


# Whitespace

Operation	Keystroke	Function	Note
Text Whitespace Modes	The following Emacs command control how whitespace is shown or hidden. There are mode text modes, see the <b>Text Modes</b> table.		
<b>Toggle Whitespace Mode</b>  ( Also in: $\Sigma$ Text Modes)	<f11> t w m	(whitespace-mode &optional ARG)	Toggle whitespace visualization (Whitespace mode). With a prefix argument ARG, enable Whitespace mode if ARG is positive, and disable it otherwise. The kind of whitespace visualized is determined by the list variable <b>whitespace-style</b> , <b>whitespace-newline</b> .
<b>Hide/Show trailing whitespaces</b>  ( Also in: $\Sigma$ Text Modes)	<f11> t w T	(pel-toggle-show-trailing-whitespace)	Toggle highlight of the trailing whitespaces in current buffer.  Toggles the value of the variable <b>show-trailing-whitespace</b> .
<b>Hide/Show trailing empty lines</b>  ( Also in: $\Sigma$ Text Modes)	<f11> t w e	(pel-toggle-indicate-empty-lines)	Toggle highlight of empty lines.  Toggles the value of the variable <b>indicate-empty-lines</b> .
<b>Toggle individual elements of whitespace-style</b>  ( Also in: $\Sigma$ Text Modes)	<f11> t w o	(whitespace-toggle-options ARG)	<ul style="list-style-type: none"><li>If local whitespace-mode is off, toggle the option given by ARG and turn on local whitespace-mode.</li><li>If local whitespace-mode is on, toggle the option given by ARG and restart local whitespace-mode.</li></ul> The argument can be:  <ul style="list-style-type: none"><li>f toggle face visualization</li><li>t toggle TAB visualization</li><li>s toggle SPACE and HARD SPACE visualization</li><li>r toggle trailing blanks visualization</li><li>l toggle "long lines" visualization</li><li>L toggle "long lines" tail visualization</li><li>n toggle NEWLINE visualizaiton</li><li>e toggle empty line at bob and/or eob visualization</li><li>C-i toggle indentation SPACeS visualization (via 'indent-tabs-mode')</li><li>I toggle indentation SPACeS visualization</li><li>i toggle indentation TABs visualization</li><li>C-t toggle big indentation visualization</li><li>C-a toggle SPACeS after TAB visualization (via 'indent-tabs-mode')</li><li>A toggle SPACeS after TAB: SPACeS visualization</li><li>a toggle SPACeS after TAB: TABs visualization</li><li>C-b toggle SPACeS before TAB visualization (via 'indent-tabs-mode')</li><li>B toggle SPACeS before TAB: SPACeS visualization</li><li>b toggle SPACeS before TAB: TABs visualization</li></ul>
Deleting Whitespace	The following Emacs commands delete whitespaces. The deleted characters are not copied in the kill ring. These commands are also described in the <b>Copy/Kill/Delete</b> table.		
<b>Delete all spaces between 2 words</b>	M- \	(delete-horizontal-space &optional BACKWARD-ONLY)	Delete all spaces and tabs around point. Only works when cursor is on the spaces between the words or on the first character of the second word.
<b>Delete all spaces but one between words</b>	M- SPC	(just-one-space &optional N)	Delete all spaces and tabs around point, leaving one space (or N spaces). If N is negative, delete newlines as well, leaving -N spaces. <ul style="list-style-type: none"><li>This command ensures that words are separated by just one space character.</li><li>The cursor may be between the words but can also be on the fist character of the word.</li><li>At the end of the word it inserts a space.</li></ul>
<b>Cycle spacing around point</b>	<f11> t SPC	(cycle-spacing &optional N PRESERVE-NL-BACK MODE)	Manipulate whitespace around point in a smart way.  <ul style="list-style-type: none"><li>The first call in a sequence acts like 'just-one-space'. It deletes all spaces and tabs around point, leaving one space (or N spaces). N is the prefix argument. If N is negative, it deletes newlines as well, leaving -N spaces. (If PRESERVE-NL-BACK is non-nil, it does not delete newlines before point.)</li><li>The second call in a sequence deletes all spaces.</li><li>The third call in a sequence restores the original whitespace (and point).</li></ul> The easiest way to use this command for the second or third call (or further) is to issue it once and then use the repeat command ( <b>C-x z</b> or <f5>).
