

Package: shinipsum (via r-universe)

October 17, 2026

Title Lorem-Ipsum-Like Helpers for Fast Shiny Prototyping

Version 0.1.1

Description Prototype your shiny apps quickly with these
Lorem-Ipsum-like Helpers.

License MIT + file LICENSE

URL <https://github.com/Thinkr-open/shinipsum>

BugReports <https://github.com/Thinkr-open/shinipsum/issues>

Depends R (>= 2.10)

Imports attempt, datasets, DT, dygraphs, ggplot2 (>= 3.0.0),
htmltools, magrittr, plotly, stats, utils

Suggests hexbin, MASS, testthat

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Config/Needs/website ThinkR-open/golemversetemplate

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev

Repository <https://thinkr-open.r-universe.dev>

Date/Publication 2026-09-17 07:13:07 UTC

RemoteUrl <https://github.com/ThinkR-open/shinipsum>

RemoteRef HEAD

RemoteSha 66ae9a5741c1dbbd5343c5f010f054b9006d919f

Contents

ipsum_examples	2
lorem	3
lorem_words	3
random_DT	3

random_dygraph	4
random_ggplot	4
random_ggplotly	5
random_image	5
random_image_ext	6
random_lm	7
random_mock	7
random_print	8
random_table	8
random_text	9
Index	10

ipsum_examples	<i>Get a shinipsum example</i>
----------------	--------------------------------

Description

Get a shinipsum example

Usage

ipsum_examples(which = NULL)

Arguments

which The example to run. If empty, all the available examples are listed.

Value

A path to the example.

Examples

ipsum_examples()

lorem	<i>Lorem text</i>
-------	-------------------

Description

A long lorem ipsum text

Usage

lorem

Format

An object of class character of length 1.

lorem_words	<i>Lorem test as vector</i>
-------------	-----------------------------

Description

A long vector of words

Usage

lorem_words

Format

An object of class character of length 13657.

random_DT	<i>A Random DT</i>
-----------	--------------------

Description

This function creates a random DT::datatable, and can be passed into renderDT & DTOutput.

Usage

```
random_DT(  
  nrow,  
  ncol,  
  type = c("random", "numeric", "character", "numchar"),  
  ...  
)
```

Arguments

nrow	number of row of the output
ncol	number of cols of the output
type	type of the columns, can be "random", "numeric", "character", "numchar". Default is random.
...	arguments to be passed to DT::datatable

Value

a DT

random_dygraph	<i>A Random Dygraph</i>
----------------	-------------------------

Description

This function returns a dygraph object, which can be passed to renderDygraph and dygraphOutput

Usage

```
random_dygraph(...)
```

Arguments

...	args passed to dygraph
-----	------------------------

Value

a dygraph

random_ggplot	<i>A Random ggplot</i>
---------------	------------------------

Description

This function returns a ggplot object, which can be passed to renderPlot and plotOutput

Usage

```
random_ggplot(  
  type = c("random", "point", "bar", "boxplot", "col", "tile", "line", "bin2d",  
    "contour", "density", "density_2d", "dotplot", "hex", "freqpoly", "histogram",  
    "ribbon", "raster", "violin", "ts"),  
  nBars = NULL  
)
```

Arguments

type	type of the geom. Can be any of "random", "point", "bar", "boxplot","col", "tile", "line", "bin2d", "contour", "density", "density_2d", "dotplot", "hex", "freqpoly", "histogram", "ribbon", "raster", "violin", "ts" (alias "timeseries") and defines the geom of the ggplot. Default is "random", and chooses a random geom for you. The "ts" type returns a time-series oriented plot, with a Date on the x axis.
nBars	number of bars to draw. Only valid together with type = "bar", and errors otherwise. When NULL (default) one of the built-in datasets is used instead.

Value

a ggplot

random_ggplotly	<i>A Random ggplotly</i>
-----------------	--------------------------

Description

This function returns a ggplotly object, which can be passed to renderPlotly and plotlyOutput

Usage

```
random_ggplotly(...)
```

Arguments

... arg to pass to random_ggplot.

Value

a ggplotly

random_image	<i>A Random Image</i>
--------------	-----------------------

Description

This function returns a random image that can be passed into renderImage and plotOutput.

Usage

```
random_image(width = NULL, height = NULL, alt = NULL)
```

Arguments

width	image width passed through to the rendered tag. NULL (default) leaves the attribute unset (#9).
height	image height, same semantics as width (#9).
alt	alt attribute for accessibility. NULL (default) leaves the attribute unset (#9).

