

Indenting & Tab

Description	Keystroke	Function	Note
Indentation under Emacs	Emacs controls indentation via major modes and the tab key may take different and surprising behaviour for people just starting with Emacs. More information is available in the tables of the various major modes.		
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 . <ul style="list-style-type: none">👉 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 by PEL to the mode specific value when the buffer is opened.
Behaviour of Tab	In Emacs the behaviour of the <tab> key depends on the major mode of the current buffer. This key is rebound by several major mode. By default, in text modes, tabs are set to 8 spaces, inserting hard tabs. However, if there is text in the above lines, the tab moves to the spot under the word above. Note that if a line is full of text (without any space), then the tab stops controlled by the ruler take effect again.		
Indent current line (or region)	<tab>	(indent-for-tab-command &optional ARG)	Indent the current line or region, or insert a tab, as appropriate. <ul style="list-style-type: none">This function either inserts a tab, or indents the current line, or performs symbol completion, depending on ‘tab-always-indent’. The function called to actually indent the line or insert a tab is given by the variable ‘indent-line-function’.If a prefix argument is given, after this function indents the current line or inserts a tab, it also rigidly indents the entire balanced expression which starts at the beginning of the current line, to reflect the current line’s indentation.In most major modes, if point was in the current line’s indentation, it is moved to the first non-whitespace character after indenting; otherwise it stays at the same position relative to the text.If ‘transient-mark-mode’ is turned on and the region is active, this function instead calls ‘indent-region’. In this case, any prefix argument is ignored. ⚠️🔧 The behaviour of the tab key vastly differ between major modes. This ranges from not moving the cursor at all if the indentation is identified as correct for the current context, to cycling through various potential positions to just what someone new to Emacs would expect. Much more has to be documented on the behaviour of that key and how it can be controlled and customized. It’s quite possible that the best way to document its behaviour would be to place a description inside the table of each major mode.
		indent-for-tab-command &optional ARG)	In Lisp related modes. <ul style="list-style-type: none">indent-line-function = indent-relative.tab-always-indent = t 🔧The above values that I got in Emacs via inspection do not explain tab behaviour in the Emacs Lisp mode (which is to indent the code according to Emacs Lisp semantics, a very useful feature when writing Lisp code). In this mode tab corrects the indentation of the code at the current line, which may be to indent, de-indent or do nothing.
		(c-indent-line-or-region &optional ARG REGION)	In C related modes. Indent active region, current line, or block starting on this line. <ul style="list-style-type: none">In Transient Mark mode, when the region is active, reindent the region.Otherwise, with a prefix argument, rigidly reindent the expression starting on the current line.Otherwise reindent just the current line.
Indent lines of list after point (CLBC s3.lisp)	C-M-q	<ul style="list-style-type: none">(indent-sexp &optional ENDPOS)(c-indent-exp &optional SHUTUP-P)	Indent each line of the list starting just after point. The command used depends on the major mode of the current buffer.
Insert spaces or tabs to next defined tab-stop column See also: <ul style="list-style-type: none">⌘I - C⌘I - C++⌘I - D	M-i	(tab-to-tab-stop)	Insert spaces or tabs to next defined tab-stop column. <ul style="list-style-type: none">The exact location of the next tab stop is identified by the value of the tab-stop-list and tab-width for the current buffer.👉 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 by PEL to the mode specific value when the buffer is opened.
Insert an indented line below current line See also: ⌘ Align	<ul style="list-style-type: none">M-RET<f11> <tab> RET	(pel-newline-and-indent-below)	Insert an indented line just below current line. <ul style="list-style-type: none">The command can also align text vertically if this special mode was activated for the buffer with the <f11> M-RET .To see the current behaviour use <f11> t a ? : it displays whether the M-RET command aligns text or not.
Toggle text alignment on pel-newline-and-indent-below See also: ⌘ Align	<f11> M-RET	(pel-toggle-newline-indent-align)	Toggle variable ‘pel-newline-does-align’ for the local buffer. This toggles the way function ‘pel-newline-and-indent-below’ operates.
Show state of pel-newline-and-indent-below See also: ⌘ Align	<f11> t a ?	(pel-show-if-newline-aligns)	Display the behaviour of M-RET in the current buffer: show if that command aligns text or not. Print the information in the echo area.
Change the tab stops	M-x edit-tab-stops		Opens a *Tab Stops* buffer . Identify the tab stops in the first line with colons. Use C-c C-c to activate and exit the buffer. Again, the tab stop take effect at the top of the buffer,
Change the tab width	M-: (setq tab-width N)		The variable <i>tab-width</i> normally defaults to 8 in emacs. It can be set locally inside a buffer with (setq tab-width N) or globally with (setq-default tab-width N). The M-: keystroke allows evaluating a lisp expression interactively (as the above two). <ul style="list-style-type: none">Note that any literal tab in the buffer impact the location of the column where the next character shows. When changing the tab-width, the layout shown in the window that contains literal tabs will be modified according to the new tab-width value.👉 With PEL, remember that the local value of tab-width is controlled by PEL and set to the value of a mode specific user option, such as pel-c-tab-width in c-mode buffers.
Make <tab> insert space/tab	M-: (setq indent-tabs-mode nil/t)		By default, pressing <tab> insert literal (hard) tabs inside the file. The indent-tabs-mode variable controls that: set to t it inserts tabs, set to nil it inserts spaces.
Next line, indented	C-j	(electric-newline-and-maybe-indent)	Add new line and indent next line. Indentation is controlled by the variable left-margin . Pressing Tab anywhere on the line also indents the line properly.

Description	Keystroke	Function	Note
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: ⌘ Whitespace	<f11> t w SPC	(untabify START END &optional ARG)	Convert all tabs in region to multiple spaces, preserving columns. <ul style="list-style-type: none"> 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 <i>untabify</i> function to replace all tabs by spaces in that region.
Replace multiple spaces with tabs in a region See also: ⌘ Whitespace	<f11> t w <tab>	(tabify START END &optional ARG)	Convert multiple spaces in region to tabs when possible. <ul style="list-style-type: none"> 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.

Indentation — References

Title & URL	Description
Understanding GNU Emacs and Tabs	Overview description of how Emacs handle the Tab key, often used for strict indentation in many editors. In Emacs it can do much more.
GNU Emacs Manual - Indentation	
GNU Emacs Manual - Indentation for Programs	
Indentation Styles	
Indentation Styles @ Wikipedia	
StackOverflow - Emacs BSD/Allman Style with 4 Space Tabs?	
GNU Emacs Manual - Styles	
Emacs BSD/Allman Style with 4 Space Tabs?	
Emacs: Linux Kernel Style but with Allman/BSD Style Braces?	
Emacs Wiki - Indenting C	
Indent preprocessor directives as C code in emacs	Does not fully address the way I want to have multi-indentations for pre-processor
elisp code - ppindent.el	Implements pre-processor indentation with the # always in the first column. Not yet exactly what I want.
Demystify C++ Metaprograms using Emacs	
Programming in C++, Rules and Recommendations	ellemtel style