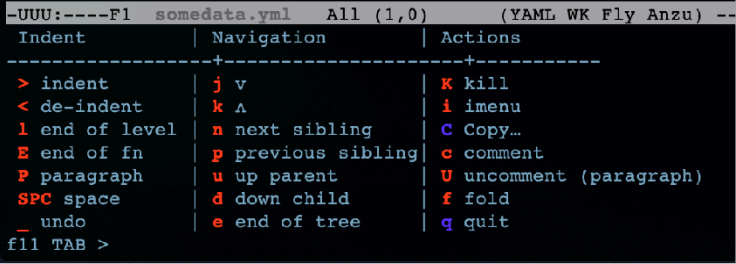











CWL - Common Workflow Language - Data Serialization Support

Operation	Keystroke	Function	Note
<div>Editing <a href="#">CWL</a> files</div> <div>See also: <a href="#">📄 YAML</a></div>	<div>The CWL format is a <a href="#">subset</a> of <a href="#">YAML</a>. The external package <a href="#">cwl-mode</a> provides a major mode for CWL files. The minor modes and commands that help manage <a href="#">YAML</a> files also help manage CWL files.</div> <ul style="list-style-type: none"><li>Aside from the first 2 key bindings listed to access help and customization buffers for <a href="#">YAML</a>, the key bindings listed in this page and their related commands are also described in other <a href="#">PEL</a> PDF pages. The links to these pages are on the first column.</li></ul> <div>📦 The <a href="#">cwl-mode</a> external package provides a major mode support for CWL. <a href="#">🔗</a> <a href="#">PEL</a> provides access to it when the <a href="#">pel-use-cwl</a> user-option is turned on (set to <a href="#">t</a>).</div> <ul style="list-style-type: none"><li><a href="#">PEL</a> associates the following file extensions with <a href="#">cwl-mode</a>: <a href="#">.cwl</a>.</li></ul>		
<div>Open this PDF file.</div> <div>See also: <a href="#">🔗 Help/Info</a></div>	<div>&lt;f11&gt; SPC M-c &lt;f1&gt;</div> <div>&lt;f12&gt; &lt;f1&gt;</div>	<div>(<a href="#">pel-help-pdf</a> &amp;optional OPEN-WEB-PAGE)</div>	Open the <a href="#">M</a> <a href="#">CWL</a> local PDF. If the prefix argument (like <a href="#">C-u</a> or <a href="#">M--</a> ) is used, then it opens the remote GitHub hosted raw PDF instead. If the <a href="#">pel-flip-help-pdf-arg</a> user-option is set it's the other way around.
<div>🔗 Customize <a href="#">PEL</a> CWL control</div>	<div>&lt;f11&gt; SPC M-c &lt;f2&gt;</div> <div>&lt;f12&gt; &lt;f2&gt;</div>	<div>(<a href="#">pel-customize-pel</a> &amp;optional OTHER-WINDOW)</div>	Customize <a href="#">PEL</a> CWL support. <ul style="list-style-type: none"><li>If <a href="#">OTHER-WINDOW</a> is non-nil (use <a href="#">C-u</a>), display in other window.</li></ul>
<div>🔗 Customize Emacs CWL control</div>	<div>&lt;f11&gt; SPC M-c &lt;f3&gt;</div> <div>&lt;f12&gt; &lt;f3&gt;</div>	<div>(<a href="#">pel-customize-library</a> &amp;optional OTHER-WINDOW)</div>	Customize Emacs CWL support groups: <a href="#">yaml</a> , <a href="#">fly check</a> , <a href="#">indent-tools</a> and <a href="#">smartparens</a> <ul style="list-style-type: none"><li>If <a href="#">OTHER-WINDOW</a> is non-nil (use <a href="#">C-u</a>), display in other window.</li></ul>
<div><a href="#">Flycheck</a></div> <div>See also: <a href="#">🔗 SyntaxCheck</a></div>	<div><a href="#">Flycheck</a> is a minor mode for on-the-fly syntax checking.</div> <div>📦 The <a href="#">flycheck</a> external package <a href="#">🔗</a> is activated by <a href="#">PEL</a> when the <a href="#">pel-use-flycheck</a> user-option is turned on or another activated <a href="#">PEL</a> user-option requires it.</div> <div>🔧 Aside from the following 2 key bindings that <a href="#">PEL</a> provides to toggle the <a href="#">flycheck</a> mode, <a href="#">flycheck</a> key prefix is <a href="#">C-c</a> ! as set by its <a href="#">flycheck-keymap-prefix</a> user-option. You can change it for a different key prefix.</div>		
Toggle flycheck mode for current buffer	<f11> ! !	( <a href="#">flycheck-mode</a> &optional ARG)	Toggle flycheck minor-mode for the current buffer.
