

# Package ‘rmarkdown’

September 17, 2014

**Type** Package

**Title** Dynamic Documents for R

**Version** 0.3.3

**Date** 2014-09-08

**Author** JJ Allaire, Jonathan McPherson, Yihui Xie, Hadley Wickham, Joe Cheng, Jeff Allen

**Maintainer** JJ Allaire <jj@rstudio.com>

**Description** Convert R Markdown documents into a variety of formats including HTML, MS Word, PDF, and Beamer.

**Depends** R (>= 3.0)

**Imports** tools, utils, knitr (>= 1.6), yaml (>= 2.1.5), htmltools (>= 0.2.4), caTools

**Suggests** shiny (>= 0.10.1), testthat, digest

**SystemRequirements** pandoc (>= 1.12.3) -<http://johnmacfarlane.net/pandoc>

**URL** <http://rmarkdown.rstudio.com>

**License** GPL-3

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-09-17 18:12:29

## R topics documented:

rmarkdown-package . . . . .	2
beamer_presentation . . . . .	3
compile_notebook . . . . .	5
default_output_format . . . . .	6
draft . . . . .	6
html_document . . . . .	8

html_document_base . . . . .	10
html_fragment . . . . .	11
html_vignette . . . . .	12
includes . . . . .	13
ioslides_presentation . . . . .	14
knitr_options . . . . .	20
knitr_options_html . . . . .	20
knitr_options_pdf . . . . .	21
md_document . . . . .	21
metadata . . . . .	23
output_format . . . . .	23
pandoc_args . . . . .	24
pandoc_available . . . . .	25
pandoc_convert . . . . .	26
pandoc_options . . . . .	27
pandoc_path_arg . . . . .	28
pandoc_self_contained_html . . . . .	28
pandoc_template . . . . .	29
pdf_document . . . . .	29
relative_to . . . . .	31
render . . . . .	32
render_delayed . . . . .	34
render_supporting_files . . . . .	35
rmarkdown_format . . . . .	36
rmd_metadata . . . . .	37
run . . . . .	37
slidy_presentation . . . . .	39
tufte_handout . . . . .	41
word_document . . . . .	41
<b>Index</b>	<b>43</b>

---

rmarkdown-package

*R Markdown Document Conversion*


---

## Description

Convert R Markdown documents into a variety of formats including HTML, MS Word, PDF, and Beamer.

## Details

The **rmarkdown** package includes high level functions for converting to a variety of formats. For example:

```
render("input.Rmd", html_document())
render("input.Rmd", pdf_document())
```

You can also specify a plain markdown file in which case knitting will be bypassed:

```
render("input.md", html_document())
```

Additional options can be specified along with the output format:

```
render("input.Rmd", html_document(toc = TRUE))
render("input.Rmd", pdf_document(latex.engine = "lualatex"))
render("input.Rmd", beamer_presentation(incremental = TRUE))
```

You can also include arbitrary pandoc command line arguments along with the other options:

```
render("input.Rmd", pdf_document(toc = TRUE, "--listings"))
```

### See Also

[render](#), [html\\_document](#), [pdf\\_document](#), [word\\_document](#), [beamer\\_presentation](#)

---

beamer\_presentation    *Convert to a Beamer presentation*

---

### Description

Format for converting from R Markdown to a Beamer presentation.

### Usage

```
beamer_presentation(toc = FALSE, slide_level = NULL, incremental = FALSE,
  fig_width = 10, fig_height = 7, fig_crop = TRUE, fig_caption = FALSE,
  theme = "default", colortheme = "default", fonttheme = "default",
  highlight = "default", template = "default", keep_tex = FALSE,
  includes = NULL, pandoc_args = NULL)
```

### Arguments

toc	TRUE to include a table of contents in the output (only level 1 headers will be included in the table of contents).
slide_level	The heading level which defines individual slides. By default this is the highest header level in the hierarchy that is followed immediately by content, and not another header, somewhere in the document. This default can be overridden by specifying an explicit <code>slide.level</code> .
incremental	TRUE to render slide bullets incrementally. Note that if you want to reverse the default incremental behavior for an individual bullet you can precede it with <code>&gt;</code> . For example: <code>&gt; - Bullet Text</code>
fig_width	Default width (in inches) for figures
fig_height	Default width (in inches) for figures

fig_crop	TRUE to automatically apply the pdfcrop utility (if available) to pdf figures
fig_caption	TRUE to render figures with captions
theme	Beamer theme (e.g. "AnnArbor").
colortheme	Beamer color theme (e.g. "dolphin").
fonttheme	Beamer font theme (e.g. "structurebold").
highlight	Syntax highlighting style. Supported styles include "default", "tango", "pygments", "kate", "monochrome", "espresso", "zenburn", and "haddock". Pass NULL to prevent syntax highlighting.
template	Pandoc template to use for rendering. Pass "default" to use the rmarkdown package default template; pass NULL to use pandoc's built-in template; pass a path to use a custom template that you've created. See the documentation on <a href="#">pandoc templates</a> for more details.
keep_tex	Keep the intermediate tex file used in the conversion to PDF
includes	Named list of additional content to include within the document (typically created using the <a href="#">includes</a> function).
pandoc_args	Additional command line options to pass to pandoc

### Details

Creating Beamer output from R Markdown requires that LaTeX be installed.

For more information on markdown syntax for presentations see [producing slide shows with pandoc](#).

R Markdown documents can have optional metadata that is used to generate a document header that includes the title, author, and date. For more details see the documentation on R Markdown [metadata](#).

R Markdown documents also support citations. You can find more information on the markdown syntax for citations within the pandoc documentation on [citations](#) and [footnotes](#).

### Value

R Markdown output format to pass to [render](#)

### Examples

```
## Not run:

library(rmarkdown)

# simple invocation
render("pres.Rmd", beamer_presentation())

# specify an option for incremental rendering
render("pres.Rmd", beamer_presentation(incremental = TRUE))

## End(Not run)
```

## Description

R Markdown can also compile R scripts to a notebook which includes commentary, source code, and script output. Notebooks can be compiled to any output format including HTML, PDF, and MS Word.

## Overview

To compile a notebook from an R script you simply pass the script to `render`. For example:

```
rmarkdown::render("analysis.R")
rmarkdown::render("analysis.R", "pdf_document")
```

The first call to `render` creates an HTML document, whereas the second creates a PDF document. By default the name of the script, username, and current date and time are included in the header of the generated notebook. You can override this default behavior by including explicit metadata in a specially formatted R comment:

```
## ---
## title: "Crop Analysis Q3 2013"
## author: "John Smith"
## date: "May 3rd, 2014"
## ---
```

## Including Markdown

Note that the R comment used above to add a title, author, and date includes a single-quote as a special prefix character. This is a **roxygen2** style comment, and it's actually possible to include many such comments in an R script, all of which will be converted to markdown content within the generated notebook. For example:

```
## A script comment that includes markdown formatting.
```

Rather than displaying as an R comment in the compiled notebook any **roxygen2** style comment will be treated as markdown and rendered accordingly.

## knitr Spin

Including markdown within R comments is possible because `render` calls the `knitr spin` function to convert the R script to an Rmd file. The `spin` function also enables you to add knitr chunk options with another special comment prefix (`#+`).

Here's an example of a script that uses the various features of `spin`:

<https://github.com/yihui/knitr/blob/master/inst/examples/knitr-spin.R>

For more details on `knitr::spin` see the following documentation:

<http://yihui.name/knitr/demo/stitch/>

---

`default_output_format` *Determine the default output format for an R Markdown document*

---

### Description

Read the YAML metadata (and any common `_output.yaml` file) for the document and return the output format that will be generated by a call to [render](#).

