













# ⚠ Emacs support for the Lua Programming Language ⚠

Description	Keystroke	Function	Note
<a href="#">Lua Editing</a>	<p>Emacs has built-in support for Lua. The Lua-mode is one of the cc-modes.</p> <ul style="list-style-type: none"> <li>Since Lua syntax is very close to C syntax, Emacs implements <b>Lua-mode</b> as a descendent of cc-mode.</li> </ul> <p>PEL provides extra support, described in this table, when  the <b>pel-use-Lua</b> user-option is set to t.</p> <ul style="list-style-type: none"> <li>Most cc-mode available capabilities are available to <b>Lua-mode</b>. PEL integrates a lot of those capabilities, but PEL support for Lua is in its early stages and all available key bindings are not yet identified in this table as they should be. <span>⚠</span></li> </ul>		
Last updated on:	2025-03-19		
<b>Open this PDF file.</b> See also: <a href="#">🔗 Help/Info</a>	<div> <div>&lt;f11&gt; SPC u &lt;f1&gt;</div> <div>&lt;f12&gt; &lt;f1&gt;</div> </div>	<div> <div>(pel-help-pdf &amp;optional OPEN-WEB-PAGE)</div> </div>	Open the <b>🔗 - Lua</b> 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="#">🔗 Customize</a> PEL Lua support	<div> <div>&lt;f11&gt; SPC u &lt;f2&gt;</div> <div>&lt;f12&gt; &lt;f2&gt;</div> </div>	<div> <div>(pel-customize-pel &amp;optional OTHER-WINDOW)</div> </div>	Customize PEL Lua support. <ul style="list-style-type: none"> <li>If OTHER-WINDOW is non-nil (use <b>C-u</b>), display in another window.</li> </ul>
<a href="#">🔗 Customize</a> Emacs Lua support	<div> <div>&lt;f11&gt; SPC u &lt;f3&gt;</div> <div>&lt;f12&gt; &lt;f3&gt;</div> </div>	<div> <div>(pel-customize-library &amp;optional OTHER-WINDOW)</div> </div>	Customize Emacs Lua support (which is currently placed in C group): <b>C</b> <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>Select Lua-mode for extension-less file</b>  The <b>&lt;f12&gt;</b> key is available only until a PEL controlled major mode is activated. Then it becomes a buffer prefix key.	<div> <div>&lt;f12&gt;</div> </div>	<div> <div>(pel-as &amp;optional FORCE)</div> </div>	Inside a fundamental-mode buffer, interactively select major mode for the buffer. Re-do it with arg.  see <a href="#">Create extension-less executable scripts with PEL</a> .
<b>Comments</b>	<p>This command is mostly used to set the major mode of a buffer in fundamental-mode’, when the <b>&lt;f12&gt;</b> key binding is available for it. After being used once in a buffer the major mode is selected and the PEL key binding will not be available when PEL supports the major mode. For Lua file, select <b>Lua</b>. It will insert a shebang line specified by  <b>pel-lua-shebang-line</b> user option. PEL defines the (as &amp;optional FORCE) alias unless  <b>pel-has-alias-as</b> user-option is set to nil. You can use <b>M-x as</b> to invoke it.</p>		
<b>Toggle display of comments in buffer or active region</b> See also: <a href="#">🔗 Comments</a>	<div> <div>&lt;f11&gt; ; ;</div> </div>	<div> <div>(hide/show-comments-toggle &amp;optional START END)</div> </div>	Toggle hiding/showing of comments in the active region or whole buffer. <ul style="list-style-type: none"> <li>If the region is active then toggle in the region. Otherwise, in the whole buffer.</li> </ul> <div>  This requires the <a href="#">hide-comnt.el</a> package (see <a href="#">🔗 Comments</a>).  PEL activates it when the <b>pel-use-hide-comnt</b> user option is t.         </div>
