Package: rmdpartials (via r-universe)

October 9, 2024

Title Partial 'rmarkdown' Documents to Prettify your Reports

Description Use 'rmarkdown' partials, also know as child documents in 'knitr', so you can make components for HTML, PDF, and Word documents. The package provides various helper functions to make certain functions easier. You may want to use this package, if you want to flexibly summarise objects using a combination of figures, tables, text, and HTML widgets. Unlike HTML widgets, the output is Markdown and can hence be turn into other output formats than HTML.

Version 0.6.1

Depends R (>= 3.0.1)

Language en-GB

URL https://github.com/rubenarslan/rmdpartials

BugReports https://github.com/rubenarslan/rmdpartials/issues

License MIT + file LICENSE

Imports stats, digest, utils, knitr, rlang

Suggests grDevices, graphics, testthat, rmarkdown, covr, dplyr, DT, ggplot2, pkgdown

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

Roxygen list(markdown = TRUE)

VignetteBuilder knitr

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

RemoteUrl https://github.com/rubenarslan/rmdpartials

RemoteRef HEAD

RemoteSha a635ef9738c219ce4ac4c354888a69fe7e8c3969

Contents

`as.partial` *Convert text or file to a partial*

Description

This adds the `knit_asis` class to a markdown chunk, so that it can be rendered in the viewer and simply echoed in other knitr chunks. Won't preserve figures unless the path happens to be the same or you explicitly pass it to the knit_meta argument.

Usage

```
as.partial(text = NULL, knit_meta = list())
```

Arguments

`text`	will be returned with the class "knit_asis"
`knit_meta`	you can pass a path to figures and other resources here

Value

Returns its input as text with class "knit_asis"

Examples

```
my_partial <- as.partial("## Headline
Text")
```

enlarge_plot *Generate a small plot that will be enlarged in a modal when clicked*

Description

Generate a small plot that will be enlarged in a modal when clicked

Usage

```
enlarge_plot(
  plot,
  large_plot = plot,
  plot_name = NULL,
  width_small = 2,
  height_small = 2,
  width_large = 7,
  height_large = 7,
  ...
)
```

Arguments

plot	a plot
large_plot	a larger version of the same plot. defaults to the first plot if left empty, but this only works for ggplot2 and similar, not base plots
plot_name	optional: specify a meaningful plot name (needs to be unique in the document)
width_small	width for the small plot
height_small	height for the small plot
width_large	width for the large plot
height_large	height for the large plot
...	passed to partial()

Value

Returns markdown/HTML text with class "knit_asis"

Examples

```
## Not run:
if(!requireNamespace("pkgdown", quietly = TRUE) || !pkgdown::in_pkgdown()) {
# will generate files in a temporary directory
if (requireNamespace("ggplot2")) {
dist <- ggplot2::qplot(stats::rbeta(200, 3, 4))
enlarge_plot(dist,
large_plot = dist + ggplot2::theme_classic(base_size = 18))
} else {
```

```
graphics::hist(stats::rbeta(200, 3, 4))
dist <- grDevices::recordPlot()
enlarge_plot(dist)
}
}

## End(Not run)
```

`knit_child_debug` *Get some debugging information on various potential problems when making partials*

Description

Get some debugging information on various potential problems when making partials

Usage

```
knit_child_debug(...)
```

Arguments

`...`	passed to `partial()`

Value

Returns markdown/HTML text with class "knit_asis"

Examples

```
if(!requireNamespace("pkgdown", quietly = TRUE) || !pkgdown::in_pkgdown()) {
knit_child_debug()
}
```

`partial` *Knit a child document and output as is (render markup)*

Description

This modifies and extends the `knitr::knit_child()` function. Defaults change as follows:

- the environment defaults to the calling environment, or if passed, to arguments passed via ...
- the output receives the class `knit_asis`, so that the output will be rendered "as is" by knitr when calling inside a chunk (no need to set `results='asis'` as a chunk option).
- defaults to `quiet = TRUE`

- the package additionally renders `knit_asis` objects in the viewer when printed to make previewing partials easier. This is achieved using `rmarkdown::render()` and done in a temporary directory (only when used interactively/not in child mode).
- the package takes care of some troubles behind the scenes that you might find yourself in if you nest partials (by trying to resolve path ambiguities, using text instead of files for sources, and some functionality to prevent iteratively overwriting generated figures and other files)