### Usage

```
default_output_format(input, encoding = getOption("encoding"))
```

### Arguments

<code>input</code>	Input file (Rmd or plain markdown)
<code>encoding</code>	The encoding of the input file; see <a href="#">file</a>

### Details

This function is useful for front-end tools that require additional knowledge of the output to be produced by [render](#) (e.g. to customize the preview experience).

### Value

A named list with a `name` value containing the format name and an `options` value that is a list containing all the options for the format and their values. An option's default value will be returned if the option isn't set explicitly in the document.

---

`draft` *Create a new document based on a template*

---

### Description

Create (and optionally edit) a draft of an R Markdown document based on a template.

### Usage

```
draft(file, template, package = NULL, create_dir = "default", edit = TRUE)
```

## Arguments

<code>file</code>	File name for the draft
<code>template</code>	Template to use as the basis for the draft. This is either the full path to a template directory or the name of a template directory within the <code>rmarkdown/templates</code> directory of a package.
<code>package</code>	(Optional) Name of package where the template is located.
<code>create_dir</code>	TRUE to create a new directory for the document (the "default" setting leaves this behavior up to the creator of the template).
<code>edit</code>	TRUE to edit the template immediately

## Details

The `draft` function creates new R Markdown documents based on templates that are either located on the filesystem or within an R package. The template and its supporting files will be copied to the location specified by `file`.

## Value

The file name of the new document (invisibly)

## Note

An R Markdown template consists of a directory that contains a description of the template, a skeleton Rmd file used as the basis for new documents, and optionally additional supporting files that are provided along with the skeleton (e.g. a logo graphic).

If the template directory is contained within a package then it should be located at `inst/rmarkdown/templates`. For example, a package named **pubtools** that wanted to provide a template named `quarterly_report` would need to provide the following files within the `pubtools/inst/rmarkdown/templates` directory:

```
quarterly_report/template.yaml
quarterly_report/skeleton/skeleton.Rmd
```

The `template.yaml` file should include a `name` field. If you want to ensure that a new directory is always created for a given template, then you can add the `create_dir` field to the `template.yaml` file. For example:

```
create_dir: true
```

The `skeleton/skeleton.Rmd` file should include the initial contents you want for files created from this template. Additional files can be added to the `skeleton` directory, for example:

```
skeleton/logo.png
```

These files will automatically be copied to the directory containing the new R Markdown draft.

**Examples**

```
## Not run:

rmarkdown::draft("Q4Report.Rmd",
                 template="/opt/rmd/templates/quarterly_report")

rmarkdown::draft("Q4Report.Rmd",
                 template="quarterly_report", package="pubtools")

## End(Not run)
```

---

html\_document

---

*Convert to an HTML document*


---

**Description**

Format for converting from R Markdown to an HTML document.

**Usage**

```
html_document(toc = FALSE, toc_depth = 3, number_sections = FALSE,
              fig_width = 7, fig_height = 5, fig_retina = if (!fig_caption) 2,
              fig_caption = FALSE, smart = TRUE, self_contained = TRUE,
              theme = "default", highlight = "default", mathjax = "default",
              template = "default", css = NULL, includes = NULL, keep_md = FALSE,
              lib_dir = NULL, pandoc_args = NULL, ...)
```

**Arguments**

toc	TRUE to include a table of contents in the output
toc_depth	Depth of headers to include in table of contents
number_sections	TRUE to number section headings
fig_width	Default width (in inches) for figures
fig_height	Default width (in inches) for figures
fig_retina	Scaling to perform for retina displays (defaults to 2 when <code>fig_caption</code> is FALSE, which currently works for all widely used retina displays). Set to NULL to prevent retina scaling. Note that this will always be NULL when <code>keep_md</code> is specified (this is because <code>fig_retina</code> relies on outputting HTML directly into the markdown document).
fig_caption	TRUE to render figures with captions
smart	Produce typographically correct output, converting straight quotes to curly quotes, — to em-dashes, – to en-dashes, and ... to ellipses.



self_contained	Produce a standalone HTML file with no external dependencies, using data: URIs to incorporate the contents of linked scripts, stylesheets, images, and videos. Note that even for self contained documents MathJax is still loaded externally (this is necessary because of it's size).
theme	Visual theme ("default", "cerulean", "journal", "flatly", "readable", "spacelab", "united", or "cosmo"). Pass NULL for no theme (in this case you can use the css parameter to add your own styles).
highlight	Syntax highlighting style. Supported styles include "default", "tango", "pygments", "kate", "monochrome", "espresso", "zenburn", "haddock", and "textmate". Pass NULL to prevent syntax highlighting.
mathjax	Include mathjax. The "default" option uses an https URL from the official MathJax CDN. The "local" option uses a local version of MathJax (which is copied into the output directory). You can pass an alternate URL or pass NULL to exclude MathJax entirely.
template	Pandoc template to use for rendering. Pass "default" to use the rmarkdown package default template; pass NULL to use pandoc's built-in template; pass a path to use a custom template that you've created. Note that if you don't use the "default" template then some features of html_document won't be available (see the Templates section below for more details).
css	One or more css files to include
includes	Named list of additional content to include within the document (typically created using the <a href="#">includes</a> function).
keep_md	Keep the markdown file generated by knitting.
lib_dir	Directory to copy dependent HTML libraries (e.g. jquery, bootstrap, etc.) into. By default this will be the name of the document with <code>_files</code> appended to it.
pandoc_args	Additional command line options to pass to pandoc
...	Additional function arguments to pass to the base R Markdown HTML output formatter

## Details

R Markdown documents can have optional metadata that is used to generate a document header that includes the title, author, and date. For more details see the documentation on [R Markdown metadata](#).

R Markdown documents also support citations. You can find more information on the markdown syntax for citations within the pandoc documentation on [citations](#) and [footnotes](#).

## Value

R Markdown output format to pass to [render](#)

## Templates

You can provide a custom HTML template to be used for rendering. The syntax for templates is described in the documentation on [pandoc templates](#). You can also use the basic pandoc template by passing `template = NULL`.

Note however that if you choose not to use the "default" HTML template then several aspects of HTML document rendering will behave differently:

- The theme parameter does not work (you can still provide styles using the css parameter).
- For the highlight parameter, the default highlighting style will resolve to "pygments" and the "textmate" highlighting style is not available
- MathJax will not work if self\_contained is TRUE (these two options can't be used together in normal pandoc templates).

Due to the above restrictions, you might consider using the includes parameter as an alternative to providing a fully custom template.

### Examples

```
## Not run:

library(rmarkdown)

render("input.Rmd", html_document())

render("input.Rmd", html_document(toc = TRUE))

## End(Not run)
```

---

html\_document\_base      *Base output format for HTML-based output formats*

---

### Description

Creates an HTML base output format suitable for passing as the base\_format argument of the [output\\_format](#) function.

