






























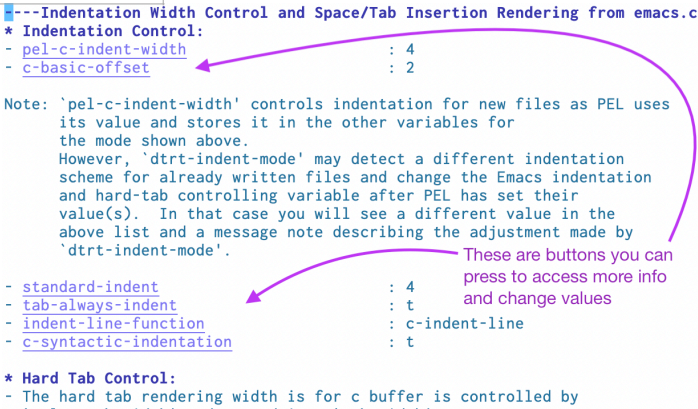

























## Indenting & Tab

Description	Keystroke	Function	Note															
<div><div>Indentation under Emacs</div><div><ul style="list-style-type: none"><li>Use <a href="#">hard tabs or spaces for indentation</a></li><li><a href="#">Set hard-tab visual width</a></li><li><a href="#">Adjust tab stop</a></li><li><a href="#">Show tab/indent settings</a></li><li><a href="#">Align on return</a></li><li><a href="#">Insert hard-tab</a></li><li><a href="#">To next tab stop</a></li><li><a href="#">tabify &amp; untabify</a></li><li><a href="#">Behaviour of tab key</a></li><li><a href="#">Insert newline, split line</a></li><li><a href="#">Indent region</a></li><li><a href="#">Delete indentation</a></li><li><a href="#">Move to 1st non-blank</a></li><li>Indenting /un-indenting rigidly</li><li><a href="#">Text alignment on newline</a></li><li><a href="#">Indent-tools</a></li><li><a href="#">dtrt-indent</a>, <a href="#">Indent-bars</a></li><li><a href="#">Smart-shift</a>, <a href="#">Smart-tabs</a></li><li>References</li></ul></div></div>	<div>Emacs controls indentation and behaviour of the tab key according to various rules controlled by the buffer's <a href="#">major mode</a>.</div> <div><ul style="list-style-type: none"><li>The modes default behaviour may be modified by the use of <a href="#">minor modes</a>.<ul style="list-style-type: none"><li>Several major modes identify one or several variable that control indentation levels and behaviour. Some use default indentation variables.</li><li>Some programming languages (<a href="#">such as Go</a>) impose hard-tab for indentation, using tab for indentation and space for alignment. Works very nicely.</li><li>There are several indentation schemes: hard-tab only for indentation, spaces only for indentation, hard-tabs for indentation space for alignment, free mix of hard-tabs and spaces for indentation and whitespace.</li><li>Emacs can support all those schemes. It can tabify or untabify source code.</li><li>Emacs controls the <i>display rendering</i> of hard tabs via the <b>tab-width</b> customizable user-option variable.</li><li>The indentation width is often independent from the tab width but not always. Again it depends on the major mode used.</li></ul></li></ul></div> <div>PEL supports all indentation schemes and various indentation mechanisms and has its own extensions. For each explicitly supported mode you can configure the indentation with and use of hard tabs for the mode. This is used for all files you create. It also support automatic adjustment for files you edit that have been created by others and use a different indentation width and use of hard-tabs.</div> <div><ul style="list-style-type: none"><li>It provides activation of following external packages by turning on the corresponding pel-use customizable user-option (setting it to <b>t</b>):<table><tr><td> The <a href="#">dtrt-indent</a> external package</td><td> <b>pel-use-dtrt-indent</b></td><td>Detects indentation width &amp; use of hard-tabs in file, then adjust settings to adapt.</td></tr><tr><td> The <a href="#">indent-bars</a> external package</td><td> <b>pel-use-indent-bars</b></td><td>Provides ability to show indent bar at each indentation level.</td></tr><tr><td> The <a href="#">indent-tools</a> external package</td><td> <b>pel-use-indent-tools</b></td><td>Provides commands to alter indentation of code.</td></tr><tr><td> The <a href="#">smart-shift</a> external package</td><td> <b>pel-use-smart-shift</b></td><td>Provides ability to shift text areas left/right by indentation levels.</td></tr><tr><td> The <a href="#">smart-tabs</a> external package</td><td> <b>pel-use-smart-tabs</b></td><td>Use it when using tabs for indentation and spaces for alignment.</td></tr></table></li></ul></div>			 The <a href="#">dtrt-indent</a> external package	 <b>pel-use-dtrt-indent</b>	Detects indentation width & use of hard-tabs in file, then adjust settings to adapt.	 The <a href="#">indent-bars</a> external package	 <b>pel-use-indent-bars</b>	Provides ability to show indent bar at each indentation level.	 The <a href="#">indent-tools</a> external package	 <b>pel-use-indent-tools</b>	Provides commands to alter indentation of code.	 The <a href="#">smart-shift</a> external package	 <b>pel-use-smart-shift</b>	Provides ability to shift text areas left/right by indentation levels.	 The <a href="#">smart-tabs</a> external package	 <b>pel-use-smart-tabs</b>	Use it when using tabs for indentation and spaces for alignment.