Toggle flycheck mode for all buffers	<f11> ! M-!	( <a href="#">global-flycheck-mode</a> &optional ARG)	Toggle Flycheck mode in all buffers. <ul style="list-style-type: none"><li>Flycheck mode is enabled in all buffers where ‘flycheck-mode-on-safe’ would do it.</li></ul>
• <a href="#">Flycheck</a> buffer/file			
Syntax Check current buffer	C-c ! c	( <a href="#">flycheck-buffer</a> )	Start checking syntax in the current buffer. <ul style="list-style-type: none"><li>Get a syntax checker for the current buffer with ‘flycheck-get-checker-for-buffer’, and start it.</li></ul>
Check syntax of current file	C-c ! C-c	( <a href="#">flycheck-compile</a> CHECKER)	Run CHECKER via ‘compile’. <ul style="list-style-type: none"><li>CHECKER must be a valid syntax checker. Interactively, prompt for a syntax checker to run.</li><li>Instead of highlighting errors in the buffer, this command pops up a separate buffer with the entire output of the syntax checker tool, just like ‘compile’.</li></ul>
<div>Highlight current column</div>	<div>The following command provide a vertical line across the entire window at the cursor location.</div> <ul style="list-style-type: none"><li>Useful when creating tables or checking indentation manually.</li><li><a href="#">vline</a> also provides the <a href="#">vline-global-mode</a> to activate the vertical line in all buffers; <a href="#">PEL</a> has no binding for it because it slows Emacs too much.</li></ul>		
<div>Toggle <a href="#">Vline</a> Mode</div> <div>See also:</div> <ul style="list-style-type: none"><li>🔗 <a href="#">Highlight</a></li><li>🔗 <a href="#">Hide/Show</a></li></ul>	<ul style="list-style-type: none"><li>&lt;f11&gt; h  </li><li>&lt;f11&gt; 9</li></ul>	( <a href="#">vline-mode</a> &optional ARG)	Toggle the display of a vertical line spanning the entire window at the cursor column. <div>📦 Requires: <a href="#">vline.el</a> <a href="#">🔗</a> <a href="#">PEL</a> activates it when <a href="#">pel-use-vline</a> user option is <a href="#">t</a>.</div>
<div>Indented Text Folding</div>	<div>The following command folds (hide or show) all lines that are indented more than the current line.</div> <ul style="list-style-type: none"><li>You can also use the <a href="#">f</a> key inside the <a href="#">indent-tools</a> Hydra, shown below, to fold indented sections.</li></ul>		
<div>Toggle hiding lines more indented than current line</div> <div>See also: <a href="#">🔗 Hide/Show</a></div>	<f11> M-/ M-/	( <a href="#">pel-toggle-hide-indent</a> )	Toggle hiding lines more indented than current line. <ul style="list-style-type: none"><li>Affects the entire buffer. Not syntax sensitive. Can be used anywhere.</li></ul> <div>⚠️ Do not modify the buffer while lines are hidden, it's allowed but its using selective display and you don't see what you change.</div>
