

# Package ‘md4r’

May 7, 2026

**Title** Markdown Parser Implemented using the 'MD4C' Library

**Version** 0.5.3.0

**Description** Provides an R wrapper for the 'MD4C' (Markdown for 'C') library. Functions exist for parsing markdown ('CommonMark' compliant) along with support for other common markdown extensions (e.g. 'GitHub' flavored markdown, 'LaTeX' equation support, etc.). The package also provides a number of higher level functions for exploring and manipulating markdown abstract syntax trees as well as translating and displaying the documents.

**License** MIT + file LICENSE

**URL** <https://rundel.github.io/md4r/>, <https://github.com/rundel/md4r>

**BugReports** <https://github.com/rundel/md4r/issues>

**Imports** checkmate, cli, glue, purrr, Rcpp, stringr, textutils, tibble

**Suggests** diffmatchpatch, rlang, styler, testthat (>= 3.0.0), withr

**LinkingTo** Rcpp

**Config/testthat/edition** 3

**Copyright** John MacFarlane, RStudio, PBC; Martin Mitáš, Colin Rundel

**Encoding** UTF-8

**Config/roxygen2/version** 8.0.0

**NeedsCompilation** yes

**Author** Colin Rundel [aut, cre],  
Martin Mitáš [cph] (md4c author: md4c.c, md4c.h, specs/md4c/),  
RStudio, PBC [cph] (httpuv\_url\_tools.cpp),  
John MacFarlane [cph] (specs/gfm/spec.txt, specs/md4c/spec.txt)

**Maintainer** Colin Rundel <rundel@gmail.com>

**Depends** R (>= 4.1.0)

**Repository** CRAN

**Date/Publication** 2026-05-07 17:33:02 UTC

## Contents

|                          |   |
|--------------------------|---|
| flags . . . . .          | 2 |
| md_block . . . . .       | 3 |
| md_node_modify . . . . . | 4 |
| md_span . . . . .        | 5 |
| md_text . . . . .        | 6 |
| parse_md . . . . .       | 7 |
| to_html . . . . .        | 8 |
| to_md . . . . .          | 9 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>10</b> |
|--------------|-----------|

---

|       |                              |
|-------|------------------------------|
| flags | <i>Markdown parser flags</i> |
|-------|------------------------------|

---

### Description

The `md4c` library supports a number of markdown variants / options. The parsing of these is controlled by flags passed to the parser. The following functions provide commonly used utilities for these flags.

### Usage

```
flags_available()
```

```
flags_describe()
```

```
flags_used(md)
```

### Arguments

md                    Markdown ast object

### Value

`flags_available()` returns a character vector of available flags accepted by `parse_md()`.

`flags_describe()` returns a tibble with columns `flag` and `description` describing each flag.

`flags_used()` returns a character vector of flags used in a parsed markdown document.

### Examples

```
flags_available()
```

```
flags_describe()
```

```
md_file = system.file("examples/commonmark.md", package = "md4r")
```

```
md = parse_md(md_file)
```

```
flags_used(md)
```

---

|          |                                       |
|----------|---------------------------------------|
| md_block | <i>Tools for creating block nodes</i> |
|----------|---------------------------------------|

---

### Description

These functions are used to create block nodes. Blocks nodes are used to represent block level elements in markdown.

### Usage

```
md_block_doc(..., flags = "MD_DIALECT_COMMONMARK")
md_block_quote(...)
md_block_html(...)
md_block_p(...)
md_block_hr(...)
md_block_code(..., info = "", lang = "", fence_char = "`")
md_block_ul(..., tight = 1, mark = "*")
md_block_ol(..., tight = 1L, start = 1, mark_delimiter = ".")
md_block_li(..., is_task = 0, task_mark = " ")
md_block_h(..., level)
md_block_table(..., col_count, body_row_count, head_row_count = 1)
md_block_thead(...)
md_block_tbody(...)
md_block_tr(...)
md_block_th(..., align = c("default", "left", "center", "right"))
md_block_td(..., align = c("default", "left", "center", "right"))
```

