↖ Topic Index

Markdown Markup Support

Description	Keystroke	Function	Note
Editing Markdown files **Markdown:** • basics • syntax • Github Markdown **CommonMark Spec** • interactive test **GitHub Flavored Markdown Spec** See also: ∑ Speedbar	📦 The **Markdown markup language** is supported by the **markdown-mode** external package. • **Supported file extensions**: `.md, .markdown, .mkd, .mdown, .mkdn, .mdwn.` With PEL, activate markdown support by turning on (setting to **t**) the following user-options: • **pel-use-markdown** user-option, which activates markdown support as a whole, • **pel-use-markdon-mode** user-option, which activates using the **markdown-mode external package** as the package that handles markdown. PEL provides support for the following extra packages that are used with the **markdown-mode** and provide more functionality: 📦 The **grip-mode** external package — activated by **pel-use-grip-mode** user-option 📦 The **impatient-showdown** external package — activated by **pel-use-impatient-showdown** user-option. 📦 The **markdown-preview-eww** external package — activated by **pel-use-preview-eww** user-option. ⚠ **markdown-live-preview-mode** is better 📦 The **markdown-preview-mode** external package — activated by **pel-use-markdown-preview-mode** user-option. 📦 The **markdown-toc** external package — activated by **pel-use-markdown-toc** user-option. 📦 The **vmd-mode** external package — activated by **pel-use-vmd-mode** user-option. 📦 The **remark-mode** external package — activated by **pel-use-remark-mode** user-option. The packages with a ⚠ are not recommended. Note: **markdown-live-review-mode** is from **markdown-mode** external package. **∑ Speedbar Support:** • PEL activates ∑ Speedbar support for markdown when the **pel-use-speedbar** user-option is turned on (set to **t**).		
Open this PDF file. See also: **∑ Help/Info**	`<f11> SPC M-m <f1>` `<f12> <f1>`	(**pel-help-pdf** &optional OPEN-WEB-PAGE)	Open the **Markdown** local PDF. If the prefix argument (like **C-u** or **M--**) is used, then it opens the remote GitHub hosted raw PDF instead. If the **pel-flip-help-pdf-arg** user-option is set it's the other way around.
∑ Customize PEL Markdown support	`<f11> SPC M-m <f2>` `<f12> <f2>`	(**pel-customize-pel** &optional OTHER-WINDOW)	Customize PEL Markdown support: open pel-pkg-for-markdown group. • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
∑ Customize Emacs Markdown support	`<f11> SPC M-m <f3>`	(**pel-customize-library** &optional OTHER-WINDOW)	Customize Markdown external packages support: 1. **markdown** 2. **grip** 3. **impatient-showdown** 4. **markdown-preview** 5. **markdown-toc** 6. **vmd** 7. **edit-indirect** 8. **htmlize** 9. **simple-https** • If OTHER-WINDOW is non-nil (use **C-u**), display in another window.
Markdown-mode	The markdown-mode is automatically invoked for files with the .md and .markdown file extensions.		
Activate markdown-mode	`M-x markdown-mode`	(**markdown-mode**)	Activate the major mode for editing Markdown files.
Markup Commands	The following commands insert text markup. With PEL these commands are also accessible via the **<f12>** and the **<f11> SPC M-m** prefix keys. ☛The **<f11> SPC M-m** prefix key provides access to markdown markup commands even if the buffer is not editing a markdown file. ☛ However if the markdown-mode file is not already loaded you will have to first load it explicitly with: `M-x load-library markdown-mode`		
Bold	• **C-c C-s b** • **<f12> b**	(**markdown-insert-bold**)	Insert markup to make a region or word bold. • If there is an active region, make the region bold. • If the point is at a non-bold word, make the word bold. • If the point is at a bold word or phrase, remove the bold markup. • Otherwise, simply insert bold delimiters and place the point in between them.
Italic	• **C-c C-s i** • **C-c C-s e** • **<f12> i**	(**markdown-insert-italic**)	Insert markup to make a region or word italic. • If there is an active region, make the region italic. • If the point is at a non-italic word, make the word italic. • If the point is at an italic word or phrase, remove the italic markup. • Otherwise, simply insert italic delimiters and place the point in between them.
Insert Code fragment	• **C-c C-s c** • **<f12> c**	(**markdown-insert-code**)	Insert markup to make a region or word an inline code fragment. • If there is an active region, make the region an inline code fragment. • If the point is at a word, make the word an inline code fragment. • Otherwise, simply insert code delimiters and place the point in between them.
Insert GFM Code fragment - Query for language	• **C-c C-s C** • **<f12> C**	(**markdown-insert-gfm-code-block** &optional LANG EDIT)	Insert GFM code block for language LANG. • If LANG is nil, the language will be queried from user. • If a region is active, wrap this region with the markup instead. • If the region boundaries are not on empty lines, these are added automatically in order to have the correct markup. • When EDIT is non-nil (e.g., when **C-u** is given), edit the code block in an indirect buffer after insertion.
Insert footnote	• **C-c C-s f** • **C-c C-a f** • **<f12> f**	(**markdown-insert-footnote**)	Insert footnote with a new number and move point to footnote definition.
Insert foldable block	• **C-c C-s F** • **<f12> F**	(**markdown-insert-foldable-block**)	Insert details disclosure element to make content foldable. • If a region is active, wrap this region with the disclosure element. More detais here 'https://developer.mozilla.org/en-US/docs/Web/HTML/Element/details'.
Insert <kbd> tags	• **C-c C-s k** • **<f12> k**	(**markdown-insert-kbd**)	Insert markup to wrap region or word in <kbd> tags. • If there is an active region, use the region. If the point is at a word, use the word. Otherwise, simply insert <kbd> tags and place the point in between them.
Start a pre-formatted section	• **C-c C-s p** • **<f12> p**	(**markdown-insert-pre**)	Start a preformatted section (or apply to the region). • If Transient Mark mode is on and a region is active, it is marked as preformatted text.
Format region as pre-formatted text	• **C-c C-s P** • **<f12> P**	(**markdown-pre-region** BEG END)	Format the region as preformatted text. • Arguments BEG and END specify the beginning and end of the region.
Insert block-quote	• **C-c C-s q** • **<f12> q**	(**markdown-insert-blockquote**)	Start a blockquote section (or blockquote the region). • If Transient Mark mode is on and a region is active, it is used as the blockquote text.
Block-quote the region	• **C-c C-s Q** • **<f12> Q**	(**markdown-blockquote-region** BEG END)	Blockquote the region. • Arguments BEG and END specify the beginning and end of the region.
Insert strike-through	• **C-c C-s s** • **<f12> s**	(**markdown-insert-strike-through**)	Insert markup to make a region or word strikethrough. • If there is an active region, make the region strikethrough. If the point is at a non-bold word, make the word strikethrough. • If the point is at a strikethrough word or phrase, remove the strikethrough markup. Otherwise, simply insert bold delimiters and place the point in between them.
Insert table See section "Modify tables" below for more.	• **C-c C-s t** • **<f12> t**	(**markdown-insert-table** &optional ROWS COLUMNS ALIGN)	Insert an empty pipe table. • Optional arguments ROWS, COLUMNS, and ALIGN specify number of rows and columns and the column alignment.

