Package: cleanrmd (via r-universe)

July 13, 2024

Title Clean Class-Less 'R Markdown' HTML Documents

Version 0.1.1.9000

Description A collection of clean 'R Markdown' HTML document templates using classy-looking classless CSS styles. These documents use a minimal set of dependencies but still look great, making them suitable for use a package vignettes or for sharing results via email.

License MIT + file LICENSE

URL https://pkg.garrickadenbuie.com/cleanrmd/, https://github.com/gadenbuie/cleanrmd

BugReports https://github.com/gadenbuie/cleanrmd/issues

Depends R (>= 2.10)

Imports htmltools, jsonlite, rmarkdown, utils

Suggests testthat (>= 2.1.0)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Repository https://gadenbuie.r-universe.dev

RemoteUrl https://github.com/gadenbuie/cleanrmd

RemoteRef main

RemoteSha eb32fc687b6b367c420dbd32a2a087e52c5bca3b

Contents

`cleanrmd_themes` *List cleanrmd themes*

Description

Lists the available themes in **cleanrmd**

Usage

```
cleanrmd_themes()
```

Value

A character string of available theme names.

cleanrmd theme list

cleanrmd includes the following no-class CSS themes:

- almond
- awsm.css
- axist
- bamboo
- bullframe
- holiday
- kacit
- latex.css
- markdown-splendor
- markdown-retro
- markdown-air
- markdown-modest
- marx
- minicss
- new.css
- no-class
- picocss
- sakura
- sakura-vader
- semantic
- simplecss
- style-sans

- style-serif
- stylize
- superstylin
- tacit
- vanilla
- water
- water-dark
- writ

Examples

```
cleanrmd_themes()
```

html_document_clean *Clean Rmarkdown HTML Document*

Description

Clean HTML documents with R Markdown.

Usage

```
html_document_clean(
  ...,
  theme = "no-class",
  css = NULL,
  toc = FALSE,
  toc_depth = 3,
  title_in_header = TRUE,
  mathjax = NULL,
  use_fontawesome = FALSE,
  fig_width = 10,
  fig_height = 7,
  fig_retina = 2,
  keep_md = FALSE,
  dev = "png",
  highlight = "default",
  pandoc_args = NULL,
  extra_dependencies = NULL,
  md_extensions = NULL,
  self_contained = !is.null(theme)
)
```

Arguments

`...` Additional function arguments to pass to the base R Markdown HTML output formatter `html_document_base`

`theme` The class-less CSS theme to use, one of `cleanrmd_themes()`.

`css` CSS and/or Sass files to include. Files with an extension of .sass or .scss are compiled to CSS via `sass::sass()`. Also, if `theme` is a `bslib::bs_theme()` object, Sass code may reference the relevant Bootstrap Sass variables, functions, mixins, etc.

`toc` `TRUE` to include a table of contents in the output

`toc_depth` Depth of headers to include in table of contents

`title_in_header`

If `TRUE` (default), the title, subtitle, author, and date are placed in a `<header>` tag. This is semantically correct HTML but some CSS frameworks work better with this structure than others.

`mathjax` Include mathjax. The "default" option uses an https URL from a MathJax CDN. The "local" option uses a local version of MathJax (which is copied into the output directory). You can pass an alternate URL or pass `NULL` to exclude MathJax entirely.

`use_fontawesome`

Should links to FontAwesome be included in the HTML document's `<head>`? Only enable if you are including FontAwesome icons in your HTML document.

`fig_width` Default width (in inches) for figures

`fig_height` Default height (in inches) for figures

`fig_retina` Scaling to perform for retina displays (defaults to 2, which currently works for all widely used retina displays). Set to `NULL` to prevent retina scaling. Note that this will always be `NULL` when `keep_md` is specified (this is because `fig_retina` relies on outputting HTML directly into the markdown document).

`keep_md` Keep the markdown file generated by knitting.

`dev` Graphics device to use for figure output (defaults to png)

`highlight` Syntax highlight engine and style, either a built-in Pandoc highlighting theme, a theme provided by **rmarkdown**, or a prismjs theme (see below). Pass `NULL` to prevent syntax highlighting.

Pandoc themes. By default, uses Pandoc's highlighting style. Pandoc's built-in styles include "tango", "pygments", "kate", "monochrome", "espresso", "zenburn", "haddock" and "breezedark".

Two custom styles are also included, "arrow", an accessible color scheme, and "rstudio", which mimics the default IDE theme. Alternatively, supply a path to a `.theme` to use a custom Pandoc style. Note that custom themes require Pandoc 2.0+.

