

# Package ‘bookdown’

October 13, 2020

**Type** Package

**Title** Authoring Books and Technical Documents with R Markdown

**Version** 0.21

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

**Description**

Output formats and utilities for authoring books and technical documents with R Markdown.

**License** GPL-3

**Imports** htmltools (>= 0.3.6), knitr (>= 1.22), rmarkdown (>= 2.4),  
xfun (>= 0.13), tinytex (>= 0.12), yaml (>= 2.1.19)

**Suggests** htmlwidgets, rstudioapi, miniUI, rsconnect (>= 0.4.3), servr  
(>= 0.13), shiny, testit (>= 0.9), tuftes, webshot

**URL** <https://github.com/rstudio/bookdown>

**BugReports** <https://github.com/rstudio/bookdown/issues>

**SystemRequirements** Pandoc (>= 1.17.2)

**LazyData** TRUE

**RoxygenNote** 7.1.1

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Yihui Xie [aut, cre] (<<https://orcid.org/0000-0003-0645-5666>>),  
JJ Allaire [ctb],  
Albert Kim [ctb],  
Alessandro Samuel-Rosa [ctb],  
Andrzej Oles [ctb],  
Atsushi Yasumoto [ctb] (<<https://orcid.org/0000-0002-8335-495X>>),  
Aust Frederik [ctb] (<<https://orcid.org/0000-0003-4900-788X>>),  
Bastiaan Quast [ctb],  
Ben Marwick [ctb],  
Chester Ismay [ctb],  
Christophe Dervieux [ctb],  
Clifton Franklund [ctb],  
Daniel Emaasit [ctb],

David Shuman [ctb],  
 Dean Attali [ctb],  
 Drew Tyre [ctb],  
 Ellis Valentiner [ctb],  
 Frans van Dunne [ctb],  
 Hadley Wickham [ctb],  
 Jeff Allen [ctb],  
 Jennifer Bryan [ctb],  
 Jonathan McPhers [ctb],  
 JooYoung Seo [ctb] (<<https://orcid.org/0000-0002-4064-6012>>),  
 Joyce Robbins [ctb],  
 Junwen Huang [ctb],  
 Kevin Cheung [ctb],  
 Kevin Ushey [ctb],  
 Kim Seonghyun [ctb],  
 Kirill Muller [ctb],  
 Luciano Selzer [ctb],  
 Matthew Lincoln [ctb],  
 Maximilian Held [ctb],  
 Michael Sachs [ctb],  
 Michal Bojanowski [ctb],  
 Nathan Werth [ctb],  
 Noam Ross [ctb],  
 Peter Hickey [ctb],  
 Romain Lesur [ctb] (<<https://orcid.org/0000-0002-0721-5595>>),  
 Sahir Bhatnagar [ctb],  
 Steve Simpson [ctb],  
 Thierry Onkelinx [ctb] (<<https://orcid.org/0000-0001-8804-4216>>),  
 Vincent Fulco [ctb],  
 Yixuan Qiu [ctb],  
 Zhuoer Dong [ctb],  
 RStudio, PBC [cph],  
 Bartek Szopka [ctb] (The jQuery Highlight plugin),  
 Zeno Rocha [cph] (clipboard.js library),  
 jQuery Foundation [cph] (jQuery library),  
 jQuery contributors [ctb, cph] (jQuery library; authors listed in  
 inst/resources/AUTHORS),  
 MathQuill contributors [ctb] (The MathQuill library; authors listed in  
 inst/resources/AUTHORS),  
 FriendCode Inc [cph, ctb] (The gitbook style, with modifications)

**Repository** CRAN

**Date/Publication** 2020-10-13 05:00:06 UTC

## R topics documented:

bookdown_site . . . . .	3
calibre . . . . .	3

<i>bookdown_site</i>	3
clean_book . . . . .	4
epub_book . . . . .	4
fence_theorems . . . . .	5
gitbook . . . . .	6
html_chapters . . . . .	7
html_document2 . . . . .	9
pdf_book . . . . .	11
publish_book . . . . .	12
render_book . . . . .	13
serve_book . . . . .	14

**Index** **16**

*bookdown\_site*                      *R Markdown site generator for bookdown*

**Description**

Implementation of custom R Markdown site generator for bookdown.

