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 **Text Modes** table.		
Open this PDF file. See also: ∑ **Help/Info**	`<f11> t w <f1>`	(**pel-help-pdf** &optional OPEN-WEB-PAGE)	Open the local copy of the ∑ **Whitespace** PDF file unless a command prefix (like **C-u**) was used. In that case it opens the Github-hosted file instead.
∑ **Customize** Emacs whitespace management control	`<f11> t w <f3>`	(**pel-customize-library** &optional OTHER-WINDOW)	Customize Emacs handling of whitespaces.
Toggle Whitespace Mode See also: • ∑ **Text Modes**	• `<f11> t w m` • `<f11> t m w`	(**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 ***whitespace-style***, ***whitespace-newline***.
Hide/Show trailing whitespaces See also: • ∑ **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 ***show-trailing-whitespace***.
Hide/Show trailing empty lines See also: • ∑ **Text Modes**	`<f11> t w e`	(**pel-toggle-indicate-empty-lines**)	Toggle highlight of empty lines. Toggles the value of the variable ***indicate-empty-lines***.
Toggle individual elements of whitespace-style See also: • ∑ **Text Modes**	`<f11> t w o`	(**whitespace-toggle-options** ARG)	• If local whitespace-mode is off, toggle the option given by ARG and turn on local whitespace-mode. • If local whitespace-mode is on, toggle the option given by ARG and restart local whitespace-mode. The argument can be: • `f` toggle face visualization • `t` toggle TAB visualization • `s` toggle SPACE and HARD SPACE visualization • `r` toggle trailing blanks visualization • `l` toggle "long lines" visualization • `L` toggle "long lines" tail visualization • `n` toggle NEWLINE visualization • `e` toggle empty line at bob and/or eob visualization • `C-i` toggle indentation SPACEs visualization (via 'indent-tabs-mode') • `I` toggle indentation SPACEs visualization • `i` toggle indentation TABs visualization • `C-t` toggle big indentation visualization • `C-a` toggle SPACEs after TAB visualization (via 'indent-tabs-mode') • `A` toggle SPACEs after TAB: SPACEs visualization • `a` toggle SPACEs after TAB: TABs visualization • `C-b` toggle SPACEs before TAB visualization (via 'indent-tabs-mode') • `B` toggle SPACEs before TAB: SPACEs visualization • `b` toggle SPACEs before TAB: TABs visualization
Controlling use of hard tabs or spaces for indentation	The use of hard tabs or spaces for indentation is controlled by the Emacs (customizable) variable **indent-tabs-mode**. Like several Emacs variable this variable has global impact, but this can be overridden by directory local value, file local value and buffer local value allowing fine control over set of files and buffers. PEL provides the following related commands.		
Toggle use of hard tabs and only spaces for indentation in the current buffer See also: ∑ **Indentation**	`<f11> t w I`	(**pel-toggle-indent-tabs-mode** &optional ARG)	Toggle use of hard tabs or spaces for indentation in current buffer. • Beep on each change to warn user of the change and display new value. • If ARG is positive set to use hard tabs, otherwise force use of spaces only.
Replacing Tabs with spaces or spaces with tabs	The following two commands can be used to replace hard tabs in a file with the corresponding number of space characters while retaining the same indentation and vice-versa.		
Replace tabs with spaces in a region See also: ∑ **Indentation**	`<f11> t w SPC`	(**untabify** START END &optional ARG)	Convert all tabs in region to multiple spaces, preserving columns. • If called interactively with prefix ARG, convert for the entire buffer. • First select a region (Use C-x h for selecting the whole file). Then use the *untabify* function to replace all tabs by spaces in that region.
Replace multiple spaces with tabs in a region See also: ∑ **Indentation**	`<f11> t w <tab>`	(**tabify** START END &optional ARG)	Convert multiple spaces in region to tabs when possible. • A group of spaces is partially replaced by tabs when this can be done without changing the column they end at. • If called interactively with prefix ARG, convert for the entire buffer.
Deleting Whitespace See also: ∑ **Cut & Paste**	The following Emacs commands delete whitespaces. The deleted characters are not copied in the kill ring. These commands are also described in the **Copy/Kill/Delete** table.		
Delete all spaces between 2 words	`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.
Delete all spaces but one beween words	`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. • This command ensures that words are separated by just one space character. • The cursor may be between the words but can also be on the fist character of the word. • At the end of the word it inserts a space.
Cycle spacing around point	`<f11> t w .`	(**cycle-spacing** &optional N PRESERVE-NL-BACK MODE)	Manipulate whitespace around point in a smart way. • 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.) • The second call in a sequence deletes all spaces. • The third call in a sequence restores the original whitespace (and point). 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 (**C-x z** or <f5>).

Operation	Keystroke	Function	Note
Delete all trailing whitespaces See also: • ⨌ **Cut & Paste**	**<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. • 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. • 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.
Remove non required whitespaces	**<f11> t w c**	(**whitespace-cleanup**)	Cleanup some blank problems (non-required whitespace) in all buffer or at region. • 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.
Delete empty/whitespace lines in region or all buffer	• **<f11> DEL M-SPC** • **<f11> ⌫ M-SPC**	(**pel-delete-all-empty-lines** &optional BEGIN END)	Delete all empty lines from marked area or the entire buffer if nothing is marked.
Delete Indentation, join this line to the previous one See also: • ⨌ **Cut & Paste** • ⨌ **Indentation**	**M-^**	(**delete-indentation** &optional ARG)	Join this line to previous and fix up whitespace at join. • If there is a fill prefix, delete it from the beginning of this line. • With argument, join this line to following line.
Delete all spaces between point and next non-white	**C-⌦**	(**pel-delete-to-next-visible**)	Delete all whitespace between point and next non-whitespace character. • Note: on macOS laptop, type: "Fn C-delete".
Delete/Insert Blank Lines	The following commands provide special techniques for inserting and deleting blank lines.		
Delete consecutive blank lines	**C-x C-o**	(**delete-blank-lines**)	• On blank line, delete all surrounding blank lines, leaving just one. • On isolated blank line, delete that one. • On nonblank line, delete any immediately following blank lines.
Insert empty line above current line	**M-L**	(**pel-insert-line-above** N)	Insert N lines above current line. • Move point to the indentation level of the first of the new lines. • If N is negative, do not move point.
Insert newline, leave point before it See also: ⨌ **Drawing**	**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.
Insert Literal Tab	**C-q <tab>**	(**quoted-insert** ARG) <tab>	Inserts a hard tab inside the file but moves the cursor to the column that represents the next multiple of **tab-width**.
		☝With PEL, for several major modes, the value of the tab-width variable is controlled by a mode specific user options variable, like pel-c-tab-width for buffers in c-mode. In those buffers the value of tab-width will be set by PEL to the mode specific value when the buffer is opened.	
Kill Whitespace	The PEL package provides a command to kill the whitespace, retaining it in the kill ring.		
Kill whitespace at point	**<f11> - SPC**	(**pel-kill-whitespace-at-point**)	Kill all whitespace at and around point. • Copy removed characters to kill ring.
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. • First select a region, then issue the command. For example, to align assignment of variables over the equal sign use **=** as the *regexp*. • The PEL package creates the **ar** alias for **align-regexp**, so it's also possible to invoke it with **M-x ar RET**

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	