

# Markdown Cheat Sheet (for Pandoc + Emacs)

## Paragraph and Block

Line-break in Paragraph	use backslash	Line1\ Line2
dito	Two blanks at line end	Line1__
<b>Preformatted</b> <code>&lt;p&gt;&lt;br/&gt;</code> (no wrap, no indent)	„ “ at begin of every line.	Line1   Line2
<b>Code</b> <code>&lt;code&gt;&lt;pre&gt;</code> (no wrap, CSS indent) special chars quoted <b>C-c &gt;</b> or <b>C-c &lt;</b> shift/unshift Optionally with line numbers and label	4 leading blanks, empty lines allowed. 1st/last line <code>```</code> or <code>~~~</code> , with lang or <code>{.lang}</code>  Also: <code>startFrom="10"</code>	<code>__ _ _ _code</code>  <code>```java</code> <code>code</code> <code>```</code>  <code>```{.java .numberLines #label}</code>
<b>Indented Paragraph</b> (with wrap + indent) <code>&lt;blockquote&gt;&lt;p&gt;</code> <b>C-c C-s b</b> , <b>C-c C-s B</b>	„>“ at begin of at least first line, after empty line. Nested: „>>“, „>>>“	<code>&gt; Line1</code> <code>Line2</code>  <code>&gt;&gt; Line3</code>
Multi-line Comment	YAML block with #	<code>---</code> <code># comment</code> <code>...</code>
Horizontal Line <b>C-c -</b>	Isolated line, <code>&gt;=3 -_*</code>	<code>---</code>

## Characters

Quote special characters <code>&lt;code&gt;</code> <b>C-c C-s c</b>	Backslash, backquote. Literal backquote: 2x + blank	<code>\\, a\\*b\\*c</code> <code>\\, `a*b*c`</code> <code>``_abc``</code>
dito with syntax	end with <code>{.lang}</code>	<code>`SELECT * FROM table`{.sql}</code>
Non-breaking space	backslash with blank	<code>3\\ km</code>
Hyphen (with <code>--smart</code> )	Short / long	<code>-- / ---</code>
Emphasis (italic) <b>C-c C-s e</b>	* around/within word _ around word not between blanks	<code>*word* w*or*d</code> <code>_word_</code> , <b>Not:</b> <code>w_or_d</code> <b>Not:</b> <code>a * b _ c</code>
Strong (bold) <b>C-c C-s s</b>	Dito <code>**</code> and <code>__</code>	<code>**word**</code>
Very strong (italic+bold)	Dito <code>***</code> and <code>___</code>	<code>***word***</code> , but also <code>w___or___d</code>
Strike-through	2 tilde	<code>~~deleted~~</code>
Superscript	caret (blanks quoted!)	<code>E=m*c^2^</code> , <code>n^a\ +\ b^</code>
Subscript	tilde (blanks quoted!)	<code>H~2~0</code>

## Document Structure

Heading, <b>C-c C-n</b> navigate	Use #, ## etc.	# AHeading1 {#id .cls key=val} ## AHeading2
Heading unnumbered	Overrides -numbers-sections	# AHeading1 {.unnumbered} ## AHeading2 {-}
Unordered List (lines)	Use * + - (even mixed)	* item1 + item2 - item3
Unordered List (para) <u>Lists:</u> <b>C-c</b> – decrease level <b>C-c</b> = increase level	Dito, with empty line (creates <code>&lt;li&gt;&lt;p&gt;</code> ), indent first line by 4	* item1  _ _ _ _continued (paragraph 2)
Terminate a List e.g. unordered list followed by code	Empty line, then <code>&lt;--</code> <code>--&gt;</code> closes the preceding list	* item1  <!-- -->  ____code
Nested list	Use 4 blanks per level	- item 1 _ _ _ _* item 1.1
Ordered List <code>&lt;ol&gt;</code>	Use 1., a., 1), (1). First number = start. A. Item (2 blanks!)	1. Item 3. Item A. _ _Item
Numbered Examples <u>Lists:</u> <b>Alt-RET</b> create item <b>Alt-Up</b> , <b>Alt-Down</b> move <b>Alt-Left</b> outdent <b>Alt-Right</b> indent <b>C-c C-k</b> kill element	Define label: (@) or (@label), gets numbered across paragraphs, @label inserts the number.	(@slides) About slides  A normal paragraph...  (@pdf) About PDF  See topic @slides.
Definition List <code>&lt;dl&gt;</code>	Term, \n, : + 3 blanks, following lines indented or not.	Term :_ _ _Paragraph line1 line2...

## Links

Emacs commands typically use the active region for the link text etc.