**Usage**

```
bookdown_site(input, ...)
```

**Arguments**

input	Website directory (or the name of a file within the directory).
...	Currently unused.

*calibre*                              *A wrapper function to convert e-books using Calibre*

**Description**

This function calls the command `ebook-convert` in Calibre (<https://calibre-ebook.com>) to convert e-books.

**Usage**

```
calibre(input, output, options = "")
```

**Arguments**

input	The input filename.
output	The output filename or extension (if only an extension is provided, the output filename will be the input filename with its extension replaced by output; for example, <code>calibre('foo.epub', 'mobi')</code> generates 'foo.mobi').
options	A character vector of additional options to be passed to <code>ebook-convert</code> .

**Value**

The output filename.

---

clean_book	<i>Clean up the output files and directories from the book</i>
------------	--

---

**Description**

After a book is rendered, there will be a series of output files and directories created in the book root directory, typically including ‘\*\_files/’, ‘\*\_cache/’, ‘\_book/’, and some HTML/LaTeX auxiliary files. These filenames depend on the book configurations. This function identifies these files and directories, and delete them if desired, so you can rebuild the book with a clean source.

**Usage**

```
clean_book(clean = getOption("bookdown.clean_book", FALSE))
```

**Arguments**

clean	Whether to delete the possible output files. If FALSE, simply print out a list of files/directories that should probably be deleted. You can set the global option <code>bookdown.clean_book = TRUE</code> to force this function to delete files. You are recommended to take a look at the list of files at least once before actually deleting them, i.e. run <code>clean_book(FALSE)</code> before <code>clean_book(TRUE)</code> .
-------	--

---

epub_book	<i>The EPUB e-book format</i>
-----------	-------------------------------

---

**Description**

Convert a book to the EPUB format, which is is an e-book format supported by many readers, such as Amazon Kindle Fire and iBooks on Apple devices.

**Usage**

```
epub_book(
  fig_width = 5,
  fig_height = 4,
  dev = "png",
  fig_caption = TRUE,
  number_sections = TRUE,
  toc = FALSE,
  toc_depth = 3,
  stylesheet = NULL,
  cover_image = NULL,
```

```

    metadata = NULL,
    chapter_level = 1,
    epub_version = c("epub3", "epub"),
    md_extensions = NULL,
    pandoc_args = NULL,
    template = "default"
)

```

### Arguments

`fig_width`, `fig_height`, `dev`, `fig_caption`  
Figure options (width, height, the graphical device, and whether to render figure captions).

`number_sections`  
Whether to number sections.

`toc`, `toc_depth`  
Whether to generate a table of contents, and its depth.

`stylesheet`  
A character vector of paths to CSS stylesheets to be applied to the eBook.

`cover_image`  
The path to a cover image.

`metadata`  
The path to the EPUB metadata file.

`chapter_level`  
The level by which the e-book is split into separate “chapter” files.

`epub_version`  
Whether to use version 3 or 2 of EPUB.

`md_extensions`  
A character string of Pandoc Markdown extensions.

`pandoc_args`  
A vector of additional Pandoc arguments.

`template`  
Pandoc template to use for rendering. Pass “default” to use Pandoc’s built-in template; pass a path to use a custom template. The default pandoc template should be sufficient for most use cases. In case you want to develop a custom template, we highly recommend to start from the default EPUB templates at <https://github.com/jgm/pandoc-templates/>.

### Note

Figure/table numbers cannot be generated if sections are not numbered (`number_sections = FALSE`).

---

<code>fence_theorems</code>	<i>Convert the syntax of theorem and proof environments from code blocks to fenced Divs</i>
-----------------------------	---

---

### Description

This function converts the syntax ‘````{theorem,label,...}`’ to ‘`::: {.theorem #label ...}`’ (Pandoc’s fenced Div) for theorem environments.

### Usage

```
fence_theorems(input, text = xfun::read_utf8(input), output = NULL)
```

**Arguments**

input	Path to an Rmd file that contains theorem environments written in the syntax of code blocks.
text	A character vector of the Rmd source. When text is provided, the input argument will be ignored.
output	The output file to write the converted input content. You can specify output to be identical to input, which means the input file will be overwritten. If you want to overwrite the input file, you are strongly recommended to put the file under version control or make a backup copy in advance.