Description	Keystroke	Function	Note
Insert inline URI	• **C-c C-a u** • **<f12> u**	(**markdown-insert-uri** &optional URI)	Insert markup for an inline URI. • If there is an active region, use it as the URI. If the point is at a URI, wrap it with angle brackets. If the point is at an inline URI, remove the angle brackets. Otherwise, simply insert angle brackets place the point between them.
Insert wiki link	• **C-c C-s w** • **C-c C-a w** • **<f12> w**	(**markdown-insert-wiki-link**)	Insert a wiki link of the form [[WikiLink]]. • If there is an active region, use the region as the link text. • If the point is at a word, use the word as the link text. • If there is no active region and the point is not at word, simply insert link markup.
Add GFM Checkbox	• **C-c C-s [** • **<f12> [**	(**markdown-insert-gfm-checkbox**)	Add GFM checkbox at point. • Returns t if added. Returns nil if non-applicable.
Insert a new list item	• **M-RET** • **C-c C-j** • **C-c C-x m** • **<f12> l**	(**markdown-insert-list-item** &optional ARG)	Insert a new list item. • If the point is inside unordered list, insert a bullet mark. If the point is inside ordered list, insert the next number followed by a period. Use the previous list item to determine the amount of whitespace to place before and after list markers. • With a **C-u** prefix (i.e., when ARG is (4)), decrease the indentation by one level. • With two **C-u** prefixes (i.e., when ARG is (16)), increase the indentation by one level.

Insert/Update link	• **C-c C-l**
	• **C-c C-s l**
	• **C-c C-a L**
	• **C-c C-a l**
	• **C-c C-a r**
	• **<f12> L**
		(**markdown-insert-link**)	Insert new or update an existing link, with interactive prompts.
			• If the point is at an existing link or URL, update the link text, URL, reference label, and/or title. Otherwise, insert a new link. The type of link inserted (inline, reference, or plain URL) depends on which values are provided:
			  • If a URL and TEXT are given, insert an inline link: `[TEXT](URL)`.
			  • If [REF] and TEXT are given, insert a reference link: `[TEXT][REF]`.
			  • If only TEXT is given, insert an implicit reference link: `[TEXT][]`.
			  • If only a URL is given, insert a plain link: `<URL>`.
			  • In other words, to create an implicit reference link, leave the URL prompt empty and to create a plain URL link, leave the link text empty.
			• If there is an active region, use the text as the default URL, if it seems to be a URL, or link text value otherwise.
			• If a given reference is not defined, this function will additionally prompt for the URL and optional title. In this case, the reference definition is placed at the location determined by 'markdown-reference-location'. In addition, it is possible to have the 'markdown-link-make-text-function' function, if non-nil, define the default link text before prompting the user for it.
			• If 'markdown-disable-tooltip-prompt' is non-nil, the user will not be prompted to add or modify a tooltip text.
			• Through updating the link, this function can be used to convert a link of one type (inline, reference, or plain) to another type by selectively adding or removing information via the prompts.