 The <a href="#">dtrt-indent</a> external package	 <b>pel-use-dtrt-indent</b>	Detects indentation width & use of hard-tabs in file, then adjust settings to adapt.																
 The <a href="#">indent-bars</a> external package	 <b>pel-use-indent-bars</b>	Provides ability to show indent bar at each indentation level.																
 The <a href="#">indent-tools</a> external package	 <b>pel-use-indent-tools</b>	Provides commands to alter indentation of code.																
 The <a href="#">smart-shift</a> external package	 <b>pel-use-smart-shift</b>	Provides ability to shift text areas left/right by indentation levels.																
 The <a href="#">smart-tabs</a> external package	 <b>pel-use-smart-tabs</b>	Use it when using tabs for indentation and spaces for alignment.																
Last updated on:	2025-11-03	 See <a href="#">Showing Information About Indentation and Hard Tab Control under PEL</a>																
<div>Open this PDF file.</div> <div>See also: <a href="#">↗ Help/Info</a></div>	<f11> <tab> <f1>	(pel-help-pdf &optional OPEN-WEB-PAGE)	Open the <a href="#">↗ Indentation</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.															
<a href="#">↗</a> <b>Customize PEL highlighting control</b>	<f11> <tab> <f2>	(pel-customize-pel &optional OTHER-WINDOW)	Customize PEL support for indentation management <ul style="list-style-type: none"><li>If OTHER-WINDOW is non-nil (use <b>C-u</b>) , display in other window.</li></ul>															
<a href="#">↗</a> <b>Customize Emacs indentation control</b>	<f11> <tab> <f3>	(pel-customize-library &optional OTHER-WINDOW)	Customize Emacs indentation control groups: indent, <a href="#">Indent-bars</a> <a href="#">indent-tools</a> , <a href="#">dtrt-indent</a> , <a href="#">smart-shift</a> . <ul style="list-style-type: none"><li>If OTHER-WINDOW is non-nil (use <b>C-u</b>), display in another window.</li></ul>															
<b>Use hard tabs or spaces for indentation</b>	<div>The use of hard tabs or spaces for indentation is controlled by the Emacs (customizable) variable <b>indent-tabs-mode</b>.</div> <div><ul style="list-style-type: none"><li>Like several Emacs variable this variable has <a href="#">global impact</a>, but this can be overridden by <a href="#">directory local</a> value, <a href="#">file local</a> value and <a href="#">buffer local</a> value</li></ul></div> <div>See also: <a href="#">↗ Whitespace</a></div>																	
<b>Toggle use of hard tabs for indentation in the current buffer</b>	<ul style="list-style-type: none"><li>&lt;f11&gt; &lt;tab&gt; m</li><li>&lt;f11&gt; t w I</li></ul>	(pel-toggle-indent-tabs-mode &optional ARG)	Toggle whether indentation can insert hard tab characters in the current buffer. <ul style="list-style-type: none"><li>Beep on each change to warn user of the change and display new value.</li><li>If ARG is positive set to use hard tabs, otherwise force use of spaces only.</li><li>This uses Emacs <b>indent-tabs-mode</b> function and provides more feedback.</li></ul>															
<b>Hard-tab “width”</b>	The current-buffer value of tab-width affects the <i>visual rendering</i> of the hard-tab character. It does not affect the content of the buffer or file.																	
<div>Set visual rendering of hard tabs for the current buffer</div> <div><ul style="list-style-type: none"><li>Does not change buffer/file content</li></ul></div>	<f11> <tab> w	(pel-set-tab-width N)	Change the <b>tab-width</b> of the current buffer, only affecting the display rendering of hard tabs inserted in the buffer text. Prompts for a new value in the [2, 8] range. <ul style="list-style-type: none"><li>This modifies a buffer local value of the the <b>tab-width</b> user-option. The change is temporary and affects the current buffer only.</li><li>PEL provides a specialized user-option to set the default value of <b>tab-width</b> for several major modes. For example, to change the tab width used for all Go source code files, change the '<b>pel-go-tab-width</b>' user-option variable instead. See the documentation of each major mode for more information.</li></ul>															
<b>Tab stops</b>	Emacs keeps track of a set of “ <i>tab-stop</i> ” columns that can be used as reference points to align text. Something similar to <b>typewriter tab-stop</b> .																	