Prism themes. To use the prismjs syntax highlighter, pass one of "prism", "prism-coy", "prism-dark", "prism-funky", "prism-okaidia", "prism-solarizedlight", "prism-tomorrow", or "prism-twilight". To use an alternate Prism theme file, pass the URL or path to the theme's CSS file prefixed with "prism:", e.g. `prism:my-theme.css`. Note that the Prism dependencies are not embedded into self-contained documents so they will require an active internet connection to work.

`pandoc_args` Additional command line options to pass to pandoc

`extra_dependencies`
Extra dependencies as a list of the `html_dependency` class objects typically generated by `htmltools:htmlDependency()`.

`md_extensions` Markdown extensions to be added or removed from the default definition of R Markdown. See the `rmarkdown_format` for additional details.

`self_contained` Produce a standalone HTML file with no external dependencies, using data: URIs to incorporate the contents of linked scripts, stylesheets, images, and videos. Note that even for self contained documents MathJax is still loaded externally (this is necessary because of its size).

Value

An R Markdown output format that can be used with `output:` in an `.Rmd` or for use with `rmarkdown::render()`.

MathJax

Note that MathJax is disabled by default to reduce the overall size of the final document. You can enable MathJax by setting `mathjax = "default"`, see `rmarkdown::html_document()` for more options.

See Also

`use_cleanrmd()` for using cleanrmd themes in places other than in R Markdown documents

Examples

```
if (interactive()) {
  html_document_clean()
}

## Not run:
rmarkdown::render("input.Rmd", html_document_clean())

## End(Not run)
```

`use_cleanrmd` *Use a clean CSS theme from cleanrmd*

Description

Provides a **cleanrmd** CSS theme using **htmltools**. You can use this CSS theme anywhere that HTML dependencies are handled via `htmltools::htmlDependency()`, for example in R Markdown or Quarto documents or in Shiny apps.

Usage

```
use_cleanrmd(name = NULL)

cleanrmd_theme_dependency(name = NULL)
```

Arguments

name	The name of the theme, see `cleanrmd_themes()` for a list of available themes. If `NULL`, all themes will be loaded with a simple drop down theme picker.

Value

`use_cleanrmd()` returns an `htmltools::tagList()` with an `htmltools::htmlDependency()`. `cleanrmd_theme_dependency()` returns only the `htmltools::htmlDependency()`.

Functions

- `use_cleanrmd()`: Use a clean CSS theme in the current document or app (general usage)
- `cleanrmd_theme_dependency()`: Use a clean CSS theme dependency (advanced usage)

R Markdown documents

In R Markdown (static or Shiny prerendered), you should use the `html_document_clean()` output format to use a cleanrmd theme.

```
---
output:
  cleanrmd::html_document_clean:
    theme: NULL # or pick a specific theme
    self_contained: false
---
```

Quarto documents

You can also use **cleanrmd** in Quarto documents or apps (using `server: shiny`). You'll need to turn off the themes provided by Quarto with `theme: none` and then call `cleanrmd::use_cleanrmd()` in a code chunk in your document.

~~~
---
title: "Untitled"
format:
  html:
    theme: none
#server: shiny
---

```{r cleanrmd, echo=FALSE}
cleanrmd::use_cleanrmd("bamboo")
```
~~~

Shiny apps

In Shiny apps, you'll need to use `shiny::basicPage()` rather than `shiny::fluidPage()`. Then call `use_cleanrmd()` in your app to use a cleanrmd theme.

```
library(shiny)

ui <- shiny::basicPage(
  cleanrmd::use_cleanrmd(),

  h2("Old Faithful Geyser Data"),

  sliderInput(
    "bins",
    "Number of bins:",
    min = 1,
    max = 50,
    value = 30
  ),
  plotOutput("distPlot")
)
```

Examples

```
page <- htmltools::withTags(
  main(
    h2("Small Demo"),
    p("Clean, simple, classy but class-less:"),
    ul(
      li("Works almost anywhere"),
      li("Small and simple"),
      li("Easy to extend"),
      li("Good enough but not perfect")
    )
  )
)

# no styles
if (interactive()) {
  htmltools::browsable(page)
}

# all clean styles
page_clean <- htmltools::tagList(page, use_cleanrmd())
if (interactive()) {
  htmltools::browsable(page_clean)
}

# one clean style
page_water <- htmltools::tagList(page, use_cleanrmd("water"))
if (interactive()) {
  htmltools::browsable(page_water)
}

cleanrmd_theme_dependency("bamboo")
```

Index