

# Package ‘commonmark’

December 1, 2018

**Type** Package

**Title** High Performance CommonMark and Github Markdown Rendering in R

**Version** 1.7

**URL** <http://github.com/jeroen/commonmark> (devel)

<https://github.github.com/gfm/> (spec)

**BugReports** <http://github.com/jeroen/commonmark/issues>

**Description** The CommonMark specification defines a rationalized version of markdown syntax. This package uses the 'cmark' reference implementation for converting markdown text into various formats including html, latex and groff man. In addition it exposes the markdown parse tree in xml format. Also includes opt-in support for GFM extensions including tables, autolinks, and strikethrough text.

**License** BSD\_2\_clause + file LICENSE

**Suggests** curl, testthat, xml2

**RoxygenNote** 6.0.1

**NeedsCompilation** yes

**Author** Jeroen Ooms [aut, cre],  
John MacFarlane [cph] (Author of cmark)

**Maintainer** Jeroen Ooms <jeroen@berkeley.edu>

**Repository** CRAN

**Date/Publication** 2018-12-01 12:30:03 UTC

## R topics documented:

commonmark . . . . .	2
extensions . . . . .	3

<b>Index</b>	<b>4</b>
--------------	----------

---

 commonmark

*Parse and render markdown text*


---

## Description

Converts markdown text to several formats using John MacFarlane's [cmark](#) reference implementation. Supported output formats include html, latex, groff man, and normalized "commonmark" markdown. In addition the markdown parse tree can be returned in xml format.

## Usage

```
markdown_html(text, hardbreaks = FALSE, smart = FALSE, normalize = FALSE,
  sourcepos = FALSE, extensions = FALSE)
```

```
markdown_xml(text, hardbreaks = FALSE, smart = FALSE, normalize = FALSE,
  sourcepos = FALSE, extensions = FALSE)
```

```
markdown_man(text, hardbreaks = FALSE, smart = FALSE, normalize = FALSE,
  width = 0, extensions = FALSE)
```

```
markdown_commonmark(text, hardbreaks = FALSE, smart = FALSE,
  normalize = FALSE, width = 0, extensions = FALSE)
```

```
markdown_text(text, hardbreaks = FALSE, smart = FALSE, normalize = FALSE,
  width = 0, extensions = FALSE)
```

```
markdown_latex(text, hardbreaks = FALSE, smart = FALSE, normalize = FALSE,
  width = 0, extensions = FALSE)
```

## Arguments

text	Markdown text
hardbreaks	Treat newlines as hard line breaks. If this option is specified, hard wrapping is disabled regardless of the value given with width.
smart	Use smart punctuation. See details.
normalize	Consolidate adjacent text nodes.
sourcepos	Include source position attribute in output.
extensions	Enables Github extensions. Can be TRUE (all) FALSE (none) or a character vector with a subset of available <a href="#">extensions</a> .
width	Specify wrap width (default 0 = nowrap).

## Details

Support for extensions (including tables and autolink) is provided via the Github [fork](#) of cmark. For now these are opt-in and have to be enabled with the extensions parameter. See also the manual page on [extensions](#).

When smart punctuation is enabled, straight double and single quotes will be rendered as curly quotes, depending on their position. Moreover -- will be rendered as – (en-dash), --- will be rendered as — (em-dash), and ... will be rendered as … (ellipses).

### Examples

```
md <- readLines("https://raw.githubusercontent.com/yihui/knitr/master/NEWS.md")
html <- markdown_html(md)
xml <- markdown_xml(md)
man <- markdown_man(md)
tex <- markdown_latex(md)
cm <- markdown_commonmark(md)
text <- markdown_text(md)
```

---

extensions

*Github CommonMark Extensions*

---

### Description

The Github fork of cmark supports several markdown extensions which provide features which are not (yet) in the official commonmark spec.

### Usage

```
list_extensions()
```

### Details

Currently the following extensions are supported:

- **table** support rendering of tables
- **strikethrough** via ~sometext~ syntax
- **autolink** automatically turn URLs into hyperlinks
- **tagfilter** blacklist html tags: title textarea style xmp iframe noembed noframes script plaintext.

See also the spec for [GitHub Flavored Markdown](#).

### Examples

```
print(list_extensions())
```

# Index

`commonmark`, [2](#)

`extensions`, [2](#), [3](#)

`list_extensions` (`extensions`), [3](#)

`markdown` (`commonmark`), [2](#)

`markdown_commonmark` (`commonmark`), [2](#)

`markdown_html` (`commonmark`), [2](#)

`markdown_latex` (`commonmark`), [2](#)

`markdown_man` (`commonmark`), [2](#)

`markdown_text` (`commonmark`), [2](#)

`markdown_xml` (`commonmark`), [2](#)