<div>Change the tab stops</div> <div><ul style="list-style-type: none"><li>used by <b>M-i</b></li></ul></div>	<f11> <tab> e	(edit-tab-stops)	Opens a <b>*Tab Stops* buffer</b> . Identify the tab stops in the first line with colons. Use <b>C-c C-c</b> to activate and exit the buffer. <ul style="list-style-type: none"><li>The tab stop take effect at the top of the buffer, as used by <b>M-i</b></li></ul>															
<div>Show Indentation settings</div> <div><div>For several major modes, a PEL mode-specific user-option controls the the value of the <b>tab-width</b> variable for the mode. For example, <b>pel-c-tab-width</b> is used for c-mode buffers.</div><div>Note however, that indentation for C buffer is controlled by the <b>c-basic-offset</b> user-option.</div><div>Other major modes may provide mode specific control variables and the printed information will differ.</div></div>	<ul style="list-style-type: none"><li>&lt;f11&gt; &lt;tab&gt; ?</li><li>&lt;f11&gt; ? &lt;tab&gt;</li></ul>	(pel-show-indent &optional APPEND)	<div>Print info about indentation control in a <b>*pel-indent-info* help-mode</b> buffer.</div> <div><ul style="list-style-type: none"><li>Buffer-specific values of relevant user-options as buttons to use to get more info and change their customized values. Includes major-mode specific ones.</li><li>Clear previous buffer content. Use prefix arg (like <b>C-u</b>) to append instead.</li></ul></div> <div></div>															
<div>Toggle text alignment on pel-newline-and-indent-below</div> <div>See also: <a href="#">↗ Align</a></div>	<f11> M-RET	(pel-toggle-newline-indent-align)	Toggle variable <i>pel-newline-does-align</i> for the local buffer. <div>It affects the way function '<b>pel-newline-and-indent-below</b>' operates.</div> <div><ul style="list-style-type: none"><li>If <i>pel-newline-does-align</i> is t, it aligns several syntactic element in the current block: the comments, the assignments.</li><li> set modes that automatically activates <i>pel-newline-does-align</i> by adding the major mode to <b>pel-modes-activating-align-on-return</b> user option.</li><li>This affects the behaviour of the following commands: <b>pel-cc-newline</b> (assigned to <b>RET</b> in CC modes like c-mode, c++-mode and d-mode). <b>pel-newline-and-indent-below</b> (assigned the <b>M-RET</b>)</li><li>See the list with &lt;f11&gt; t a ?</li></ul></div>															
<div>Show state of text modes</div> <div><ul style="list-style-type: none"><li>whether hard tabs are used for indentation, tab-width</li><li>whether newline aligns text</li><li>electric-quote-mode</li><li>delete-selection-mode</li><li>enriched-mode</li><li>overwrite-mode</li><li>case folding</li><li>subword, superword, glass modes</li><li>visible-mode, smart-dash-mode</li><li>paragraph definition</li></ul></div>	<f11> t m ?	(pel-show-text-modes)	Display the state of the various text modes in the mini buffer. <div> Quickly show several settings inside the mode line: the tab settings, text alignment on newline, whitespace mode, etc...</div> <div><ul style="list-style-type: none"><li>When <b>indent-tabs-mode</b> is active, Emacs inserts a number of hard tabs and spaces.<ul style="list-style-type: none"><li>The number of hard tabs instead depends on the amount of characters required for indentation and the <b>tab-width</b>.</li></ul></li><li>If indent-tabs-mode is not active, then Emacs inserts only space characters.</li><li> PEL provides user-options of the form <b>pel-&lt;mode&gt;-use-tabs</b> which is used to initialize the indent-tabs-mode supported major modes buffers.</li></ul></div> <div>Text Modes Status:<ul style="list-style-type: none"><li>- Local indent-tabs-mode : off: use spaces, Tab width = 8</li><li>- Local newline does align : off . Automatically activated by modes (&lt;f11&gt; t a &lt;f2&gt;): (c-mode c++-mode sh-mode)</li><li>- Local electric-quote-mode: off , electric-quote-local-mode: not loaded.</li><li>- whitespace-mode : not loaded, show-trailing-whitespace : off , indicate-empty-lines: off.</li><li>- enriched-mode : not loaded.</li><li>- overwrite mode : off , delete-selection-mode : off.</li><li>- case-fold-search : on , sort-fold-case : not loaded.</li><li>- subword mode : off , superword mode : on , glass-mode: not loaded.</li><li>- visible-mode : off , smart-dash-mode : not loaded.</li><li>- Sentences end with 2 space characters.</li><li>- paragraph-start : "^L\\ [ ]*\$"</li><li>- paragraph-separate: "[^L]*\$"</li></ul></div>															
Output example ➡																		



Description	Keystroke	Function	Note																																																																								
Indenting and un-indenting rigidly	The following commands provide non-semantic indentation of the current line or marked region. <ul style="list-style-type: none"><li>The first command allows you to use further keystrokes to fine-tune the indentation back and forth using cursor keys. That's probably all you ever need to use.</li><li>Currently, PEL also provides the last 2 commands that indent or un-indent the current line or marked region. Once used, the region remains marked to allow further use of the command.</li></ul>																																																																										
