

# Package ‘xaringan’

September 17, 2019

**Type** Package

**Title** Presentation Ninja

**Version** 0.12

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

**Imports** htmltools, knitr (>= 1.21), servr (>= 0.13), xfun (>= 0.6), 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** 6.1.1

**NeedsCompilation** no

**Author** Yihui Xie [aut, cre] (<<https://orcid.org/0000-0003-0645-5666>>),

Benjie Gillam [ctb],

Claus Thorn Ekstrøm [ctb],

Daniel Anderson [ctb],

Dawei Lang [ctb],

Garrick Aden-Buie [ctb],

John Little [ctb],

Joseph Casillas [ctb],

Michael Wayne Kearney [ctb],

Nan-Hung Hsieh [ctb],

Ole Petter Bang [ctb] (CSS in <rmarkdown/templates/xaringan/resources/default.css>),

Patrick Schratz [ctb],

Paul Lemmens [ctb],

Sean Lopp [ctb],

Lucy D'Agostino McGowan [ctb] (<<https://orcid.org/0000-0001-7297-9359>>),

Emi Tanaka [ctb],  
 Yongfu Liao [ctb],  
 Malcolm Barrett [ctb] (<<https://orcid.org/0000-0003-0299-5825>>),  
 Alessandro Gasparini [ctb] (<<https://orcid.org/0000-0002-8319-7624>>),  
 Yue Jiang [ctb] (<<https://orcid.org/0000-0002-9798-5517>>)

**Maintainer** Yihui Xie <[xie@yihui.name](mailto:xie@yihui.name)>

**Repository** CRAN

**Date/Publication** 2019-09-17 09:50:03 UTC

## R topics documented:

|                                |   |
|--------------------------------|---|
| decktape . . . . .             | 2 |
| infinite_moon_reader . . . . . | 3 |
| moon_reader . . . . .          | 4 |
| summon_remark . . . . .        | 5 |

**Index** 7

---

|          |   |
|----------|---|
| decktape | <i>Convert HTML presentations to PDF via DeckTape</i> |
|----------|---|

---

## 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

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

```
if (interactive()) {
  xaringan::decktape("https://slides.yihui.name/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()` 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](http://naruto.wikia.com/wiki/Infinite_Tsukuyomi)

## See Also

servr: [httpw](#)

---

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

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

... 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 `` 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>

## 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

decktape, [2](#)

html\_document, [5](#)

http, [4](#)

inf\_mr(infinite\_moon\_reader), [3](#)

infinite\_moon\_reader, [3](#)

moon\_reader, [3](#), [4](#), [5](#)

summon\_remark, [5](#), [5](#)

tsukuyomi(moon\_reader), [4](#)