Insert/Update Image	• **C-c C-i**
	• **<f12> I**
		(**markdown-insert-image**)	Insert new or update an existing image, with interactive prompts.
			• If the point is at an existing image, update the alt text, URL, reference label, and/or title. Otherwise, insert a new image.
			• The type of image inserted (inline or reference) depends on which values are provided:
			  • If a URL and ALT-TEXT are given, insert an inline image: `![ALT-TEXT](URL)`.
			  • If [REF] and ALT-TEXT are given, insert a reference image: `![ALT-TEXT][REF]`.
			• If there is an active region, use the text as the default URL, if it seems to be a URL, or alt text value otherwise.
			• If a given reference is not defined, this function will additionally prompt for the URL and optional title. In this case, the reference definition is placed at the location determined by 'markdown-reference-location'.
			• Through updating the image, this function can be used to convert an image of one type (inline or reference) to another type by selectively adding or removing information via the prompts.

Insert/replace Header	• **C-c C-s h**
	• **C-c C-t h**
	• **<f12> h**
		(**markdown-insert-header-dwim** &optional ARG SETEXT)	Insert or replace header markup.
			• The level and type of the header are determined automatically by the type and level of the previous header, unless a prefix argument is given via ARG.
			• With a numeric prefix valued 1 to 6, insert a header of the given level, with the type being determined automatically (note that only level 1 or 2 setext headers are possible).
			  • With a C-u prefix (i.e., when ARG is (4)), promote the heading by one level.
			  • With two C-u prefixes (i.e., when ARG is (16)), demote the heading by one level.
			  • When SETEXT is non-nil, prefer setext-style headers when possible (levels one and two).
			• When there is an active region, use it for the header text. When the point is at an existing header, change the type and level according to the rules above. Otherwise, if the line is not empty, create a header using the text on the current line as the header text.
			• Finally, if the point is on a blank line, insert empty header markup (atx) or prompt for text (setext).
			• See '**markdown-insert-header**' for more details about how the header text is determined.

Insert/replace Header. Prefer setext.	• **C-c C-s H** • **C-c C-t H** • **<f12> H**	(**markdown-insert-header-setext-dwim** &optional ARG)	Insert or replace header markup, with preference for setext. • See '**markdown-insert-header-dwim**' for details, including how ARG is handled.
Insert/replace horizontal rule	• **C-c C-s -** • **<f12> -**	(**markdown-insert-hr** ARG)	Insert or replace a horizontal rule. • By default, use the first element of 'markdown-hr-strings'. • When ARG is non-nil, as when given a prefix, select a different element as follows. • When prefixed with C-u, use the last element of 'markdown-hr-strings' instead. • When prefixed with an integer from 1 to the length of 'markdown-hr-strings', use the element in that position instead.
Insert setext-style level-1 header	• **C-c C-s !** • **C-c C-t !** • **C-c C-t t** • **<f12> !**	(**markdown-insert-header-setext-1**)	Insert a setext-style (underlined) first-level header. See '**markdown-insert-header**'.
Insert setext-style level-2 header	• **C-c C-s @** • **C-c C-t @** • **C-c C-t s** • **<f12> @**	(**markdown-insert-header-setext-2**)	Insert a setext-style (underlined) second-level header. See '**markdown-insert-header**'.
Insert level-1 atx header	• **C-c C-s 1** • **C-c C-t 1** • **<f12> 1**	(**markdown-insert-header-atx-1**)	Insert a first level atx-style (hash mark) header. See '**markdown-insert-header**'.
Insert level-2 atx header	• **C-c C-s 2** • **C-c C-t 2** • **<f12> 2**	(**markdown-insert-header-atx-2**)	Insert a level two atx-style (hash mark) header. See '**markdown-insert-header**'.
Insert level-3 atx header	• **C-c C-s 3** • **C-c C-t 3** • **<f12> 3**	(**markdown-insert-header-atx-3**)	Insert a level three atx-style (hash mark) header. See '**markdown-insert-header**'.

Description	Keystroke	Function	Note
Insert level-4 atx header	• `C-c C-s 4` • `C-c C-t 4` • `<f12> 4`	(**markdown-insert-header-atx-4**)	Insert a level four atx-style (hash mark) header. See '**markdown-insert-header**'.
Insert level-5 atx header	• `C-c C-s 5` • `C-c C-t 5` • `<f12> 5`	(**markdown-insert-header-atx-5**)	Insert a level five atx-style (hash mark) header. See '**markdown-insert-header**'.
Insert level-6 atx header	• `C-c C-s 6` • `C-c C-t 6` • `<f12> 6`	(**markdown-insert-header-atx-6**)	Insert a level six atx-style (hash mark) header. See '**markdown-insert-header**'.
Itemize all previous lines same indention level	• `<f12> M--` • `<M-f12> M--`	(**pel-itemize-lines** &optional ITEM-PREFIX-STRING)	Prepend each of the previous lines with a ITEM-PREFIX-STRING that is "- " by default. ☝ When writing a list of items, instead of manually typing the "- " prefix on each line, type each line without them and then use this command to itemize each of the lines above the current one. • It indents all lines above the current line that are at the same indentation level as the current position.
Modify tables	The following commands provide support for tables. Note that there's several general purpose commands (described in the sections above) that operate on table as well: • Insert a table: `C-c C-s t` • Move 1 cell backward: `S-<tab>` • Move a column to the right: `C-c C-x r` • Move a column to the left: `C-c C-x l` • Move current row up: `C-c C-x u` • Move current row down: `C-c C-x d`		