Indent/Unindent rigidly  See also: <a href="#">🔗 Key-Chords</a>	<ul style="list-style-type: none"><li><b>C-x &lt;tab&gt;</b></li><li><b>&lt;f11&gt; &lt;tab&gt; &lt;tab&gt;</b></li><li><b>&lt;tab&gt;q</b></li></ul>	<b>(pel-indent-rigidly &amp;optional N)</b>  -----  PEL uses the above instead of the standard: <b>(indent-rigidly START END ARG &amp;optional INTERACTIVE)</b>	Enter a mode to indent rigidly the marked region or current line N times. <ul style="list-style-type: none"><li><b>If a region is marked</b>, it uses 'indent-rigidly' and provides the same prompts to control indentation changes.</li><li><b>If no region is marked</b>, it operates on current line(s) identified by the numeric argument N (or if not specified N=1):<ul style="list-style-type: none"><li>N = [-1, 0, 1] : operate on current line</li><li>N &gt; 1 : operate on the current line and N-1 lines below.</li><li>N &lt; -1 : operate on the current line and (abs N) -1 lines above.</li></ul></li></ul> Once the command is issued use the keys listed below to indent or un-indent by indent-width step or by single column.  ----- Indent all lines starting in the region. <ul style="list-style-type: none"><li>If called interactively with no prefix argument, activate a transient mode in which the indentation can be adjusted interactively by typing <b>&lt;left&gt;</b>, <b>&lt;right&gt;</b>, <b>S-&lt;left&gt;</b>, or <b>S-&lt;right&gt;</b>.</li></ul>																																																																								
See also: <ul style="list-style-type: none"><li><a href="#">🔗 I - C</a></li><li><a href="#">🔗 I - C++</a></li><li><a href="#">🔗 I - D</a></li><li><a href="#">🔗 M reStructuredText</a></li></ul>	Both of these commands activate a transient mode where Emacs prompts for extra keys to control how to indent or un-indent. The capabilities are controlled by the variable <i>indent-rigidly-map</i> with by default provides: <table><tr><td><b>S-&lt;right&gt;</b></td><td>indent-rigidly-right-to-tab-stop</td><td><b>S-&lt;left&gt;</b></td><td>indent-rigidly-left-to-tab-stop</td></tr><tr><td><b>&lt;right&gt;</b></td><td>indent-rigidly-right</td><td><b>&lt;left&gt;</b></td><td>indent-rigidly-left</td></tr></table> Typing any other key deactivates the transient mode. <ul style="list-style-type: none"><li>The <b>S-&lt;right&gt;</b> and <b>S-&lt;left&gt;</b> keys indent/de-indent to the next tab-stop position, which is controlled by the <b>tab-width</b> user option.<ul style="list-style-type: none"><li>With PEL, for several major modes, the indentation is controlled by a mode-specific user option variable . For example, for buffers in c-mode, the value of <b>pel-c-tab-width</b> is automatically stored into tab-width when the buffer is opened.</li></ul></li></ul>  If you use the cua-mode: the cua-mode uses <b>C-x</b> , to invoke this command when cua-mode is active, type it really fast or type <b>C-x C-x &lt;tab&gt;</b> (or use the PEL binding <b>&lt;f11&gt; &lt;tab&gt; &lt;tab&gt;</b> ).			<b>S-&lt;right&gt;</b>	indent-rigidly-right-to-tab-stop	<b>S-&lt;left&gt;</b>	indent-rigidly-left-to-tab-stop	<b>&lt;right&gt;</b>	indent-rigidly-right	<b>&lt;left&gt;</b>	indent-rigidly-left																																																																
<b>S-&lt;right&gt;</b>	indent-rigidly-right-to-tab-stop	<b>S-&lt;left&gt;</b>	indent-rigidly-left-to-tab-stop																																																																								
<b>&lt;right&gt;</b>	indent-rigidly-right	<b>&lt;left&gt;</b>	indent-rigidly-left																																																																								
	  With PEL, the <b>&lt;tab&gt;q</b> key-chord is available when <b>pel-use-key-chord</b> is non-nil. See <a href="#">🔗 Key-Chords</a> .  Command numeric prefix <b>is available</b> with the key-chord binding.																																																																										
Indent line(s) rigidly	<ul style="list-style-type: none"><li><b>&lt;f6&gt; &lt;tab&gt;</b></li><li><b>&lt;f11&gt; &lt;tab&gt; c</b></li></ul>	<b>(pel-indent-lines &amp;optional N)</b>	Indent current or marked lines by N indentation levels																																																																								
	<ul style="list-style-type: none"><li>Works with point anywhere on the line.</li><li>All lines touched by the region are indented.</li><li>A special argument N can specify more than one indentation level. It defaults to 1.</li><li>If a negative number is specified, 'pel-unindent-lines' is used.</li><li>If a region is marked, the function does not deactivate it to allow repeated execution of the command. It also modifies the region to include all characters in all affected lines.</li><li>Use <b>C-g</b> to de-activate the region.</li><li>Handles presence of hard tabs:<ul style="list-style-type: none"><li>If indent-tabs-mode is non-nil the indentation is created with a mix of hard-tabs and space characters.</li><li>If indent-tabs-mode is nil, any hard tab in the indentation of the marked lines is replaced by the proper number of spaces. Hard tabs after first non-whitespace character on the line are left.</li></ul></li></ul>																																																																										
Un-indent line(s) rigidly	<ul style="list-style-type: none"><li><b>&lt;backtab&gt;</b></li><li><b>&lt;f6&gt; &lt;backtab&gt;</b></li><li><b>&lt;f11&gt; &lt;tab&gt; C</b></li></ul>	<b>(pel-unindent-lines &amp;optional N)</b>	<ul style="list-style-type: none"><li>Un-indent current line or marked lines by N indentation levels.</li></ul>																																																																								
	<ul style="list-style-type: none"><li>Works with point is anywhere on the line.</li><li>All lines touched by the region are un-indented.</li><li>If region was marked, the function does not deactivate it to allow repeated execution of the command.</li><li>If a region was marked, the function does not deactivate it to allow repeated execution of the command. It also modifies the region to include all characters in all affected lines</li><li>Use <b>C-g</b> to de-activate the region.</li><li>Handles presence of hard tabs:<ul style="list-style-type: none"><li>If indent-tabs-mode is non-nil the indentation is created with a mix of hard-tabs and space characters.</li><li>If indent-tabs-mode is nil, any hard tab in the indentation of the marked lines is replaced by the proper number of spaces. Hard tabs after first non-whitespace character on the line are left.</li></ul></li></ul>																																																																										