### Usage

```
html_document_base(smart = TRUE, theme = NULL, self_contained = TRUE,
  lib_dir = NULL, mathjax = "default", pandoc_args = NULL,
  template = "default", dependency_resolver = NULL,
  copy_resources = FALSE, extra_dependencies = NULL,
  bootstrap_compatible = FALSE, ...)
```

### Arguments

smart	Produce typographically correct output, converting straight quotes to curly quotes, — to em-dashes, – to en-dashes, and ... to ellipses.
theme	Visual theme ("default", "cerulean", "journal", "flatly", "readable", "spacelab", "united", or "cosmo"). Pass NULL for no theme (in this case you can use the css parameter to add your own styles).

self_contained	Produce a standalone HTML file with no external dependencies, using data: URIs to incorporate the contents of linked scripts, stylesheets, images, and videos. Note that even for self contained documents MathJax is still loaded externally (this is necessary because of it's size).
lib_dir	Directory to copy dependent HTML libraries (e.g. jquery, bootstrap, etc.) into. By default this will be the name of the document with <code>_files</code> appended to it.
mathjax	Include mathjax. The "default" option uses an https URL from the official MathJax CDN. The "local" option uses a local version of MathJax (which is copied into the output directory). You can pass an alternate URL or pass NULL to exclude MathJax entirely.
pandoc_args	Additional command line options to pass to pandoc
template	Pandoc template to use for rendering. Pass "default" to use the rmarkdown package default template; pass NULL to use pandoc's built-in template; pass a path to use a custom template that you've created. Note that if you don't use the "default" template then some features of <code>html_document</code> won't be available (see the Templates section below for more details).
dependency_resolver	A dependency resolver
copy_resources	Copy resources
extra_dependencies	Extra dependencies
bootstrap_compatible	Bootstrap compatible
...	Additional function arguments to pass to the base R Markdown HTML output formatter

**Value**

HTML base output format.

---

html_fragment	<i>Convert to an HTML fragment.</i>
---------------	-------------------------------------

---

**Description**

An html fragment is suitable for inclusion into an external html paage. See [html\\_document](#) for full details - this is a minor variation that assumes you will include the output into an existing document (e.g. a blog post).

**Usage**

```
html_fragment(number_sections = FALSE, fig_width = 7, fig_height = 5,
  fig_retina = if (!fig_caption) 2, fig_caption = FALSE, smart = TRUE,
  keep_md = FALSE, pandoc_args = NULL, ...)
```

**Arguments**

number_sections	TRUE to number section headings
fig_width	Default width (in inches) for figures
fig_height	Default width (in inches) for figures
fig_retina	Scaling to perform for retina displays (defaults to 2 when fig_caption is FALSE, which currently works for all widely used retina displays). Set to NULL to prevent retina scaling. Note that this will always be NULL when keep_md is specified (this is because fig_retina relies on outputting HTML directly into the markdown document).
fig_caption	TRUE to render figures with captions
smart	Produce typographically correct output, converting straight quotes to curly quotes, — to em-dashes, – to en-dashes, and ... to ellipses.
keep_md	Keep the markdown file generated by knitting.
pandoc_args	Additional command line options to pass to pandoc
...	Additional function arguments to pass to the base R Markdown HTML output formatter

**Value**

R Markdown output format to pass to [render](#)

---

html_vignette	<i>Convert to an HTML vignette.</i>
---------------	-------------------------------------

---

**Description**

A HTML vignette is a lightweight alternative to [html\\_document](#) suitable for inclusion in packages to be released to CRAN. It reduces the size of a basic vignette from 100k to around 10k.

**Usage**

```
html_vignette(fig_width = 3, fig_height = 3, css = NULL, ...)
```

**Arguments**

fig_width	Default width (in inches) for figures
fig_height	Default width (in inches) for figures
css	One or more css files to include
...	Additional function arguments to pass to the base R Markdown HTML output formatter

**Details**

Compared to `html_document`, it:

- never uses retina figures
- has a smaller default figure size
- uses a custom css stylesheet

**Value**

R Markdown output format to pass to [render](#)

---

includes	<i>Include content within output</i>
----------	--------------------------------------

---

**Description**

Specify additional content to be included within an output document.

**Usage**

```
includes(in_header = NULL, before_body = NULL, after_body = NULL)
```

```
includes_to_pandoc_args(includes, filter = identity)
```

**Arguments**

<code>in_header</code>	One or more files with content to be included in the header of the document.
<code>before_body</code>	One or more files with content to be included before the document body.
<code>after_body</code>	One or more files with content to be included after the document body.
<code>includes</code>	Includes to convert to pandoc args
<code>filter</code>	Filter to pre-process includes with

**Details**

Non-absolute paths for resources referenced from the `in_header`, `before_body`, and `after_body` parameters are resolved relative to the directory of the input document.

**Value**

Includes list or pandoc args

**Examples**

```
## Not run:

library(rmarkdown)

html_document(includes = includes(before_body = "header.htm"))

pdf_document(includes = includes(after_body = "footer.tex"))

## End(Not run)
```

---

ioslides\_presentation *Convert to an ioslides Presentation*

---

**Description**

Format for converting from R Markdown to an **ioslides** presentation.

**Usage**

```
ioslides_presentation(logo = NULL, incremental = FALSE, fig_width = 7.5,
  fig_height = 4.5, fig_retina = if (!fig_caption) 2, fig_caption = FALSE,
  smart = TRUE, self_contained = TRUE, widescreen = FALSE, smaller = FALSE,
  transition = "default", mathjax = "default", css = NULL, includes = NULL,
  keep_md = FALSE, lib_dir = NULL, pandoc_args = NULL, ...)
```

**Arguments**

logo	Path to file that includes a logo for use in the presentation (should be square and at least 128x128)
incremental	TRUE to render slide bullets incrementally. Note that if you want to reverse the default incremental behavior for an individual bullet you can precede it with >. For example: > - Bullet Text
fig_width	Default width (in inches) for figures
fig_height	Default width (in inches) for figures
fig_retina	Scaling to perform for retina displays (defaults to 2, which currently works for all widely used retina displays). Set to NULL to prevent retina scaling. Note that this will always be NULL when keep_md is specified (this is because fig_retina relies on outputting HTML directly into the markdown document).
fig_caption	TRUE to render figures with captions
smart	Produce typographically correct output, converting straight quotes to curly quotes, — to em-dashes, – to en-dashes, and ... to ellipses.

<code>self_contained</code>	Produce a standalone HTML file with no external dependencies, using data: URIs to incorporate the contents of linked scripts, stylesheets, images, and videos. Note that even for self contained documents MathJax is still loaded externally (this is necessary because of it's size).
<code>widescreen</code>	Display presentation with wider dimensions.
<code>smaller</code>	Use smaller text on all slides. You can also enable this for individual slides by adding the <code>.smaller</code> attribute to the slide header (see <i>Presentation Size</i> below for details).
<code>transition</code>	Speed of slide transitions. This can be "default", "slower", "faster", or a numeric value with a number of seconds (e.g. 0.5)
<code>mathjax</code>	Include mathjax. The "default" option uses an https URL from the official MathJax CDN. The "local" option uses a local version of MathJax (which is copied into the output directory). You can pass an alternate URL or pass NULL to exclude MathJax entirely.
<code>css</code>	One or more css files to include
<code>includes</code>	Named list of additional content to include within the document (typically created using the <code>includes</code> function). If a <code>before_body</code> include is specified then it will replace the standard title slide entirely.
<code>keep_md</code>	Keep the markdown file generated by knitting.
<code>lib_dir</code>	Directory to copy dependent HTML libraries (e.g. jquery, bootstrap, etc.) into. By default this will be the name of the document with <code>_files</code> appended to it.
<code>pandoc_args</code>	Additional command line options to pass to pandoc
<code>...</code>	Additional function arguments to pass to the base R Markdown HTML output formatter

**Value**

R Markdown output format to pass to `render`

**Slide Basics**

You can create a slide show broken up into sections by using the `#` and `##` heading tags (you can also create a new slide without a header using a horizontal rule (-----)). For example here's a simple slide show:

```

---
title: "Habits"
author: John Doe
date: March 22, 2005
output: ioslides_presentation
---

# In the morning

## Getting up

```

```

- Turn off alarm
- Get out of bed

## Breakfast

- Eat eggs
- Drink coffee

# In the evening

## Dinner

- Eat spaghetti
- Drink wine

-----

!picture of spaghetti(images/spaghetti.jpg)

## Going to sleep

- Get in bed
- Count sheep

```

You can add a subtitle to a slide or section by including text after the pipe (|) character. For example:

```
## Getting up | What I like to do first thing
```

## Display Modes

The following single character keyboard shortcuts enable alternate display modes:

- 'f' enable fullscreen mode
- 'w' toggle widescreen mode
- 'o' enable overview mode
- 'h' enable code highlight mode
- 'p' show presenter notes

Pressing Esc exits all of these modes. See the sections below on *Code Highlighting* and *Presenter Mode* for additional detail on those modes.

