Package: xaringan (via r-universe)

August 23, 2024

Type Package

Title Presentation Ninja

Version 0.30.1

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

Depends R (>= 3.5.0)

Imports htmltools, knitr (>= 1.30), servr (>= 0.30), xfun (>= 0.18), rmarkdown (>= 2.8)

Suggests rstudioapi, jsonlite, testit

License MIT + file LICENSE

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

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

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.3.1

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

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

RemoteRef HEAD

RemoteSha 0f942ca1ee772944f23b4bb2f550a376053e4e9a

Contents

decktape *Convert HTML presentations to PDF via DeckTape*

Description

This function can use either the `decktape` command or the hosted docker image of the **decktape** library to convert HTML slides to PDF (including slides produced by **xaringan**).

Usage

```
decktape(
  file,
  output,
  args = "--chrome-arg=--allow-file-access-from-files",
  docker = Sys.which("decktape") == "",
  version = "",
  open = FALSE
)
```

Arguments

`file`	The path to the HTML presentation file. When `docker = FALSE`, this path could be a URL to online slides.
`output`	The desired output path of the PDF file.
`args`	Command-line arguments to be passed to `decktape`.
`docker`	Whether to use Docker (`TRUE`) or use the `decktape` command directly (`FALSE`). By default, if **decktape** has been installed in your system and can be found via `Sys.which('decktape')`, it will be uesd directly.
`version`	The **decktape** version when you use Docker.
`open`	Whether to open the resulting PDF with your system PDF viewer.

Value

The output file path (invisibly).

Note

For some operating systems you may need to add yourself to the `docker` group and restart your machine if you use DeckTape via Docker. By default, the latest version of the **decktape** Docker image is used. In case of errors, you may want to try older versions (e.g., `version = '2.8.0'`).

References

DeckTape: https://github.com/astefanutti/decktape. Docker: https://www.docker.com.

Examples

```
xaringan::decktape("https://slides.yihui.org/xaringan", "xaringan.pdf", docker = FALSE)
```

`infinite_moon_reader` *Serve and live reload slides*

Description

Use the **servr** package to serve and reload slides on change. `inf_mr()` and `mugen_tsukuyomi()` are aliases of `infinite_moon_reader()`.

Usage

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

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

mugen_tsukuyomi(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.
`...`	Passed to `rmarkdown::render()`.

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`. Use `servr::daemon_stop()` to perform a genjutsu Kai and break the spell.

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

https://naruto.fandom.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(),
  anchor_sections = FALSE,
  ...
)

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()`. With **rmarkdown** >= 2.8, Sass files (filenames ending with '`.scss`' or '`.sass`') can also be used, and they will be processed by the **sass** package, which needs to be installed.
`self_contained`	Whether to produce a self-contained HTML file by embedding all external resources into the HTML file. See the 'Note' section below.
`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). Please note that if you use the default remote latest version of remark.js, your slides will not work when you do not have Internet access. They might also be broken after a newer version of remark.js is released. If these issues concern you, you should download remark.js locally (e.g., via `summon_remark()`), and use the local version instead.

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; alternatively, `autoplay` can be a list of the form `list(interval = N, loop = TRUE)`, so the slides will go to the next page every `N` milliseconds, and optionally go back to the first page to restart the play when `loop = TRUE`. 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`. To alter the set of classes applied to the title slide, you can optionally set `titleSlideClass` to a vector of classes; the default is `c("center", "middle", "inverse")`.

anchor_sections, ...
: 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.

For the option `self_contained = TRUE`, it encodes images as base64 data in the HTML output file. The image path should not contain the string `")"` when the image is written with the syntax `![](PATH)` or `background-image: url(PATH)`, and should not contain the string `"/>"` when it is written with the syntax `<img src="PATH" />`. Rendering slides in the self-contained mode can be time-consuming when you have remote resources (such as images or JS libraries) in your slides because these resources need to be downloaded first. We strongly recommend that you download remark.js (via `summon_remark()`) and use a local copy instead of the default `chakra` argument when `self_contained = TRUE`, so remark.js does not need to be downloaded each time you compile your slides.

When the slides are previewed via `xaringan::inf_mr()`, `self_contained` will be temporarily changed to `FALSE` even if the author of the slides set it to `TRUE`. This will make it faster to preview slides locally (by avoiding downloading remote resources explicitly and base64 encoding them). You can always click the Knit button in RStudio or call `rmarkdown::render()` to render the slides in the self-contained mode (these approaches will respect the `self_contained` setting).

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

https://naruto.fandom.com/wiki/Tsukuyomi

Examples

```
# rmarkdown::render('foo.Rmd', 'xaringan::moon_reader')
```

`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