

Document: **Markdown Test Document**

Author: **bsp**

Date: **06-Oct-2018**

1 Introduction

`md` is an extended re-implementation of the [markdown¹](#) utility script, originally released in 2004 by John Gruber.

`md` implements most of the markdown syntax, with some notable changes / omissions:

- Four spaces at the beginning of a line do **not** produce a code block
- A newline in a text paragraph inserts a line break
 - Similar to [asciidoc²](#), for example
- Inline HTML is not supported
 - I think this defeats the purpose of an output-format agnostic markup language.
- Lists may not contain block quotes
 - but they can contain buffer refs (which can contain arbitrary markdown)

`md` also adds a couple of **extensions**:

- Support for [pandoc³](#) style document headers (% title, % author(s), % date)
- Support for fenced code blocks (`~~~`)
- Nested lists
 - Sub-lists are started when their indent level is higher than the previous' list indentation
 - Consequently, the minimum indent to start a sub-list is one char, not three like in [pandoc³](#).
- Ordered list sequence start and explicit list item numbers
 - Ordered lists can start at an offset, and list item sequence numbers can be skipped
- Tables
 - Configurable table cell alignment
 - Configurable table cell widths
 - Multi-line table cells
- Variables
 - Preprocessor variables can be set in the markdown source or on the command line
 - Variable contents can be declared *inline* or be read from external files
- Post processor includes (raw HTML)
- Buffer redirection
 - The HTML output of sections of the markdown document can be redirected to named memory buffers
 - Memory buffers can be inserted at any point in the document
 - This makes it possible to create complex document structures, e.g. tables within tables
- Image scaling
 - Override width, height, or both
 - Scaled images will link to the original-size image
- Sub-titled paragraphs
 - Can be used to add a sub-title to e.g. an image or a table

- Redefinable link references
 - Link targets can be redefined
 - Previous (forward) references will be resolved as soon as a link reference definition becomes available
- Admonition paragraphs (`NOTE:`, `IMPORTANT:`, `TIP:`, `CAUTION:`, `WARNING:`)
 - Similar to [asciidoc²](#), for example
 - Single or multi-line (multi-paragraph)
 - Small or large (e.g. `note:`, `NOTE:`, ...)
- Footnotes
 - Local footnotes tables can be inserted right after the footnote definitions
 - Global footnotes typically occur at the end of the document but it's possible to split these into multiple tables
- Template mode
 - Load (complex) HTML template and insert buffer contents.
 - Document body will be stored in `body` buffer
 - Table of contents will be stored in `toc` buffer
 - Global footnotes will be stored in `footnote` buffer
 - Additional content can be stored in user defined buffers
- Custom CSS classes
 - Each paragraph can be assigned an additional, custom CSS class
- Special char entities (©, ®, ←, ⇐, ⇒, →, ..)
- Multi-column print layout (`+++col`, `+++/col`)
- Citations
- Single-file, inline-data mode (`-il`, `-inline-data`)
 - Inline additional content (CSS, images) via data URIs
 - The resulting self-contained document can be archived or sent via email

2 Table of Contents

- 1 [Introduction](#)
- 2 [Table of Contents](#)
- 3 [Command line options](#)
 - 3.1 [Examples](#)
- 4 [Plain text paragraphs](#)
- 5 [Multi-column layout \(print\)](#)
- 6 [Text styles](#)
 - 6.1 [Text style table](#)
- 7 [Escaped chars](#)
- 8 [Forced line breaks](#)
 - 8.1 [Empty line with two spaces \(
\)](#)
- 9 [Forced page breaks \(print\)](#)
 - 9.1 [Linefeed paragraph via backslash](#)
- 10 [Special char entities](#)
- 11 [Variables](#)
 - 11.1 [Set variable to inlined content](#)
 - 11.2 [Set variable to file content](#)
 - 11.3 [Externally defined vars](#)
 - 11.4 [Built-in special variables](#)
- 12 [Include markdown](#)
 - 12.1 [Example](#)
 - 12.2 [Parametrization](#)
- 13 [Include HTML](#)
 - 13.1 [Example](#)
- 14 [Buffer redirection](#)
 - 14.1 [Example](#)
 - 14.2 [Special buffer names](#)
- 15 [Pre-formatted text \(code block\)](#)
 - 15.1 [Indented code blocks](#)
- 16 [Block quotes](#)
 - 16.1 [Citations](#)
 - 16.1.1 [Nested citations](#)

17 Admonition paragraphs

17.1 NOTE:

17.2 IMPORTANT:

17.3 TIP:

17.4 CAUTION:

17.5 WARNING:

17.6 TODO:

17.7 Multi-paragraph admonitions

17.8 Admonition icon resources

17.9 Alternative icon resource path

17.10 Inlined icon resources

18 Header 1 <test>

18.1 Header 2

18.1.1 Header 3

18.1.1.1 Header 4

18.1.1.1.1 Header 5

18.1.1.1.1.1 Header 6

18.1.1.1.1.1.1 Header 7

18.1.1.1.1.1.1.1 Header 8

18.1.1.1.1.1.1.1.1 Header 9

19 Header 1 via =====

19.1 Header 2 via ---

20 Horizontal rules

20.1 Dash

20.2 Asterisk

20.3 Asterisk (book)

20.4 Pagebreak (print)

21 Lists

21.1 Unordered, nested lists

21.2 Ordered, nested lists

21.2.1 Start offset

21.2.2 Skip

21.2.3 Auto-enumeration

21.3 List sub-titles

22 [Links](#)

22.1 [Direct Links](#)

22.2 [Direct Links with description](#)

22.3 [Document internal links](#)

22.4 [Automatic Links](#)

22.5 [Link refs](#)

22.5.1 [Links \(redefined refs\)](#)

23 [Footnotes](#)

23.1 [Local footnotes](#)

23.2 [Global footnotes](#)

23.3 [Footnote-only links](#)

