Package 'markdown'

May 8, 2026

Type Package

Title Render Markdown with 'commonmark'

Version 2.0

Description Render Markdown to full and lightweight HTML/LaTeX documents with the 'commonmark' package. This package has been superseded by 'litedown'.

Depends R (>= 2.11.1)

Imports utils, xfun, litedown (>= 0.6)

Suggests knitr, rmarkdown (>= 2.18), yaml, RCurl

License MIT + file LICENSE

URL https://github.com/rstudio/markdown

BugReports https://github.com/rstudio/markdown/issues

RoxygenNote 7.3.2

Encoding UTF-8

NeedsCompilation no

Author Yihui Xie [aut, cre] (ORCID: <https://orcid.org/0000-0003-0645-5666>),
JJ Allaire [aut],
Jeffrey Horner [aut],
Henrik Bengtsson [ctb],
Jim Hester [ctb],
Yixuan Qiu [ctb],
Kohske Takahashi [ctb],
Adam November [ctb],
Nacho Caballero [ctb],
Jeroen Ooms [ctb],
Thomas Leeper [ctb],
Joe Cheng [ctb],
Andrzej Oles [ctb],
Posit Software, PBC [cph, fnd]

Maintainer Yihui Xie <xie@yihui.name>

Repository CRAN

Date/Publication 2025-03-23 19:30:03 UTC

Contents

`markdown-package` *Markdown rendering for R*

Description

Markdown is a plain-text formatting syntax that can be converted to XHTML or other formats. This package provides wrapper functions (mainly `mark()`) based on the **commonmark** package.

Author(s)

Maintainer: Yihui Xie <xie@yihui.name> (ORCID)

Authors:

- JJ Allaire
- Jeffrey Horner

Other contributors:

- Henrik Bengtsson [contributor]
- Jim Hester [contributor]
- Yixuan Qiu [contributor]
- Kohske Takahashi [contributor]
- Adam November [contributor]
- Nacho Caballero [contributor]
- Jeroen Ooms [contributor]
- Thomas Leeper [contributor]
- Joe Cheng [contributor]
- Andrzej Oles [contributor]
- Posit Software, PBC [copyright holder, funder]

See Also

Useful links:

- https://github.com/rstudio/markdown
- Report bugs at https://github.com/rstudio/markdown/issues

`html_format` *R Markdown output formats*

Description

Convenience functions for R Markdown v2 users.

Usage

```
html_format(
  meta = NULL,
  template = NULL,
  options = NULL,
  keep_md = FALSE,
  keep_tex = FALSE,
  latex_engine = "xelatex"
)

latex_format(
  meta = NULL,
  template = NULL,
  options = NULL,
  keep_md = FALSE,
  keep_tex = FALSE,
  latex_engine = "xelatex"
)
```

Arguments

`meta`, `template`, `options`	Arguments to be passed to `mark()`.
`keep_md`, `keep_tex`	Whether to keep the intermediate '`.md`' and '`.tex`' files generated from '`.Rmd`'.
`latex_engine`	The LaTeX engine to compile '`.tex`' to '`.pdf`'. This argument and `keep_tex` are for `latex_format()` only, and ignored in `html_format()`.

Details

We refer to this **markdown** package plus **knitr** as "R Markdown v1", and the **rmarkdown** package as "R Markdown v2". The former uses **commonmark** to convert Markdown, and the latter uses Pandoc. However, the latter also accept custom converting tools in addition to Pandoc. The output formats here provide the custom converting function `mark()` to **rmarkdown**, so that users can take advantage of `rmarkdown::render()` and the Knit button in RStudio. It is absolutely not necessary to rely on **rmarkdown**. The only point is convenience. If you do not use `rmarkdown::render()` or the Knit button, you can definitely just call `markdown::mark()` directly.

mark *Render Markdown to an output format*

Description

This is a wrapper function based on `litedown::mark()`. You should use `litedown::mark()` directly.

Usage

```
mark(
  file = NULL,
  output = NULL,
  text = NULL,
  format = c("html", "latex"),
  options = NULL,
  template = FALSE,
  meta = list()
)

mark_html(..., template = TRUE)

mark_latex(..., template = TRUE)
```

Arguments

`file`, `output`, `text`, `options`, `meta`	Passed to `litedown::mark()`.
`format`	Output format name.
`template`	Whether to use a built-in template, or path to a custom template.
`...`	Arguments to be passed to `mark()`.

markdown_options *Markdown rendering options*

Description

A wrapper function of `litedown::markdown_options()`.

Usage

```
markdown_options()
```

rpubsUpload *Upload an HTML file to RPubs*

Description

This function uploads an HTML file to rpubs.com. If the upload succeeds a list that includes an `id` and `continueUrl` is returned. A browser should be opened to the `continueUrl` to complete publishing of the document. If an error occurs then a diagnostic message is returned in the `error` element of the list.

Usage

```
rpubsUpload(
  title,
  htmlFile,
  id = NULL,
  properties = list(),
  method = getOption("rpubs.upload.method", "auto")
)
```

Arguments

`title`	The title of the document.
`htmlFile`	The path to the HTML file to upload.
`id`	If this upload is an update of an existing document then the id parameter should specify the document id to update. Note that the id is provided as an element of the list returned by successful calls to `rpubsUpload`.
`properties`	A named list containing additional document properties (RPubs doesn't currently expect any additional properties, this parameter is reserved for future use).
`method`	Method to be used for uploading. "internal" uses a plain http socket connection; "curl" uses the curl binary to do an https upload; "rcurl" uses the RCurl package to do an https upload; and "auto" uses the best available method searched for in the following order: "curl", "rcurl", and then "internal". The global default behavior can be configured by setting the `rpubs.upload.method` option (the default is "auto").

Value

A named list. If the upload was successful then the list contains a `id` element that can be used to subsequently update the document as well as a `continueUrl` element that provides a URL that a browser should be opened to in order to complete publishing of the document. If the upload fails then the list contains an `error` element which contains an explanation of the error that occurred.

Examples

```
## Not run:
# upload a document
result <- rpubsUpload("My document title", "Document.html")
if (!is.null(result$continueUrl))
    browseURL(result$continueUrl) else stop(result$error)

# update the same document with a new title
updateResult <- rpubsUpload("My updated title", "Document.html", result$id)

## End(Not run)
```

`smartypants` *Convert some ASCII strings to HTML entities*

Description

Transform ASCII strings `(c)` (copyright), `(r)` (registered trademark), `(tm)` (trademark), and fractions `n/m` into *smart* typographic HTML entities.

Usage

```
smartypants(text)
```

Arguments

`text`	A character vector of the Markdown text.

Value

A character vector of the transformed text.

Examples

```
cat(smartypants("1/2 (c)\n"))
```

Index