**Value**

If output = NULL, the converted text is returned, otherwise the text is written to the output file.

**References**

Learn more about [theorems and proofs](#) and [custom blocks](#) in the **bookdown** book.

---

 gitbook

*The GitBook output format*


---

**Description**

This output format function ported a style provided by GitBook (<https://www.gitbook.com>) for R Markdown.

**Usage**

```
gitbook(
  fig_caption = TRUE,
  number_sections = TRUE,
  self_contained = FALSE,
  lib_dir = "libs",
  pandoc_args = NULL,
  ...,
  template = "default",
  split_by = c("chapter", "chapter+number", "section", "section+number", "rmd", "none"),
  split_bib = TRUE,
  config = list(),
  table_css = TRUE
)
```

**Arguments**

fig_caption, number_sections, self_contained, lib_dir, pandoc_args	... Arguments to be passed to rmarkdown::html_document() (... not including toc, and theme).
...	Other arguments to be passed to base_format. For html_book() and tufte_html_book(), ... is passed to html_chapters().
template	Pandoc template to use for rendering. Pass "default" to use the bookdown default template; pass a path to use a custom template. The default template should be sufficient for most use cases. In case you want to develop a custom template, we highly recommend to start from the default template: <a href="https://github.com/rstudio/bookdown/blob/master/inst/templates/gitbook.html">https://github.com/rstudio/bookdown/blob/master/inst/templates/gitbook.html</a> .
split_by	How to name the HTML output files from the book: rmd uses the base filenames of the input Rmd files to create the HTML filenames, e.g. generate 'chapter1.html' for 'chapter1.Rmd'; none means do not split the HTML file (the book will be a single HTML file); chapter means split the file by the first-level headers; section means the second-level headers. For chapter and section, the HTML filenames will be determined by the header ID's, e.g. the filename for the first chapter with a chapter title # Introduction will be 'introduction.html'; for chapter+number and section+number, the chapter/section numbers will be prepended to the HTML filenames, e.g. '1-introduction.html' and '2-1-literature.html'.
split_bib	Whether to split the bibliography onto separate pages where the citations are actually used.
config	A list of configuration options for the gitbook style, such as the font/theme settings.
table_css	TRUE to load gitbook's default CSS for tables. Choose FALSE to unload and use customized CSS (for example, bootstrap) via the css option. Default is TRUE.

---

html_chapters	<i>Build book chapters into separate HTML files</i>
---------------	---

---

**Description**

Split the HTML output into chapters while updating relative links (e.g. links in TOC, footnotes, citations, figure/table cross-references, and so on). Functions `html_book()` and `tufte_html_book()` are simple wrapper functions of `html_chapter()` using a specific base output format.

**Usage**

```
html_chapters(
  toc = TRUE,
  number_sections = TRUE,
  fig_caption = TRUE,
  lib_dir = "libs",
  template = bookdown_file("templates/default.html"),
```

```

pandoc_args = NULL,
...,
base_format = rmarkdown::html_document,
split_bib = TRUE,
page_builder = build_chapter,
split_by = c("section+number", "section", "chapter+number", "chapter", "rmd", "none")
)

html_book(...)

tufte_html_book(...)

```

### Arguments

<code>toc</code> , <code>number_sections</code> , <code>fig_caption</code> , <code>lib_dir</code> , <code>template</code> , <code>pandoc_args</code>	See <code>rmarkdown::html_document</code> , <code>tufte::tufte_html</code> , or the documentation of the <code>base_format</code> function.
<code>...</code>	Other arguments to be passed to <code>base_format</code> . For <code>html_book()</code> and <code>tufte_html_book()</code> , <code>...</code> is passed to <code>html_chapters()</code> .
<code>base_format</code>	An output format function to be used as the base format.
<code>split_bib</code>	Whether to split the bibliography onto separate pages where the citations are actually used.
<code>page_builder</code>	A function to combine different parts of a chapter into a page (an HTML character vector). See <code>build_chapter</code> for the specification of this function.
<code>split_by</code>	How to name the HTML output files from the book: <code>rmd</code> uses the base filenames of the input Rmd files to create the HTML filenames, e.g. generate <code>'chapter1.html'</code> for <code>'chapter1.Rmd'</code> ; <code>none</code> means do not split the HTML file (the book will be a single HTML file); <code>chapter</code> means split the file by the first-level headers; <code>section</code> means the second-level headers. For <code>chapter</code> and <code>section</code> , the HTML filenames will be determined by the header ID's, e.g. the filename for the first chapter with a chapter title <code># Introduction</code> will be <code>'introduction.html'</code> ; for <code>chapter+number</code> and <code>section+number</code> , the chapter/section numbers will be prepended to the HTML filenames, e.g. <code>'1-introduction.html'</code> and <code>'2-1-literature.html'</code> .