<b>Delete all trailing whitespaces</b>	<f11> t w t	(delete-trailing-whitespace &optional START END)	Delete trailing whitespace in the entire (or narrowed part of the) buffer or in the marked region. <ul style="list-style-type: none"><li>This command deletes whitespace characters after the last non-whitespace character in each line between START and END. It does not consider formfeed characters to be whitespace.</li><li>If this command acts on the entire buffer, it also deletes all trailing lines at the end of the buffer if the variable 'delete-trailing-lines' is non-nil.</li></ul>
<b>Remove non required whitespaces</b>	<f11> t w c	(whitespace-cleanup)	Cleanup some blank problems (non-required whitespace) in all buffer or at region. <ul style="list-style-type: none"><li>It usually applies to the whole buffer, but in transient mark mode when the mark is active, it applies to the region. It also applies to the region when it is not in transient mark mode, the mark is active and C-u was pressed just before calling 'whitespace-cleanup' interactively.</li></ul>
<b>Delete Indentation, join this line to the previous one</b>	M- ^	(delete-indentation &optional ARG)	Adjoin the current line with the line above. It joins two lines by deleting the intervening newline, along with any indentation following it.
<b>Delete all spaces between point and next non-white</b>	C- [X]	(pel-delete-to-next-visible)	Delete all whitespace between point and next non-whitespace character. <ul style="list-style-type: none"><li>Note: on macOS laptop, type: "Fn C-delete".</li></ul>
Delete/Insert Blank Lines	The following commands provide special techniques for inserting and deleting blank lines.		
<b>Delete consecutive blank lines</b>	C- x C- o	(delete-blank-lines)	<ul style="list-style-type: none"><li>On blank line, delete all surrounding blank lines, leaving just one.</li><li>On isolated blank line, delete that one.</li><li>On nonblank line, delete any immediately following blank lines.</li></ul>
<b>Insert newline, leave point before it</b>	C- o	(open-line N)	Insert a newline and leave point before it. If there is a fill prefix and/or a 'left-margin' (an emacs variable), insert them on the new line if the line would have been blank. With arg N, insert N newlines.
Kill Whitespace	The PEL package provides a command to kill the whitespace, retaining it in the kill ring.		
<b>Kill whitespace at point</b>	<f11> - SPC	(pel-kill-whitespace-at-point)	Kill all whitespace at and around point. <ul style="list-style-type: none"><li>Copy removed characters to kill ring.</li></ul>

Operation	Keystroke	Function	Note
Align text vertically	The following command can be used to align text vertically. Useful for source code.		
Align a set of lines on some text  (See Also:  Align)	<f11> t a r	(align-regexp BEG END REGEXP &optional GROUP SPACING REPEAT)	Align the current region using an ad-hoc rule read from the minibuffer. BEG and END mark the limits of the region. Interactively, this function prompts for the regular expression REGEXP to align with. <ul style="list-style-type: none"> <li>First select a region, then issue the command. For example, to align assignment of variables over the equal sign use = as the <i>regexp</i>.</li> <li>The PEL package creates the <b>ar</b> alias for <b>align-regexp</b>, so it's also possible to invoke it with <b>M-x ar &lt;RET&gt;</b></li> </ul>

## Whitespace — Reference

Topic/URL	Comment
GNU Emacs manual - Useless Whitespace	
Emacs Wiki - Deleting Whitespace	
Stack Overflow - aligning or prettifying code in emacs	
Stack Exchange - Understanding of emacs align-regexp	