Comments

Action	Key binding	Command	Notes
Comment Commands	🔨 This is an early version of this page. More information is required on this topic.		
Insert, realign, comment/ uncomment region	`M-;`	(**comment-dwim** ARG)	Insert or realign comment on current line. • If the region is active, comment or uncomment the region instead. • **C-u M-;** executes comment-kill
Comment/uncomment current line	• `C-x C-;` • `<f11> ; l`	(**comment-line** N)	Comment/uncomment current line (the comment is at the beginning of the line). Since C-; is not an ASCII Control character, C-x C-; does not work in Terminal. Use the alternate keystroke.
Comment the region as a box	`<f11> ; B`	(**comment-box** BEG END &optional ARG)	Comment the marked region in a comment box.
Kill comment on current line	`<f11> ; k`	(**comment-kill** ARG)	Kill the complete comment text on the line (whether it is at the beginning of the line or after some code) and remember in kill ring. • With numeric argument, kill comment on as many lines staring with the current one. • For one line this can also be executed with **C-u M-;**
Set comment column to current column	`C-x ;`	(**comment-set-column** ARG)	Set the *comment-column* variable (shown in the ruler as '#'). • With no argument: set to the current column.
Set comment column to previous comment and adjust/ insert a comment	`C-u C-x ;`	(**comment-set-column** 1)	• With any positive argument set comment-column to the same column as the previous comment and align/create a comment on this line at that column. • With - as arg, kill any comment on current line.
New line & aligned comment	• `M-j` • `C-M-j`	Mapped to: • C, C++ modes: (**c-indent-new-comment-line** &optional SOFT ALLOW-AUTO-FILL) • Buffer Menu, text mode, python mode: (**comment-dwim** ARG)	When M-j is typed, on a line that contains a comment, a new line is entered and a comment is placed in the same column, with point placed after the same number of spaces.
Set comment start string	`<f11> ; b`	(**pel-comment-start**)	Show and set comment start string for current mode. • Controlled by variable: ***comment-start***
Set comment end string	`<f11> ; e`	(**pel-comment-end**)	Show and set comment end string for current mode. • Controlled by variable: ***comment-end*** • By default it's "". It must be "" if the comment ends at the end of the line.
Set comment continue (middle) string	`<f11> ; m`	(**pel-comment-middle**)	Show and set comment continue/middle string for current mode. • Controlled by variable: ***comment-continue*** • By default it is *nil*. Can be set to any string. • This is used for multi-line comments
Set Fill Column	`C-x f`	(**set-fill-column** ARG)	When no prefix value: prompts for column. • If with **C-u** prefix: use current column. • If with prefix value: use that value. ☞ Note: this command and other fill specific commands are described in the **Fill/Justification** table.
Toggle auto fill inside comments only	`<f11> t f C`	(**pel-auto-fill-only-comments**)	Toggle the ***comment-auto-fill-only-comments*** variable to control whether filling is done everywhere or only inside the comments. ☞ Note: this command and other fill specific commands are described in the **Fill/Justification** table.
See all comment control variables	• `C-h v comment- <tab><tab>` • `<f1> v comment- <tab><tab>`		Lists all variables that have a name that starts with the word "comment-". 👉 See the Help table for more information on getting help from within Emacs.

Comments — References

GNU Emacs Manual - Manipulating Comments		
GNU Emacs Lisp — Tips on Writing Comments	Emacs Lisp comments conventions/guidelines.	