24 [Images](#)

24.1 [Direct image links](#)

24.2 [Scaled image links](#)

24.3 [Referenced images](#)

24.4 [Sub-titled images](#)

24.5 [Left-aligned images](#)

25 [Tables](#)

25.1 [Simple table](#)

25.2 [Simple table with cell alignment](#)

25.3 [Cell alignment in --- style table](#)

25.4 [Tables with different text styles, links, and images](#)

25.5 [Multi-line tables](#)

25.6 [Column widths](#)

25.7 [Headerless tables](#)

25.8 [Complex tables \(1\)](#)

25.9 [Complex tables \(2\)](#)

26 [CSS classes](#)

26.1 [Basic text styles and paragraphs](#)

26.2 [Blockquotes and citations](#)

26.3 [Images](#)

26.4 [Lists](#)

26.5 [Horizontal rules and printed page breaks](#)

26.6 [Subtitled paragraphs](#)

26.7 [Tables](#)

26.8 [Footnotes](#)

26.9 [Table of contents](#)

26.10 [Admonitions](#)

26.11 [Class name prefixes](#)

26.12 [Custom CSS classes](#)

27 [Creating a print version \(PDF\) of the document](#)

27.1 [Pandoc / LaTeX / wkhtmltopdf](#)

27.2 [Microsoft Print to PDF \(Windows 10\)](#)

27.3 [CutePDF \(Windows 98/ME/2000/XP/2003/Vista/7/2012/8/8.1/10\)](#)

27.4 [Prince](#)

3 Command line options

`md` supports the following command line options:

<code>-o outputfile</code>	Set output pathname. The default output name is derived from the input file (replace suffix by <code>.html</code>)
<code>-css cssurl</code>	Set CSS URL. The default CSS name is derived from the input file (replace suffix by <code>.css</code>)
<code>-Dvar=content</code>	Set variable (see Variables)
<code>-Dvar='<file>'</code>	Set variable to content of <i>file</i>
<code>-Bbuf=content</code>	Set buffer (see Buffers)
<code>-Bbuf='<file>'</code>	Set buffer to content of <i>file</i>
<code>-p</code> <code>--partial</code>	Enable partial document generation (skip <code><html></code> and <code><body></code>)
<code>-cp</code> <code>class_prefix</code>	Set CSS class prefix
<code>-ip</code> <code>icon_prefix</code>	Set icon resource prefix (default is <code>icons/</code>)
<code>-x</code> <code>template_file</code>	Use HTML template file. The generated document body will be placed in buffer <code>body</code> and can be inserted at any point in the template via <code>\$(buf!body)</code> .
	 This automatically enables the <i>partial</i> processing mode.
<code>-ae</code> <code>-ascii-</code> <code>entities</code>	Use plain ASCII for special character entities (unicode may confuse html-to-pdf tools)
<code>-ne</code> <code>-no-</code> <code>enumeration</code>	Disable header enumeration
<code>-eo</code> <code>-enumeration-</code> <code>offset nr</code>	Set header enumeration offset (1..n)
<code>-il</code> <code>-inline-data</code>	Inline external files via data URIs and create self-contained document.
<code>-h</code> <code>-help</code>	Show usage

3.1 Examples

```
$ tks app:md myfile.md
```

Example 1

Markdown input will be read from `myfile.md` and HTML output will be written to `myfile.html`, using the `myfile.css` stylesheet.

```
$ tks app:md -o myoutput.html -css mystylesheet.css input.md
```

Example 2

Markdown input will be read from `myfile.md` and HTML output will be written to `myoutput.html`, using the `mystylesheet.css` stylesheet.

```
$ tks app:md -x mytemplate.html myfile.md
```

Example 3

Markdown input will be read from `myfile.md` and HTML output will be written to `myfile.html`, using the `mytemplate.html` template file and the `myfile.css` stylesheet.

4 Plain text paragraphs

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.



Paragraphs are separated by empty lines.

A line break is inserted when a non-blank line is followed by another non-blank line.

5 Multi-column layout (print)

The `+++col` and `+++/col` char sequences are used to enable / disable the multi-column layout mode.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Lorem ipsum dolor sit amet, consetetur

sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

- Lorem ipsum dolor sit amet
- Consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.



The multi-column layout will only appear in the print version of the document.

6 Text styles

Some *italic text*^{superscript text}

Plain text *Some emphasized text* more plain text more emphasized text^{superscript text} remaining plain text.

Some _{subscript} text. **strong text** or **strong text**. ~~strikethrough~~, `fixed`, `more fixed text`

A new paragraph with more regular text.

6.1 Text style table

Example	Output
Emphasis	<i>Emphasis</i>
Emphasis2	<u>Emphasis2</u>
Strong	Strong
__Strong__	<u>Strong2</u>
~~Strike-through~~	Strike-through
`Fixed`	<code>Fixed</code>
``Fixed``	<code>Fixed2</code>
~Subscript~	_{Subscript}
^Superscript^	^{Superscript}

Text styles

7 Escaped chars

Escaped control chars: \ backslash, ` backtick, * asterisk, _ underscore, {} curly braces, [] square braces, () parenthesis, # hash, + plus, - minus, . dot, ! exclamation mark, ~ tilde, | pipe, < less than, > greater than.

Unescaped control chars like * asterisk or _ underscore. Freestanding chars like * will be treated as plain text.

Some escaped * asterisk. **strong text with escaped asterisk*** subscript with escaped ~ tilde.

8 Forced line breaks

8.1 Empty line with two spaces (
)

Forced line breaks are inserted when an empty line starts with two spaces:

Line 1

Line 5.

9 Forced page breaks (print)

A page break in the print version of the document can be inserted via

+++



The default CSS automatically inserts a page break before each 1st level header.



Page breaks will only appear in the print version of the document.

9.1 Linefeed paragraph via backslash

Under certain circumstances, for example when using custom CSS classes for advanced layouts, it may be necessary to force the end of the current paragraph.