## Incremental Bullets

You can render bullets incrementally by adding the incremental option:

```

---
output:

```



```
ioslides_presentation:  
  incremental: true  
---
```

If you want to render bullets incrementally for some slides but not others you can use this syntax:

```
> - Eat eggs  
> - Drink coffee
```

### Presentation Size

You can display the presentation using a wider form factor using the `widescreen` option. You can specify that smaller text be used with the `smaller` option. For example:

```
---  
output:  
  ioslides_presentation:  
    widescreen: true  
    smaller: true  
---
```

You can also enable the `smaller` option on a slide-by-slide basis by adding the `.smaller` attribute to the slide header:

```
## Getting up {.smaller}
```

### Adding a Logo

You can add a logo to the presentation using the `logo` option (the logo should be square and at least 128x128). For example:

```
---  
output:  
  ioslides_presentation:  
    logo: logo.png  
---
```

A 128x128 version of the logo graphic will be added to the title slide and an icon version of the logo will be included in the bottom-left footer of each slide.

### Build Slides

Slides can also have a `.build` attribute that indicate that their content should be displayed incrementally. For example:

```
## Getting up {.build}
```

Slide attributes can be combined if you need to specify more than one, for example:

```
## Getting up {.smaller .build}
```

## Code Highlighting

It's possible to select subsets of code for additional emphasis by adding a special "highlight" comment around the code. For example:

```
### <b>
x <- 10
y <- x * 2
### </b>
```

The highlighted region will be displayed with a bold font. When you want to help the audience focus exclusively on the highlighted region press the 'h' key and the rest of the code will fade away.

## Tables

The ioslides template has an attractive default style for tables so you shouldn't hesitate to add tables for presenting more complex sets of information. Pandoc markdown supports several syntaxes for defining tables which are described in the [pandoc markdown specification](#).

## Advanced Layout

You can center content on a slide by adding the `.flexbox` and `.vcenter` attributes to the slide title. For example:

```
## Dinner {.flexbox .vcenter}
```

You can horizontally center content by enclosing it in a `div` tag with class `centered`. For example:

```
<div class="centered">
This text is centered.
</div>
```

You can do a two-column layout using the `columns-2` class. For example:

```
<div class="columns-2">
  ![Image](image.png)

  - Bullet 1
  - Bullet 2
  - Bullet 3
</div>
```

Note that content will flow across the columns so if you want to have an image on one side and text on the other you should make sure that the image has sufficient height to force the text to the other side of the slide.

## Text Color

You can color content using base color classes red, blue, green, yellow, and gray (or variations of them e.g. red2, red3, blue2, blue3, etc.). For example:

```
<div class="red2">  
This text is red  
</div>
```

## Presenter Mode

A separate presenter window can also be opened (ideal for when you are presenting on one screen but have another screen that's private to you). The window stays in sync with the main presentation window and also shows presenter notes and a thumbnail of the next slide. To enable presenter mode add `?presentme=true` to the URL of the presentation, for example:

```
mypresentation.html?presentme=true
```

The presenter mode window will open and will always re-open with the presentation until it's disabled with:

```
mypresentation.html?presentme=false
```

To add presenter notes to a slide you include it within a "notes" div. For example:

```
<div class="notes">  
This is my *note*.  
  
- It can contain markdown  
- like this list  
  
</div>
```

## Printing and PDF Output

You can print an ioslides presentation from within browsers that have good support for print CSS (i.e. as of this writing Google Chrome has the best support). Printing maintains most of the visual styles of the HTML version of the presentation.

To create a PDF version of a presentation you can use Print to PDF from Google Chrome.

---

knitr\_options      *Knitr options for an output format*

---

**Description**

Define the knitr options for an R Markdown output format.

**Usage**

```
knitr_options(opts_knit = NULL, opts_chunk = NULL, knit_hooks = NULL)
```

**Arguments**

opts_knit	List of package level knitr options (see <a href="#">opts_knit</a> )
opts_chunk	List of chunk level knitr options (see <a href="#">opts_chunk</a> )
knit_hooks	List of hooks for R code chunks, inline R code, and output (see <a href="#">knit_hooks</a> )

**Value**

An list that can be passed as the knitr argument of the [output\\_format](#) function.

**See Also**

[output\\_format](#)

---

knitr\_options\_html      *Knitr options for an HTML output format*

---

**Description**

Define knitr options for an R Markdown output format that creates HTML output.

**Usage**

```
knitr_options_html(fig_width, fig_height, fig_retina, keep_md)
```

**Arguments**

fig_width	Default width (in inches) for figures
fig_height	Default width (in inches) for figures
fig_retina	Scaling to perform for retina displays (defaults to 2 when <code>fig_caption</code> is FALSE, which currently works for all widely used retina displays). Set to NULL to prevent retina scaling. Note that this will always be NULL when <code>keep_md</code> is specified (this is because <code>fig_retina</code> relies on outputting HTML directly into the markdown document).
keep_md	Keep the markdown file generated by knitting.

**Value**

An list that can be passed as the `knitr` argument of the `output_format` function.

**See Also**

[knitr\\_options](#), [output\\_format](#)

---

knitr_options_pdf	<i>Knitr options for a PDF output format</i>
-------------------	----------------------------------------------

---

**Description**

Define knitr options for an R Markdown output format that creates PDF output.

**Usage**

```
knitr_options_pdf(fig_width, fig_height, fig_crop)
```

**Arguments**

<code>fig_width</code>	Default width (in inches) for figures
<code>fig_height</code>	Default width (in inches) for figures
<code>fig_crop</code>	TRUE to automatically apply the <code>pdfcrop</code> utility (if available) to pdf figures

**Value**

An list that can be passed as the `knitr` argument of the `output_format` function.

**See Also**

[knitr\\_options](#), [output\\_format](#)

---

md_document	<i>Convert to a markdown document</i>
-------------	---------------------------------------

---

**Description**

Format for converting from R Markdown to another variant of markdown (e.g. strict markdown or github flavored markdown)

**Usage**

```
md_document(variant = "markdown_strict", preserve_yaml = FALSE,
  toc = FALSE, toc_depth = 3, fig_width = 7, fig_height = 5,
  fig_retina = NULL, includes = NULL, pandoc_args = NULL)
```

**Arguments**

variant	Markdown variant to produce (defaults to "markdown_strict"). Other valid values are "markdown_github", "markdown_mmd", "markdown_phpextra", or even "markdown" (which produces pandoc markdown). You can also compose custom markdown variants, see the documentation on <a href="#">pandoc's markdown</a> for details.
preserve_yaml	Preserve YAML front matter in final document.
toc	TRUE to include a table of contents in the output
toc_depth	Depth of headers to include in table of contents
fig_width	Default width (in inches) for figures
fig_height	Default width (in inches) for figures
fig_retina	Scaling to perform for retina displays. Defaults to NULL which performs no scaling. A setting of 2 will work for all widely used retina displays, but will also result in the output of <img> tags rather than markdown images due to the need to set the width of the image explicitly.
includes	Named list of additional content to include within the document (typically created using the <a href="#">includes</a> function).
pandoc_args	Additional command line options to pass to pandoc

**Details**

R Markdown documents can have optional metadata that is used to generate a document header that includes the title, author, and date. For more details see the documentation on [R Markdown metadata](#).

