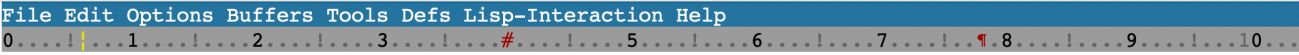







## Filling and Justification

Description	Keystroke	Function	Note
<b>Filling, Justification, Position (and navigation) Information</b> <ul style="list-style-type: none"> <li><a href="#">Filling Text</a></li> <li><a href="#">Text Justification</a></li> <li><a href="#">Center text horizontally</a></li> </ul>	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. <ul style="list-style-type: none"> <li>The ruler-mode provides this functionality.</li> <li>You can also use a vertical line highlighting a specific column: see <a href="#">↗ Highlight</a></li> </ul> Emacs filling uses several variables identifying the column where text must wrap. The main variable is the <b>‘fill-column’</b> 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 <b>pel-show-fill-colum</b> command, shown below.		
Last updated on:	2025-11-05		
<b>Open this PDF file.</b> See also: <a href="#">↗ Help/Info</a>	<ul style="list-style-type: none"> <li><b>&lt;f11&gt; t f &lt;f1&gt;</b></li> <li><b>&lt;f11&gt; t j &lt;f1&gt;</b></li> </ul>	(pel-help-pdf &optional OPEN-WEB-PAGE)	Open the <a href="#">↗ Filling/Justification</a> local PDF. If the prefix argument (like <b>C–u</b> or <b>M––</b> ) is used, then it opens the remote GitHub hosted raw PDF instead. If the <b>pel-flip-help-pdf-arg</b> user-option is set it’s the other way around.
<b>Customize Emacs file and justification control</b> See also: <a href="#">↗ Customize</a>	<ul style="list-style-type: none"> <li><b>&lt;f11&gt; t f &lt;f3&gt;</b></li> <li><b>&lt;f11&gt; t j &lt;f3&gt;</b></li> </ul>	(pel-customize-library &optional OTHER-WINDOW)	Customize Emacs fill and justification control.
<b>Show the ruler</b>  See also: <a href="#">↗ Highlight</a>	<b>&lt;f11&gt; b -</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.
Example of the ruler in terminal mode showing the ruler-mode markers <b>➡➡</b>  ... and the point (cursor) <b>➡➡</b>	<ul style="list-style-type: none"> <li>The ruler markers are:               <ul style="list-style-type: none"> <li><b> </b> Current cursor position <b>!</b> Marks every 5 positions.</li> <li><b>#</b> Current <i>comment-column</i> value, set by <b>C–x</b> ; It controls where the cursor goes when creating a new comment on a line.</li> <li><b>¶</b> Current <i>fill-column</i> value, set by <b>C–x f</b>, controls where the cursor wraps.</li> <li><b>G</b> Current <i>goal-column</i> value, controlled by <b>C–x C–n</b>. The horizontal position when going up and down lines via <b>C–p</b> and <b>C–n</b>. 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 <a href="#">set-goal-column command</a> in the navigation table.</li> </ul> </li> </ul>		
			
