

Package: pixture (via r-universe)

May 18, 2026

Type Package

Title Responsive HTML Image Gallery

Version 0.2.1

Date 2025-11-09

Maintainer Roy Francis <roy.m.francis@outlook.com>

Description HTML image gallery in R with responsive layout and lightbox.

URL <https://github.com/royfrancis/pixture>
<https://royfrancis.r-universe.dev/pixture>
<http://royfrancis.github.io/pixture/>

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 4.0)

Imports htmlwidgets, shiny

Suggests testthat

BugReports <https://github.com/royfrancis/pixture/issues>

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Language en-US

Config/pak/sysreqs cmake make libuv1-dev zlib1g-dev

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

Date/Publication 2025-12-19 13:10:44 UTC

RemoteUrl <https://github.com/royfrancis/pixture>

RemoteRef main

RemoteSha 8072b6d7c0223c88ac3c0bd021a18d61257b2cf4

Contents

launchApp	2
pixcarousel	3
pixcarousel-shiny	5
pixfigure	6
pixfigure-shiny	7
pixgallery	8
pixgallery-shiny	10

Index	12
--------------	-----------

launchApp	<i>Interactive shiny app</i>
-----------	------------------------------

Description

Launches interactive shiny demo app

Usage

```
launchApp(app = "pixgallery", ...)
```

Arguments

app	A character string denoting the app. Options are "pixfigure", "pixgallery", "pixgallery-simple", "pixcarousel".
...	Parameters are passed to runApp().

Value

This function does not return anything

Examples

```
## Not run:
library(pixfigure)
launchApp()

## End(Not run)
```

pixcarousel	<i>Create a carousel</i>
-------------	--------------------------

Description

Creates an image carousel

Usage

```
pixcarousel(  
  path,  
  caption = NULL,  
  caption_valign = "none",  
  caption_halign = "left",  
  link = TRUE,  
  caption_lightbox = TRUE,  
  gap = "0px",  
  border_radius = "0px",  
  shuffle = FALSE,  
  fit = "cover",  
  position = "center",  
  carousel = list(),  
  lightbox = list(),  
  h = "400px",  
  width = NULL,  
  height = NULL,  
  elementId = NULL  
)
```

Arguments

path	A character vector of full paths to images.
caption	A character vector of captions for the images. If used, it must be equal to the length of path.
caption_valign	A character denoting position of the caption. Options are 'none', 'top', 'center' or 'bottom'.
caption_halign	A character denoting horizontal justification of the caption. Options are 'left', 'center' or 'right'.
link	A logical or character vector. What happens when you click on the thumbnail? TRUE opens up the lightbox, FALSE to disable the lightbox. A character vector of custom URLs equal to length of path.
caption_lightbox	A logical or character vector. TRUE shows caption in lightbox, FALSE disables caption in lightbox. A character vector of characters equal to length of path.
gap	A character denoting spacing between thumbnails in valid CSS units. Applied to left and right of each image.

<code>border_radius</code>	A character denoting corner radius of the carousel in valid CSS units.
<code>shuffle</code>	A logical indicating whether images are randomly shuffled.
<code>fit</code>	String. Passed to <code>object-fit</code> CSS property.
<code>position</code>	String. Passed to <code>object-position</code> CSS property.
<code>carousel</code>	A list of options to customize the carousel. See details.
<code>lightbox</code>	A list of options to customize the lightbox. See details.
<code>h</code>	A character denoting height of the image thumbnails in valid CSS units.
<code>width</code>	A character denoting width of the widget in valid CSS units.
<code>height</code>	A character denoting height of the widget in valid CSS units.
<code>elementId</code>	A character string denoting parent container ID.

Details

carousel options

The carousel can be customized by passing a list of options to the `carousel` parameter. Here is an example;

```
pixcarousel(  
  paths,  
  carousel = list(  
    slidesToShow = 3,  
    slidesToScroll = 2  
  )  
)
```

The available options for carousel can be found at <https://github.com/NickPiscitelli/Glider.js> or <https://nickpiscitelli.github.io/Glider.js/>.

lightbox options:

The lightbox can be customized by passing a list of options to the `lightbox` parameter. Here is an example;

```
pixcarousel(  
  paths,  
  lightbox = list(  
    touchNavigation = FALSE,  
    loop = TRUE  
  )  
)
```

The available options for glightbox can be found at <https://github.com/biati-digital/glightbox?tab=readme-ov-file#lightbox-options>.