Convert region into a table	`C-c C-c |`	(**markdown-table-convert-region** BEGIN END &optional SEPARATOR)	Convert region from BEGIN to END to table with SEPARATOR.
			• If every line contains at least one TAB character, the function assumes that the material is tab separated (TSV). If every line contains a comma, comma-separated values (CSV) are assumed. If not, lines are split at whitespace into cells.
			• You can use a prefix argument to force a specific separator:
			  • **C-u** once forces CSV,
			  • **C-u** twice forces TAB,
			  • **C-u** three times will prompt for a regular expression to match the separator,
			  • a numeric argument N indicates that at least N consecutive spaces, or alternatively a TAB should be used as the separator.

Transpose table at point	`C-c C-c t`	(**markdown-table-transpose**)	Transpose table at point. • Horizontal separator lines will be eliminated.
Sort table lines	`C-c C-c ^`	(**markdown-table-sort-lines** &optional SORTING-TYPE)	Sort table lines according to the column at point. • The position of point indicates the column to be used for sorting, and the range of lines is the range between the nearest horizontal separator lines, or the entire table of no such lines exist. If point is before the first column, user will be prompted for the sorting column. If there is an active region, the mark specifies the first line and the sorting column, while point should be in the last line to be included into the sorting. • The command then prompts for the sorting type which can be alphabetically or numerically. Sorting in reverse order is also possible. • If SORTING-TYPE is specified when this function is called from a Lisp program, no prompting will take place. SORTING-TYPE must be a character, any of (?a ?A ?n ?N) where the capital letters indicate that sorting should be done in reverse order.
Table of Contents	Use the following commands to generate a table of contents inside the document. 📦 These require the **markdown-toc** external package 🔄 activated by **pel-use-markdown-toc** user-option. ☝ If you want to automatically refresh the table of content on buffer save, set the **pel-use-markdown-toc** user-option to 'update-toc-on-save.		
Insert Generated Table of Content	• `<f12> M-t M-t` • `<M-f12> M-t M-t`	(**markdown-toc-generate-toc** &optional REPLACE-TOC-P)	Generate a TOC for markdown file at current point. • Deletes any previous TOC. • If called interactively with prefix arg REPLACE-TOC-P, replaces previous TOC.
Refresh Table of Contents	• `<f12> M-t M-r` • `<M-f12> M-t M-r`	(**markdown-toc-generate-or-refresh-toc**)	Generate a TOC for markdown file at current point or refreshes an already generated TOC. • Same as `C-u <f12> M-t`
Detele Table of Contents	• `<f12> M-t M-d` • `<M-f12> M-t M-d`	(**markdown-toc-delete-toc**)	Deletes a previously generated TOC.
Navigate to the TOC linked header	• `<f12> M-t M-f` • `<M-f12> M-t M-f`	(**markdown-toc-follow-link-at-point**)	On a given toc link, navigate to the current markdown header. • If the toc is misindented (according to **markdown-toc-indentation-space** user-option) or if not on a toc link, this does nothing.
Other			
Kill thing at point (without markup)	`C-c C-k`	(**markdown-kill-thing-at-point**)	Kill thing at point and add important text, without markup, to kill ring. • Possible things to kill include (roughly in order of precedence): • inline code, headers, horizontal rules, links (add link text to kill ring), images (add alt text to kill ring), angle uri, email addresses, bold, italics, reference definition (add URI to kill ring), footnote markers and text (kill both marker and text, add text to kill ring), and list items.
Context sensitive return	`RET`	(**markdown-enter-key**)	Handle RET depending on the context. • If the point is at a table, move to the next row. Otherwise, indent according to value of 'markdown-indent-on-enter'. When it is nil, simply call 'newline'. Otherwise, indent the next line following RET using 'markdown-indent-line'. Furthermore, when it is set to 'indent-and-new-item and the point is in a list item, start a new item with the same indentation. If the point is in an empty list item, remove it (so that pressing RET twice when in a list simply adds a blank line).
Modify Indentation and levels	The following commands move markup elements.		
Move *thing* at point up: • **list item** • **table row** • **current heading**	• `C-c <up>` • `C-c C-x u`	(**markdown-move-up**)	Move thing at point up. • When in a list item, call '**markdown-move-list-item-up**'. • When in a table, call '**markdown-table-move-row-up**'. • Otherwise, move the current heading subtree up with 'markdown-move-subtree-up'.
Move *thing* at point down: • **list item** • **table row** • **current heading**	• `C-c <down>` • `C-c C-x d`	(**markdown-move-down**)	Move thing at point down. • When in a list item, call '**markdown-move-list-item-down**'. • Otherwise, move the current heading subtree up with '**markdown-move-subtree-down**'.
Indent region	`C-c >`	(**markdown-indent-region** BEG END ARG)	Indent the region from BEG to END using some heuristics. • When ARG is non-nil, outdent the region instead. • See '**markdown-indent-line**' and 'markdown-indent-line'.
Outdent region	`C-c <`	(**markdown-outdent-region** BEG END)	Call 'markdown-indent-region' on region from BEG to END with prefix.