Usage

```
partial(
  input = NULL,
  ...,
  text = NULL,
  output = NULL,
  quiet = TRUE,
  options = NULL,
  envir = parent.frame(),
  name = NULL,
  cacheable = NA,
  show_code = FALSE,
  use_strings = TRUE,
  render_preview = needs_preview(),
  preview_output_format = NULL
)
```

Arguments

`input`	if you specify a file path here, it will be read in before being passed to knitr (to avoid a working directory mess)
`...`	ignored, but you can use it to clarify which variables will be used in the rmd partial
`text`	passed to `knitr::knit_child()`
`output`	if you specify a file path here, where to put the file
`quiet`	passed to `knitr::knit_child()`
`options`	defaults to NULL.
`envir`	passed to `knitr::knit_child()`
`name`	a name to use for cacheing and figure paths. Randomly generated if left unspecified.
`cacheable`	whether the results of this partial can be cached in knitr
`show_code`	whether to print the R code for the partial or just the results (sets the chunk option echo = FALSE while the chunk is being rendered)
`use_strings`	whether to read in the child file as a character string (solves working directory problems but harder to debug)
`render_preview`	true if interactive mode is auto-detected, false when actually knitting the partial as a child
`preview_output_format`	defaults to `rmarkdown::html_document()` with self_contained set to true

Details

Why default to the calling environment? Typically this function defaults to the global environment. This makes sense if you want to use knit children in the same context as the rest of the document. However, you may also want to use knit children to respect conventional scoping rules inside functions to e.g. summarise a regression using a set of commands (e.g. plot some diagnostic graphs and a summary for a regression nicely formatted).

Some caveats:

- the function has to return to the top-level. There's no way to `cat()` this from loops or an if-condition without without setting `results='asis'`. You can however concatenate these objects with `paste.knit_asis()`
- currently not yet producing expected results in RStudio notebooks in interactive use

Value

Returns rendered markdown with the class "knit_asis". When used interactively, the knit_meta attributes will additionally contain the path of a rendered preview in a temporary directory.

Examples

```
# super simple partial example
partial(text = "Test")

# an example of a wrapper function that calls partial with an argument
# ensures distinct paths for cache and figures, so that these calls can be looped in parallel
regression_diagnostics <- function(regression, ...) {
   partial(system.file("_regression_diagnostics.Rmd",
           package = "rmdpartials", mustWork = TRUE),
           regression = regression, ...)
}
```

`paste.knit_asis` *Paste and output as is (render markup)*

Description

Helper function for `knit_asis` objects, useful when e.g. `partial()` was used in a loop.

Usage

```
paste.knit_asis(..., sep = "\n\n\n", collapse = "\n\n\n")
```

Arguments

`...`	passed to `base::paste()`
`sep`	defaults to two empty lines, passed to `base::paste()`
`collapse`	defaults to two empty lines, passed to `base::paste()`

Details

Works like `base::paste()` with both the sep and the collapse argument set to two empty lines

Value

Returns text with the class "knit_asis"

Examples

```
paste.knit_asis("# Headline 1", "## Headline 2")
```

---

print.knit_asis *Print* knit_asis *as rendered HTML in the viewer*

Description

Print knit_asis as rendered HTML in the viewer

Usage

```
## S3 method for class 'knit_asis'
print(x, ...)
```

Arguments

x	the knit_asis object
...	ignored

Value

Invisibly returns its input, either prints its input or sends it to a viewer, if one is defined

Examples

```
text <- paste(c("### Headline",
"Text"), collapse = "\n")
print(knitr::asis_output(text))
```

`regression_diagnostics` *Show the estimated coefficients in a regression and diagnostics*

Description

Show the estimated coefficients in a regression and diagnostics

Usage

```
regression_diagnostics(regression, ...)
```

Arguments

`regression`	an lm object
`...`	passed to `partial()`

Value

Returns markdown/HTML text with class "knit_asis"

Examples

```
## Not run:
# will generate files in a temporary directory
if(!requireNamespace("pkgdown", quietly = TRUE) || !pkgdown::in_pkgdown()) {
data("ChickWeight")
regression <- lm(weight ~ Time, data = ChickWeight)
regression_diagnostics(regression)
}

## End(Not run)
```

Index