A line that only consists of the `\` character will insert a linefeed div (CSS class `.lf`):

line 1
\
\
\
line 2

⇒

line 1

line 2

10 Special char entities

♥ hello, world 😊

Code	Output
(C)	©
(R)	®
(TM)	™
--	—
...	…
->	→
\Rightarrow	⇒
<-	←
<=	⇐
<=>	⇌
< !=>	⇏

Special char entities

Code	Output
% :) %	😊
% ;) %	😏
% : %	😐
% ; (%	😞
% : D %	😄
% : P %	😜
% : O %	😮
% > (%	😡
% : ' (%	😓
% 8) %	😬
% .) %	😛

Smileys

Code	Output
% < 3 %	♥
% * %	*

Other entities

11 Variables

`md`'s pre-processor can be used to set variables, and insert them at any point in the document.

Variables are set via the `$(var:<varname>=<value>)` syntax.

Variable contents are inserted via the `$(var:<varname>)` syntax.

11.1 Set variable to inlined content

Set test to 42:

```
$(var:test=42)
```

Variable `test` now contains >42<.

```
Set variable `test` to 3.1415 in one line:
$(var:test=3.1415)>$(var:test)<
```

Set variable `test` to 3.1415 in one line: >3.1415<

Set variable `test` to "hello, world.": Var content: >hello, world.<

11.2 Set variable to file content

```
Set variable `test` to file content: $(var:test=<res/var_content.txt).
Var content: >$(var:test)<
```

Set variable `test` to file content: . Var content: >A longer var content that is read from a file and spans multiple lines.

Var Line 3

Var Line 4<

11.3 Externally defined vars

Variables can also be defined via the `-D` command line option

- either directly, e.g. `-Dmyextvar="my externally defined var"`
- or via file content, e.g. `-Dmyextvar='<var_content.txt'`

`myextvar` is currently defined as >(VAR "myextvar" IS UNDEFINED)<

11.4 Built-in special variables

The following variables have special meaning:

Name	Description
header	<p>A built-in markdown snippet that creates a simple document header:</p> <pre> noheader Document: : :\$(buf!title) Author: : :\$(buf!author) Date: : :\$(buf!date) .header </pre>
localdate	Current date, e.g. 06-Oct-2018
localtime	Current time, e.g. `12:46:16"
localdatetime	Current date and time, e.g. 06-Oct-2018 12:46:16
localyear	Current year, e.g. 2018
localmonth	Current month, e.g. 10
localmonthnameshort	Current month name, e.g. Oct
localmonthnamelong	Current month name, e.g. October
localdayofmonth	Current day of month, e.g. 6
localdayofmonthname	Current day of month, e.g. 6th
localdayofweeknameshort	Current day of week, e.g. Sat
localdayofweeknamelong	Current day of week, e.g. Saturday

Special variables

12 Include markdown

12.1 Example

```
Markdown content can be included from an external file:
>$(inc:res/var_content_3.md)<
```

Markdown content can be included from an external file: >Content of *external* **markdown** file.<



Alternatively, the value of a variable can be set to the content of a file via the `-Dmyvar='<myfile.md'` cmdline option, then the variable can be inserted at any point via `$(var:myvar)`.

12.2 Parametrization

Along with [variables](#) and / or [redefined link refs](#), this preprocessor feature can be used to parametrize the included markdown content by

- redefining variables before the include
- redefining referenced links before or directly after the include

13 Include HTML

In the postprocessor pass, additional, pre-generated HTML content can be included from files or memory buffers.

13.1 Example

```
Raw HTML content can be included from an external file:  
>$(inc!res/html_content.html)<
```

Raw HTML content can be included from an external file: >**hello, *HTML* world.** <



Postprocessor instructions use the `$(<cmd>!<arg>)` syntax.



The reason for using a different syntax (`!` instead of `:`) is that this way the preprocessor can generate postprocessor instructions.

14 Buffer redirection

Sections of the document can be redirected to memory buffers by enclosing them in `>>>` and `<<<`:

```
>>>buf_id
..buffer content..
<<<
```

These buffers can then be inserted at any point in the generated document via the `$(buf!buf_id)` post-processor command.

When creating multiple buffers, the `<<<` sequence before the next buffer can be omitted:



```
>>>buf1
buf1 content
>>>buf2
buf2 content
<<<
```

14.1 Example

```
- A list that contains a nested buffer
  - $(buf!mybuf)
- Another list item
```

results in:

- A list that contains a nested buffer
 - This is the content of buffer `mybuf`.

It may contain arbitrary markdown text, except for `>>>` and `<<<`.

Here's a code block:

```
some code
```

And a table:

a	b
a1	b1

- Another list item

A list with a nested buffer

14.2 Special buffer names

The following buffer names have special meanings:

Buffer	Description
body	Stores the main document body in template mode.
toc	Stores the table of contents
footnote	Stores the current (global) footnote table
w3validator	Stores a link to the validator.w3.org service.
title	Document title (set via % header)
author	Document author(s) (set via % header)
date	Document title (set via % header)

Special buffers



When the tool is run in template mode (`-x`), the output buffer is created by loading the provided template file and post processing it.

The generated markdown content can be inserted at any point in the template by inserting a buffer (e.g. `$(buf!body)`), or by including an external file (e.g. `$(inc!myfile.html)`).

15 Pre-formatted text (code block)

Code blocks are created by enclosing a section of the document in `~~~`:

```
~~~
..fenced code block contents..
~~~
```

⇒

```
..fenced code block contents..
```

15.1 Indented code blocks

Unlike the [original markdown](#) implementation, four (or more) spaces at the beginning of a line do **not** produce pre-formatted code blocks.