Description	Keystroke	Function	Note
Outdent or delete	`DEL`	(**markdown-outdent-or-delete** ARG)	Handle BACKSPACE by cycling through indentation points. • When BACKSPACE is pressed, if there is only whitespace before the current point, then outdent the line one level. Otherwise, do normal delete by repeating 'backward-delete-char-untabify' ARG times.
Promote or move element left	• `C-c C--` • `C-c C-x l` • `C-c <left>`	(**markdown-promote**)	Promote or move element at point to the left. • Depending on the context, this function will promote a heading or list item at the point, move a table column to the left, or cycle markup. ⚠ `C-c <left>` can be shadowed by winner-undo.
Demote or move element right	• `C-c C-=` • `C-c C-x r` • `C-c <right>`	(**markdown-demote**)	Demote or move element at point to the right. • Depending on the context, this function will demote a heading or list item at the point, move a table column to the right, or cycle or remove markup. ⚠ `C-c <right>` can be shadowed by winner-redo.
Navigation See also: ⨌ Navigation	Use one of the following commands to navigate inside mark-down buffer. ☛ As described in ➢**Legend** , the key bindings shown in `orange` are not available in terminal mode, `green` ones are provided by PEL.		
Move to next inline/ reference/link	`M-n`	(**markdown-next-link**)	Jump to next inline, reference, or wiki link. • If successful, return point. Otherwise, return nil. • See 'markdown-wiki-link-p' and 'markdown-previous-wiki-link'.
Move to previous inline/ reference/link	`M-p`	(**markdown-previous-link**)	Jump to previous wiki link. If successful, return point. Otherwise, return nil. See 'markdown-wiki-link-p' and 'markdown-next-wiki-link'.
Move to next end of block	• `C-M-}` • `<f12> }`	(**markdown-forward-block** &optional ARG)	Move forward to the next end of a Markdown block. • Moves across complete code blocks, list items, and blockquotes, but otherwise stops at blank lines, headers, and horizontal rules. • With argument ARG, do it ARG times; a negative argument ARG = -N means move backward N blocks.
Move to start of current block	• `C-M-{` • `<f12> {`	(**markdown-backward-block** &optional ARG)	Move the point to the start of the current Markdown block. • Moves across complete code blocks, list items, and blockquotes, but otherwise stops at blank lines, headers, and horizontal rules. • With argument ARG, do it ARG times; a negative argument ARG = -N means move forward N blocks.
Move to the end of the current or next section	`C-M-e` • `<f12> <right>`	(**end-of-defun** &optional ARG)	Move point to the end of the current section
Move to the next section title	`<f12> <down>`	(**pel-beginning-of-next-defun** &optional SILENT DONT-PUSH-MARK)	Move point forward to the beginning of the next section title. • Push mark. Jump back with `M-`` or `<f6><f6>`
Move to the title of the current or previous section	`C-M-a` • `<f12> <up>` • `<f12< <left>`	(**beginning-of-defun** &optional ARG)	Move point to the header (title) of the current or previous section.
Move to next list item/ header	`C-c C-n`	(**markdown-outline-next**)	Move to next list item, when in a list, or next visible heading.
Move to previous list item/header	`C-c C-p`	(**markdown-outline-previous**)	Move to previous list item, when in a list, or previous visible heading.
Move to next list item/ heading same level	`C-c C-f`	(**markdown-outline-next-same-level**)	Move to next list item or heading of same level
Move to previous list item/heading same level	`C-c C-b`	(**markdown-outline-previous-same-level**)	Move to previous list item or heading of same level.
Move to previous list item/heading	`C-c C-u`	(**markdown-outline-up**)	Move to previous list item, when in a list, or previous heading.
Move to the start of current paragraph	`M-{`	(**markdown-backward-paragraph** &optional ARG)	Move the point to the start of the current paragraph. • With argument ARG, do it ARG times; • a negative argument ARG = -N means move forward N blocks.
Move to the next end of paragraph	`M-}`	(**markdown-forward-paragraph** &optional ARG)	Move forward to the next end of a paragraph. • With argument ARG, do it ARG times; • a negative argument ARG = -N means move backward N blocks.
Jump between links and definitions, footnote marker and name	`C-c C-d`	(**markdown-do**)	Do something sensible based on context at point. • Jumps between reference links and definitions; between footnote markers and footnote text.
Mark an Area	The following commands mark a specific region of text.		
Mark current paragraph	`M-h`	(**markdown-mark-paragraph**)	Put mark at end of this block, point at beginning. • The block marked is the one that contains point or follows point. • Interactively, if this command is repeated or (in Transient Mark mode) if the mark is active, it marks the next block after the ones already marked.
Mark current subtree	`C-c C-M-h`	(**markdown-mark-subtree**)	Mark the current subtree. This puts point at the start of the current subtree, and mark at the end.
Mark current block	`C-c M-h`	(**markdown-mark-block**)	Put mark at end of this block, point at beginning. • The block marked is the one that contains point or follows point. • Interactively, if this command is repeated or (in Transient Mark mode) if the mark is active, it marks the next block after the ones already marked.
Comments	Global comment commands can be used in markdown buffers to comment or un-comment lines and regions.		
Comment See also: ⨌ Comments	`M-;`	(**comment-dwim** ARG)	Comment line or region. • Uncomment line or region if it's already commented.
Insert commented line See also: • ⨌ Comments • ⨌ Filling/Justification	• `<f11> i l` • `<f6> l`	(**pel-insert-line** &optional LINELEN)	Insert a (commented) line before/at current line. • If point is at the beginning of the line insert it there. • If point is in the middle of a line, move point at beginning of line before inserting it. • The number of dash characters of the line is specified by LINELEN: • If LINELEN is not specified the buffer's fill-column value is used. ☛ fill-column is customizable and can be used as a file or directory variable.
Control Visibility	The following commands can be used to collapse or expand the view of markdown sections, operating like an outline document, similar to org-mode.		
Cycle visibility, indent, insert tab	`<tab>`	(**markdown-cycle** &optional ARG)	Visibility cycling for Markdown mode. • This function is called with a '`C-u`' or if ARG is t, perform global visibility cycling. • If the point is at an atx-style header, cycle visibility of the corresponding subtree. • Otherwise, indent the current line or insert a tab, as appropriate, by calling 'indent-for-tab-command'.
• **In table: move 1 cell backward** • **otherwise: cycle global visibility**	`S-<tab>`	(**markdown-shifttab**)	Handle S-TAB keybinding based on context. • When in a table, move backward one cell. • Otherwise, cycle global heading visibility.

