

# Package ‘xaringan’

March 8, 2022

**Type** Package

**Title** Presentation Ninja

**Version** 0.23

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

**Imports** htmltools, knitr (>= 1.30), servr (>= 0.13), xfun (>= 0.18), rmarkdown (>= 2.8)

**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** 7.1.2

**NeedsCompilation** no

**Author** Yihui Xie [aut, cre] (<<https://orcid.org/0000-0003-0645-5666>>),  
Alessandro Gasparini [ctb] (<<https://orcid.org/0000-0002-8319-7624>>),  
Benjie Gillam [ctb],  
Claus Thorn Ekstrøm [ctb],  
Daniel Anderson [ctb],  
Dawei Lang [ctb],  
Deo Salil [ctb],  
Emi Tanaka [ctb],  
Garrick Aden-Buie [ctb] (<<https://orcid.org/0000-0002-7111-0077>>),  
Iñaki Ucar [ctb] (<<https://orcid.org/0000-0001-6403-5550>>),  
John Little [ctb],  
Joselyn Chávez [ctb] (<<https://orcid.org/0000-0002-4974-4591>>),  
Joseph Casillas [ctb],  
Lucy D'Agostino McGowan [ctb] (<<https://orcid.org/0000-0001-7297-9359>>),  
Malcolm Barrett [ctb] (<<https://orcid.org/0000-0003-0299-5825>>),  
Matthew Mark Strasiotto [ctb] (mstr3336),

Michael Wayne Kearney [ctb],  
 Nan-Hung Hsieh [ctb],  
 Ole Petter Bang [ctb] (CSS in  
 rmarkdown/templates/xaringan/resources/default.css),  
 Patrick Schratz [ctb],  
 Paul Klemm [ctb] (<<https://orcid.org/0000-0002-5985-1737>>),  
 Paul Lemmens [ctb],  
 Sean Lopp [ctb],  
 Silvia Canelon [ctb] (<<https://orcid.org/0000-0003-1709-1394>>),  
 Susan VanderPlas [ctb] (<<https://orcid.org/0000-0002-3803-0972>>),  
 Tuo Wang [ctb],  
 Waldir Leoncio [ctb],  
 Yongfu Liao [ctb],  
 Yue Jiang [ctb] (<<https://orcid.org/0000-0002-9798-5517>>),  
 Zhian N. Kamvar [ctb] (<<https://orcid.org/0000-0003-1458-7108>>)

**Maintainer** Yihui Xie <xie@yihui.name>

**Repository** CRAN

**Date/Publication** 2022-03-08 15:50:02 UTC

## R topics documented:

decktape . . . . .	2
infinite_moon_reader . . . . .	3
moon_reader . . . . .	4
summon_remark . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

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**

file	The path to the HTML presentation file. When <code>docker = FALSE</code> , 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 <code>decktape</code> .
docker	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.
version	The <b>decktape</b> 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()` 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.
...	Passed to <code>rmarkdown::render()</code> .

**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**

[https://naruto.fandom.com/wiki/Infinite\\_Tsukuyomi](https://naruto.fandom.com/wiki/Infinite_Tsukuyomi)

**See Also**

`servr::http`

---

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 <a href="https://remarkjs.com">https://remarkjs.com</a> . 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 <code>css = c('default', 'extra.css')</code> , it means default.css in this package and a user-provided extra.css. To find out all built-in CSS files, use <code>xaringan::list_css()</code> . With <b>rmarkdown</b> >= 2.8, Sass files (filenames ending with '.scss' or '.sass') can also be used, and they will be processed by the <b>sass</b> 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 <b>Mustache Karl (TM)</b> 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 <code>list(times = n, img = path)</code> : 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 <code>summon_remark()</code> ), and use the local version instead.
nature	(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 remark.js, 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; alternatively, <code>autoplay</code> can be a list of the form <code>list(interval = N, loop = TRUE)</code> , 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 <code>loop = TRUE</code> . You can also set <code>countdown</code> to a number (the number of milliseconds) to include a countdown timer on each slide. If using <code>autoplay</code> , you can optionally set <code>countdown</code> to TRUE to include a countdown equal to <code>autoplay</code> . To alter the set of classes applied to the title slide, you can optionally set <code>titleSlideClass</code> to a vector of classes; the default is <code>c("center", "middle", "inverse")</code> .
anchor_sections, ...	For <code>tsukuyomi()</code> , arguments passed to <code>moon_reader()</code> ; for <code>moon_reader()</code> , arguments passed to <code>rmarkdown::html_document()</code> .

## 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 `` or `background-image: url(PATH)`, and should not contain the string `/>` when it is written with the syntax ``. 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**

<code>version</code>	The version of <code>remark.js</code> (e.g. <code>latest</code> , <code>0.13</code> , or <code>0.14.1</code> ).
<code>to</code>	The destination directory.

# Index

decktape, [2](#)

html\_document, [5](#)

http, [4](#)

inf\_mr, [6](#)

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

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

moon\_reader, [4](#), [4](#), [6](#)

render, [4](#)

summon\_remark, [5](#), [6](#), [6](#)

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