R Markdown documents also support citations. You can find more information on the markdown syntax for citations within the pandoc documentation on [citations](#) and [footnotes](#).

**Value**

R Markdown output format to pass to [render](#)

**Examples**

```
## Not run:

library(rmarkdown)

render("input.Rmd", md_document())

render("input.Rmd", md_document(variant = "github_flavored_markdown"))

## End(Not run)
```

---

metadata	<i>The YAML metadata of the current R Markdown document</i>
----------	-------------------------------------------------------------

---

**Description**

The object `metadata` stores the YAML metadata of the current R Markdown document as a list, which you may use in the R code chunks, e.g. `metadata$title` (the title of the document), `metadata$author`, and `metadata$foo` (if you have a YAML field named `foo`), etc.

**Usage**

```
metadata
```

**Format**

```
list()
```

---

output_format	<i>Define an R Markdown output format</i>
---------------	-------------------------------------------

---

**Description**

Define an R Markdown output format based on a combination of knitr and pandoc options.

**Usage**

```
output_format(knitr, pandoc, keep_md = FALSE, clean_supporting = TRUE,
  pre_processor = NULL, post_processor = NULL, base_format = NULL)
```

**Arguments**

<code>knitr</code>	Knitr options for an output format (see <a href="#">knitr_options</a> )
<code>pandoc</code>	Pandoc options for an output format (see <a href="#">pandoc_options</a> )
<code>keep_md</code>	Keep the markdown file generated by knitting. Note that if this is TRUE then <code>clean_supporting</code> will always be FALSE.
<code>clean_supporting</code>	Cleanup any supporting files after conversion see <a href="#">render_supporting_files</a>
<code>pre_processor</code>	An optional pre-processor function that receives the <code>metadata</code> , <code>input_file</code> , <code>runtime</code> , <code>knit_meta</code> , and <code>files_dir</code> and can return additional arguments to pass to pandoc.
<code>post_processor</code>	An optional post-processor function that receives the <code>metadata</code> , <code>input_file</code> , <code>output_file</code> , <code>clean</code> , and <code>verbose</code> parameters, and can return an alternative <code>output_file</code> .
<code>base_format</code>	An optional format to extend.

**Value**

An R Markdown output format definition that can be passed to [render](#).

**See Also**

[render](#), [knitr\\_options](#), [pandoc\\_options](#)

**Examples**

```
## Not run:
output_format(knitr = knitr_options(opts_chunk = list(dev = 'png')),
              pandoc = pandoc_options(to = "html"))

## End(Not run)
```

---

pandoc\_args

*Functions for generating pandoc command line arguments*

---

**Description**

Functions that assist in creating various types of pandoc command line arguments (e.g. for templates, table of contents, highlighting, and content includes)

**Usage**

```
pandoc_variable_arg(name, value)

pandoc_include_args(in_header = NULL, before_body = NULL,
                   after_body = NULL)

pandoc_highlight_args(highlight, default = "tango")

pandoc_toc_args(toc, toc_depth = 3)
```

**Arguments**

name	Name of template variable to set.
value	Value of template variable.
in_header	One or more files with content to be included in the header of the document.
before_body	One or more files with content to be included before the document body.
after_body	One or more files with content to be included after the document body.
highlight	The name of a pandoc syntax highlighting theme.
default	The highlighting theme to use if "default" is specified.
toc	TRUE to include a table of contents in the output.
toc_depth	Depth of headers to include in table of contents.



**Details**

Non-absolute paths for resources referenced from the `in_header`, `before_body`, and `after_body` parameters are resolved relative to the directory of the input document.

**Value**

A character vector with pandoc command line arguments

**Examples**

```
## Not run:  
  
library(rmarkdown)  
  
pandoc_include_args(before_body = "header.htm")  
pandoc_include_args(before_body = "header.tex")  
  
pandoc_highlight_args("kate")  
  
pandoc_toc_args(toc = TRUE, toc_depth = 2)  
  
## End(Not run)
```

---

pandoc_available	<i>Check whether pandoc is available</i>
------------------	------------------------------------------

---

**Description**

Determine whether pandoc is currently available on the system, optionally checking for a specific version or greater.

**Usage**

```
pandoc_available(version = NULL)
```

**Arguments**

version	Required version of pandoc
---------	----------------------------

**Details**

The system path as well as the version of pandoc shipped with RStudio (if running under RStudio) are scanned for pandoc and the highest version available is used.

**Value**

Logical indicating whether a version of pandoc is available

**Examples**

```
## Not run:
library(rmarkdown)

if (pandoc_available())

  cat("pandoc is available!\n")

if (pandoc_available("1.12.3"))
  cat("required version of pandoc is available!\n")

## End(Not run)
```

---

pandoc_convert	<i>Convert a document with pandoc</i>
----------------	---------------------------------------

---

**Description**

Convert documents to and from various formats using the pandoc utility.

**Usage**

```
pandoc_convert(input, to = NULL, from = NULL, output = NULL,
  citeproc = FALSE, options = NULL, verbose = FALSE, wd = NULL)
```

**Arguments**

input	Character vector containing paths to input files (files must be UTF-8 encoded)
to	Format to convert to (if not specified, you must specify output)
from	Format to convert from (if not specified then the format is determined based on the file extension of input).
output	Output file (if not specified then determined based on format being converted to)
citeproc	TRUE to run the pandoc-citeproc filter (for processing citations) as part of the conversion
options	Character vector of command line options to pass to pandoc.
verbose	TRUE to show the pandoc command line which was executed
wd	Working directory in which code will be executed. If not supplied, defaults to the common base directory of input

**Details**

Supported input and output formats are described in the [pandoc user guide](#).

The system path as well as the version of pandoc shipped with RStudio (if running under RStudio) are scanned for pandoc and the highest version available is used.

**Examples**

```
## Not run:
library(rmarkdown)

# convert markdown to various formats
pandoc_convert("input.md", to = "html")
pandoc_convert("input.md", to = "pdf")

# process citations
pandoc_convert("input.md", to = "html", citeproc = TRUE)

# add some pandoc options
pandoc_convert("input.md", to="pdf", options = c("--listings"))

## End(Not run)
```

---

pandoc_options	<i>Pandoc options for an output format</i>
----------------	--------------------------------------------

---

**Description**

Define the pandoc options for an R Markdown output format.

**Usage**

```
pandoc_options(to, from = rmarkdown_format(), args = NULL,
  keep_tex = FALSE, ext = NULL)
```

**Arguments**

<code>to</code>	Pandoc format to convert to
<code>from</code>	Pandoc format to convert from
<code>args</code>	Character vector of command line arguments to pass to pandoc
<code>keep_tex</code>	Keep the intermediate tex file used in the conversion to PDF (applies only to 'latex' and 'beamer' target formats)
<code>ext</code>	File extension (e.g. ".tex") for output file (if NULL chooses default based on to). This is typically used to force the final output of a latex or beamer conversion to be .tex rather than .pdf.

**Details**

The `from` argument should be used very cautiously as it's important for users to be able to rely on a stable definition of supported markdown extensions.

**Value**

An list that can be passed as the `pandoc` argument of the `output_format` function.