<a href="#">Indent-tools</a>	The <a href="#">indent-tools</a> external package provides several commands to indent, un-indent and navigate across indented text levels. <ul style="list-style-type: none"><li>It provides a minor mode and a key <a href="#">hydra</a> that provides all of these commands.</li></ul>  The <a href="#">indent-tools</a> external package  PEL activates it when the <b>pel-use-indent-tools</b> user-option is turned on (set to <b>t</b> ). <ul style="list-style-type: none"><li>This also automatically activates the <a href="#">hydra</a> external package.</li></ul>  PEL provide a global key binding to its key <a href="#">hydra</a> and provides the ability to activate the proposed key binding globally and for python mode: <ul style="list-style-type: none"><li><b>pel-indent-tools-key-bound</b> : activates the <b>C-c &gt;</b> key binding either globally or for python-mode only.</li></ul>																																																																										
Open the indent-tools hydra  See also: <a href="#">🔗 I - Python</a>	<ul style="list-style-type: none"><li><b>&lt;f11&gt; &lt;tab&gt; &lt;f7&gt;</b></li><li><b>&lt;f7&gt; &lt;tab&gt;</b></li><li><b>C-c &gt;</b></li></ul>	<b>(indent-tools-hydra/body)</b>	Activate the "indent-tools-hydra" hydra.   With PEL, this key binding is only available when: <ul style="list-style-type: none"><li>globally, when <b>pel-indent-tools-key-bound</b> is set to <b>globally</b>,</li><li>in python-mode only when <b>pel-indent-tools-key-bound</b> is set to <b>python</b>.</li><li>The actual key is selected by indent-tools <b>indent-tools-keymap-prefix</b> user-option, the default is <b>C-c &gt;</b></li></ul>																																																																								
See also: <a href="#">🔗 Hide/Show</a>	<div>The heads for the associated hydra are:<table><tr><td>&gt;:</td><td>'indent-tools-indent',</td></tr><tr><td>&lt;:</td><td>'indent-tools-demote',</td></tr><tr><td>E:</td><td>'indent-tools-indent-end-of-defun',</td></tr><tr><td>c:</td><td>'indent-tools-comment',</td></tr><tr><td>U:</td><td>'indent-tools-uncomment',</td></tr><tr><td>P:</td><td>'indent-tools-indent-paragraph',</td></tr><tr><td>l:</td><td>'indent-tools-indent-end-of-level',</td></tr><tr><td>K:</td><td>'indent-tools-kill-tree',</td></tr><tr><td>C:</td><td>'indent-tools-copy-hydra/body',</td></tr><tr><td>s:</td><td>'indent-tools-select',</td></tr><tr><td>e:</td><td>'indent-tools-goto-end-of-tree',</td></tr><tr><td>u:</td><td>'indent-tools-goto-parent',</td></tr><tr><td>d:</td><td>'indent-tools-goto-child',</td></tr><tr><td>S:</td><td>'indent-tools-select-end-of-tree',</td></tr><tr><td>n:</td><td>'indent-tools-goto-next-sibling',</td></tr><tr><td>p:</td><td>'indent-tools-goto-previous-sibling',</td></tr><tr><td>i:</td><td>'helm-imenu',</td></tr><tr><td>j:</td><td>'forward-line',</td></tr><tr><td>k:</td><td>'previous-line',</td></tr><tr><td>SPC:</td><td>'indent-tools-indent-space',</td></tr><tr><td>_:</td><td>'undo-tree-undo',</td></tr><tr><td>L:</td><td>'recenter-top-bottom',</td></tr><tr><td>f:</td><td>'yafolding-toggle-element',</td></tr><tr><td>q:</td><td>exit</td></tr></table></div> <div><table><tr><th>Indent</th><th>Navigation</th><th>Actions</th></tr><tr><td>&gt; indent</td><td>j v</td><td>K kill</td></tr><tr><td>&lt; de-indent</td><td>k A</td><td>i imenu</td></tr><tr><td>l end of level</td><td>n next sibling</td><td>C Copy...</td></tr><tr><td>E end of fn</td><td>p previous sibling</td><td>c comment</td></tr><tr><td>P paragraph</td><td>u up parent</td><td>U uncomment (paragraph)</td></tr><tr><td>SPC space</td><td>d down child</td><td>f fold</td></tr><tr><td>_ undo</td><td>e end of tree</td><td>q quit</td></tr></table>  The <b>f</b> key toggles the element folding. Press once to hide the sub-tree, press-again to display it back.</div>			>:	'indent-tools-indent',	<:	'indent-tools-demote',	E:	'indent-tools-indent-end-of-defun',	c:	'indent-tools-comment',	U:	'indent-tools-uncomment',	P:	'indent-tools-indent-paragraph',	l:	'indent-tools-indent-end-of-level',	K:	'indent-tools-kill-tree',	C:	'indent-tools-copy-hydra/body',	s:	'indent-tools-select',	e:	'indent-tools-goto-end-of-tree',	u:	'indent-tools-goto-parent',	d:	'indent-tools-goto-child',	S:	'indent-tools-select-end-of-tree',	n:	'indent-tools-goto-next-sibling',	p:	'indent-tools-goto-previous-sibling',	i:	'helm-imenu',	j:	'forward-line',	k:	'previous-line',	SPC:	'indent-tools-indent-space',	_:	'undo-tree-undo',	L:	'recenter-top-bottom',	f:	'yafolding-toggle-element',	q:	exit	Indent	Navigation	Actions	> indent	j v	K kill	< de-indent	k A	i imenu	l end of level	n next sibling	C Copy...	E end of fn	p previous sibling	c comment	P paragraph	u up parent	U uncomment (paragraph)	SPC space	d down child	f fold	_ undo	e end of tree	q quit
>:	'indent-tools-indent',																																																																										
<:	'indent-tools-demote',																																																																										
E:	'indent-tools-indent-end-of-defun',																																																																										
c:	'indent-tools-comment',																																																																										
U:	'indent-tools-uncomment',																																																																										
P:	'indent-tools-indent-paragraph',																																																																										