### Arguments

|       |  |
|-------|--|
| ...   | Child nodes that will be contained within the block - all must inherit from md_node. |
| flags | Used by md_block_doc to specify the document parser flag(s).                         |

|                |  |
|----------------|--|
| info           | Used by md_block_code nodes to specify the code block info string.   |
| lang           | Used by md_block_code nodes to specify the code block language.  |
| fence_char     | Used by md_block_code nodes to specify the code block fence character.                                       |
| tight          | Used by md_block_ul or md_block_ol nodes to specify whether the list is tight or loose.                      |
| mark           | Used by md_block_ul nodes to specify the list marker.  |
| start          | Used by md_block_ol nodes to specify the start number of the list.   |
| mark_delimiter | Used by md_block_ol nodes to specify the delimiter between the value and text. Only "." and ")" are allowed. |
| is_task        | Used by md_block_li nodes to specify whether the item is a task.   |
| task_mark      | Used by md_block_li task nodes to specify the task mark character. Only " ", "x", and "X" are allowed.       |
| level          | Used by md_block_h nodes to specify the heading level.   |
| col_count      | Used by md_block_table nodes to specify the number of columns.   |
| body_row_count | Used by md_block_table nodes to specify the number of body rows.   |
| head_row_count | Used by md_block_table nodes to specify the number of header rows. Should only be 1.                         |
| align          | Used by md_block_td or md_block_th nodes to specify the alignment of the table contents.                     |

### Value

Returns a list with a class of specified type along with md\_span and md\_node.

### See Also

[md\\_span](#), [md\\_text](#)

---

md\_node\_modify      *Tools for modifying md\_node objects*

---

### Description

Overrides the \$ operator so the content and attributes of md\_node objects can be read and updated by name. Use node\$content to access or replace the children of a block or span node (or the text of a text node), and node\$<attr> to read or update an existing node attribute (e.g. level, href, mark).

Only attributes already present on the node may be replaced, and the new value is validated against the same rules used by the corresponding md\_block\_\* / md\_span\_\* constructor. The class and names attributes cannot be set this way.

**Usage**

```
## S3 method for class 'md_node'
x$name

## S3 replacement method for class 'md_node'
x$name <- value

## S3 replacement method for class 'md_text'
x$name <- value
```

**Arguments**

|       |   |
|-------|---|
| x     | An object that inherits from md_node.   |
| name  | A character string giving the attribute to access, or "content" to access the node's children / text content. |
| value | Replacement value for the content or attribute.   |

**Value**

A modified md\_node object.

**Examples**

```
md = md_text_normal("Hello World")
md$content = "Nice to meet you"
md

md = md_block_ul(
  md_block_li(md_text_normal("bullet 1")),
  md_block_li(md_text_normal("bullet 2"))
)
md$mark = "-"
md$content[[1]]$content[[1]]$content = "first"
to_md(md) |> cat(sep = "\n")
```

**Description**

These functions are used to create span nodes. Span nodes are used to represent inline elements in markdown and include things like links, images, code, emphasized text, strong text, and more.

**Usage**

```
md_span_em(...)
md_span_strong(...)
md_span_a(..., href, title)
md_span_img(..., src, title)
md_span_code(...)
md_span_del(...)
md_span_latexmath(...)
md_span_latexmath_display(...)
md_span_wikilink(..., target)
md_span_u(...)
```

**Arguments**

|        |   |
|--------|---|
| ...    | Child nodes that will be contained within the span - all must inherit from <code>md_node</code> . |
| href   | Used by <code>md_span_a</code> nodes to provide the href attribute.                               |
| title  | Used by <code>md_span_a</code> or <code>md_span_img</code> nodes to provide a title attribute.    |
| src    | Used by <code>md_span_img</code> nodes to provide the src attribute.                              |
| target | Used by <code>md_span_wikilink</code> nodes to provide the target attribute.                      |

**Value**

Returns a list with a class of specified type along with `md_span` and `md_node`.

**See Also**

[md\\_block](#), [md\\_text](#)

---

md\_text

*Tools for creating text nodes*

---

**Description**

These functions are used to create text nodes. Text nodes are used to represent textual elements in markdown.

**Usage**

```
md_text_normal(x)
md_text_nullchar()
md_text_break()
md_text_softbreak()
md_text_entity(x)
md_text_code(x)
md_text_html(x)
md_text_latexmath(x)
```

**Arguments**

x                   Text content of the node.

**Value**

Returns a character vector with a class of specified type along with `md_text` and `md_node`.