Code blocks can be indented, and the indentation will not be visible in the output file:

```
~~~~~
another code block.
  (indented in the markdown source file)
    - raw text list item 1
      - raw text list 1.1
~~~~~
] Indented code block example
```

⇒

```
another code block.
  (indented in the markdown source file)
    - raw text list item 1
      - raw text list 1.1
```

Indented code block example

Some more text.

16 Block quotes

Block quotes are created by prepending a line with one or more `>` strings (the number of strings determines the block level).

```
> a block quote
```

⇒

a block quote

```
> > a double block quote
```

⇒

a double block quote



The `>` characters must be separated by spaces to distinguish them from the buffer redirection char sequence (`>>>`).

```
> a block quote
> > directly followed by a double quote
> and back to a single quote
```

⇒

a block quote

directly followed by a double quote

and back to a single quote

Block quotes can be used with all paragraph types, e.g.

a	b
a1	b1

or

- list item 1
- list item 2
- list item 3

or

A block quote that contains a fenced code block:

```
..code block content..
```

some text

16.1 Citations

```
> The secret of getting ahead is getting started. The secret of getting
started is breaking your complex overwhelming tasks into small
manageable tasks, and then starting on the first one.
> -- Mark Twain[-][^marktwain]
```

⇒

“*The secret of getting ahead is getting started. The secret of getting started is breaking your complex overwhelming tasks into small manageable tasks, and then starting on the first one.*”
— Mark Twain⁴

16.1.1 Nested citations

A block quote

“*with a citation*”
— me

back to 1st level quote

17 Admonition paragraphs

17.1 NOTE:

```
NOTE: This is a note.
```

⇒



This is a note.

```
note: This is a small note.
```

⇒



This is a small note.

17.2 IMPORTANT:

```
IMPORTANT: This is an important note.
```

⇒



This is an important note.

```
important: This is a small, important note.
```

⇒



This is a small, important note.

17.3 TIP:

```
TIP: This is a tip.
```

⇒



This is a tip.

```
tip: This is a small tip.
```

⇒



This is a small tip.

17.4 CAUTION:

CAUTION: Take caution !

⇒



Take caution !

caution: Take (a bit of) caution !

⇒



Take (a bit of) caution !

17.5 WARNING:

WARNING: This is a warning.

⇒



This is a warning.

warning: This is a small warning.

⇒



This is a small warning.

17.6 TODO:

TODO: This is a to-do.

⇒



This is a to-do.

todo: This is a small to-do.

⇒



This is a small to-do.

17.7 Multi-paragraph admonitions

TIP:

Admonitions can span multiple lines.

They may contain arbitrary markdown, except for further admonitions.

Lists:

- list item 1
- list item 2
- list item 3

Tables:

```
|a|b
|a1|b1
```

```
> block quote
```

```
===
```

⇒

Admonitions can span multiple lines.

They may contain arbitrary markdown, except for further admonitions.

Lists:

- list item 1
- list item 2
- list item 3



Tables:

a	b
a1	b1

```
| block quote
```

17.8 Admonition icon resources

By default, admonition icons will link to the following files:

Type	File
note:	icons/note_s.png
NOTE:	icons/note_l.png
important:	icons/important_s.png
IMPORTANT:	icons/important_l.png
tip:	icons/tip_s.png
TIP:	icons/tip_l.png
caution:	icons/caution_s.png
CAUTION:	icons/caution_l.png
warning:	icons/warning_s.png
WARNING:	icons/warning_l.png
todo:	icons/todo_s.png
TODO:	icons/todo_l.png

Icon resources

17.9 Alternative icon resource path

The default icon path prefix (`icons/`) can be changed via the `-ip` [command line switch](#).

17.10 Inlined icon resources

When using the `-il`, `-inline-data` command line option, all external images, and the CSS file, will be embedded in the output document.

While this may increase the file size considerably (depending on the number of images and their sizes), this makes it possible to easily distribute the self-contained document via email, similar to a PDF.

18 Header 1 <test>

The syntax for *atx*-style headers is:

```
# myheader
```

The number of # characters determines the header level (1..9).

18.1 Header 2

18.1.1 Header 3

18.1.1.1 Header 4

18.1.1.1.1 Header 5

18.1.1.1.1.1 Header 6

18.1.1.1.1.1.1 Header 7

18.1.1.1.1.1.1.1 Header 8

18.1.1.1.1.1.1.1.1 Header 9

19 Header 1 via =====

The syntax for *setext*-style level 1 headers is:

```
My Header
=====
```

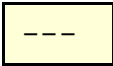
19.1 Header 2 via ---

The syntax for *setext*-style level 2 headers is:

```
My Header
-----
```

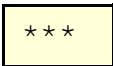
20 Horizontal rules

20.1 Dash



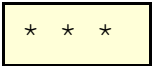
⇒

20.2 Asterisk



⇒

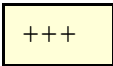
20.3 Asterisk (book)



⇒

* * *

20.4 Pagebreak (print)



⇒

21 Lists

```
- list item 1
- list item 2
- list item 3
```

⇒

- list item 1
- list item 2
- list item 3

```
* list item 1
* list item 2
* list item 3
```

⇒

- list item 1
- list item 2
- list item 3

```
1) list item 1
2) list item 2
3) list item 3
```

⇒

1. list item 1
2. list item 2
3. list item 3

```
1. list item 1
2. list item 2
3. list item 3
```

⇒

1. list item 1
2. list item 2
3. list item 3

21.1 Unordered, nested lists

An unordered, nested list:

```
- list item 1
  - sublist item 1.1
    - sublist item 1.1.1
      - sublist item 1.1.1.1
        List items can span multiple lines (when indented).

        This includes newlines.
        Some *formatted* text.
    - sublist item 1.1.2
  - sublist item 1.2
- list item 2
- list item 3
```

⇒

- list item 1
 - sublist item 1.1
 - sublist item 1.1.1
 - sublist item 1.1.1.1

List items can span multiple lines (when indented).