l:	'indent-tools-indent-end-of-level',																																																																										
K:	'indent-tools-kill-tree',																																																																										
C:	'indent-tools-copy-hydra/body',																																																																										
s:	'indent-tools-select',																																																																										
e:	'indent-tools-goto-end-of-tree',																																																																										
u:	'indent-tools-goto-parent',																																																																										
d:	'indent-tools-goto-child',																																																																										
S:	'indent-tools-select-end-of-tree',																																																																										
n:	'indent-tools-goto-next-sibling',																																																																										
p:	'indent-tools-goto-previous-sibling',																																																																										
i:	'helm-imenu',																																																																										
j:	'forward-line',																																																																										
k:	'previous-line',																																																																										
SPC:	'indent-tools-indent-space',																																																																										
_:	'undo-tree-undo',																																																																										
L:	'recenter-top-bottom',																																																																										
f:	'yafolding-toggle-element',																																																																										
q:	exit																																																																										
Indent	Navigation	Actions																																																																									
> indent	j v	K kill																																																																									
< de-indent	k A	i imenu																																																																									
l end of level	n next sibling	C Copy...																																																																									
E end of fn	p previous sibling	c comment																																																																									
P paragraph	u up parent	U uncomment (paragraph)																																																																									
SPC space	d down child	f fold																																																																									
_ undo	e end of tree	q quit																																																																									



Description	Keystroke	Function	Note
<a href="#">dtrt-indent</a>  ★★	<div>  The <a href="#">dtrt-indent</a> external package            <a href="#">pel-use-dtrt-indent</a> </div> <ul style="list-style-type: none"> <li>A very useful global minor mode. When activated by <b>pel-use-dtrt-indent</b>, when you open a file it analyzes the indentation scheme and use of hard tabs inside the file. It compares it to the indentation width imposed by the mode specific indentation settings imposed by the appropriate customization user-options for the mode. If the current file indentation scheme differs from what you request for new files, it adjusts the settings to the current content of the file, adapting to the current style used in the files and prints a message showing what it changed.</li> </ul> <div>  This is extremely useful when editing <i>foreign files</i>, files written by others, that use an indentation scheme that differs from what you use when you create your own files:           <ul style="list-style-type: none"> <li>All new files you create will use the indentation width and use of hard-tabs you identified in your PEL customizable user-options.</li> <li>When you open a file written by others, files that use a different indentation scheme, <code>dtrt-indent-mode</code> will automatically detect the difference and will adjust the relevant user-options variables to adapt to the style, you won't have to modify your PEL settings for these files.               <ul style="list-style-type: none"> <li>Normally these <i>foreign files</i> should be files located inside repositories of foreign groups.</li> </ul> </li> <li>Note that inside a development project and a development group you should use a consistent style and that style should be identified by your PEL customization for the relevant modes. But at least now you have a way to deal with the other files and that does not require you to continuously changing your PEL customization to do it.</li> </ul> </div> <p><code>dart-indent</code> also provides <code>command</code> to show how it infers the indentation and use of hard tabs and commands you can use to force a different indentation with or less the indentation width and <code>hard--tab</code> use again. Use that after you modify the file's content.</p>		
Show indentation style detection data	<b>&lt;f11&gt; &lt;tab&gt; d ?</b>	( <a href="#">dtrt-indent-diagnosis</a> )	Show guess indentation for the current buffer and output diagnostics.
Guess indentation scheme again	<b>&lt;f11&gt; &lt;tab&gt; d d</b>	( <a href="#">dtrt-indent-try-set-offset</a> )	Try adjusting the current buffer's indentation offset. <ul style="list-style-type: none"> <li>Use this after modifying the file's content, manually adjusting the indentation to the with and style you want.</li> </ul>
Force a new indentation width	<b>&lt;f11&gt; &lt;tab&gt; d D</b>	( <a href="#">dtrt-indent-set</a> INDENT)	Prompt for a new indentation offset (width) and force the indentation offset for the current buffer to INDENT.
