

# Package ‘shinipsum’

July 23, 2025

**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), magrittr,  
plotly, stats, utils

**Suggests** testthat

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.3

**NeedsCompilation** no

**Author** Colin Fay [cre, aut] (ORCID: <<https://orcid.org/0000-0001-7343-1846>>),  
Sebastien Rochette [aut] (ORCID:  
<<https://orcid.org/0000-0002-1565-9313>>),  
ThinkR [cph, fnd]

**Maintainer** Colin Fay <contact@colinfay.me>

**Repository** CRAN

**Date/Publication** 2024-02-09 15:50:05 UTC

## Contents

ipsum_examples	2
lorem	2
lorem_words	3
random_DT	3
random_dygraph	4

random_ggplot . . . . .	4
random_ggplotly . . . . .	5
random_image . . . . .	5
random_lm . . . . .	6
random_print . . . . .	6
random_table . . . . .	7
random_text . . . . .	7

<b>Index</b>	<b>8</b>
--------------	----------

---

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", "tile", "violin")
)
```

**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", "tile", "violin" and defines the geom of the ggplot. Default is "random", and chooses a random geom for you.

**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()
```

**Value**

an image

---

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_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)
```

**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.

**Value**

a text

# Index

## \* datasets

lorem, [2](#)

lorem\_words, [3](#)

ipsum\_examples, [2](#)

lorem, [2](#)

lorem\_words, [3](#)

random\_DT, [3](#)

random\_dygraph, [4](#)

random\_ggplot, [4](#)

random\_ggplotly, [5](#)

random\_image, [5](#)

random\_lm, [6](#)

random\_print, [6](#)

random\_table, [7](#)

random\_text, [7](#)