### Value

An R Markdown output format object to be passed to `bookdown::render_book()`.

### Note

These functions are expected to be used in conjunction with `render_book()`. It is almost meaningless if they are used with `rmarkdown::render()`. Functions like `html_document2` are designed to work with the latter.

If you want to use a different template, the template must contain three pairs of HTML comments: `<!--bookdown:title:start-->` and `<!--bookdown:title:end-->` to mark the title section of the book (this section will be placed only on the first page of the rendered book);

'<!--bookdown:toc:start-->' and '<!--bookdown:toc:end-->' to mark the table of contents section (it will be placed on all chapter pages); '<!--bookdown:body:start-->' and '<!--bookdown:body:end-->' to mark the HTML body of the book (the HTML body will be split into separate pages for chapters). You may open the default HTML template (`bookdown::bookdown_file('templates/default.html')`) to see where these comments were inserted.

---

html_document2	<i>Output formats that allow numbering and cross-referencing figures/tables/equations</i>
----------------	---

---

### Description

These are simple wrappers of the output format functions like `rmarkdown::html_document()`, and they added the capability of numbering figures/tables/equations/theorems and cross-referencing them. See 'References' for the syntax. Note you can also cross-reference sections by their ID's using the same syntax when sections are numbered. In case you want to enable cross reference in other formats, use `markdown_document2` with `base_format` argument.

### Usage

```
html_document2(
  ...,
  number_sections = TRUE,
  pandoc_args = NULL,
  base_format = rmarkdown::html_document
)

html_fragment2(..., number_sections = FALSE)

html_notebook2(..., number_sections = FALSE)

html_vignette2(..., number_sections = FALSE)

ioslides_presentation2(..., number_sections = FALSE)

slidy_presentation2(..., number_sections = FALSE)

tufte_html2(..., number_sections = FALSE)

pdf_document2(...)

beamer_presentation2(..., number_sections = FALSE)

tufte_handout2(...)

tufte_book2(...)
```

```

markdown_document2(
  number_sections = TRUE,
  fig_caption = TRUE,
  md_extensions = NULL,
  pandoc_args = NULL,
  ...,
  base_format = rmarkdown::md_document
)

context_document2(...)

github_document2(...)

odt_document2(...)

powerpoint_presentation2(...)

rtf_document2(...)

word_document2(...)

```

### Arguments

`...`, `fig_caption`, `md_extensions`, `pandoc_args`  
 Arguments to be passed to a specific output format function. For a function `foo2()`, its arguments are passed to `foo()`, e.g. `...` of `html_document2()` are passed to `rmarkdown::html_document()`.

`number_sections`  
 Whether to number section headers: if `TRUE`, figure/table numbers will be of the form `X.i`, where `X` is the current first-level section number, and `i` is an incremental number (the `i`-th figure/table); if `FALSE`, figures/tables will be numbered sequentially in the document from 1, 2, ..., and you cannot cross-reference section headers in this case.

`base_format`     An output format function to be used as the base format.

### Value

An R Markdown output format object to be passed to `rmarkdown::render()`.

### Note

These output formats are used to generate single output files, such as a single HTML output file (unlike `gitbook`, which generates multiple HTML output files by default).

The functions ‘`tufte_*()`’ are wrappers of functions in the **tufte** package.

### References

<https://bookdown.org/yihui/bookdown/>

pdf\_book

*Convert R Markdown to a PDF book*

## Description

Convert R Markdown files to PDF after resolving the special tokens of **bookdown** (e.g., the tokens for references and labels) to native LaTeX commands.