<b>Show the values of the fill columns</b>	<b>&lt;f11&gt; t f ?</b>	(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.]
<b>Toggle ruler line on fill column</b>  See also: <ul style="list-style-type: none"> <li><a href="#">↗ Highlight</a></li> <li><a href="#">↗ Spell Checking</a></li> </ul>	<ul style="list-style-type: none"> <li><b>&lt;f11&gt; h \</b></li> <li><b>&lt;f11&gt; &lt;f5&gt; \</b></li> <li><b>&lt;f11&gt; 8</b></li> </ul>	<ul style="list-style-type: none"> <li>(fci-mode &amp;optional ARG)</li> </ul>	Toggle a vertical ruler line shown at the buffer’s fill-column. <ul style="list-style-type: none"> <li>For <b>Emacs &lt; 27.1</b>:  requires the <a href="#">fill-column-indicator external package</a>  activated by PEL <b>use-fill-column-indicator</b> user option set to <b>t</b>.               <ul style="list-style-type: none"> <li> <b>fci-mode</b> interferes with pop-up menu displays in terminal-mode, at least with the one used by <b>flyspell-correct-word-before-point</b>: the menu lines become all jagged, they do not line up vertically. The problem does not affect Emacs running in graphics mode.</li> </ul> </li> <li>For <b>Emacs &gt;= 27.1</b>: the built-in <b>display-fill-column-indicator-mode</b> is used.</li> </ul>
<b>Filling Text</b>  See also: <a href="#">↗ Text Modes</a>	<ul style="list-style-type: none"> <li>For fill and refill mode to work properly on text:               <ul style="list-style-type: none"> <li>Sentences must be separated by <b>2</b> spaces (unless <i>sentence-end-double-space</i> is t): use <b>&lt;f11&gt; t m s</b> to change this.</li> <li>Paragraphs must be separated by 1 empty line.</li> </ul> </li> <li>Identify major modes where the auto-fill-mode is automatically activated in the <b>pel-modes-activating-auto-fill-mode</b> user-option.               <ul style="list-style-type: none"> <li>Access the relevant customization buffer with <b>&lt;f11&gt; t &lt;f2&gt;</b></li> </ul> </li> </ul>		
<b>Toggle end of sentence space count</b>	<b>&lt;f11&gt; t m s</b>	(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.
<b>Set Fill Column</b>	<ul style="list-style-type: none"> <li><b>C–x f</b></li> <li><b>&lt;f11&gt; t f c</b></li> </ul>	(set-fill-column ARG)	When no prefix value: prompts for column. If with <b>C–u</b> prefix: use current column. <ul style="list-style-type: none"> <li>If with numeric prefix value: use that value.</li> </ul>
<b>Toggle auto-fill mode</b>	<ul style="list-style-type: none"> <li><b>&lt;f11&gt; t f C</b></li> <li><b>&lt;f11&gt; RET</b></li> </ul>	(auto-fill-mode &optional ARG)	Toggle automatic line breaking (Auto Fill mode). <ul style="list-style-type: none"> <li>With prefix argument, enable Auto Fill mode if prefix argument is positive, and disable it otherwise.</li> <li>When Auto Fill mode is enabled, inserting a space at a column beyond ‘current-fill-column’ automatically breaks the line at a previous space.</li> </ul>
<b>Toggle auto fill inside comments only</b>	<b>&lt;f11&gt; t f ;</b>	(pel-auto-fill-only-comments)	Toggle the <b>comment-auto-fill-only-comments</b> variable to control whether filling is done everywhere or only inside the comments.
<b>Togge refill mode</b>	<b>&lt;f11&gt; t f f</b>	(refill-mode &optional ARG)	Toggle automatic refilling (Refill mode). <ul style="list-style-type: none"> <li>With a prefix argument ARG, enable Refill mode if ARG is positive, and disable it otherwise.</li> <li>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.</li> </ul>
<b>Set fill prefix</b>	<ul style="list-style-type: none"> <li><b>C–x .</b></li> <li><b>&lt;f11&gt; t f .</b></li> </ul>	(set-fill-prefix)	Set the fill prefix to the current line up to point. <ul style="list-style-type: none"> <li>Filling expects lines to start with the fill prefix and reinserts the fill prefix in each resulting line.</li> <li>The fill prefix is then inserted before any character typed in the line.</li> </ul>
<b>Fill Lisp code section</b>	<b>&lt;f11&gt; t f l</b>	(lisp-fill-paragraph &optional JUSTIFY)	Like <b>M–q</b> , but handle Emacs Lisp comments and docstrings. <ul style="list-style-type: none"> <li>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.</li> <li>Temporarily binds ‘fill-column’ to ‘emacs-lisp-docstring-fill-column’ which is used to identify the fill column for Emacs Lisp docstring.</li> </ul>
<b>Fill region</b>	<b>&lt;f11&gt; t f r</b>	(fill-region FROM TO &optional JUSTIFY NOSQUEEZE TO-EOP)	Fill each of the paragraphs in the region. The ‘fill-column’ variable controls the width. <ul style="list-style-type: none"> <li>A prefix arg means justify as well.</li> </ul>
<b>Fill region as one paragraph</b>	<b>&lt;f11&gt; t f q</b>	(fill-region-as-paragraph FROM TO &optional JUSTIFY NOSQUEEZE SQUEEZE-AFTER)	Fill the region as one paragraph. <ul style="list-style-type: none"> <li>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.</li> <li>In most cases, the variable ‘fill-column’ controls the width.</li> <li>It leaves point at the beginning of the line following the paragraph.</li> <li>Normally performs justification according to the ‘current-justification’ function, but with a prefix arg, does full justification instead.</li> <li>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 <b>&lt;f11&gt; t m s</b>.</li> </ul>
<b>Fill current paragraph</b> <ul style="list-style-type: none"> <li> Refill also properly refill a multi-line comment and strings.</li> </ul>	<ul style="list-style-type: none"> <li><b>M–q</b></li> <li><b>&lt;f11&gt; t f p</b></li> </ul>	(fill-paragraph &optional JUSTIFY REGION)	Fill paragraph at or after point. To justify as well: <b>C–u M–q</b> <ul style="list-style-type: none"> <li>In refill mode, refill is done automatically. In auto-fill mode, it’s done on reaching fill column.</li> <li>The <b>M–q</b> key is also used in major modes for programming languages but is mapped to a different command for several of them.</li> </ul>
<b>Fill region, each indentation starts a new “paragraph”</b>	<b>&lt;f11&gt; t f i</b>	(fill-individual-paragraphs MIN MAX &optional JUSTIFY CITATION-REGEXP)	Fill paragraphs of uniform indentation within the region. <ul style="list-style-type: none"> <li>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.</li> </ul>
<b>Fill region considering only paragraph-separator lines as starting a new paragraph</b>	<b>&lt;f11&gt; t f n</b>	(fill-nonuniform-paragraphs MIN MAX &optional JUSTIFYF CITATION-REGEXP)	Fill paragraphs within the region, allowing varying indentation within each. <ul style="list-style-type: none"> <li>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.</li> </ul>

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

## Filling and Justification — References

Operation	Keystroke
<b>Text Filling</b>	
<b>GNU Emacs Lisp - Text - Filling</b>	
<b>GNU Emacs Lisp - Text - Auto Filling</b>	
<b>GNU Emacs Lisp - Text - Adaptive Filling</b>	
<b>GNU Emacs Lisp - Text - Margins</b>	
<b>Emacs hard wrap lines</b>	
<b>Emacs wiki - Ruler Mode</b>	
<b>Emacs wiki - refilling mode</b>	
<b>Emacs wiki - Auto Fill Mode</b>	
<b>How to set multiline comments - StackOverflow</b>	
<b>Line Wraps</b>	
<b>Truncate Lines @ EmacsWiki</b>	Describes how to set/disable line wrapping globally. Also explains how to control it for a given mode.
<b>Line Wrap @ EmacsWiki</b>	Describes various modes: AutoFillMode, LongLines, VisualLineMode
<b>longlines-mode @ emacs horror</b>	Discuss difference between longlines-mode and visual-line-mode
<b>ErgoEmacs: Show Cursor Position</b>	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).