Examples

```
library(pixtuture)
paths <- c(
  "https://images.pexels.com/photos/572897/pexels-photo-572897.jpeg",
  "https://images.pexels.com/photos/7604425/pexels-photo-7604425.jpeg",
  "https://images.pexels.com/photos/4666748/pexels-photo-4666748.jpeg",
  "https://images.pexels.com/photos/4932184/pexels-photo-4932184.jpeg",
  "https://images.pexels.com/photos/4210900/pexels-photo-4210900.jpeg",
  "https://images.pexels.com/photos/3126574/pexels-photo-3126574.jpeg",
  "https://images.pexels.com/photos/167699/pexels-photo-167699.jpeg",
  "https://images.pexels.com/photos/1376201/pexels-photo-1376201.jpeg"
)
pixcarousel(paths)

# local example
## Not run:
library(pixtuture)
paths <- list.files(path=system.file("extdata/images", package="pixtuture"), full.names=TRUE)
pixcarousel(paths)

## End(Not run)
```

pixcarousel-shiny *Shiny bindings for pixcarousel*

Description

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

Usage

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

renderPixcarousel(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 pixcarousel
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.

 pixfigure

Create a simple layout of images

Description

Create a simple sequential layout of images

Usage

```

pixfigure(
  path,
  caption = NULL,
  link = TRUE,
  caption_lightbox = TRUE,
  h = "auto",
  w = "100%",
  fit = "cover",
  position = "center",
  lightbox = list(),
  width = "100%",
  height = "100%",
  elementId = NULL
)

```

Arguments

path	A character vector of full paths to images.
caption	A character vector of captions for the images. If used, it must be equal to the length of path.
link	A logical or character vector. What happens when you click on the image? TRUE opens up the lightbox, FALSE to disable the lightbox. A character vector of custom URLs equal to length of path.
caption_lightbox	A logical or character vector. TRUE shows caption in lightbox, FALSE disables caption in lightbox. A character vector of characters equal to length of path.
h	A character denoting height of the image in valid CSS units.
w	A character denoting width of the image in valid CSS units.
fit	String. Passed to object-fit CSS property.
position	String. Passed to object-position CSS property.
lightbox	A list of options to customize the lightbox. See details.
width	A character denoting width of the widget as a string in valid CSS units.
height	A character denoting height of the widget as a string in valid CSS units.
elementId	A character string denoting parent container ID.

Details

lightbox options

The lightbox can be customized by passing a list of options to the `lightbox` parameter. Here is an example;

```
pixfigure(  
  paths,  
  lightbox = list(  
    touchNavigation = FALSE,  
    loop = TRUE  
  )  
)
```

The available options for lightbox can be found at <https://github.com/biati-digital/glightbox?tab=readme-ov-file#lightbox-options>.

Examples

```
library(pixtuture)  
paths <- c(  
  "https://images.pexels.com/photos/572897/pexels-photo-572897.jpeg",  
  "https://images.pexels.com/photos/7604425/pexels-photo-7604425.jpeg",  
  "https://images.pexels.com/photos/4666748/pexels-photo-4666748.jpeg"  
)  
pixfigure(paths)  
  
# local example  
## Not run:  
library(pixtuture)  
paths <- list.files(path=system.file("extdata/images", package="pixtuture"), full.names=TRUE)  
pixfigure(paths)  
  
## End(Not run)
```

pixfigure-shiny

Shiny bindings for pixfigure

Description

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

Usage

```
pixfigureOutput(outputId, width = "100%", height = "auto")  
  
renderPixfigure(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 pixfigure
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.

pixgallery	<i>Create an image gallery</i>
------------	--------------------------------

Description

Create an image gallery

Usage

