min = "10", max = "20", value = "12",
          oninput =
            "this.nextElementSibling.value = this.value;
             Shiny.setInputValue('slider', this.value);"
        ),
        tags$output("12", style = "font-weight: bold; color: blue"),
        br(), hr(), br(),
        materialSwitch("facets", "Facets?", status = "info"),
        conditionalPanel(
          condition = "input.facets",
          awesomeRadio(
            "direction", label = NULL, status = "info",
            choices = c("by row" = "row", "by column" = "column"),
          )
        ),
        br(), hr(), br(),
        actionButton(
          "btn", "Add slide", class = "btn-primary btn-block",
          onclick = "document.getElementById('SWIPER').swiper.appendSlide(
            '<div class=\"swiper-slide rlogo\"></div>');
            Shiny.setInputValue('newslide', true, {priority: 'event'});"
        )
      )
    ),
    style = "margin-left: 10%; margin-right: 10%; font-size: 2rem;"
  ),
  div(
    plotOutput("ggplot", width = "85%", height = "400px"),
```

```
    align = "center"
  )
)

ui <- fluidPage(
  tags$head(
    tags$style(HTML(
      ".shiny-plot-output {
          border: 2px solid royalblue;
      }
      .shiny-text-output {
          font-size: 30px;
          font-style: italic;
      }
      .recalculating {
         display: none; /* otherwise there's a flash */
      }
      .rlogo {
         width: 100%;
         height: 100%;
         background-image: url(https://www.r-project.org/logo/Rlogo.png);
         background-repeat: no-repeat;
         background-size: contain;
         background-position: center;
      }"
    ))
  ),
  br(), br(), br(),
  fluidRow(
    column(
      12,
      swipeR(
        wrapper, id = "SWIPER", effect = "flip", rewind = TRUE,
        height = "450px", width = "90%",
        navigationColor = "black", paginationColor = "black",
        on = list(
          afterInit = htmlwidgets::JS(
            "function(swiper) {
               setTimeout(function(){ Shiny.setInputValue('index', 1); }, 0);
            }"
          ),
          slideChange = htmlwidgets::JS(
            "function(swiper) {
               Shiny.setInputValue('index', swiper.activeIndex + 1);
            }"
          )
        )
      )
    ),
    column(
      12,
      textOutput("slideIndex")
    )
```

```
  )
)

server <- function(input, output, session) {

  ggtheme <- reactive({
    size <- input[["slider"]]
    size <- if(is.null(size)) 12 else as.integer(size)
    switch(
      input[["theme"]],
      "Calc"            = theme_calc(base_size = size),
      "Clean"           = theme_clean(base_size = size),
      "Economist"       = theme_economist(base_size = size),
      "Excel"           = theme_excel_new(base_size = size),
      "FiveThirtyEight" = theme_fivethirtyeight(base_size = size),
      "Foundation"      = theme_foundation(base_size = size),
      "Google Docs"     = theme_gdocs(base_size = size),
      "Highcharts"      = theme_hc(base_size = size),
      "Pander"          = theme_pander(base_size = size),
      "Solarized"       = theme_solarized(base_size = size),
      "Stata"           = theme_stata(base_size = size),
      "Wall Street"     = theme_wsj(base_size = size)
    )
  })

  output[["ggplot"]] <- renderPlot({
    gg <- ggplot(iris, aes(x = Sepal.Length, y = Petal.Length, color = Species)) +
      geom_point(size = 6) + ggtheme()
    if(input[["facets"]]) {
      if(input[["direction"]] == "row") {
        gg <- gg + facet_grid(rows = vars(Species))
      } else {
        gg <- gg + facet_grid(cols = vars(Species))
      }
    }
    gg
  })

  nSlides <- reactiveVal(2)
  observeEvent(input[["newslide"]], {
    nSlides(nSlides() + 1)
  })

  output[["slideIndex"]] <- renderText({
    paste0(input[["index"]], "/", nSlides())
  })

}

if(interactive()) shinyApp(ui, server)
```

swipeR-shiny *Shiny bindings for swipeR carousels*

Description

Output and render functions for using swipeR within Shiny applications.

Usage

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

renderSwipeR(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 swipeR carousel
env	the environment in which to evaluate expr
quoted	Boolean, whether expr is a quoted expression (with quote()); this is useful if you want to save an expression in a variable

Value

swipeROutput returns an output element that can be included in a Shiny UI, and renderSwipeR returns a shiny.render.function object that can be assigned to an output slot in a Shiny server.

swipeRwrapper *List of DOM elements for a carousel*

Description

Enclose a list of DOM elements in a HTML div element to be passed to the swipeR function.

Usage

```
swipeRwrapper(...)
```

Arguments

...	HTML elements, one for each slide

Value

A shiny.tag object.

Index