Description	Keystroke	Function	Note
Narrowing and Indirect buffer	The following commands narrow specific areas of text. To widen them back use the widen command, mapped to **C-x C-n w**. See: ∑ **Narrowing** for more information.		
Narrow current block	**C-x n b**	(**markdown-narrow-to-block**)	Make text outside current block invisible. • The current block is the one that contains point or follows point
Narrow current subtree	**C-x n s**	(**markdown-narrow-to-subtree**)	Narrow buffer to the current subtree.
Edit block inside an indirect buffer	**C-c '**	(**markdown-edit-code-block**)	Edit Markdown code block in an indirect buffer using the mode of the target programming language. It opens the indirect buffer and use the major-mode associated with that programming language. 📦 Requires and installs if allowed, the **edit-indirect** external package. PEL installs it when **pel-use-edit-indirect** is turned on (set to **t**).
Consistency Operations	The following commands help complete your markdown document, making sure that all the makeup is valid.		
Complete markup at point	**C-c C-]**	(**markdown-complete**)	Complete markup of object near point or in region when active. Handle all objects in 'markdown-complete-alist', in order. See '**markdown-complete-at-point**' and '**markdown-complete-region**'.
Complete the markup of all objects in current buffer	**C-c C-c]**	(**markdown-complete-buffer**)	Complete markup for all objects in the current buffer.
Show all undefined references	**C-c C-c c**	(**markdown-check-refs** &optional SILENT)	Show all undefined Markdown references in current 'markdown-mode' buffer. • Links which have empty reference definitions are considered to be defined. • If SILENT is non-nil, do not message anything when no such references found.
Update numbering of ordered lists	**C-c C-c n**	(**markdown-cleanup-list-numbers**)	Update the numbering of ordered lists.
Show all unused references	**C-c C-c u**	(**markdown-unused-refs** &optional SILENT)	Show all unused Markdown references in current 'markdown-mode' buffer. • If SILENT is non-nil, do not message anything when no such references found.

Render markdown files to HTML and open then inside Emacs or inside a browser.	The **markdown-mode** external package provides the ability to convert markdown files to HTML and open the result inside Emacs or inside the system's default browser. Other external package provide additional rendering features.
	• Several Markdown processor command line utilities are available to render markdown files into HTML.
	• The following Markdown processors are available and must be installed on your system separately:
	  • **markdown.pl,** the original Perl script from **John Gruber Daring Fireball Markdown** page.
	  • **markdown_py,** the Python implementation of markdown.pl
	  • **MultiMarkDown**
	  • **Pandoc**
	• Once you have selected the tool to use, you may have to modify the **markdown-command** user-option to identify it.
	  • The default value is: `markdown`.
	  • With PEL, you can open the markdown-mode customization group quickly with **<f12> <f3> 1** .		

