

Package: chunkhooks (via r-universe)

September 11, 2024

Title Chunk Hooks for 'R Markdown'

Version 0.0.1

Maintainer Atsushi Yasumoto <atusy.rpkg@gmail.com>

Description Set chunk hooks for 'R Markdown' documents
<<https://rmarkdown.rstudio.com/>>, and improve user experience.
For example, change units of figure sizes, benchmark chunks,
and number lines on code blocks.

License MIT + file LICENSE

URL <https://chunkhooks.atusy.net>, <https://github.com/atusy/chunkhooks>

Encoding UTF-8

Language en-US

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Imports knitr, measurements, prettyunits

Suggests spelling, testthat, rmarkdown

VignetteBuilder knitr

Repository <https://atusy.r-universe.dev>

RemoteUrl <https://github.com/atusy/chunkhooks>

RemoteRef HEAD

RemoteSha 68f4f41cda2667e9a9237fb25f56fdc0a6810fb2

Contents

benchmark	2
comment_out	3
comment_syntax	4
comment_title	4
hook_figure_unit	5
hook_numberLines	6
hook_title_comment	6

benchmark	<i>Benchmark chunks</i>
-----------	-------------------------

Description

hook_benchmark sets a hook to benchmark chunks with the benchmark=TRUE option. The name of the trigger chunk option can be changed via the chunk_option parameter. The result is printed right after chunk outputs. See examples for the default printing format by format_benchmark.

Usage

```
hook_benchmark(
  trigger = "benchmark",
  default = NULL,
  format = format_benchmark,
  .set = TRUE
)

format_benchmark(result, options)

benchmarks
```

Arguments

trigger	A name of chunk option that triggers benchmark (default: "benchmark"). In order to trigger benchmark, specify TRUE as a value of this option.
default	A default value for the chunk option that triggers the hook (default: NULL).
format	A function to format a benchmark result (default: format_benchmark). It must accept two arguments, where the first is the benchmark result and the second is the list of current chunk options. NULL suppresses printing.
.set	TRUE or FALSE to set the hook.
result	A result of benchmark
options	A list of current chunk options

Format

An object of class environment of length 0.

Details

benchmarks records the results of benchmarks from chunks as a list named by chunk labels. If one requires complex formatting of benchmark results, then suppress automatic formatting by hook_benchmark(format = NULL). Then, retrieve benchmark results from benchmarks[["chunk-label"]]. Furthermore, format can happen conditionally by utilizing current chunk options via the second argument of the formatting function.

Value

invisible hook function

Examples

```
# Set a hook that triggers benchmarks if the `time` chunk option is not `NULL`.
hook_benchmark("time")

# Example of the default output format
# Input is sec. Output is prettified.
format_benchmark(1234, options = list(label = "example-chunk"))
```

comment_out	<i>Comment out codes based on their languages.</i>
-------------	--

Description

Comment out codes based on their languages.

Usage

```
comment_out(x, engine, extra_syntax = character(0L))
```

Arguments

x	A character vector
engine	A string that defines the engine of the x. See names(comment_syntax) for the predefined engines.
extra_syntax	A named character vector which defines extra syntax to comment out. Each values should contain %s which is the placeholder to insert x. If definitions collides among comment_syntax and extra_syntax, then the latter has priority.

See Also

[comment_syntax\(\)](#)

Examples

```
# Default behaviors
comment_out("R language", "R")
comment_out("Python language", "python")

# A customized behavior
comment_out(
  "Python language",
  "python",
  extra_syntax = c("python" = '""%s""')
)
```

comment_syntax	<i>Comment syntax.</i>
----------------	------------------------

Description

A named vector which defines comment syntax. This is internally used by `hook_title_comment()`, `comment_title()`, and `comment_out()`.