This includes newlines.
Some *formatted* text.
 - sublist item 1.1.2
 - sublist item 1.2
- list item 2
- list item 3

21.2 Ordered, nested lists

An ordered list:

```

1) list item 1
  1. sublist item 1.1
    1. sublist item 1.1.1
      1) sublist item 1.1.1.1
    2. sublist item 1.1.2
  2. sublist item 1.2
2) list item 2
3) list item 3
    
```

⇒

- 1. list item 1
 - 1. sublist item 1.1
 - 1. sublist item 1.1.1
 - 1. sublist item 1.1.1.1
 - 2. sublist item 1.1.2
 - 2. sublist item 1.2
- 2. list item 2
- 3. list item 3

21.2.1 Start offset

Ordered lists can start at an offset:

- 4. list item 4
- 5. list item 5

21.2.2 Skip

It's also possible to skip sequence numbers:

- 6. list item 6
- 8. list item 8 (*skip 7*)
- 10. list item 10 (*skip 9*)
- 12. list item 12 (*skip 11*)
- 14. list item 14 (*skip 13*)

21.2.3 Auto-enumeration

```
. one  
. two  
. three
```

⇒

1. one
2. two
3. three

21.3 List sub-titles

An unordered list with a sub title:

- list item 1
- list item 2
- list item 3

Unordered list sub title

22 Links

22.1 Direct Links

```
See [this example page] (http://www.example.com/doc3.html)
```

See [this example page](http://www.example.com/doc3.html).

22.2 Direct Links with description

```
Also see [this other example page] (http://www.example.com/doc4.html
"doc4 link description").
```

Also see [this other example page](http://www.example.com/doc4.html).



The link description is shown as a tooltip when hovering over the link.

22.3 Document internal links

```
A [document internal link] (#header-1-test).
```

A [document internal link](#header-1-test).

```
A link to the [#introduction][].
```

A link to the [introduction](#introduction).

22.4 Automatic Links

```
See <http://www.example.com/doc5.html>.
```

See <http://www.example.com/doc5.html>.

Write an email to info@example.com (⇒ will be obfuscated (entity encoding)+ `mailto:` will be auto-prepended.).

An (obfuscated) FTP link: <ftp://myuser:mypasswd@ftp.example.com>.

22.5 Link refs

Link declaration:

```
[example.com]: http://www.example.com
```

Link reference:

```
A link to [example.com][].
```

⇒

A link to [example.com](http://www.example.com).

```
A link to [another example site][example].
[example]: http://www.example.com
```

⇒

A link to [another example site](http://www.example.com).

22.5.1 Links (redefined refs)

Referenced link targets can be redefined:

```
See [this][1] or [that][2] document.

[1]: http://www.example.com/doc1a.html
[2]: http://www.example.com/doc1b.html
```

⇒

See [this](http://www.example.com/doc1a.html) or [that](http://www.example.com/doc1b.html) document.

```
Also see [here][1] or [there][2].

[1]: http://www.example.com/doc2a.html
[2]: http://www.example.com/doc2b.html
```

⇒

Also see [here](http://www.example.com/doc2a.html) or [there](http://www.example.com/doc2b.html).

23 Footnotes

23.1 Local footnotes

```
See [this][^1] or [that][^2] document.

[^1]: http://www.example.com/doc3a.html "Document 3a"
[^2]: http://www.example.com/doc3b.html "Document 3b"
[^^]
```

⇒

See [this](#)⁵ or [that](#)⁶ document.

```
5 http://www.example.com/doc3a.html Document 3a
6 http://www.example.com/doc3b.html Document 3b
```

The local footnote table is inserted when the list of footnote definitions is followed by `[^^]`.



```
[^1]: http://www.example.com/doc3a.html "Document 3a"
[^2]: http://www.example.com/doc3b.html "Document 3b"
[^^]
```

This can be used to insert a footnote table at the end of each logical document section, in addition to an optional, global footnote table at the end of the document.

```
Also see [^here][] or [^there][].

[^here]: http://www.example.com/doc4a.html
[^there]: http://www.example.com/doc4b.html
[^^]
```

⇒

Also see [here](#)⁷ or [there](#)⁸.

```
7 http://www.example.com/doc4a.html
8 http://www.example.com/doc4b.html
```


23.2 Global footnotes

The syntax for creating a global footnote is the same as for local footnotes.

A footnote becomes global when it has not been consumed in a [local footnote table](#)⁹.

The global footnote table is created by inserting the `footnote` buffer at any point in the document:



```
$(buf!footnote)
```

While the buffer can be generated / inserted multiple times, once inserted, a footnote will be marked as *done*.

Each global footnote table will only contain footnotes that have not been displayed in previous tables.

23.3 Footnote-only links

On rare occasions it can be useful to link only to the footnote, e.g.

```
The sky is blue[-][^bluesky].
```

⇒

The sky is blue¹⁰.

24 Images

24.1 Direct image links

```
Here's an image: ![kitty cat (alt text)](res/image.png "kitty cat (title)")
```



Here's an image:

24.2 Scaled image links

```
Here's a scaled image: ![kitty cat (alt text)](res/image.png "kitty cat (title)" w=50).
```



Here's a scaled image:

24.3 Referenced images

```
Here's a referenced image: ![kitty cat (alt text)][kitty].
```



Here's a referenced image:

```
Here's a referenced, scaled image: ![kitty cat (alt text)][kitty_scl].
```



Here's a referenced, scaled image:



Scaled images will link to the original-size image resources.

24.4 Sub-titled images

```
![kitty][]  
] Sub-titled image
```

⇒



Sub-titled image



Similar to links, the short-cut version of an image link omits the link id and the alt text is interpreted as the link id (as seen in the previous example).

24.5 Left-aligned images

```
![kitty][]  
].left  
  
This text is displayed right next to the image.  
  
It can span multiple paragraphs.  
  
- list item 1  
- list item 2  
  
More regular text below the image.  
].none
```

⇒



This text is displayed right next to the image.