<div><a href="#">Indent-tools</a></div>	<div>The <a href="#">indent-tools</a> external package provides several commands to indent, un-indent and navigate across indented text levels.</div> <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> <div>📦 The <a href="#">indent-tools</a> external package <a href="#">🔗</a> <a href="#">PEL</a> activates it when the <a href="#">pel-use-indent-tools</a> user-option is turned on (set to <a href="#">t</a>).</div> <ul style="list-style-type: none"><li>This also automatically activates the <a href="#">hydra</a> external package.</li></ul> <div>🔧 <a href="#">PEL</a> 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:</div> <ul style="list-style-type: none"><li><a href="#">pel-indent-tools-key-bound</a> : activates the <a href="#">C-c</a> &gt; key binding either globally or for python-mode only.</li></ul>		
<div>Open the indent-tools hydra</div> <div>See also: <a href="#">🔗 Indentation</a></div> <div>The indent-tools hydra provide keys you can use to navigate across the indented CWL elements.</div> <div>See also: <a href="#">🔗 Hide/Show</a></div>	<div>&lt;f11&gt; &lt;tab&gt; &gt;</div> <div>C-c &gt;</div>	<div>(<a href="#">indent-tools-hydra/body</a>)</div>	<div>Activate the e body in the "indent-tools-hydra" hydra.</div> <div>🔧 With <a href="#">PEL</a>, this key binding is only available when:</div> <ul style="list-style-type: none"><li>globally, when <a href="#">pel-indent-tools-key-bound</a> is set to <a href="#">globally</a>,</li><li>in python-mode only when <a href="#">pel-indent-tools-key-bound</a> is set to <a href="#">python</a>.</li><li>The actual key is selected by indent-tools <a href="#">indent-tools-keymap-prefix</a> user-option, the default is <a href="#">C-c</a> &gt;</li></ul> <div></div> <div>👉 The <a href="#">f</a> key toggles the element folding. Press once to hide the sub-tree, press-again to display it back.</div>

Operation	Keystroke	Function	Note
<b>Smartparens Mode</b> • <a href="#">Smartparens manual</a>  See also: <a href="#">↗ ↘ Smartparens</a>	Simplify insertion of matching pairs with the <a href="#">smartparens</a> minor mode. PEL binds a set of keys, described below, to toggle activation of that mode.  This uses the <a href="#">smartparens</a> external package.  PEL activates it when <a href="#">pel-use-smartparens</a> is set to <b>t</b> . <ul style="list-style-type: none"> <li>Smartparens enhances the behaviour of certain keys, namely those that are part of any pair or tag.</li> <li>Mode line lighter: smartparens-mode: <b>SP</b> smartparens-strict-mode: <b>SP/s</b></li> </ul>		
Help on smartparens	<f11> ( ?	(sp-cheat-sheet &optional ARG)	Generate a cheat sheet of all the smartparens interactive functions. Shows inside Emacs buffer. <ul style="list-style-type: none"> <li>Without a prefix argument, print only the short documentation and examples.</li> <li>With non-nil prefix argument ARG, show the full documentation for each function.</li> <li>You can follow the links to the function or variable help page. <ul style="list-style-type: none"> <li>To get back to the full list, use M-x help-go-back.</li> </ul> </li> <li>You can use 'beginning-of-defun' and 'end-of-defun' to jump to the previous/next entry.</li> <li>Examples are fontified using the 'font-lock-string-face' for better orientation.</li> </ul>
Describe user system	<f11> ( M-?	(sp-describe-system STARTERKIT)	Describe user's system. Prompt for starter kit: Evil, Spacemacs, Vanilla. <ul style="list-style-type: none"> <li>The output of this function can be used in bug reports.</li> </ul>
Toggle smartparens mode	<f11> ( (	(smartparens-mode &optional ARG)	Toggle smartparens mode.
