↖ Topic Index

Filling and Justification

Description	Keystroke	Function	Note
Filling, Justification, Position (and navigation) Information ○ Filling Text ○ Text Justification • Center text horizontally Last updated on:	Under some conditions, and for editing some type of files, it may be useful to see the horizontal ruler above the window of a buffer. • The ruler-mode provides this functionality. • You can also use a vertical line highlighting a specific column: see **Highlight** Emacs filling uses several variables identifying the column where text must wrap. The main variable is the **fill-column** variable. But it's not the only one. • For buffer used to show and edit Lisp or Emacs Lisp code, the **emacs-lisp-docstring-fill-column** controls text wrapping inside Lisp docstrings. • The values of the relevant fill wrapping controlling column is shown by the **pel-show-fill-colum** command, shown below. 2025-11-09		
Open this PDF file. See also: **Help/Info**	• `<f11> t f <f1>` • `<f11> t j <f1>`	(**pel-help-pdf** &optional OPEN-WEB-PAGE)	Open the **Filling/Justification** 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 Emacs file and justification control See also: **Customize**	• `<f11> t f <f3>` • `<f11> t j <f3>`	(**pel-customize-library** &optional OTHER-WINDOW)	Customize Emacs fill and justification control.
Show the ruler See also: **Highlight** Example of the ruler in terminal mode showing the ruler-mode markers ➨➨ ... and the point (cursor) ➨➨	`<f11> b -`	(**ruler-mode** &optional ARG)	Toggle display of ruler in header line (Ruler mode) in window(s) of current buffer. With a prefix argument, enable Ruler mode if ARG is positive, disable it otherwise.

	• The ruler markers are:
	• `|` Current cursor position `!` Marks every 5 positions.
	• `#` Current *comment-column* value, set by **C-x ;** It controls where the cursor goes when creating a new comment on a line.
	• `¶` Current *fill-column* value, set by **C-x f**, controls where the cursor wraps.
	• `G` Current *goal-column* value, controlled by `C-x C-n`.
	The horizontal position when going up and down lines via `C-p` and `C-n`. If nil, its G symbol will not show on the ruler and it will not have any impact on the navigation commands. See the information about the set-goal-column command in the navigation table.
	
	`File Edit Options Buffers Tools Defs Lisp-Interaction Help`
	`0....!...1....!....2....!....3....!....#....!....5....!....6....!....7....!..¶.8....!....9....!...10...`
	`.`		

Show the values of the fill columns	`<f11> t f ?`	(**pel-show-fill-columns**)	Display values of relevant user-options and variables controlling text fill and justification inside a special *pel-fill-info* buffer. In that buffer the user-options and variable names are buttons providing access to more information.
Toggle ruler line on fill column See also: • **Highlight** • **Spell Checking**	• `<f11> h \` • `<f11> <f5> \` • `<f11> 8`	(**fci-mode** &optional ARG)	Toggle a vertical ruler line shown at the buffer's fill-column. • For **Emacs < 27.1:** 📦 requires the **fill-column-indicator external package** activated by PEL **use-fill-column-indicator** user option set to **t**. • ⚠️ **fci-mode** interferes with pop-up menu displays in terminal-mode, at least with the one used by **flyspell-correct-word-before-point:** the menu lines become all jagged, they do not line up vertically. The problem does not affect Emacs running in graphics mode.
		(**display-fill-column-indicator-mode** &optional ARG)	• For **Emacs >= 27.1**: the built-in **display-fill-column-indicator-mode** is used.
Filling Text See also: **Text Modes**	• For fill and refill mode to work properly on text: • Sentences must be separated by **2** spaces (unless *sentence-end-double-space* is t): use `<f11> t m s` to change this. • Paragraphs must be separated by 1 empty line. • Identify major modes where the auto-fill-mode is automatically activated in the **pel-modes-activating-auto-fill-mode** user-option. • Access the relevant customization buffer with `<f11> t <f2>`		
Toggle end of sentence space count	`<f11> t m s`	(**pel-toggle-sentence-end**)	Toggle the value of sentence-end-double-space variable to change the number of required spaces after a period to identify at the end of a sentence in text: from 2 (the default) to 1 and vice-versa.
Set Fill Column	• `C-x f` • `<f11> t f c`	(**set-fill-column** ARG)	When no prefix value: prompts for column. If with **C-u** prefix: use current column. • If with numeric prefix value: use that value.
Toggle auto-fill mode	• `<f11> t f C` • `<f11> RET`	(**auto-fill-mode** &optional ARG)	Toggle automatic line breaking (Auto Fill mode). • With prefix argument, enable Auto Fill mode if prefix argument is positive, and disable it otherwise. • When Auto Fill mode is enabled, inserting a space at a column beyond 'current-fill-column' automatically breaks the line at a previous space.
Toggle auto-fill inside comments only	`<f11> t f ;`	(**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.
Toggle refill mode	`<f11> t f f`	(**refill-mode** &optional ARG)	Toggle automatic refilling (Refill mode). • With a prefix argument ARG, enable Refill mode if ARG is positive, and disable it otherwise. • Refill mode is a buffer-local minor mode. When enabled, the current paragraph is refilled as you edit. Self-inserting characters only cause refilling if they would cause auto-filling.
Set fill prefix	• `C-x .` • `<f11> t f .`	(**set-fill-prefix**)	Set the fill prefix to the current line up to point. • Filling expects lines to start with the fill prefix and reinserts the fill prefix in each resulting line. • The fill prefix is then inserted before any character typed in the line.
Fill region	`<f11> t f r`	(**fill-region** FROM TO &optional JUSTIFY NOSQUEEZE TO-EOP)	Fill each of the paragraphs in the region. The 'fill-column' variable controls the width. • A prefix arg means justify as well.
Fill region as one paragraph	`<f11> t f q`	(**fill-region-as-paragraph** FROM TO &optional JUSTIFY NOSQUEEZE SQUEEZE-AFTER)	Fill the region as one paragraph. • It removes any paragraph breaks in the region and extra newlines at the end, indents and fills lines between the margins given by the 'current-left-margin' and 'current-fill-column' functions. • In most cases, the variable 'fill-column' controls the width. • It leaves point at the beginning of the line following the paragraph. • Normally performs justification according to the 'current-justification' function, but with a prefix arg, does full justification instead. • If 'sentence-end-double-space' is non-nil, then period followed by one space does not end a sentence, so don't break a line there. You can change this using the (pel-toggle-sentence-end) bound to `<f11> t m s`.
Fill current paragraph • ☝ Refill also properly refill a multi-line comment and strings.	• `M-q` • `<f11> t f p`	(**fill-paragraph** &optional JUSTIFY REGION)	Fill paragraph at or after point. To justify as well: **C-u M-q** • In refill mode, refill is done automatically. In auto-fill mode, it's done on reaching fill column. • The `M-q` key is also used in major modes for programming languages but is mapped to a different command for several of them.
Fill Lisp code section	`<f11> t f l`	(**lisp-fill-paragraph** &optional JUSTIFY)	Like `M-q`, but handle Emacs Lisp comments and docstrings. • If any of the current line is a comment, fill the comment or the paragraph of it that point is in, preserving the comment's indentation and initial semicolons. • Temporarily binds `fill-column' to `emacs-lisp-docstring-fill-column' which is used to identify the fill column for Emacs Lisp docstring.
Fill region, each indentation starts a new "paragraph"	`<f11> t f i`	(**fill-individual-paragraphs** MIN MAX &optional JUSTIFY CITATION-REGEXP)	Fill paragraphs of uniform indentation within the region. • This command divides the region into "paragraphs", treating every change in indentation level or prefix as a paragraph boundary, then fills each paragraph using its indentation level as the fill prefix.
Fill region considering only paragraph-separator lines as starting a new paragraph	`<f11> t f n`	(**fill-nonuniform-paragraphs** MIN MAX &optional JUSTIFYP CITATION-REGEXP)	Fill paragraphs within the region, allowing varying indentation within each. • This command divides the region into "paragraphs", only at paragraph-separator lines, then fills each paragraph using as the fill prefix the smallest indentation of any line in the paragraph.

