



Filling and Justification

Description	Keystroke	Function	Note
Filling, Justification, Position (and navigation) Information	<p>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.</p> <p>Emacs filling uses several variables identifying the column where text must wrap. The main variable is the <i>‘fill-column’</i> 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-column command, shown below.</p>		
Show the ruler	<f11> b -	(ruler-mode &optional ARG)	<p>Toggle display of ruler in header line (Ruler mode) in window(s) of current buffer.</p> <ul style="list-style-type: none"><li>With a prefix argument ARG, enable Ruler mode if ARG is positive, and disable it otherwise.</li><li>The ruler markers are:<ul style="list-style-type: none"><li>  Current cursor position</li><li>! Marks every 5 positions.</li><li># 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>¶ Current <i>fill-column</i> value, set by <b>C-x f</b>, controls where the cursor wraps.</li><li>G 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 <u>set-goal-column command</u> in the navigation table.</li></ul></li></ul>
Show the values of the fill columns	<f11> t f ?	(pel-show-fill-columns)	<p>Display value of relevant fill columns for current buffer.</p> <ul style="list-style-type: none"><li>These are:<ul style="list-style-type: none"><li>fill-column</li><li>comment-fill-column</li><li>emacs-lisp-docstring-fill-column (in a lisp related mode)</li></ul></li></ul>
Filling Text	<p>For fill and refill mode to work properly on text:</p> <ul style="list-style-type: none"><li>Sentences must be separated by 2 spaces (unless <i>sentence-end-double-space</i> is t): use &lt;f11&gt; t m s to change this.</li><li>Paragraphs must be separated by 1 empty line.</li></ul>		
Toggle end of sentence space count	<f11> t m s	(pel-toggle-sentence-end)	<p>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.</p>
Set Fill Column	C-x f	(set-fill-column ARG)	<p>When no prefix value: prompts for column.</p> <ul style="list-style-type: none"><li>If with <b>C-u</b> prefix: use current column.</li><li>If with prefix value: use that value.</li></ul> <p>⚠ When editing Emacs Lisp code, the function used to refill is different and</p>
Toggle auto-fill mode	<f11> t f A	(auto-fill-mode &optional ARG)	<p>Toggle automatic line breaking (Auto Fill mode).</p> <p>With a prefix argument, enable Auto Fill mode if the prefix argument is positive, and disable it otherwise.</p> <p>When Auto Fill mode is enabled, inserting a space at a column beyond ‘current-fill-column’ automatically breaks the line at a previous space.</p>
Togge refill mode	<f11> t f R	(refill-mode &optional ARG)	<p>Toggle automatic refilling (Refill mode).</p> <p>With a prefix argument ARG, enable Refill mode if ARG is positive, and disable it otherwise.</p> <p>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.</p>
Toggle auto fill inside comments only	<f11> t f C	(pel-auto-fill-only-comments)	<p>Toggle the <i>comment-auto-fill-only-comments</i> variable to control whether filling is done everywhere or only inside the comments.</p>
Set fill prefix	C-x .	(set-fill-prefix)	<p>Set the fill prefix to the current line up to point.</p> <p>Filling expects lines to start with the fill prefix and reinserts the fill prefix in each resulting line.</p> <p>The fill prefix is then inserted before any character typed in the line.</p>
Fill region	<f11> t f r	(fill-region FROM TO &optional JUSTIFY NOSQUEEZE TO-EOP)	<p>Fill each of the paragraphs in the region. A prefix arg means justify as well. The ‘fill-column’ variable controls the width.</p>
Fill current paragraph	<ul style="list-style-type: none"><li>M-q</li><li>&lt;f11&gt; t f p</li></ul>	(fill-paragraph &optional JUSTIFY REGION)	<p>To justify as well: <b>C-u M-q</b></p> <p>Note: in refill mode this is done automatically. In auto fill mode the filling is done at the end of the line.</p>
		(lisp-fill-paragraph &optional JUSTIFY)	<p>Like M-q, but handle Emacs Lisp comments and docstrings.</p> <p>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.</p> <p>temporarily binds `fill-column’ to `emacs-lisp-docstring-fill-column’ which is used to identify the fill column for Emacs Lisp docstring.</p>
		(fill-comment-paragraph &optional JUSTIFY)	<p>Fill current comment.</p> <p>If we’re not in a comment, just return nil so that the caller can take care of filling. JUSTIFY is used as in ‘fill-paragraph’.</p>
Fill region, each indentation starts a new “paragraph”		(fill-individual-paragraphs MIN MAX &optional JUSTIFY CITATION-REGEXP)	<p>Fill paragraphs of uniform indentation within the region.</p> <p>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.</p>
Fill region considering only paragraph-separator lines as starting a new paragraph		(fill-nonuniform-paragraphs MIN MAX &optional JUSTIFYP CITATION-REGEXP)	<p>Fill paragraphs within the region, allowing varying indentation within each.</p> <p>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.</p>
Text Justification	<p>Text justification is normally used when <u>enriched text support</u> is activated via the <u>enriched-mode</u>. The standard Emacs key-chords listed below are only available in enriched-mode. The PEL key-chords are always available.</p>		
Toggle Enriched Text Mode	<f11> t m r	(enriched-mode &optional ARG)	<p>Minor mode for editing text/enriched files.</p> <p>These are files with embedded formatting information in the MIME standard text/enriched format.</p>
Set justification - full (both)	<ul style="list-style-type: none"><li>M-j b</li><li>&lt;f11&gt; t j f</li></ul>	(set-justification-full B E)	<p>Make paragraphs in the region fully justified.</p> <p>This makes lines flush on both margins by inserting spaces between words. If the mark is not active, this applies to the current paragraph.</p>
Set justification - left	<ul style="list-style-type: none"><li>M-j l</li><li>&lt;f11&gt; t j l</li></ul>	(set-justification-left B E)	<p>Make paragraphs in the region left-justified.</p> <p>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.</p>

Description	Keystroke	Function	Note
Set justification - center	<ul style="list-style-type: none"> <li>• M-j c</li> <li>• M-S</li> <li>• &lt;f11&gt; t j c</li> </ul>	(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	<ul style="list-style-type: none"> <li>• M-j r</li> <li>• &lt;f11&gt; t j r</li> </ul>	(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	<ul style="list-style-type: none"> <li>• M-j u</li> <li>• &lt;f11&gt; t j n</li> </ul>	(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’. This means adjusting the indentation so that it equals the distance between the end of the text and ‘fill-column’. The argument NLINES says how many lines to center.  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 @ emacs horror	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).