It can span multiple paragraphs.

- list item 1
- list item 2

More regular text below the image.



`.left` and `.none` are user-defined CSS classes.

It is possible to right-align images in a similar fashion (just edit the CSS).

25 Tables

- All table lines must start with the `|` pipe character.
- A `|` after the last column is optional.
- The default column alignment is `center`.

25.1 Simple table

```
| Col A      | Col B (a wide column) | Col C
| Data A1    | Data B1                | Data C1
| Data A2    | Data B2                | Data C2
| Data A3    | Data B3                | Data C3
] Simple table
```

⇒

Col A	Col B (a wide column)	Col C
Data A1	Data B1	Data C1
Data A2	Data B2	Data C2
Data A3	Data B3	Data C3

Simple table

25.2 Simple table with cell alignment

```
|:This is Column A | This is Column B | This is Column C:
| Data A1          | Data B1          | Data C1
| Data A2          | Data B2          | Data C2
| Data A3          | Data B3          | Data C3
] Simple table with cell alignment
```

⇒

This is Column A	This is Column B	This is Column C
Data A1	Data B1	Data C1
Data A2	Data B2	Data C2
Data A3	Data B3	Data C3

Simple table with cell alignment

25.3 Cell alignment in --- style table

```
|-----|-----|-----|
| This is Column A | This is Column B | This is Column C |
|:-----:|:-----:|:-----:|
| Data A1          | Data B1          | Data C1          |
| Data A2          | Data B2          | Data C2          |
| Data A3          | Data B3          | Data C3          |
|-----|-----|-----|
] Table cell alignment in --- style table
```

⇒

This is Column A	This is Column B	This is Column C
Data A1	Data B1	Data C1
Data A2	Data B2	Data C2
Data A3	Data B3	Data C3

Table cell alignment in --- style table

25.4 Tables with different text styles, links, and images

```
|-----|-----|-----|
| This is Column A | This is Column B | This is Column C |
|:-----:|:-----:|:-----:|
| `fixed`          | *emphasis1*      | _emphasis2_      |
| ``fixed2``       | **strong1**      | __strong2__      |
| ~subscript~      | ^superscript^    | ~~strikethrough~~|
| [a link][1]      | ![kitty]         | regular text      |
|-----|-----|-----|
] Table cell text styles, links, and images
```

⇒


This is Column A	This is Column B	This is Column C
fixed	<i>emphasis1</i>	<u>emphasis2</u>
fixed2	strong1	<u>strong2</u>
subscript	superscript	strikethrough
a link		regular text

Table cell text styles, links, and images

25.5 Multi-line tables

```
|-----|-----|-----|
| This is Column A | This is Column B | This is Column C |
|:-----:|:-----:|:-----:|
| Data A11 | Data A12 | Data A13 |
| Data A21 | Data A22 | Data A23 |
| Data A31 | Data A32 | Data A33 |
| | +Data A32.line2 | +Data A33.line2 |
| | +Data A32.line3 | +Data A33.line3 |
| | +Data A33.line4 | +Data A33.line4 |
| Data A41 | Data A42 | Data A43 |
|+$(inc!res/html_list.html)| | |
|-----|
] Multi-line table cells via + modifier
```

⇒

This is Column A	This is Column B	This is Column C
Data A11	Data A12	Data A13
Data A21	Data A22	Data A23
Data A31	Data A32	Data A33
	Data A32.line2	Data A33.line2
	Data A32.line3	Data A33.line3
		Data A33.line4
Data A41	Data A42	Data A43
<ul style="list-style-type: none">Item 1Item 2Item 3		

Multi-line table cells via + modifier

25.6 Column widths

```
|cols=15,10,25
|-----|-----|-----|
| This is Column A | This is Column B | This is Column C |
|:-----|:-----|:-----|
| Data A1          | Data B1          | Data C1          |
| Data A2          | Data B2          | Data C2          |
| Data A3          | Data B3          | Data C3          |
|-----|-----|-----|
] Table cell widths via cols attribute
```

⇒

This is Column A	This is Column B	This is Column C
Data A1	Data B1	Data C1
Data A2	Data B2	Data C2
Data A3	Data B3	Data C3

Table cell widths via cols attribute

25.7 Headerless tables

```
|noheader
| left | right
|.noheader_table_example
```

⇒




left right

Each table column can be styled differently by using the CSS `:nth-child(<n>)` feature, e.g.



```
.noheader_table_example .td:nth-child(1) {
  font-size: 20;
  font-weight: normal;
  background-color: #eee;
  padding-left: 0.25em;
  padding-right: 0.25em;
  border: 0px;
}
```

25.8 Complex tables (1)

This is Column A	This is Column B	This is Column C												
<p>This is some regular text.</p> <p>With multiple paragraphs.</p> <div><code>a code block.</code> <code>with multiple lines.</code></div>	<p>Data B1</p>	<div><div><div>1. list item 1b</div><div>2. list item 2b</div><div><div><div>one</div><div>two</div></div><div><div>1.1</div><div>1.2</div></div><div><div>2.1</div><div>2.2</div></div></div><div>○ list item 2b.2</div><div>3. list item 3b</div></div></div>												
<p>Data A2</p>	<div><div><div>1. list item 1a</div><div>2. list item 2a</div><div><div>○ sub list item 2a.1</div></div><div>3. list item 3a</div></div></div>	<p>Data C2</p>												
<p>Data A3</p>	<p>Data B3</p>	<table><tr><th>First</th><th>Second</th><th>Third</th></tr><tr><td>some</td><td>table</td><td>content</td></tr><tr><td>another</td><td>row</td><td>of content</td></tr><tr><td></td><td></td><td></td></tr></table>	First	Second	Third	some	table	content	another	row	of content			
First	Second	Third												
some	table	content												
another	row	of content												
														

Complex tables via buffer redirection

25.9 Complex tables (2)

This is Column A

This is Column B

This is some regular text.