**See Also**

[md\\_block](#), [md\\_span](#)

---

parse\_md

*Parse markdown*

---

**Description**

Parse either a literal markdown string or a markdown file given a path. Different dialects and features are supported via the `flags` argument. See [flags\\_describe\(\)](#) for possible flags and their usage. `parse_md()` defaults parsing using the commonmark spec while `parse_gfm()` uses the GitHub flavored markdown spec.

**Usage**

```
parse_md(md, flags = "MD_DIALECT_COMMONMARK")
parse_gfm(md, flags = "MD_DIALECT_GITHUB")
```

**Arguments**

`md` Character. Either literal string of markdown or a path to a markdown file.  
`flags` Character vector. Dialect flags used by the parser.

**Value**

Both functions return a markdown ast, a list with the `md_block_doc` class.

**Examples**

```
parse_md(system.file("examples/commonmark.md", package = "md4r"))  
parse_gfm(system.file("examples/github.md", package = "md4r"))
```

---

|                      |                        |
|----------------------|------------------------|
| <code>to_html</code> | <i>Convert to html</i> |
|----------------------|------------------------|

---

**Description**

Converts a markdown object (full ast or node) to HTML.

**Usage**

```
to_html(md, ...)
```

**Arguments**

`md` Markdown object  
`...` Unused, for extensibility.

**Value**

Returns a character vector of HTML lines representing the markdown object.

**Examples**

```
md_file = system.file("examples/commonmark.md", package = "md4r")  
md = parse_md(md_file)  
cat(to_html(md), sep="\n")
```

---

|       |                            |
|-------|----------------------------|
| to_md | <i>Convert to markdown</i> |
|-------|----------------------------|

---

**Description**

Converts a markdown object (full ast or node) to markdown text.

**Usage**

```
to_md(md, ...)
```

**Arguments**

|     |                            |
|-----|----------------------------|
| md  | Markdown object            |
| ... | Unused, for extensibility. |

**Value**

Returns a character vector of markdown lines representing the markdown object.

**Examples**

```
md_file = system.file("examples/commonmark.md", package = "md4r")
md = parse_md(md_file)
cat(to_md(md), sep="\n")
```

# Index

`$.md_node (md_node_modify)`, 4  
`$<- .md_node (md_node_modify)`, 4  
`$<- .md_text (md_node_modify)`, 4

`flags`, 2  
`flags_available (flags)`, 2  
`flags_describe (flags)`, 2  
`flags_describe()`, 7  
`flags_used (flags)`, 2

`md_block`, 3, 6, 7  
`md_block_code (md_block)`, 3  
`md_block_doc (md_block)`, 3  
`md_block_h (md_block)`, 3  
`md_block_hr (md_block)`, 3  
`md_block_html (md_block)`, 3  
`md_block_li (md_block)`, 3  
`md_block_ol (md_block)`, 3  
`md_block_p (md_block)`, 3  
`md_block_quote (md_block)`, 3  
`md_block_table (md_block)`, 3  
`md_block_tbody (md_block)`, 3  
`md_block_td (md_block)`, 3  
`md_block_th (md_block)`, 3  
`md_block_thead (md_block)`, 3  
`md_block_tr (md_block)`, 3  
`md_block_ul (md_block)`, 3  
`md_node_modify`, 4  
`md_span`, 4, 5, 7  
`md_span_a (md_span)`, 5  
`md_span_code (md_span)`, 5  
`md_span_del (md_span)`, 5  
`md_span_em (md_span)`, 5  
`md_span_img (md_span)`, 5  
`md_span_latexmath (md_span)`, 5  
`md_span_latexmath_display (md_span)`, 5  
`md_span_strong (md_span)`, 5  
`md_span_u (md_span)`, 5  
`md_span_wikilink (md_span)`, 5  
`md_text`, 4, 6, 6  
`md_text_break (md_text)`, 6  
`md_text_code (md_text)`, 6  
`md_text_entity (md_text)`, 6  
`md_text_html (md_text)`, 6  
`md_text_latexmath (md_text)`, 6  
`md_text_normal (md_text)`, 6  
`md_text_nullchar (md_text)`, 6  
`md_text_softbreak (md_text)`, 6

`parse_gfm (parse_md)`, 7  
`parse_md`, 7

`to_html`, 8  
`to_md`, 9