## Usage

```
pdf_book(
  toc = TRUE,
  number_sections = TRUE,
  fig_caption = TRUE,
  pandoc_args = NULL,
  ...,
  base_format = rmarkdown::pdf_document,
  toc_unnumbered = TRUE,
  toc_appendix = FALSE,
  toc_bib = FALSE,
  quote_footer = NULL,
  highlight_bw = FALSE
)
```

## Arguments

toc, number_sections, fig_caption, pandoc_args	See rmarkdown:: <a href="#">pdf_document</a> , or the documentation of the base_format function.
...	Other arguments to be passed to base_format.
base_format	An output format function to be used as the base format.
toc_unnumbered	Whether to add unnumbered headers to the table of contents.
toc_appendix	Whether to add the appendix to the table of contents.
toc_bib	Whether to add the bibliography section to the table of contents.
quote_footer	If a character vector of length 2 and the quote footer starts with three dashes ('---'), quote_footer[1] will be prepended to the footer, and quote_footer[2] will be appended; if NULL, the quote footer will not be processed.
highlight_bw	Whether to convert colors for syntax highlighting to black-and-white (grayscale).

## Details

This function is based on rmarkdown::[pdf\\_document](#) (by default) with better default arguments. You can also change the default format to other LaTeX/PDF format functions using the base\_format argument.

The global R option `bookdown.post.latex` can be set to a function to post-process the LaTeX output. This function takes the character vector of the LaTeX output as its input argument, and should return a character vector to be written to the `.tex` output file. This gives you full power to post-process the LaTeX output.

### Note

This output format can only be used with `render_book()`.

---

publish\_book

*Publish a book to the web*

---

### Description

Publish a book to the web. Note that you should be sure to render all versions of the book before publishing, unless you have specified `render = TRUE`.

### Usage

```
publish_book(
  name = NULL,
  account = NULL,
  server = NULL,
  render = c("none", "local", "server")
)
```

### Arguments

name	Name of the book (this will be used in the URL path of the published book). Defaults to the <code>book_filename</code> in <code>_bookdown.yml</code> if not specified.
account	Account name to publish to. Will default to any previously published to account or any single account already associated with server.
server	Server to publish to (by default <code>beta.rstudioconnect.com</code> but any RStudio Connect server can be published to).
render	Rendering behavior for site: "none" to upload a static version of the current contents of the site directory; "local" to render the site locally then upload it; "server" to render the site on the server. Note that for "none" and "local" R scripts ( <code>.R</code> ) and markdown documents ( <code>.Rmd</code> and <code>.md</code> ) will not be uploaded to the server.

render\_book

*Render multiple R Markdown documents into a book***Description**

Render multiple R Markdown files under the current working directory into a book. It can be used in the RStudio IDE (specifically, the knit field in YAML). The `preview_chapter()` function is a wrapper of `render_book(preview = TRUE)`.

**Usage**

```
render_book(
  input,
  output_format = NULL,
  ...,
  clean = TRUE,
  envir = parent.frame(),
  clean_envir = !interactive(),
  output_dir = NULL,
  new_session = NA,
  preview = FALSE,
  config_file = "_bookdown.yml"
)

preview_chapter(..., envir = parent.frame())
```

**Arguments**

<code>input</code>	An input filename (or multiple filenames). If <code>preview = TRUE</code> , only files specified in this argument are rendered, otherwise all R Markdown files specified by the book are rendered.
<code>output_format, ..., clean, envir</code>	Arguments to be passed to <code>rmarkdown::render()</code> . For <code>preview_chapter()</code> , <code>...</code> is passed to <code>render_book()</code> . See <code>rmarkdown::render()</code> and <a href="#">the bookdown reference book</a> for details on how output formatting options are set from YAML or parameters supplied by the user when calling <code>render_book()</code> .
<code>clean_envir</code>	This argument has been deprecated and will be removed in future versions of <b>bookdown</b> .
<code>output_dir</code>	The output directory. If <code>NULL</code> , a field named <code>output_dir</code> in the configuration file <code>'_bookdown.yml'</code> will be used (possibly not specified, either, in which case a directory name <code>'_book'</code> will be used).
<code>new_session</code>	Whether to use new R sessions to compile individual Rmd files (if not provided, the value of the <code>new_session</code> option in <code>'_bookdown.yml'</code> is used; if this is also not provided, <code>new_session = FALSE</code> ).

preview	Whether to render and preview the input files specified by the input argument. Previewing a certain chapter may save compilation time as you actively work on this chapter, but the output may not be accurate (e.g. cross-references to other chapters will not work).
config_file	The book configuration file.