With multiple paragraphs.

a code block.
with multiple lines.

1. list item 1b

2. list item 2b

one

two

1.1

1.2

2.1

2.2

list item 2b.2

3. list item 3b

Data A2

Data B2

Data A3

First

Second

Third

some


table

content

another

row

of content



Nested buffer table

This is Column A

This is Column B

This is some regular text.

With multiple paragraphs.

a code block.
with multiple lines.

1. list item 1b

2. list item 2b

one

two

1.1

1.2

2.1

2.2

list item 2b.2

3. list item 3b

Data A2

Data B2

Data A3

First

Second

Third

some


table

content

another

row

of content



Nested buffer table

This is Column A

This is Column B

This is some regular text.

With multiple paragraphs.

a code block.
with multiple lines.

1. list item 1b

2. list item 2b

one

two

1.1

1.2

2.1

2.2

list item 2b.2

3. list item 3b

Data A2

Data B2

Data A3

First

Second

Third

some


table

content

another

row

of content



Nested buffer table

This is Column A

This is Column B

This is some regular text.

With multiple paragraphs.

a code block.
with multiple lines.

1. list item 1b

2. list item 2b

one

two

1.1

1.2

2.1

2.2

list item 2b.2

3. list item 3b

Data A2

Data B2

Data A3

First

Second

Third

some


table

content

another

row

of content



Nested buffer table

Complex tables via buffer redirection 2

26 CSS classes

The following CSS classes can be used to modify the document's appearance:

26.1 Basic text styles and paragraphs

Description	CSS Class
Main body	<code>.body</code>
Paragraph	<code>.p</code>
Forced end of paragraph div (\)	<code>.lf</code>
Header	<code>.h[1..9]</code>
Header enumeration span	<code>.h[1..9]_enum</code>
Emphasized text	<code>.em</code>
Emphasized text via _	<code>.em2</code>
Strong text	<code>.strong</code>
Strong text via __	<code>.strong2</code>
Strikethrough text	<code>.s</code>
Fixed text	<code>.code</code>
Fixed text (double ``)	<code>.code2</code>
Subscript text	<code>.sub</code>
Superscript text	<code>.sup</code>
Preformatted text	<code>.pre</code>

Basic text styles and paragraphs

26.2 Blockquotes and citations

Description	CSS Class
Blockquote	<code>.blockquote</code>
Blockquote with a citation	<code>.blockquote_cite</code>
Start of citation	<code>.blockquote_cite_begin</code>
End of citation	<code>.blockquote_cite_end</code>
Citation footer	<code>.footer_cite</code>

Blockquotes and citations

26.3 Images

Description	CSS Class
Image	<code>.img</code>
Link to original image	<code>.oimg</code>

Images

26.4 Lists

Description	CSS Class
Unordered root list	<code>.ulroot</code>
Unordered sub list	<code>.ulsub</code>
Ordered root list	<code>.olroot</code>
Ordered sub list	<code>.olsub</code>
Root list item	<code>.liroot</code>
Root list item followed by sub-list	<code>.lirootwithsub</code>
Last root list item	<code>.lirootlast</code>
Sub list item	<code>.lisub</code>
Sub list item followed by sub-list	<code>.lisubwithsub</code>
Last sub list item	<code>.lisublast</code>

Lists

26.5 Horizontal rules and printed page breaks

Description	CSS Class
Horizontal rule (***)	<code>.hr_asterisk</code>
Horizontal rule (* * *)	<code>.hr_spaced_asterisk</code>
Horizontal rule (* * *)	<code>.hr_spaced_asterisk_inner</code>
Printed page break	<code>.hr_pagebreak</code>

Horizontal rules and printed page breaks

26.6 Subtitled paragraphs

Description	CSS Class
Sub-titled div (outer)	<code>.subdiv_outer</code>
Sub-titled div (inner)	<code>.subdiv_inner</code>
Sub-titled div (title)	<code>.subdiv_title</code>

Subtitled paragraphs

26.7 Tables

Description	CSS Class
Table	<code>.table</code>
Table header row	<code>.thtr</code>
Table header cell	<code>.thtd</code>
Table data row	<code>.tr</code>
Table data cell	<code>.td</code>

Tables

26.8 Footnotes

Description	CSS Class
Footnote link (superscript)	<code>.footnote_nr</code>
Local Footnote table	<code>.footnote_table_local</code>
Global Footnote table	<code>.footnote_table_global</code>
Footnote table row	<code>.footnote_tr</code>
Footnote table cell (number)	<code>.footnote_td_nr</code>
Footnote table cell (link)	<code>.footnote_td_link</code>
Footnote table link	<code>.footnote_a</code>

Footnotes

26.9 Table of contents

Description	CSS Class
Table Of Contents div	<code>.toc_div</code>
Table Of Contents enumeration div	<code>.toc_enum</code>

Table of contents



The CSS class names used in the table of contents are prefixed by `toc_`, e.g. `toc_lirootwithsub`.