**See Also**

[output\\_format](#), [rmarkdown\\_format](#)

---

<code>pandoc_path_arg</code>	<i>Transform path for passing to pandoc</i>
------------------------------	---------------------------------------------

---

**Description**

Transform a path for passing to pandoc on the command line. Calls `path.expand` on all platforms. On Windows, transform it to a short path name if it contains spaces, and then convert forward slashes to back slashes (as required by pandoc for some path references)

**Usage**

```
pandoc_path_arg(path)
```

**Arguments**

<code>path</code>	Path to transform
-------------------	-------------------

**Value**

Transformed path that can be passed to pandoc on the command line

---

<code>pandoc_self_contained_html</code>	<i>Create a self-contained HTML document using pandoc.</i>
-----------------------------------------	------------------------------------------------------------

---

**Description**

Create a self-contained HTML document by base64 encoding images, scripts, and stylesheets referenced by the input document.

**Usage**

```
pandoc_self_contained_html(input, output)
```

**Arguments**

<code>input</code>	Input html file to create self-contained version of.
<code>output</code>	Path to save output.

**Value**

(Invisibly) The path of the generated file.

---

pandoc\_template      *Render a pandoc template.*

---

### Description

Use the pandoc templating engine to render a text file. Substitutions are done using the metadata list passed to the function.

### Usage

```
pandoc_template(metadata, template, output, verbose = FALSE)
```

### Arguments

metadata	A named list containing metadata to pass to template.
template	Path to a pandoc template.
output	Path to save output.
verbose	TRUE to show the pandoc command line which was executed.

### Value

(Invisibly) The path of the generated file.

---

pdf\_document      *Convert to a PDF document*

---

### Description

Format for converting from R Markdown to a PDF document.

### Usage

```
pdf_document(toc = FALSE, toc_depth = 2, number_sections = FALSE,
  fig_width = 6.5, fig_height = 4.5, fig_crop = TRUE,
  fig_caption = FALSE, highlight = "default", template = "default",
  keep_tex = FALSE, latex_engine = "pdflatex", includes = NULL,
  pandoc_args = NULL)
```

**Arguments**

toc	TRUE to include a table of contents in the output
toc_depth	Depth of headers to include in table of contents
number_sections	TRUE to number section headings
fig_width	Default width (in inches) for figures
fig_height	Default width (in inches) for figures
fig_crop	TRUE to automatically apply the pdfcrop utility (if available) to pdf figures
fig_caption	TRUE to render figures with captions
highlight	Syntax highlighting style. Supported styles include "default", "tango", "pygments", "kate", "monochrome", "espresso", "zenburn", and "haddock". Pass NULL to prevent syntax highlighting.
template	Pandoc template to use for rendering. Pass "default" to use the rmarkdown package default template; pass NULL to use pandoc's built-in template; pass a path to use a custom template that you've created. See the documentation on <a href="#">pandoc templates</a> for more details.
keep_tex	Keep the intermediate tex file used in the conversion to PDF
latex_engine	LaTeX engine for producing PDF output. Options are "pdflatex", "lualatex", and "xelatex".
includes	Named list of additional content to include within the document (typically created using the <a href="#">includes</a> function).
pandoc_args	Additional command line options to pass to pandoc

**Details**

Creating PDF output from R Markdown requires that LaTeX be installed.

R Markdown documents can have optional metadata that is used to generate a document header that includes the title, author, and date. For more details see the documentation on [R Markdown metadata](#).

R Markdown documents also support citations. You can find more information on the markdown syntax for citations within the pandoc documentation on [citations](#) and [footnotes](#).

Many aspects of the LaTeX template used to create PDF documents can be customized using metadata. For example:

```

---
title: "Crop Analysis Q3 2013"
fontsize: 11pt
geometry: margin=1in
---
```

Available metadata variables include:

lang Document language code

fontsize Font size (e.g. 10pt, 11pt, 12pt)  
 documentclass LaTeX document class (e.g. article)  
 classoption Option for documentclass (e.g. oneside); may be repeated  
 geometry Options for geometry class (e.g. margin=1in); may be repeated  
 mainfont, sansfont, monofont, mathfont Document fonts (works only with xelatex and lualatex, see the latex\_engine option)  
 linkcolor, urlcolor, citecolor Color for internal, external, and citation links (red, green, magenta, cyan, blue, black)

### Value

R Markdown output format to pass to [render](#)

### Examples

```

## Not run:

library(rmarkdown)

# simple invocation
render("input.Rmd", pdf_document())

# specify an option for latex engine
render("input.Rmd", pdf_document(latex_engine = "lualatex"))

# add a table of contents and pass an option to pandoc
render("input.Rmd", pdf_document(toc = TRUE, "--listings"))

## End(Not run)

```

---

relative_to	<i>Relative path utility function Given a directory and a file, return a relative path from the directory to the file, or the unmodified file path if the file does not appear to be in the directory.</i>
-------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

### Description

Relative path utility function Given a directory and a file, return a relative path from the directory to the file, or the unmodified file path if the file does not appear to be in the directory.

### Usage

```
relative_to(dir, file)
```

### Arguments

dir	Directory
file	File

**Value**

Relative path from the directory to the file (or the unmodified file path if the file does not appear to be in the directory).

---

 render

*Render R Markdown*


---

**Description**

Render the input file to the specified output format using pandoc. If the input requires knitting then [knit](#) is called prior to pandoc.

**Usage**

```
render(input, output_format = NULL, output_file = NULL, output_dir = NULL,
       output_options = NULL, intermediates_dir = NULL,
       runtime = c("auto", "static", "shiny"),
       clean = TRUE, envir = parent.frame(), quiet = FALSE,
       encoding = getOption("encoding"))
```

**Arguments**

input	Input file (R script, Rmd, or plain markdown).
output_format	R Markdown output format to convert to. Pass "all" to render all formats defined within the file. Pass the name of a format (e.g. "html_document") to render a single format or pass a vector of format names to render multiple formats. Alternatively you can pass an output format object; e.g. <code>html_document()</code> . If NULL is passed then the output format is the first one defined in the YAML metadata of the input file (defaulting to HTML if none is specified).
output_options	List of output options that can override the options specified in metadata (e.g. could be used to force <code>self_contained</code> or <code>mathjax = "local"</code> ). Note that this is only valid when the output format is read from metadata (i.e. not a custom format object passed to <code>output_format</code> ).
output_file	Output file. If NULL then a default based on the name of the input file is chosen.
output_dir	Output directory. An alternate directory to write the output file to (defaults to the directory of the input file).
intermediates_dir	Intermediate files directory. If NULL, intermediate files are written to the same directory as the input file; otherwise.
runtime	The runtime target for rendering. <code>static</code> produces output intended for static files; <code>shiny</code> produces output suitable for use in a Shiny document (see <a href="#">run</a> ). The default, <code>auto</code> , allows the runtime target specified in the YAML metadata to take precedence, and renders for a <code>static</code> runtime target otherwise.
clean	TRUE to clean intermediate files created during rendering.



envir	The environment in which the code chunks are to be evaluated during knitting (can use <a href="#">new.env()</a> to guarantee an empty new environment).
quiet	TRUE to suppress printing of the pandoc command line.
encoding	The encoding of the input file; see <a href="#">file</a> .

### Details

For additional details on rendering R scripts see [Compiling R scripts to a notebook](#).

If no `output_format` parameter is specified then the output format is read from the YAML front-matter of the input file. For example, the following YAML would yield a PDF document:

```
output: pdf_document
```

Additional format options can also be specified in metadata. For example:

```
output:
  pdf_document:
    toc: true
    highlight: zenburn
```

Multiple formats can be specified in metadata. If no `output_format` is passed to render then the first one defined will be used:

```
output:
  pdf_document:
    toc: true
    highlight: zenburn
  html_document:
    toc: true
    theme: united
```

Formats specified in metadata can be any one of the built in formats (e.g. [html\\_document](#), [pdf\\_document](#)) or a format defined in another package (e.g. `pkg::custom_format`).