### Details

There are two ways to render a book from Rmd files. The default way (`new_session = FALSE`) is to merge Rmd files into a single file and render this file. You can also choose to render each individual Rmd file in a new R session (`new_session = TRUE`).

### Examples

```
# see https://bookdown.org/yihui/bookdown for the full documentation
if (file.exists("index.Rmd")) bookdown::render_book("index.Rmd")
## Not run:
# will use the default format defined in index.Rmd or _output.yml
bookdown::render_book("index.Rmd")
# will use the options for format defined in YAML metadata
bookdown::render_book("index.Rmd", "pdf_book")
# If you pass an output format object, it must have all the options set
bookdown::render_book("index.Rmd", bookdown::pdf_book(toc = FALSE))

## End(Not run)
```

---

serve_book	<i>Continuously preview the HTML output of a book using the <b>servr</b> package</i>
------------	--

---

### Description

When any files are modified or added to the book directory, the book will be automatically re-compiled, and the current HTML page in the browser will be refreshed. This function is based on `servr::httpw()` to continuously watch a directory.

### Usage

```
serve_book(
  dir = ".",
  output_dir = "_book",
  preview = TRUE,
  in_session = TRUE,
  quiet = FALSE,
  ...
)
```

**Arguments**

<code>dir</code>	The root directory of the book (containing the Rmd source files).
<code>output_dir</code>	The directory for output files; see <code>render_book()</code> .
<code>preview</code>	Whether to render the modified/added chapters only, or the whole book; see <code>render_book()</code> .
<code>in_session</code>	Whether to compile the book using the current R session, or always open a new R session to compile the book whenever changes occur in the book directory.
<code>quiet</code>	Whether to suppress output (e.g., the knitting progress) in the console.
<code>...</code>	Other arguments passed to <code>servr::http()</code> (not including the handler argument, which has been set internally).

**Details**

For `in_session = TRUE`, you will have access to all objects created in the book in the current R session: if you use a daemonized server (via the argument `daemon = TRUE`), you can check the objects at any time when the current R session is not busy; otherwise you will have to stop the server before you can check the objects. This can be useful when you need to interactively explore the R objects in the book. The downside of `in_session = TRUE` is that the output may be different with the book compiled from a fresh R session, because the state of the current R session may not be clean.

For `in_session = FALSE`, you do not have access to objects in the book from the current R session, but the output is more likely to be reproducible since everything is created from new R sessions. Since this function is only for previewing purposes, the cleanliness of the R session may not be a big concern. You may choose `in_session = TRUE` or `FALSE` depending on your specific applications. Eventually, you should run `render_book()` from a fresh R session to generate a reliable copy of the book output.

# Index

beamer\_presentation2 (html\_document2), 9  
bookdown\_site, 3  
build\_chapter, 8

calibre, 3  
clean\_book, 4  
context\_document2 (html\_document2), 9

epub\_book, 4

fence\_theorems, 5

gitbook, 6  
github\_document2 (html\_document2), 9

html\_book (html\_chapters), 7  
html\_chapters, 7  
html\_document, 7–9  
html\_document2, 8, 9  
html\_fragment2 (html\_document2), 9  
html\_notebook2 (html\_document2), 9  
html\_vignette2 (html\_document2), 9  
http, 14, 15

ioslides\_presentation2  
(html\_document2), 9

markdown\_document2 (html\_document2), 9

odt\_document2 (html\_document2), 9

pdf\_book, 11  
pdf\_document, 11  
pdf\_document2 (html\_document2), 9  
powerpoint\_presentation2  
(html\_document2), 9  
preview\_chapter (render\_book), 13  
publish\_book, 12

render, 10, 13  
render\_book, 8, 12, 13, 15

rtf\_document2 (html\_document2), 9

serve\_book, 14  
slidy\_presentation2 (html\_document2), 9

tufte\_book2 (html\_document2), 9  
tufte\_handout2 (html\_document2), 9  
tufte\_html, 8  
tufte\_html2 (html\_document2), 9  
tufte\_html\_book (html\_chapters), 7

word\_document2 (html\_document2), 9