26.10 Admonitions

Description	CSS Class
Admonition note: outer div	.note_s_div_outer
Admonition NOTE: outer div	.note_l_div_outer
Admonition important: outer div	.important_s_div_outer
Admonition IMPORTANT: outer div	.important_l_div_outer
Admonition tip: outer div	.tip_s_div_outer
Admonition TIP: outer div	.tip_l_div_outer
Admonition caution: outer div	.caution_s_div_outer
Admonition CAUTION: outer div	.caution_l_div_outer
Admonition warning: outer div	.warning_s_div_outer
Admonition WARNING: outer div	.warning_l_div_outer
Admonition note: table	.notediv_s_table
Admonition NOTE: table	.notediv_l_table
Admonition important: table	.important_s_div_table
Admonition IMPORTANT: table	.important_l_div_table
Admonition tip: table	.tip_s_div_table
Admonition TIP: table	.tip_l_div_table
Admonition caution: table	.caution_s_div_table
Admonition CAUTION: table	.caution_l_div_table
Admonition warning: table	.warning_s_div_table
Admonition WARNING: table	.warning_l_div_table
Admonition note: table row	.note_s_div_tr
Admonition NOTE: table row	.note_l_div_tr
Admonition important: table row	.important_s_div_tr
Admonition IMPORTANT: table row	.important_l_div_tr
Admonition tip: table row	.tip_s_div_tr
Admonition TIP: table row	.tip_l_div_tr
Admonition caution: table row	.caution_s_div_tr
Admonition CAUTION: table row	.caution_l_div_tr
Admonition warning: table row	.warning_s_div_tr
Admonition WARNING: table row	.warning_l_div_tr

Admonitions

Description	CSS Class
Admonition image ()	.infoimg
Admonition note: icon column	.note_s_div_td_icon
Admonition NOTE: icon column	.note_l_div_td_icon
Admonition important: icon column	.important_s_div_td_icon
Admonition IMPORTANT: icon column	.important_l_div_td_icon
Admonition tip: icon column	.tip_s_div_td_icon
Admonition TIP: icon column	.tip_l_div_td_icon
Admonition caution: icon column	.caution_s_div_td_icon
Admonition CAUTION: icon column	.caution_l_div_td_icon
Admonition warning: icon column	.warning_s_div_td_icon
Admonition WARNING: icon column	.warning_l_div_td_icon
Admonition note: text column	.note_s_div_td_text
Admonition NOTE: text column	.note_l_div_td_text
Admonition important: text column	.important_s_div_td_text
Admonition IMPORTANT: text column	.important_l_div_td_text
Admonition tip: text column	.tip_s_div_td_text
Admonition TIP: text column	.tip_l_div_td_text
Admonition caution: text column	.caution_s_div_td_text
Admonition CAUTION: text column	.caution_l_div_td_text
Admonition warning: text column	.warning_s_div_td_text
Admonition WARNING: text column	.warning_l_div_td_text
Admonition note: inner div	.note_s_div_inner
Admonition NOTE: inner div	.note_l_div_inner
Admonition important: inner div	.important_s_div_inner
Admonition IMPORTANT: inner div	.important_l_div_inner
Admonition tip: inner div	.tip_s_div_inner
Admonition TIP: inner div	.tip_l_div_inner
Admonition caution: inner div	.caution_s_div_inner
Admonition CAUTION: inner div	.caution_l_div_inner
Admonition warning: inner div	.warning_s_div_inner
Admonition WARNING: inner div	.warning_l_div_inner

Admonitions (cont.)

26.11 Class name prefixes

The CSS class names can be prefixed via the `-cp <class_prefix>` command line option.

26.12 Custom CSS classes

Each paragraph can be assigned a custom CSS class:

```
Some text

This is a div with a custom CSS class
].myclass

Some more text
```

⇒

Some text

This is a div with a custom CSS class

Some more text



Custom CSS classes can also be used for advanced layouts, e.g. text floating right next to an image or a table.

27 Creating a print version (PDF) of the document

While `md` itself cannot output PDF files, there are 3rd party tools that can transform the HTML output to a printable version of the document.

The `@media print` CSS selector can be used to select the paper size and margins, or fine-tune specific elements (e.g. automatically add page-breaks before or after certain elements, or prevent page breaks within certain elements).

The `+++` char sequence inserts a forced page break, while `+++col` / `+++/col` enable / disable multi-column mode.



If all that is needed is a redistributable, self-contained version of the document, the `-il` command line option can be used to embed any additional resources (images, CSS) into the HTML file.

27.1 Pandoc / LaTeX / wkhtmltopdf

These are free, open-source HTML-to-PDF solutions which seem to work fine with simpler documents but currently (October 2018) have issues with tables (LaTeX error message).

27.2 Microsoft Print to PDF (Windows 10)

Windows 10 comes with a PDF printer driver that creates an excellent rendition of the document (including multi-column paragraphs).

It does have some (minor) drawbacks, though:

- Document internal links will become plain text (which, to be honest, doesn't really matter when the document is actually printed)
- The table of contents will not contain leading dots and page numbers (some of the print-related CSS is ignored)

27.3 CutePDF (Windows 98/ME/2000/XP/2003/Vista/7/2012/8/8.1/10)

For earlier versions of Windows there are 3rd party PDF printer drivers like e.g. [CutePDF¹¹](#) (the output looks basically the same as with the Print to PDF driver).

27.4 Prince

This is a 3rd party, standalone software¹³ that is free for non-commercial use (available for Windows, MacOS, Linux, and FreeBSD).

It implements the W3 [GCPM¹²](#) spec, preserves external and document internal links, supports multi-column paragraphs, and can create a nice looking table of contents with `.` leaders and page numbers.

-
- 1 <https://daringfireball.net/projects/markdown/> a text-to-HTML conversion tool for web writers
 - 2 <http://asciidoc.org/> a text document format for writing notes, documentation, articles, books, ebooks, slideshows, web pages, man pages and blogs
 - 3 <https://pandoc.org/> a universal document converter
 - 4 Maybe he was not the first one to say this ;)
 - 9 [#local-footnotes](#) a link to the local footnote documentation (global footnote test)
 - 10 https://en.wikipedia.org/wiki/Wikipedia:You_do_need_to_cite_that_the_sky_is_blue (yep, it's due to Rayleigh scattering)
 - 11 <http://www.cutepdf.com/Products/CutePDF/writer.asp> Convert to PDF documents on the fly (FREE software for commercial and non-commercial use)
 - 12 <https://www.w3.org/TR/css-gcpm-3/> CSS Generated Content for Paged Media Module
 - 13 <https://www.princexml.com/> Convert your HTML documents to PDF
-



Document created on 06-Oct-2018 12:46:16