Value

a list compatible with `shiny::renderImage()`: `src`, plus the `width` / `height` / `alt` attributes when provided.

random_image_ext	<i>A Random External Image from the Lorem Picsum API</i>
------------------	--

Description

Returns an tag pointing at a random image served by the **Lorem Picsum** API, ready to be dropped inside a Shiny UI for quick prototyping. No network request is performed by this function: only the URL is built, the image is fetched by the browser.

Usage

```
random_image_ext(width = 400, height = 400, seed = NULL)
```

Arguments

width, height	image dimensions, in pixels. Single positive (finite, whole) numbers.
seed	optional seed making the picked image stable across calls. Any atomic scalar; it is coerced to a string and URL-encoded. When NULL (default), a fresh random image is served on every request.

Value

an [htmltools::tag](#)

Examples

```
random_image_ext()
random_image_ext(width = 400, height = 600, seed = "caramba")
```

random_lm	<i>A Random Linear Model</i>
-----------	------------------------------

Description

This function returns a model which can be passed to renderText or renderTable if pre-processed appropriately

Usage

```
random_lm(nobs = 100, nx = 2)
```

Arguments

nobs	Numeric. number of observation
nx	Numeric. number of variables. Should be lower than nobs

Value

a model output

random_mock	<i>A Mock UI Layout</i>
-------------	-------------------------

Description

Build a quick "zoning" of a Shiny UI: a row of bootstrap columns, each one drawn as an empty bordered box with a centered label. Handy to sketch the layout of an app before any real content exists.

Usage

```
random_mock(widths, heights = 3, labels = NULL)
```

Arguments

widths	integer vector of bootstrap column widths (each between 1 and 12). Its length defines the number of mock zones.
heights	integer vector giving the visual height of each zone, expressed as a number of blank lines. Recycled to the length of widths. Default 3.
labels	character vector of zone labels. Recycled to the length of widths. When NULL (default), short random lorem-ipsum labels are used.

Details

The returned object is a `htmltools::tagList()` (a self-contained CSS `<style>` block plus the columns), so it can be dropped directly inside a `shiny::fluidPage()` / `bslib::page_fluid()` and rendered as is.

Value

a `htmltools::tagList()`

Examples

```
random_mock(c(4, 8))
random_mock(c(6, 6), labels = c("a plot here", "a table here"))
```

random_print	<i>A Random print output</i>
--------------	------------------------------

Description

This function returns a random print output that can be passed to `renderPrint` and `verbatimTextOutput`.

Usage

```
random_print(type = c("character", "numeric", "integer", "model", "table"))
```

Arguments

type type of the output ("character", "numeric", "model", "table")

Value

a random print

random_table	<i>A Random Table</i>
--------------	-----------------------

Description

This function returns a table that can be passed to `renderTable` and `tableOutput`.

Usage

```
random_table(nrow, ncol, type = c("random", "numeric", "character", "numchar"))
```

Arguments

nrow	number of row of the output
ncol	number of cols of the output
type	type of the columns, can be "random", "numeric", "character", "numchar". Default is random.

Value

a table

random_text	<i>A Random Lorem Ipsum</i>
-------------	-----------------------------

Description

A Random Lorem Ipsum

Usage

```
random_text(nchars = NULL, nwords = NULL, offset = 0, var = 1)
```

Arguments

nchars	number of characters. One of the two params should be left NULL.
nwords	number of words to return. One of the two params should be left NULL.
offset	number of characters or words to offset the result by. Defaults to 0.
var	integer ≥ 1 selecting a "variant" of the text: each value returns a distinct slice of the lorem-ipsum corpus, so repeated calls (e.g. in a bulleted list) don't all look alike. Defaults to 1. The corpus wraps around when var is large.

Value

a text

Index

* datasets

lorem, [3](#)

lorem_words, [3](#)

htmltools::tag, [6](#)

htmltools::tagList(), [8](#)

ipsum_examples, [2](#)

lorem, [3](#)

lorem_words, [3](#)

random_DT, [3](#)

random_dygraph, [4](#)

random_ggplot, [4](#)

random_ggplotly, [5](#)

random_image, [5](#)

random_image_ext, [6](#)

random_lm, [7](#)

random_mock, [7](#)

random_print, [8](#)

random_table, [8](#)

random_text, [9](#)