Toggle native markup preview with EWW	• **C-c C-c l** • **<f12> M-p l**	(**markdown-live-preview-mode** &optional ARG)	Toggle native previewing on save for a specific markdown file. • Renders the file using Emacs built-in **eww** inside a separate Emacs buffer. • ☝ This works in both terminal and graphics mode!
➽ browser preview	**C-c C-c p**	(**markdown-preview** &optional OUTPUT-BUFFER-NAME)	Run 'markdown-command' on the current buffer and view output in browser. • When OUTPUT-BUFFER-NAME is given, insert the output in the buffer with that name.
➽ basename.html	**C-c C-c e**	(**markdown-export** &optional OUTPUT-FILE)	Run Markdown on the current buffer, save to file, and return the filename. • If OUTPUT-FILE is given, use that as the filename. Otherwise, use the filename generated by 'markdown-export-file-name', which will be constructed using the current filename, but with the extension removed and replaced with .html.
➽ basename.html ➽ browser preview	**C-c C-c v**	(**markdown-export-and-preview**)	Export to XHTML using 'markdown-export' and browse the resulting file.
Toggle automatic HTML rendering using local HTTP server and browser	• **<f12> M-p p** • **<M-f12> M-p p**	(**markdown-preview-mode** &optional ARG)	Toggle Markdown preview mode, where the rendering is shown inside the preferred browser. Uses a websocket server and web sockets for the implementation. • Automatically refreshes when buffer modifications are saved. • The markdown-preview customization group defines various elements such as the preview delay, the file name, style, etc… • With PEL access this quickly using the **<f12> <f3>** key.
		📦 Requires the **markdown-preview-mode** external package activated by **pel-use-markdown-preview-mode** user-option.	
Activate Github-compliant HTML rendering	• **<f12> M-p g** • **<M-f12> M-p g**	(**grip-mode** &optional ARG)	Live Markdown preview with grip. • Launches a grip process that extract style files from GitHub and generates the HTML rendering then displayed in the default browser.
		📦 Requires the **grip-mode** external package activated by **pel-use-grip-mode** user-option. • Also requires Python and the Python **grip package,** you can install with `pip install grip`	
Active Rendering	The following external packages provide browser-based rendering that is updated automatically as you modify the markdown buffer.		
Toggle automatic HTML rendering using local HTTP server and browser	• **<f12> M-p i** • **<M-f12> M-p i**	(**impatient-showdown-mode &optional ARG**)	Toggle Minor mode 'impatient-showdown-mode'. • When the mode is active it launches a local HTTP server and renders the markdown buffer in HTML in the default browser updating it when the buffer changes. • The browser automatically refreshes itself as you update the markdown buffer. • The **impatient-showdown** customization group defines various elements of the rendered page, such as background colors and used showdown javascript URLs. • With PEL access this quickly using the **<f12> <f3>** key.
		📦 Requires the **impatient-showdown** external package activated by **pel-use-impatient-showdown** user-option.	
vmd-mode: automatic HTML rendering in default browser with completion support for emojis	• **<f12> M-p v** • **<M-f12> M-p v**	(**vmd-mode** &optional ARG)	Activates vmd-mode to live-render the markdown buffer in the default browser. • ☝ If company-mode is available it provides completion for emojis.
		📦 Requires the **vmd-mode** external package activated by **pel-use-vmd-mode** user-option. • Also requires **Node.js** and **vmd** javascript package. They must be installed separately (PEL does not install these).	
Other render help	The following commands run markdown on the current buffer, creating HTML content • As HTML markup inside a *markdown-output* buffer or in the kill ring allowing you to manually create HTML content from the markup.		
➽ *markdown-output* buffer in other window	**C-c C-c m**	(**markdown-other-window** &optional OUTPUT-BUFFER-NAME)	Run 'markdown-command' on current buffer and display the resulting HTML markup inside the other window. • With OUTPUT-BUFFER-NAME: insert the output in the buffer with that name.
Open file for current buffer using command specified by markdown-open-command	**C-c C-c o**	(**markdown-open**)	Open file for the current buffer with **'markdown-open-command'**. • Use this Emacs command to provide specialized handling of your markdown files, for example extra processing. • By default the **markdown-open-command** user-option is not set.
Markdown and store result in kill ring	**C-c C-c w**	(**markdown-kill-ring-save**)	Run Markdown on file and store output in the kill ring.