Usage

```
comment_syntax
```

Format

An object of class character of length 38.

comment_title	<i>Comment title to chunk code</i>
---------------	------------------------------------

Description

Comment title to chunk code

Usage

```
comment_title(
  options = list(engine = "R"),
  template = "{engine}",
  extra_syntax = character(0L),
  ...
)
```

Arguments

options	A list of chunk options.
template	A template of title processed by <code>glue::glue()</code> . The processed value is automatically commented out by <code>comment_out()</code> .
extra_syntax	A named character vector which defines extra syntax to comment out. Each values should contain <code>%s</code> which is the placeholder to insert <code>x</code> . If definitions collides among <code>comment_syntax</code> and <code>extra_syntax</code> , then the latter has priority.
...	Arguments passed to <code>glue::glue()</code>

See Also

[comment_out\(\)](#)

Examples

```
comment_title(  
  options = list(engine = "R"),  
  template = "{engine}"  
)  
  
comment_title(  
  options = list(engine = "css", label = "mystyle"),  
  template = "{label}.{engine}"  
)
```

hook_figure_unit	<i>Change the unit of figure size</i>
------------------	---------------------------------------

Description

By default, figure size of R Markdown is specified with inches. This function changes the default unit.

Usage

```
hook_figure_unit(unit = "mm", .set = TRUE)
```

Arguments

unit	A string of an unit (default: "mm"). Available units follow. <code>measurements::conv_unit_options\$length</code>
.set	TRUE or FALSE to set the hook.

Details

As a side effect, `fig.retina` is set to NULL because of <https://github.com/yihui/knitr/issues/1876>.

Value

invisible hook function

Examples

```
hook_figure_unit('mm')
```

hook_numberLines	<i>Number lines on code blocks</i>
------------------	------------------------------------

Description

Number lines on code blocks created by chunks, i.e. source, output, message, warning, and/or error.

Usage

```
hook_numberLines(targets = "source", .set = TRUE)
```

Arguments

targets	A character vector specifying what kind of code blocks automatically number lines (default: "source"). Choices are "source", "output", "message", "warning", and/or "error". NULL disables the automation.
.set	TRUE or FALSE to set the hook.

Value

invisible hook function

Examples

```
hook_numberLines("source")
```

hook_title_comment	<i>Prepend title comment to chunks</i>
--------------------	--

Description

Prepend title comments to each chunks. By default, title is the name of the corresponding chunk engines.

Usage

```
hook_title_comment(
  trigger = "title",
  default = "{engine}",
  args = list(),
  .set = TRUE
)
```

Arguments

trigger	A name of chunk option triggers the hook.
default	A string or function as a default value of the triggering chunk option. If string, the value is given to <code>glue::glue</code> , and evaluates the expression by setting chunk options as an environment, and then automatically commented out. If function, it must have <code>options</code> parameter which receives chunk options, and return a string.
args	A list of arguments given to <code>glue::glue()</code> or callable default.
.set	TRUE or FALSE to set a default value and a hook to the trigger.

See Also

[comment_title\(\)](#), [comment_out\(\)](#), [comment_syntax\(\)](#)

Examples

```
# Set hook which comments engine names of each chunks at the beginning.
hook_title_comment()

# Customize format
# For R chunks, let title comment be something like "chunk-label.R".
# Otherwise, fall back to the default behavior.
hook_title_comment(
  default = function(options) {
    if (options$engine == "R") {
      return(sprintf("# %s.%s", options$label, options$engine))
    }
    return(comment_title(options, template = "{engine}"))
  }
)
```

Index

* datasets

benchmark, [2](#)

comment_syntax, [4](#)

benchmark, [2](#)

benchmarks (benchmark), [2](#)

comment_out, [3](#)

comment_out(), [4](#), [7](#)

comment_syntax, [4](#)

comment_syntax(), [3](#), [7](#)

comment_title, [4](#)

comment_title(), [7](#)

format_benchmark (benchmark), [2](#)

hook_benchmark (benchmark), [2](#)

hook_figure_unit, [5](#)

hook_numberLines, [6](#)

hook_title_comment, [6](#)