URL (shown) <b>C-c C-a u</b>	<URL> (also mailto)	<http://google.com/>
URL With Text. <b>C-c C-a l</b>	[text](URL "Title")	[Google] (http://google.com "T")
File Link		[text] (filename.html)
Local Link (explicit id)	for e.g. # H1 {#id}	[text] (#id)
Local Link (auto id)	for e.g. # The X	[text] (#the-x)
Local Link (text)	for e.g. # The X	See chapter [The X] See chapter [Link text][The X]
Referenced Link (Label) <b>C-c C-a L</b> , <b>C-c C-j</b> (jump)	[text][label] [label]: URL	[Google] [G] [G]: http://google.com "T"
Referenced Link (Text) (text case-insensitive)	[text][] [text]: URL	[Google] [] [google]: http://google.com "T"

## Images

Image w/Text C-c C-a i	![text](URL "Title")	! [Alt] (http://a.com/x.png "T")
Referenced Image (Label) C-c C-a I	![text][label] [label]: URL "Title"	[Alt text] [G] [G]: http://a.com/x.png "T"
Image in separate line	[text] becomes caption	
Local Images	Path relative to cwd	

## Tables

Required: blank line after table. „Upper border“ := line above first data row (not above header)

<p><b>Simple Table:</b> border = dashed lines; cells must be under upper border part (fixed width).</p> <p>Pandoc extension simple_tables</p>	<p>Each row has 1 line (long line =&gt; wrapped) no empty lines. <b>Alignment</b> := position of <b>header</b> or <b>1st row</b> to upper border line</p>	<pre> de      or          or      Hex ----- rot          red          `ff0000` grün       green        `00ff00` ----- hdr+bot  no header  no bottom </pre>
<p><b>Multi-line Table</b> like Simple Table, plus:</p> <p>Pandoc extension multiline_tables</p>	<p>Source rows multiline, sep. by <b>empty line</b>; length of upper border parts defines % of col. width (=&gt; colgroup); top/bottom border may have blanks or not; bottom border required</p>	<pre> ----- de      en          or ----- rot    red          rot    red,       nice!          nice! grün   green        grün   green ----- </pre>
<p><b>Pipe Table</b> (most widespread – PHP Markdown, Flavored GitHub)</p> <p>Pandoc extension pipe_tables</p>	<p>beg./end „ “ optional; need not be aligned; <b>Alignment</b> := position of <b>:</b> in upper border; contents wrapped; header is optional</p>	<pre>   Right   Left   Mid   Default    -----: :----- :---: -----    123   abcde   x   123   </pre>
<p><b>Grid Table</b> (visually drawn, e.g using Emacs table mode)</p> <p>Pandoc extension grid_tables</p>	<p>Output left-aligned; cells: „arbitrary block elements“; header with „+====+“ line is optional</p>	<pre> +-----+-----+   Command   Function   +-----+-----+   `drop db`   drops DB                 *item 1*   +-----+-----+ </pre>
<p>Table Caption (paragraph before/after)</p>	<p>Table: &lt;caption&gt; <u>or</u> : &lt;caption&gt;</p>	<pre>   123 abcde   x  123   : Pipe Table </pre>

## Footnotes

Embedded Footnote	^[Footnote]	Unix[^UNIX = trademark...] is
Reference Footnote C-c C-a f, C-c C-j (jump)	^[label] ^[label]: text (indented)	Unix[^unix] [^unix]: UNIX = trademark... _ _ _ _Paragraph 2 of footnote

# Emacs Markdown Mode

C-c C-c e	Export
C-c C-c v	Export and preview

# Emacs Table Mode

Recommendation: have at least one blank line above and below a Markdown table.

Only one space can be entered between words.

To form a new paragraph, the line must start with some space characters or follow a blank line

(empty)	<b>table-insert =&gt;</b> cell width(s) may be blank-separated list	+-----+ editing („activated“)   a   b   +-----+   c   c   +-----+
a b (free text) c d	<b>table-capture =&gt;</b> Vertically: C-o or C-c C-j = line-break <b>&lt;= table-release</b>	dito
+-----+ (grid table, free text)   a   b   +-----+   c   c   +-----+	<b>table-recognize</b> {, -region, table, cell} => <b>&lt;= table-unrecognize ...</b>	<b>table-generate-source =&gt;</b> <table>... <td>a</td>...</table> HTML

Menu Tools   Table	Table creation / initialization
Menu Table, context menu	Table manipulation (if point is within a table)
C->, or C-c C-c >	Widen/narrow active cell by <i>prefix</i> characters
C-}, or C-c C-c }, <u>dito</u> {	Heighten row by <i>prefix</i> lines, <u>dito</u> lower {
C-+, or C-c C-c +	Insert 1 or <i>prefix</i> rows or columns (prompted which one)
<b>table-insert-row/-column</b> <b>table-delete-row/-column</b>	Inserts before current row/column, prefix argument = how many; if point is outside the table, appends a row/column
C-!, or <b>table-fixed-width-mode</b>	In fixed width mode, typing in a cell doesn't change cell width. „\“ at end of line indicates continuation on next line.
<b>table-insert-sequence</b> STR N INCR. INTERVAL JUST. STR empty = erase, array: „A[0]“ N<0 backwards, N=0 entire table	Inserts a specified sequence string into each cell. (table-insert-sequence "0" 3 5 1 'right) = 0, 5, 10 (table-insert-sequence "a" 26 1 1 'center) = a, b, c, ..., z
C-:, or <b>table-justify</b>	Justify contents of cell(s). The table code detects justification of each cell when recognizing a table.
<b>table-split-cell-vertically/horizontally</b>	Horizontally: „split“/„left“/“right“