<a href="#">indent-bars</a>	<ul style="list-style-type: none"> <li>A minor-mode that displays vertical bars over each indentation level. The style of the bars is customizable.</li> </ul> <div>  The <a href="#">indent-bars</a> external package            PEL activates it when the <b>pel-use-indent-bars</b> user-option is turned on (set to t).         </div>		
Toggle indent-bars	<b>&lt;f11&gt; &lt;tab&gt; b b</b>	( <a href="#">indent-bars-mode</a> &optional ARG)	Toggle showing bars that indicate indentation. A minor mode.
Reset indent bar config	<b>&lt;f11&gt; &lt;tab&gt; b r</b>	( <a href="#">indent-bars-reset</a> &rest R)	Reset indent-bars config.
Reset style of indent bars	<b>&lt;f11&gt; &lt;tab&gt; b s</b>	( <a href="#">indent-bars-reset-styles</a> &rest R)	Reset all styles' colors and faces. <ul style="list-style-type: none"> <li>Useful for calling after theme changes.</li> </ul>
<a href="#">Smart-shift</a>  ⓘ Customize with: <b>&lt;f11&gt; &lt;tab&gt; s &lt;f2&gt;</b>	<p>The <a href="#">smart-shift</a> external package simplifies shifting a complete line or region of lines right or left but also up or down.</p> <ul style="list-style-type: none"> <li>It is implemented as a minor or global minor mode that must be enabled first.</li> </ul> <ul style="list-style-type: none"> <li>Automatically activate the smart-shift-mode in specified major mode by customizing the <b>pel-&lt;mode&gt;-activates-minor-modes</b> user-options.</li> <li>You can also use the commands manually or through the key bindings provided by PEL to activate the smart-shift-mode in the current buffer or globally for all buffers.</li> <li>PEL controls it through customization user-options:           <div>  The <a href="#">smart-shift</a> external package              PEL activates it when the <b>pel-use-smart-shift</b> user-option is turned on (set to t).           </div> </li> <li>PEL also provides the <b>pel-smart-shift-keybinding</b> user-option that allows you to select whether the shift keys used by smart-shift mode is the default provided keys only or whether you also want to activate another set.</li> <li>The default are always available when smart-shift mode is active: <b>C-c &lt;right&gt;</b>, <b>C-c &lt;left&gt;</b>, <b>C-c &lt;up&gt;</b> and <b>C-c &lt;down&gt;</b>.</li> <li>PEL can also activate <b>one</b> of the following extra key binding sets:           <ul style="list-style-type: none"> <li>Using the control cursor key : <b>C-c C-&lt;right&gt;</b>, <b>C-c C-&lt;left&gt;</b>, <b>C-c C-&lt;up&gt;</b> and <b>C-c C-&lt;down&gt;</b>.</li> <li>Using the <b>&lt;f9&gt;</b> key as prefix: <b>&lt;f9&gt; &lt;right&gt;</b> , <b>&lt;f9&gt; &lt;left&gt;</b> , <b>&lt;f9&gt; &lt;up&gt;</b> and <b>&lt;f9&gt; &lt;down&gt;</b></li> </ul> </li> </ul>		
Toggle smart-shift mode in current buffer	<b>&lt;f11&gt; &lt;tab&gt; s</b>	( <a href="#">smart-shift-mode</a> &optional ARG)	Activate/de-activate the smart-shift mode in the current buffer. <ul style="list-style-type: none"> <li>Activate the line-shift key bindings listed below, in the current buffer.               <ul style="list-style-type: none"> <li>With PEL, the actual key binding selected for the line shift commands depend on the value of the <b>pel-smart-shift-keybinding</b> user-option.</li> </ul> </li> </ul>
Toggle smart-shift mode globally	<b>&lt;f11&gt; &lt;tab&gt; S</b>	( <a href="#">global-smart-shift-mode</a> &optional ARG)	<ul style="list-style-type: none"> <li>Toggle Smart-Shift mode in all buffers.</li> <li>With prefix ARG, enable Global Smart-Shift mode if ARG is positive; otherwise, disable it.</li> <li>Smart-Shift mode is enabled in all buffers where 'smart-shift-mode-on' would do it.</li> </ul>
When smart-shift mode is active:	 As described above, with PEL only <b>one</b> of the extra key bindings provided by PEL can be enabled via the <b>pel-smart-shift-keybinding</b> user-option. So unlike other key binding description cells in this and other tables, only one of the last 2 key bindings is available in the smart-shift minor mode.		
Shift line or region right	<ul style="list-style-type: none"> <li><b>C-c &lt;right&gt;</b></li> <li><b>C-c C-&lt;right&gt;</b></li> <li><b>&lt;f9&gt; &lt;right&gt;</b></li> </ul>	( <a href="#">smart-shift-right</a> &optional ARG)	Shift the line or region to the ARG times to the right.
Shift line or region left	<ul style="list-style-type: none"> <li><b>C-c &lt;left&gt;</b></li> <li><b>C-c C-&lt;left&gt;</b></li> <li><b>&lt;f9&gt; &lt;left&gt;</b></li> </ul>	( <a href="#">smart-shift-left</a> &optional ARG)	Shift the line or region to the ARG times to the left.