```

pixgallery(
  path,
  caption = NULL,
  caption_valign = "none",
  caption_halign = "left",
  link = TRUE,
  caption_lightbox = TRUE,
  h = NULL,
  w = NULL,
  gap = "5px",
  border_radius = "0px",
  layout = "grid",
  shuffle = FALSE,
  lightbox = list(),
  width = "100%",
  height = "100%",
  elementId = NULL
)

```

Arguments

path	A character vector of full paths to images.
caption	A character vector of captions for the images. If used, it must be equal to the length of path.
caption_valign	A character denoting position of the caption. Options are 'none', 'top', 'center', 'bottom' or 'below'.

caption_halign	A character denoting horizontal justification of the caption. Options are 'left', 'center' or 'right'.
link	A logical or character vector. What happens when you click on the thumbnail? TRUE opens up the lightbox, FALSE to disable the lightbox. A character vector of custom URLs equal to length of path.
caption_lightbox	A logical or character vector. TRUE shows caption in lightbox, FALSE disables caption in lightbox. A character vector of characters equal to length of path.
h	A character denoting height of the image thumbnails in valid CSS units.
w	A character denoting width of the image thumbnails in valid CSS units.
gap	A character denoting spacing between thumbnails in valid CSS units.
border_radius	A character denoting corner radius of image thumbnails in valid CSS units.
layout	A character denoting gallery layout type. Options are "grid", "fixed", "mosaic", "masonry", "justified", "scroll", "elastic", "rhombus" or "hexagon". See details.
shuffle	A logical indicating whether images are randomly shuffled.
lightbox	A list of options to customize the lightbox. See details.
width	A character denoting width of the widget in valid CSS units.
height	A character denoting height of the widget in valid CSS units.
elementId	A character string denoting parent container ID.

Details

layout options

grid: Grid responsive layout. Width is fluid to fit parent. When using caption_valign options 'top', 'bottom' or 'center', height cannot be adjusted.

fixed: Same as grid but with strictly fixed dimensions. Defaults to square image. When using caption_valign options 'top', 'bottom' or 'center', note that long captions may overflow. Manually set height in this case.

mosaic: Grid layout with mixed sizes.

masonry: Column layout where images flow vertically. Image height cannot be set.

justified: Similar to masonry but images flow horizontally. Image width cannot be set. When using caption_valign options 'top', 'bottom' or 'center', note that long captions are clipped. Option 'below' may cause images to be stretched if captions are too long.

elastic: Single row layout where images are magnified on hover. Image width cannot be set. Captions cannot be displayed on thumbnails.

scroll: Single row layout with horizontal scrolling. Images go offscreen. Thumbnail height and width can be set.

rhombus: Diamond shaped layout. When using caption_valign options 'top', 'bottom' or 'center', long captions are clipped. Option 'below' is disabled.

hexagon: Hexagon shaped layout. When using caption_valign options 'top', 'bottom' or 'center', long captions are clipped. Option 'below' is disabled.

lightbox options

The lightbox can be customized by passing a list of options to the lightbox parameter. Here is an example;

```

pixgallery(
  paths,
  lightbox = list(
    touchNavigation = FALSE,
    loop = TRUE
  )
)

```

The available options for glightbox can be found at <https://github.com/biati-digital/glightbox?tab=readme-ov-file#lightbox-options>.

Examples

```

library(pixtecture)
paths <- c(
  "https://images.pexels.com/photos/572897/pexels-photo-572897.jpeg",
  "https://images.pexels.com/photos/7604425/pexels-photo-7604425.jpeg",
  "https://images.pexels.com/photos/4666748/pexels-photo-4666748.jpeg",
  "https://images.pexels.com/photos/4932184/pexels-photo-4932184.jpeg",
  "https://images.pexels.com/photos/4210900/pexels-photo-4210900.jpeg",
  "https://images.pexels.com/photos/3126574/pexels-photo-3126574.jpeg",
  "https://images.pexels.com/photos/167699/pexels-photo-167699.jpeg",
  "https://images.pexels.com/photos/1376201/pexels-photo-1376201.jpeg"
)
pixgallery(paths)

# local example
## Not run:
library(pixtecture)
paths <- list.files(path=system.file("extdata/images", package="pixtecture"), full.names=TRUE)
pixgallery(paths)

## End(Not run)

```

pixgallery-shiny

Shiny bindings for pixgallery

Description

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

Usage

```

pixgalleryOutput(outputId, width = "100%", height = "100%")

renderPixgallery(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 pixgallery
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.

Index

launchApp, [2](#)

pixcarousel, [3](#)

pixcarousel-shiny, [5](#)

pixcarouselOutput (pixcarousel-shiny), [5](#)

pixfigure, [6](#)

pixfigure-shiny, [7](#)

pixfigureOutput (pixfigure-shiny), [7](#)

pixgallery, [8](#)

pixgallery-shiny, [10](#)

pixgalleryOutput (pixgallery-shiny), [10](#)

renderPixcarousel (pixcarousel-shiny), [5](#)

renderPixfigure (pixfigure-shiny), [7](#)

renderPixgallery (pixgallery-shiny), [10](#)