Description	Keystroke	Function	Note
Toggle commands	The following commands update the way the markup rendering is done in the current buffer.		
Hide/display markup	`C-c C-x C-m`	(**markdown-toggle-markup-hiding** &optional ARG)	Toggle the display or hiding of markup. • With a prefix argument ARG, enable markup hiding if ARG is positive, and disable it otherwise. • See 'markdown-hide-markup' for additional details.
Hide/display URLs	`C-c C-x C-l`	(**markdown-toggle-url-hiding** &optional ARG)	Toggle the display or hiding of URLs. • With a prefix argument ARG, enable URL hiding if ARG is positive, and disable it otherwise.
Hide/display LaTeX math expressions	`C-c C-x C-e`	(**markdown-toggle-math** &optional ARG)	Toggle support for inline and display LaTeX math expressions. • With a prefix argument ARG, enable math mode if ARG is positive, and disable it otherwise. If called from Lisp, enable the mode if ARG is omitted or nil.
Hide/display code block fortification	`C-c C-x C-f`	(**markdown-toggle-fontify-code-blocks-natively** &optional ARG)	Toggle the native fontification of code blocks. • With a prefix argument ARG, enable if ARG is positive, and disable otherwise.
Hide/display GFM checkbox at point	`C-c C-x C-x`	(**markdown-toggle-gfm-checkbox**)	Toggle GFM checkbox at point. • Returns the resulting status as a string, either "`[x]`" or "`[ ]`". • Returns nil if there is no task list item at the point.
Hide/display inline image overlays	`C-c C-x <tab>`	(**markdown-toggle-inline-images**)	Toggle inline image overlays in the buffer.
Slide Show with remark-mode	You can turn your markdown file into a slide presentation using the local default browser. • While remark-mode is active several keys are rebound. The remark-mode commands and their bindings are listed below. • Note that remark-mode is a major-mode derived from markdown-mode. • To return to markdown-mode, execute `M-x markdown-mode` and explicitly kill the *remark browser* buffer. 📦 Requires the **remark-mode** external package 🔄 activated by **pel-use-remark-mode** user-option. • Also requires **Node.js** and **remark** javascript package. They must be installed separately (PEL does not install these). • This also interact with Google browser.		
Turn remark-mode on	• `<f12> M-p r` • `<M-f12> M-p r`	(**remark-mode**)	Turn-on remark-mode. • To activate the slide-show you must first connect: execute **remark-connect-browser**, bound to `C-c C-s c` while remark-mode is active.
Connect to the browser and show the slide	`C-c C-s c`	(**remark-connect-browser**)	Connect the <slideshow>.remark file to the in browser remark slideshow.
Skip to next slide	• `M-n` • `M-<down>`	(**remark-next-slide** &optional ARG)	Skip to next slide. • Optional argument ARG skips to next incremental slide.
Skip to previous slide	• `M-p` • `M-<up>`	(**remark-prev-slide** &optional ARG)	Skip to prev slide. Optional argument ARG skips to previous incremental slide.
Move the slide past the next slide	`M-S-<down>`	(**remark-move-slide-next**)	Move the slide past the next slide.
Move the slide in from of the previous slide	`M-S-<up>`	(**remark-move-slide-prev**)	Move the slide in front of the previous slide.
Toggle presenter mode	`C-c C-s p`	(**remark-toggle-presenter**)	Toggle remark presenter mode. Reloads the slideshow.
Create a new slide	`C-c C-s s`	(**remark-new-slide**)	Create new slide.
Create new incremental slide	`C-c C-s i`	(**remark-new-incremental-slide**)	Create new incremental slide.
Kill current slide	`C-c C-s k`	(**remark-kill-slide**)	Kill the current slide. ⚠️ This empties the current file!!
Create note for current slide	`C-c C-s n`	(**remark-create-note**)	Create note for slide.
Kill current slide buffer	`C-c C-s d`	(**remark-kill-browser**)	Kill current remark browser buffer unconditionally.

Markdown & Emacs — References

Description & URL	Notes
Markdown Markup	There are several *flavors* of Markdown, unfortunately. This is a relative newcomer to the lightweight markup and it already suffers from noticeable entropy. It could be argued that it's due to its popularity but that it unfortunately introduces complexity that would have best been avoided by using a centralized definition of the markup. It could also be argued that Markdown is not extensible enough, the way reStructuredText is, which led to this excessive diversity and artificial complexity. • See the Wikipedia Markdown Standardization note on this topic.
Markdown @ Wikipedia	
Markdown Home Page	John Gruber page where you can get a copy of the Perl-based markdown.pl script.
Markdown Home Page - Markdown syntax	The description of the syntax of the official markdown. An interesting read is Richard O'Keefe's annoyance at markup.
CommonMark Spec	The CommonMark specification. It was used as the *base* for Github and several other site despite the fact that the standard never reached version 1.0 and has remaining issues as of April 2021.
GitHub-flavored Markdown Spec	As of April 2021, this has 5 extensions over the CommonMark Spec: tables, task list items, strikethrough, auto link and disallowed raw HTML.
Markdown & reStructuredText @ GitHub	Comparison of markdown and reStructuredText
Markdown markup processors	
markdown.pl 1.0.1	From John Gruber Daring Fireball Markdown page. Implementation is written in Perl and requires Perl 5.6.0 or later.
MultiMarkDown • MultiMarkDown @ Wikipedia • MultiMarkDown @ GitHub	MultiMarkDow is an extension of Markdown, implemented in C.
Pandoc	Pandoc is a very useful program that can convert several formats into other formats. It supports Markdown, CommonMark and GitHub-flavored Markdown formats. This is implemented in Haskell.
Emacs Markdown support	Several Emacs packages support Markdown. You will need the first one: markdown-mode and possibly complement it with others.
Markdown Mode for Emacs - User Manual	Jason Blevin's markdown-mode package user manual • Project : Emacs Markdown Mode @ Github