

Package ‘lorem’

May 8, 2026

Title Generate Lorem Ipsum Text

Version 1.0.0

Description Quickly generate lorem ipsum placeholder text. Easy to integrate in RMarkdown documents. Includes an RStudio addin to insert lorem ipsum into the current document.

License MIT + file LICENSE

URL <https://github.com/gadenbuie/lorem>,
<http://pkg.garrickadenbuie.com/lorem/>

BugReports <https://github.com/gadenbuie/lorem/issues>

Depends R (>= 2.10)

Imports htmltools, knitr, stats

Suggests rstudioapi, testthat (>= 3.0.0), withr

Config/Needs/website gadenbuie/grkgdown

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Garrick Aden-Buie [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-7111-0077>>)

Maintainer Garrick Aden-Buie <garrick@adenbuie.com>

Repository CRAN

Date/Publication 2023-03-09 09:40:06 UTC

Contents

as.tags.lorem	2
ipsum	3
Index	5

`as.tags.lorem`*Render placeholder text as HTML tags*

Description

Renders `ipsum()` placeholder text as HTML tags using `htmltools::as.tags()`. By default, paragraphs are rendering `<p>` tags with `htmltools::p()`, but you may provide your own paragraph wrapper function. In general, you won't need to manually call this function, instead you can just drop `ipsum()` text into another `htmltools::tag()`.

Usage

```
## S3 method for class 'lorem'  
as.tags(x, wrapper = NULL, ...)
```

Arguments

<code>x</code>	Object to be converted.
<code>wrapper</code>	A function that takes a character string of a paragraph or chunk of placeholder text and returns an <code>htmltools::tag()</code> . By default, lorem ipsum text is wrapped in <code>htmltools::p()</code> .
<code>...</code>	Any additional parameters.

Value

Returns an `htmltools::tagList()`.

Examples

```
htmltools::div(  
  lorem::ipsum(paragraphs = 3)  
)  
  
htmltools::tags$ul(  
  htmltools::as.tags(  
    lorem::ipsum(paragraphs = 3, sentences = 1),  
    wrapper = htmltools::tags$li  
  )  
)
```

`ipsum`*Generate Lorem Ipsum Text*

Description

Generates *lorem ipsum* placeholder text for the requested number of sentences or paragraphs. You can control the number of sentences per paragraph and the average number of words per sentence, or simply enter the number of desired paragraphs for a completely random experience.

`lorem::ipsum()` uses sampling and the random number generator and makes no effort to shield the placeholder text generation from the main script, so please only use this package for temporary placeholder text.

Usage

```
ipsum(paragraphs = 1, sentences = NULL, avg_words_per_sentence = 10)
```

```
ipsum_words(n, collapse = TRUE)
```

```
ipsum_starts(n)
```

Arguments

<code>paragraphs</code>	Number of paragraphs of text to generate.
<code>sentences</code>	Number of sentences per paragraph. If <code>NULL</code> , then a random number of sentences per paragraph (approximately 3-8) will be chosen. Alternatively, <code>sentences</code> can be a vector of integers representing the number of sentences per paragraph.
<code>avg_words_per_sentence</code>	Number of expected words per sentence.
<code>n</code>	Number of words to generate
<code>collapse</code>	Should the words be collapsed into a single string, separated by spaces (default)? If <code>FALSE</code> , the chosen words are returned as a character vector.

Value

A character vector of *lorem ipsum* placeholder text, where each element in the vector is a paragraph of text.

Functions

- `ipsum()`: Generate paragraphs and sentences of *lorem ipsum* text.
- `ipsum_words()`: Generate *lorem ipsum* words, without punctuation.
- `ipsum_starts()`: Generate *lorem ipsum* starting words.

Options

You can influence, to a degree, the amount of punctuation that is included in the output using the `lorem.punctuation_valence` option. This global option should be a number between 0 and 1, or `FALSE` to disable punctuation altogether. When the value is closer to 1, more punctuation is included in the sentences. When the value is closer to 0, less punctuation will be inserted. The default value is 0.4.

Examples

```
# 1 paragraph of text
lorem::ipsum(1)

# 2 paragraphs with 2 and 3 sentences each
lorem::ipsum(2, sentences = c(2, 3))

# 2 paragraphs with short sentences
lorem::ipsum(2, avg_words_per_sentence = 4)

# 2 paragraphs with long sentences
lorem::ipsum(2, avg_words_per_sentence = 20)
```

Index

`as.tags.lorem`, 2

`htmltools::as.tags()`, 2

`htmltools::p()`, 2

`htmltools::tag()`, 2

`htmltools::tagList()`, 2

`ipsum`, 3

`ipsum()`, 2

`ipsum_starts(ipsum)`, 3

`ipsum_words(ipsum)`, 3