If there is no format defined in the YAML then [html\\_document](#) will be used.

### Value

The compiled document is written into the output file, and the path of the output file is returned.

### R Markdown

R Markdown supports all of the base pandoc markdown features as well as some optional features for compatibility with GitHub Flavored Markdown (which previous versions of R Markdown were based on). See [rmarkdown\\_format](#) for details.

### See Also

[knit](#), [output\\_format](#), [pandoc](#)

## Examples

```
## Not run:

library(rmarkdown)

# render the default (first) format defined in the file
render("input.Rmd")

# render all formats defined in the file
render("input.Rmd", "all")

# render a single format
render("input.Rmd", "html_document")

# render multiple formats
render("input.Rmd", c("html_document", "pdf_document"))

## End(Not run)
```

---

render\_delayed

*Delay Rendering for an Expression*

---

## Description

In a Shiny document, evaluate the given expression after the document has finished rendering, instead of during render.

## Usage

```
render_delayed(expr)
```

## Arguments

expr            The expression to evaluate.

## Details

This function is useful inside Shiny documents. It delays the evaluation of its argument until the document has finished its initial render, so that the document can be viewed before the calculation is finished.

Any expression that returns HTML can be wrapped in `render_delayed`.

## Value

An object representing the expression.

**Note**

expr is evaluated in a **copy** of the environment in which the render\_delayed call appears. Consequently, no side effects created by expr are visible in succeeding expressions, nor are changes to the environment after the call to render\_delayed visible to expr.

expr must be an expression that produces HTML.

**Examples**

```
## Not run:

# Add the following code to an R Markdown document

div(Sys.time())

render_delayed({
  Sys.sleep(3)      # simulate an expensive computation
  div(Sys.time())
})

div(Sys.time())

## End(Not run)
```

---

render\_supporting\_files

*Render supporting files for an input document*

---

**Description**

Render (copy) required supporting files for an input document to the \_files directory associated with the document.

**Usage**

```
render_supporting_files(from, files_dir, rename_to = NULL)
```

**Arguments**

from	Directory to copy from
files_dir	Directory to copy files into
rename_to	Optional rename of source directory after it is copied

**Value**

The relative path to the supporting files. This path is suitable for inclusion in HTMLhref and src attributes.

---

rmarkdown_format	<i>R Markdown input format definition</i>
------------------	-------------------------------------------

---

### Description

Compose a pandoc markdown input definition for R Markdown that can be passed as the from argument of [pandoc\\_options](#).

### Usage

```
rmarkdown_format(extensions = NULL)
```

### Arguments

extensions	Markdown extensions to be added or removed from the default definition of R Markdown.
------------	---------------------------------------------------------------------------------------

### Details

By default R Markdown is defined as all pandoc markdown extensions with the following tweaks for backward compatibility with the markdown package (+ features are added, - features are removed):

```
+autolink_bare_uris  
+ascii_identifier  
+tex_math_single_backslash
```

For more on pandoc markdown see the [pandoc markdown specification](#).

### Value

Pandoc markdown format specification

### See Also

[output\\_format](#), [pandoc\\_options](#)

### Examples

```
## Not run:  
rmarkdown_format("-implicit_figures")  
  
## End(Not run)
```

## Description

Rmd files include a metadata section (typically located at the top of the file) that can specify (among other things) the title, author, and date of the document. Metadata adheres to the [YAML](#) format and is delimited by lines containing three dashes (---). Here is an example metadata section:

```
---  
title: "Crop Analysis Q3 2013"  
author: Martha Smith  
date: October 23rd, 2013  
---
```

Note that the title field is quoted. This is because titles often contained embedded colons (:) and colons followed by a space need to be quoted in YAML.

## Details

When title, author, and date metadata is provided it's used to automatically create a title section within output documents. If you don't want this section included in your document then you should remove the corresponding metadata fields.

When generating PDF and Beamer output there are also a number of other metadata fields that can be included to customize the appearance and theme of PDF output. For more details see the documentation for [pdf\\_document](#) and [beamer\\_presentation](#).

## Description

Start a Shiny server for the given document, and render it for display.

## Usage

```
run(file = "index.Rmd", dir = dirname(file), auto_reload = TRUE,  
    shiny_args = NULL, render_args = NULL)
```

## Arguments

<code>file</code>	Path to the R Markdown document to launch in a web browser. Defaults to <code>index.Rmd</code> in the current working directory, but may be <code>NULL</code> to skip launching a browser.
<code>dir</code>	The directory from which to read input documents. Defaults to the parent directory of <code>file</code> .
<code>auto_reload</code>	If <code>TRUE</code> (the default), automatically reload the Shiny application when the file currently being viewed is changed on disk.
<code>shiny_args</code>	Additional arguments to <code>runApp</code> .
<code>render_args</code>	Additional arguments to <code>render</code> .

## Details

The `run` function runs a Shiny document by starting a Shiny server associated with the document. The `shiny_args` parameter can be used to configure the server; see the `runApp` documentation for details.

Once the server is started, the document will be rendered using `render`. The server will initiate a render of the document whenever necessary, so it is not necessary to call `run` every time the document changes: if `auto_reload` is `TRUE`, saving the document will trigger a render. You can also manually trigger a render by reloading the document in a Web browser.

The server will render any R Markdown (`.Rmd`) document in `dir`; the `file` argument specifies only the initial document to be rendered and viewed. You can therefore link to other documents in the directory using standard Markdown syntax, e.g. `[Analysis Page 2](page2.Rmd)`.

If you wish to share R code between your documents, place it in a file named `global.R` in `dir`; it will be sourced into the global environment.

## Value

Invisible `NULL`.

## Note

Unlike `render`, `run` does not render the document to a file on disk. In most cases a Web browser will be started automatically to view the document; see `launch.browser` in the `runApp` documentation for details.

When using an external web browser with the server, specify the name of the R Markdown file to view in the URL (e.g. `http://127.0.0.1:1234/foo.Rmd`). A URL without a filename will show `index.Rmd`, if it exists in `dir`.

## Examples

```
## Not run:  
  
# Run the Shiny document "index.Rmd" in the current directory  
rmarkdown::run()  
  
# Run the Shiny document "shiny_doc.Rmd" on port 8241
```

```
rmarkdown::run("shiny_doc.Rmd", shiny_args = list(port = 8241))
```

```
## End(Not run)
```

---

slidy\_presentation      *Convert to a slidy presentation*

---

## Description

Format for converting from R Markdown to a slidy presentation.

