Package 'xaringan'

February 19, 2018

Type Package

Title Presentation Ninja

Version 0.6

Description Create HTML5 slides with R Markdown and the JavaScript library 'remark.js' (<https://remarkjs.com>).

Imports htmltools, knitr (>= 1.16), servr (>= 0.5), xfun, rmarkdown

Suggests rstudioapi, testit

License MIT + file LICENSE

URL https://github.com/yihui/xaringan

BugReports https://github.com/yihui/xaringan/issues

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 6.0.1

NeedsCompilation no

Author Yihui Xie [aut, cre] (<https://orcid.org/0000-0003-0645-5666>),
Claus Thorn Ekstrøm [ctb],
Dawei Lang [ctb],
Garrick Aden-Buie [ctb],
Ole Petter Bang [ctb] (CSS in
rmarkdown/templates/xaringan/resources/default.css),
Patrick Schratz [ctb],
Sean Lopp [ctb]

Maintainer Yihui Xie <xie@yihui.name>

Repository CRAN

Date/Publication 2018-02-19 20:10:51 UTC

R topics documented:

`infinite_moon_reader` *Serve and live reload slides*

Description

Use the **servr** package to serve and reload slides on change. `inf_mr()` is an alias of `infinite_moon_reader()`.

Usage

```
infinite_moon_reader(moon, cast_from = ".")

inf_mr(moon, cast_from = ".")
```

Arguments

`moon`	The input Rmd file path (if missing and in RStudio, the current active document is used).
`cast_from`	The root directory of the server.

Details

The Rmd document is compiled continuously to trap the world in the Infinite Tsukuyomi. The genjutsu is cast from the directory specified by `cast_from`, and the Rinne Sharingan will be reflected off of the `moon`.

Note

This function is not really tied to the output format `moon_reader()`. You can use it to serve any single-HTML-file R Markdown output.

References

http://naruto.wikia.com/wiki/Infinite_Tsukuyomi

See Also

`servr::httw`

`moon_reader` *An R Markdown output format for remark.js slides*

Description

This output format produces an HTML file that contains the Markdown source (knitted from R Markdown) and JavaScript code to render slides. `tsukuyomi()` is an alias of `moon_reader()`.

Usage

```
moon_reader(css = c("default", "default-fonts"), self_contained = FALSE, seal = TRUE,
   yolo = FALSE, chakra = "https://remarkjs.com/downloads/remark-latest.min.js",
    nature = list(), ...)

tsukuyomi(...)
```

Arguments

`css`	A vector of CSS file paths. Two default CSS files ('`default.css`' and '`default-fonts.css`') are provided in this package, which was borrowed from https://remarkjs.com. If the character vector `css` contains a value that does not end with `.css`, it is supposed to be a built-in CSS file in this package, e.g., for `css = c('default', 'extra.css')`), it means `default.css` in this package and a user-provided `extra.css`. To find out all built-in CSS files, use `xaringan:::list_css()`.
`self_contained`	Whether to produce a self-contained HTML file.
`seal`	Whether to generate a title slide automatically using the YAML metadata of the R Markdown document (if `FALSE`, you should write the title slide by yourself).
`yolo`	Whether to insert the Mustache Karl (TM) randomly in the slides. `TRUE` means insert his picture on one slide, and if you want him to be on multiple slides, set `yolo` to a positive integer or a percentage (e.g. 0.3 means 30% of your slides will be the Mustache Karl). Alternatively, `yolo` can also be a list of the form `list(times = n, img = path)`: `n` is the number of times to show an image, and `path` is the path to an image (by default, it is Karl).
`chakra`	A path to the remark.js library (can be either local or remote).
`nature`	(Nature transformation) A list of configurations to be passed to `remark.create()`, e.g. `list(ratio = '16:9', navigation = list(click = TRUE))`; see https://github.com/gnab/remark/wiki/Configuration. Besides the options provided by remark.js, you can also set `autoplay` to a number (the number of milliseconds) so the slides will be played every `autoplay` milliseconds. You can also set `countdown` to a number (the number of milliseconds) to include a countdown timer on each slide. If using `autoplay`, you can optionally set `countdown` to `TRUE` to include a countdown equal to `autoplay`.
`...`	For `tsukuyomi()`, arguments passed to `moon_reader()`; for `moon_reader()`, arguments passed to `rmarkdown::html_document()`.

Details

Tsukuyomi is a genjutsu to trap the target in an illusion on eye contact.

If you are unfamiliar with CSS, please see the xaringan wiki on Github providing CSS slide modification examples.

Note

Do not stare at Karl's picture for too long after you turn on the `yolo` mode. I believe he has Sharingan.

Local images that you inserted via the Markdown syntax `![](path/to/image)` will not be embedded into the HTML file when `self_contained = TRUE` (only CSS, JavaScript, and R plot files will be embedded). You may also download remark.js (via `summon_remark()`) and use a local copy instead of the default `chakra` argument when `self_contained = TRUE`, because it may be time-consuming for Pandoc to download remark.js each time you compile your slides.

Each page has its own countdown timer (when the option `countdown` is set in `nature`), and the timer is (re)initialized whenever you navigate to a new page. If you need a global timer, you can use the presenter's mode (press `P`).

References

http://naruto.wikia.com/wiki/Tsukuyomi

---

summon_remark *Summon remark.js to your local disk*

Description

Download a version of the remark.js script to your local disk, so you can render slides offline. You need to change the `chakra` argument of `moon_reader()` after downloading remark.js.

Usage

```
summon_remark(version = "latest", to = "libs/")
```

Arguments

`version`	The version of remark.js (e.g. `latest`, `0.13`, or `0.14.1`).
`to`	The destination directory.

Index