Shift line or region up	<ul style="list-style-type: none"> <li><b>C-c &lt;up&gt;</b></li> <li><b>C-c C-&lt;up&gt;</b></li> <li><b>&lt;f9&gt; &lt;up&gt;</b></li> </ul>	( <a href="#">smart-shift-up</a> &optional ARG)	Shift the line or region to the ARG times to the upwards.
Shift line or region down	<ul style="list-style-type: none"> <li><b>C-c &lt;down&gt;</b></li> <li><b>C-c C-&lt;down&gt;</b></li> <li><b>&lt;f9&gt; &lt;down&gt;</b></li> </ul>	( <a href="#">smart-shift-down</a> &optional ARG)	Shift the line or region to the ARG times to the downwards
<a href="#">smart-tabs</a>	<div>  The <a href="#">smart-tabs</a> external package            PEL activates it when the <b>pel-use-smart-tabs</b> user-option is turned on (set to t).         </div> <ul style="list-style-type: none"> <li>See <a href="#">Smart-Tabs Emacs Wiki page</a> for more information.</li> </ul>		
Toggle smart-tabs mode	<b>&lt;f11&gt; &lt;tab&gt; M-s</b>	( <a href="#">smart-tabs-mode</a> &optional ARG)	Toggle smart-tabs minor mode. <ul style="list-style-type: none"> <li>Intelligently indent with tabs, align with spaces!</li> </ul>

Indentation — References

Title & URL		Description				
<u>Understanding GNU Emacs and Tabs</u>		Overview description of how Emacs handle the Tab key, often used for strict indentation in many editors. In Emacs it can do much more.				
<u>GNU Emacs Manual - Indentation</u>						
<u>GNU Emacs Manual - Indentation for Programs</u>						
<u>Indentation Basic Concepts Tutorial @ XEmacs</u>		A tutorial on indentation written by <a href="#">KaiGrossjohann</a>				
<u>Tabs or space for indentation??</u>						
<u>Indentation and hard-tab use Style</u>		<u>Advantages</u>	<u>Disadvantages</u>	<u>Popularity</u>	<u>Support by PEL</u>	<u>Supported by</u>
<ul style="list-style-type: none"><li>• Use hard-tabs for indentation.</li><li>• Uncontrolled use of space and hard-tabs for indentation.</li></ul>		Smaller file size	May render indentation differently on printer and sites such as Github. The code may look misaligned when not using the proper rendering width for hard-tab characters.	Often used for C, Emacs Lisp and languages used by users of programming editors that can easily deal with this, such as Emacs.	Yes	<ul style="list-style-type: none"><li>• PEL mode-specific customizable user-options that are stored into the Emacs relevant indentation-controlling variables.</li></ul>
<ul style="list-style-type: none"><li>• Use hard-tabs for indentation.</li><li>• Controlled use of hard-tabs for indentation, spaces for alignment.</li></ul>		<ul style="list-style-type: none"><li>• Ease of indentation width.</li><li>• Visual rendering of indentation can be narrowed or widened by simply changing the visual rendering (the width) of a hard-tab character.</li><li>• Smaller file size.</li></ul>	May render indentation differently on printer and sites such as Github.	Used by Go and some project that use C and some other programming languages.	Yes	<ul style="list-style-type: none"><li>• PEL mode-specific customizable user-options that are stored into the Emacs relevant indentation-controlling variables.</li><li>• <a href="#">smart-tabs</a></li><li>• Adjust use of hard-tabs and indentation width settings used by foreign files when <a href="#">dtrt-indent</a> is used (via <b>pel-use-dtrr-indent</b>).</li></ul>
Use spaces only. No hard tabs.		The rendering is the same everywhere; on all printers and even on Github which botches support for hard tabs.	<ul style="list-style-type: none"><li>• Cannot adjust visual width rendering of indentation, a useful feature for visually challenged people: the indentation with is directly tied to the file content.</li><li>• Larger file size.</li></ul>	Widely popular strategy for most programming languages as of 2025, despite the growing concern for the use of small indentation width (2 mostly) that is difficult to see for some people.	Yes	<ul style="list-style-type: none"><li>• PEL mode-specific customizable user-options that are stored into the Emacs relevant indentation-controlling variables.</li><li>• Adjust use of hard-tabs and indentation width settings used by foreign files when <a href="#">dtrt-indent</a> is used (via <b>pel-use-dtrr-indent</b>).</li></ul>
<u>Smarttabs @ GitHub</u>		Starttabs source code repository.				
<u>Indentation Styles for Curly Bracket Languages</u>						
<u>Indentation Styles @ Wikipedia</u>						
<u>StackOverflow - Emacs BSD/Allman Style with 4 Space Tabs?</u>						
<u>GNU Emacs Manual - Styles</u>						
<u>Emacs BSD/Allman Style with 4 Space Tabs?</u>						
<u>Emacs: Linux Kernel Style but with Allman/BSD Style Braces?</u>						
<u>Emacs Wiki - Indenting C</u>						
<u>Indent preprocessor directives as C code in emacs</u>		Does not fully address the way I want to have multi-indentations for pre-processor				
<u>elisp code - ppindent.el</u>		Implements pre-processor indentation with the # always in the first column. Not yet exactly what I want.				
<u>Demystify C++ Metaprograms using Emacs</u>						
<u>Programming in C++, Rules and Recommendations</u>		ellemtel style				