Toggle smartparens-strict mode	<f11> ( )	(smartparens-strict-mode &optional ARG)	Toggle the strict smartparens mode. <ul style="list-style-type: none"> <li>When strict mode is active, 'delete-char', 'kill-word' and their backward variants will skip over the pair delimiters in order to keep the structure always valid (the same way as 'paredit-mode' does). This is accomplished by remapping them to 'sp-delete-char' and 'sp-kill-word'. There is also function 'sp-kill-symbol' that deletes symbols instead of words, otherwise working exactly the same (it is not bound to any key by default).</li> <li>When strict mode is active, this is indicated with "/s" after the smartparens indicator in the mode list</li> </ul>
Toggle smartparens mode	<f11> ( M-(	(smartparens-global-mode &optional ARG)	Toggle Smartparens mode in all buffers. <ul style="list-style-type: none"> <li>With prefix ARG, enable Smartparens-Global mode if ARG is positive; otherwise, disable it.</li> <li>Smartparens mode is enabled in all buffers where 'turn-on-smartparens-mode' would do it.</li> </ul>
Toggle smartparens-strict mode	<f11> ( M-)	(smartparens-global-strict-mode &optional ARG)	Toggle Smartparens-Strict mode in all buffers. <ul style="list-style-type: none"> <li>With prefix ARG, enable Smartparens-Global-Strict mode if ARG is positive; otherwise, disable it.</li> <li>Smartparens-Strict mode is enabled in all buffers where 'turn-on-smartparens-strict-mode' would do it.</li> </ul>
<b>Smart-shift</b>  See also: <a href="#">↗ Indentation</a>	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. <ul style="list-style-type: none"> <li>It is implemented as a minor or global minor mode that must be enabled first. You can identify the smart-shift-mode inside one of the pel-&lt;mode&gt;-activates-minor-modes user-options to activate it automatically. 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: <ul style="list-style-type: none"> <li> The <a href="#">smart-shift</a> external package  PEL activates it when the <a href="#">pel-use-smart-shift</a> user-option is turned on (set to t).</li> <li> PEL also provides the <a href="#">pel-smart-shift-keybinding</a> user-option that allows you to select additional alternative key bindings for the smart-shift commands that shift line(s). By default the key bindings are using <b>C-c</b> as a key prefix. With PEL you can also use a control key for the cursor or change the prefix key to use the <b>&lt;f9&gt;</b> key. The 3 possible key bindings are shown below but only one of them will be available at any given time. The one available is the one selected by the user-option value.</li> </ul> </li> </ul>		
Toggle smart-shift mode in current buffer	<f11> <tab> s	(smart-shift-mode &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	<f11> <tab> S	(global-smart-shift-mode &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>
Shift line or region right	<ul style="list-style-type: none"> <li>C-c &lt;right&gt;</li> <li>C-c &lt;C-right&gt;</li> <li>&lt;f9&gt; &lt;right&gt;</li> </ul>	(smart-shift-right &optional ARG)	Shift the line or region to the ARG times to the right.  With PEL <b>one</b> of the extra key bindings can be enabled via the <b>pel-smart-shift-keybinding</b> user-option. So unlike other cells only one of the last 2 key bindings is available in the smart-shift minor mode.
Shift line or region left	<ul style="list-style-type: none"> <li>C-c &lt;left&gt;</li> <li>C-c &lt;C-left&gt;</li> <li>&lt;f9&gt; &lt;left&gt;</li> </ul>	(smart-shift-left &optional ARG)	Shift the line or region to the ARG times to the left.  With PEL <b>one</b> of the extra key bindings can be enabled via the <b>pel-smart-shift-keybinding</b> user-option. So unlike other cells only one of the last 2 key bindings is available in the smart-shift minor mode.
Shift line or region up	<ul style="list-style-type: none"> <li>C-c &lt;up&gt;</li> <li>C-c &lt;C-up&gt;</li> <li>&lt;f9&gt; &lt;up&gt;</li> </ul>	(smart-shift-up &optional ARG)	Shift the line or region to the ARG times to the upwards.  With PEL <b>one</b> of the extra key bindings can be enabled via the <b>pel-smart-shift-keybinding</b> user-option. So unlike other cells only one of the last 2 key bindings is available in the smart-shift minor mode.
Shift line or region down	<ul style="list-style-type: none"> <li>C-c &lt;down&gt;</li> <li>C-c &lt;C-down&gt;</li> <li>&lt;f9&gt; &lt;down&gt;</li> </ul>	(smart-shift-down &optional ARG)	Shift the line or region to the ARG times to the downwards  With PEL <b>one</b> of the extra key bindings can be enabled via the <b>pel-smart-shift-keybinding</b> user-option. So unlike other cells only one of the last 2 key bindings is available in the smart-shift minor mode.

## CWL & Emacs — References

Description & URL	Notes
<a href="#">Common Workflow Language</a>	Common Workflow Language (CWL) uses a <a href="#">subset of YAML</a> and provides YAML supporting tools. <ul style="list-style-type: none"> <li><b>CWL home page</b> <ul style="list-style-type: none"> <li><a href="#">CWL User Guide</a></li> <li><a href="#">CWL YAML Guide</a></li> </ul> </li> </ul>