Description	Keystroke	Function	Note
Text Justification See also: 𝕀 **Enriched Text**	Text justification is normally used when **enriched text support** is activated via the enriched-mode. • The standard Emacs key- sequences listed below are only available in enriched-mode. The PEL key-sequences are always available.		
Toggle Enriched Text Mode See also: 𝕀 **Text Modes**	• `<f11> t e e` • `<f11> t m e`	(**enriched-mode** &optional ARG)	Minor mode for editing text/enriched files. • These are files with embedded formatting information in the MIME standard text/enriched format.
Set justification - full (both)	• `M-j b` • `<f11> t j f`	(**set-justification-full** B E)	Make paragraphs in the region fully justified. • This makes lines flush on both margins by inserting spaces between words. • If the mark is not active, this applies to the current paragraph.
Set justification - left	• `M-j l` • `<f11> t j l`	(**set-justification-left** B E)	Make paragraphs in the region left-justified. • This means they are flush at the left margin and ragged on the right. • If the mark is not active, this applies to the current paragraph.
Set justification - center	• `M-j c` • `M-S` • `<f11> t j c`	(**set-justification-center** B E)	Make paragraphs in the region centered. • If the mark is not active, this applies to the current paragraph.
Set justification - right	• `M-j r` • `<f11> t j r`	(**set-justification-right** B E)	Make paragraphs in the region right-justified. • This means they are flush at the right margin and ragged on the left. • If the mark is not active, this applies to the current paragraph.
Set justification - none (un-justify)	• `M-j u` • `<f11> t j u`	(**set-justification-none** B E)	Disable automatic filling for paragraphs in the region. • If the mark is not active, this applies to the current paragraph.
Centering	The following commands center text horizontally.		
Center paragraph	`M-o M-S`	(**center-paragraph**)	Center each nonblank line in the paragraph at or after point.
Center line	`M-o M-s`	(**center-line** &optional NLINES)	Center the line point is on, within the width specified by 'fill-column'. • It adjusts the indentation so that it equals the distance between the end of the text and 'fill-column'. • With numeric prefix argument NLINES: centres that many lines. ✂ The ⌘-- key is normally bound here in graphics mode. PEL re-assigns it.

Filling and Justification — References

Operation	Keystroke
Text Filling	
GNU Emacs Lisp - Text - Filling	
GNU Emacs Lisp - Text - Auto Filling	
GNU Emacs Lisp - Text - Adaptive Filling	
GNU Emacs Lisp - Text - Margins	
Emacs hard wrap lines	
Emacs wiki - Ruler Mode	
Emacs wiki - refilling mode	
Emacs wiki - Auto Fill Mode	
How to set multiline comments - StackOverflow	
Line Wraps	
Truncate Lines @ EmacsWiki	Describes how to set/disable line wrapping globally. Also explains how to control it for a given mode.
Line Wrap @ EmacsWiki	Describes various modes: AutoFillMode, LongLines, VisualLineMode
longlines-mode @ emacshorror	Discuss difference between longlines-mode and visual-line-mode
ErgoEmacs: Show Cursor Position	Xah Lee's ErgoEmacs web site has a picture of the ruler at the top of this page. I did not see one inside Emacs manuals (yet, let me know if you find one).