## Usage

```
slidy_presentation(incremental = FALSE, duration = NULL, footer = NULL,
  font_adjustment = 0, fig_width = 8, fig_height = 6, fig_retina = if
  (!fig_caption) 2, fig_caption = FALSE, smart = TRUE,
  self_contained = TRUE, highlight = "default", mathjax = "default",
  template = "default", css = NULL, includes = NULL, keep_md = FALSE,
  lib_dir = NULL, pandoc_args = NULL, ...)
```

## Arguments

incremental	TRUE to render slide bullets incrementally. Note that if you want to reverse the default incremental behavior for an individual bullet you can precede it with >. For example: > - Bullet Text
duration	Duration (in minutes) of the slide deck. This value is used to add a countdown timer to the slide footer.
footer	Footer text (e.g. organization name and/or copyright)
font_adjustment	Increase or decrease the default font size (e.g. -1 or +1). You can also manually adjust the font size during the presentation using the 'S' (smaller) and 'B' (bigger) keys.
fig_width	Default width (in inches) for figures
fig_height	Default width (in inches) for figures
fig_retina	Scaling to perform for retina displays (defaults to 2 when fig_caption is FALSE, which currently works for all widely used retina displays). Set to NULL to prevent retina scaling. Note that this will always be NULL when keep_md is specified (this is because fig_retina relies on outputting HTML directly into the markdown document).
fig_caption	TRUE to render figures with captions
smart	Produce typographically correct output, converting straight quotes to curly quotes, — to em-dashes, – to en-dashes, and ... to ellipses.

self_contained	Produce a standalone HTML file with no external dependencies, using data: URIs to incorporate the contents of linked scripts, stylesheets, images, and videos. Note that even for self contained documents MathJax is still loaded externally (this is necessary because of it's size).
highlight	Syntax highlighting style. Supported styles include "default", "tango", "pygments", "kate", "monochrome", "espresso", "zenburn", and "haddock". Pass NULL to prevent syntax highlighting.
mathjax	Include mathjax. The "default" option uses an https URL from the official MathJax CDN. The "local" option uses a local version of MathJax (which is copied into the output directory). You can pass an alternate URL or pass NULL to exclude MathJax entirely.
template	Pandoc template to use for rendering. Pass "default" to use the rmarkdown package default template; pass NULL to use pandoc's built-in template; pass a path to use a custom template that you've created. See the documentation on <a href="#">pandoc templates</a> for more details.
css	One or more css files to include
includes	Named list of additional content to include within the document (typically created using the <a href="#">includes</a> function).
keep_md	Keep the markdown file generated by knitting.
lib_dir	Directory to copy dependent HTML libraries (e.g. jquery, bootstrap, etc.) into. By default this will be the name of the document with <code>_files</code> appended to it.
pandoc_args	Additional command line options to pass to pandoc
...	Additional function arguments to pass to the base R Markdown HTML output formatter

### Details

For more information on markdown syntax for presentations see [producing slide shows with pandoc](#).

### Value

R Markdown output format to pass to [render](#)

### Examples

```
## Not run:

library(rmarkdown)

# simple invocation
render("pres.Rmd", slidy_presentation())

# specify an option for incremental rendering
render("pres.Rmd", slidy_presentation(incremental = TRUE))

## End(Not run)
```



---

tufte_handout	<i>Tufte handout format (PDF)</i>
---------------	-----------------------------------

---

**Description**

Template for creating a handout according to the style of Edward R. Tufte and Richard Feynman.

**Usage**

```
tufte_handout(fig_width = 4, fig_height = 2.5, fig_crop = TRUE,
  highlight = "default", keep_tex = FALSE, includes = NULL,
  pandoc_args = NULL)
```

**Arguments**

fig_width	Default width (in inches) for figures
fig_height	Default width (in inches) for figures
fig_crop	TRUE to automatically apply the pdfcrop utility (if available) to pdf figures
highlight	Syntax highlighting style. Supported styles include "default", "tango", "pygments", "kate", "monochrome", "espresso", "zenburn", and "haddock". Pass NULL to prevent syntax highlighting.
keep_tex	Keep the intermediate tex file used in the conversion to PDF
includes	Named list of additional content to include within the document (typically created using the <a href="#">includes</a> function).
pandoc_args	Additional command line options to pass to pandoc

---

word_document	<i>Convert to an MS Word document</i>
---------------	---------------------------------------

---

**Description**

Format for converting from R Markdown to an MS Word document.

**Usage**

```
word_document(fig_width = 5, fig_height = 4, fig_caption = FALSE,
  highlight = "default", reference_docx = "default", pandoc_args = NULL)
```

## Arguments

<code>fig_width</code>	Default width (in inches) for figures
<code>fig_height</code>	Default width (in inches) for figures
<code>fig_caption</code>	TRUE to render figures with captions
<code>highlight</code>	Syntax highlighting style. Supported styles include "default", "tango", "pygments", "kate", "monochrome", "espresso", "zenburn", and "haddock". Pass NULL to prevent syntax highlighting.
<code>reference_docx</code>	Use the specified file as a style reference in producing a docx file. For best results, the reference docx should be a modified version of a docx file produced using pandoc. Pass "default" to use the rmarkdown default styles.
<code>pandoc_args</code>	Additional command line options to pass to pandoc

## Details

R Markdown documents can have optional metadata that is used to generate a document header that includes the title, author, and date. For more details see the documentation on [R Markdown metadata](#).

R Markdown documents also support citations. You can find more information on the markdown syntax for citations within the pandoc documentation on [citations](#) and [footnotes](#).

## Value

R Markdown output format to pass to [render](#)

## Examples

```
## Not run:  
  
library(rmarkdown)  
  
# simple invocation  
render("input.Rmd", word_document())  
  
# specify an option for syntax highlighting  
render("input.Rmd", word_document(highlight = "zenburn"))  
  
## End(Not run)
```

# Index

## \*Topic **datasets**

- metadata, [23](#)
- beamer\_presentation, [3](#), [3](#), [37](#)
- compile\_notebook, [5](#)
- Compiling R scripts to a notebook, [33](#)
- default\_output\_format, [6](#)
- draft, [6](#)
- file, [6](#), [33](#)
- html\_document, [3](#), [8](#), [11](#), [12](#), [33](#)
- html\_document\_base, [10](#)
- html\_fragment, [11](#)
- html\_vignette, [12](#)
- includes, [4](#), [9](#), [13](#), [15](#), [22](#), [30](#), [40](#), [41](#)
- includes\_to\_pandoc\_args (includes), [13](#)
- ioslides\_presentation, [14](#)
- knit, [32](#), [33](#)
- knit\_hooks, [20](#)
- knitr\_options, [20](#), [21](#), [23](#), [24](#)
- knitr\_options\_html, [20](#)
- knitr\_options\_pdf, [21](#)
- md\_document, [21](#)
- metadata, [4](#), [9](#), [22](#), [23](#), [30](#), [42](#)
- new.env, [33](#)
- opts\_chunk, [20](#)
- opts\_knit, [20](#)
- output\_format, [10](#), [20](#), [21](#), [23](#), [27](#), [28](#), [33](#), [36](#)
- pandoc\_args, [24](#)
- pandoc\_available, [25](#)
- pandoc\_convert, [26](#)
- pandoc\_highlight\_args (pandoc\_args), [24](#)
- pandoc\_include\_args (pandoc\_args), [24](#)
- pandoc\_options, [23](#), [24](#), [27](#), [36](#)
- pandoc\_path\_arg, [28](#)
- pandoc\_self\_contained\_html, [28](#)
- pandoc\_template, [29](#)
- pandoc\_toc\_args (pandoc\_args), [24](#)
- pandoc\_variable\_arg (pandoc\_args), [24](#)
- path.expand, [28](#)
- pdf\_document, [3](#), [29](#), [33](#), [37](#)
- relative\_to, [31](#)
- render, [3-6](#), [9](#), [12](#), [13](#), [15](#), [22](#), [24](#), [31](#), [32](#), [38](#), [40](#), [42](#)
- render\_delayed, [34](#)
- render\_supporting\_files, [23](#), [35](#)
- rmarkdown (rmarkdown-package), [2](#)
- rmarkdown-package, [2](#)
- rmarkdown\_format, [28](#), [33](#), [36](#)
- rmd\_metadata, [37](#)
- run, [32](#), [37](#)
- runApp, [38](#)
- slidy\_presentation, [39](#)
- tufte\_handout, [41](#)
- word\_document, [3](#), [41](#)