<b>Generic code skeletons</b> <ul style="list-style-type: none"> <li><a href="#">tempo skeletons</a></li> </ul> See also: <ul style="list-style-type: none"> <li><a href="#">🔗 Inserting Text</a></li> <li><a href="#">T Templates</a></li> </ul>	Several mechanisms have been developed to allow easy insertion of predefined text in Emacs. <span>⚠</span> PEL does not yet define skeletons for Lua. You can use the generic one. <ul style="list-style-type: none"> <li>Emacs provides the built-in skeleton mechanism and the <a href="#">tempo skeletons</a>.               <ul style="list-style-type: none"> <li>PEL supports both. They are used a little bit differently. PEL provides <b>generic</b> tempo skeletons you can use for Lua until PEL adds Lua-specific skeletons.                   <ul style="list-style-type: none"> <li>PEL provides key bindings to the tempo skeletons: the generic code templates, accessible via the <b>&lt;f6&gt;</b> prefix key, and the language-specific code templates, accessible via the <b>&lt;f12&gt;</b> key prefix.</li> </ul> </li> </ul> </li></ul>		
<a href="#">🔗 Customize</a> PEL Text Insertions control for Lua code skeletons.	<div> <div>&lt;f6&gt; &lt;f2&gt;</div> <div>&lt;f12&gt; &lt;f12&gt; &lt;f2&gt;</div> </div>	<div> <div>(pel-customize-pel &amp;optional OTHER-WINDOW)</div> <div>(pel-customize-generic-skels &amp;optional OTHER-WINDOW)</div> </div>	Open the customization groups that control the format of the various skeletons including the generic skeleton used by the <b>&lt;f6&gt; h</b> key and the <b>&lt;f12&gt;&lt;f12&gt; h</b> key (see below). <ul style="list-style-type: none"> <li>If OTHER-WINDOW is non-nil (use <b>C-u</b>) , display in other window.</li> </ul>
<b>Insert generic file module header block — Language agnostic</b>  After inserting the template, navigate though areas that must be filled with: <ul style="list-style-type: none"> <li>forward: <b>C-c .</b></li> <li>backward: <b>C-c ,</b></li> </ul>	<div> <div>&lt;f6&gt; h</div> <div>&lt;f12&gt; &lt;f12&gt; h</div> </div>	<div> <div>(pel-generic-file-header)</div> </div>	Insert a file header block at the top of the file. Works only for buffer visiting a file. <div>  The command key binding <b>&lt;f6&gt; h</b> is available only 1 second after Emacs has started.            As mentioned above PEL does not yet define Lua-specific skeletons, this uses the generic one.         </div> <div>  Specify the format of the header via the user-options in the <b>pel-pkg-generic-code-style</b> customization group accessible via <b>&lt;f6&gt; &lt;f2&gt;</b> <ul style="list-style-type: none"> <li>Inside a <b>Lua</b> buffer, <b>&lt;f12&gt; &lt;f2&gt;</b> provides access to the following customization groups:</li> </ul>  After inserting a template, use <b>tempo-forward-mark</b> and <b>tempo-backward-mark</b> to move to the beginning of each section that must be filled.         </div>
<b>Toggle pel-tempo-mode</b>	<div> <div>&lt;f6&gt; SPC</div> <div>&lt;f12&gt; &lt;f12&gt; SPC</div> </div>	<div> <div>(pel-tempo-mode &amp;optional ARG)</div> </div>	Toggle PEL tempo mode on/off. <p>PEL tempo mode activates <b>C-c .</b> and <b>C-c ,</b> , as well as to <b>C-c C-.</b> and <b>C-c C-,</b> key bindings to navigate across tempo mark hot-spots. When pel-tempo-mode is active the pel-tempo-mode lighter (🔗) is shown on the status bar. The second set of keys are only available in graphics mode.</p>  The pel-generic-file-header command inserts the text using a tempo skeleton: the PEL tempo mode is automatically activated by typing <b>&lt;f6&gt; h</b> .
<b>Expand any tag in template</b>  Note: PEL default skeleton does not use tags.	<div> <div>&lt;f6&gt; &lt;f12&gt;</div> <div>&lt;f12&gt; &lt;f12&gt; &lt;f12&gt;</div> </div>	<div> <div>(tempo-complete-tag &amp;optional SILENT)</div> </div>	Look for a tag and expand it. All the tags in the tag lists in ‘ <b>tempo-local-tags</b> ’ (this includes ‘tempo-tags’) are searched for a match for the text before the point. The way the string to match for is determined can be altered with the variable ‘tempo-match-finder’. If ‘tempo-match-finder’ returns nil, then the results are the same as no match at all. <ul style="list-style-type: none"> <li>If a single match is found, the corresponding template is expanded in place of the matching string.</li> <li>If a partial completion or no match at all is found, and SILENT is non-nil, the function will give a signal.</li> <li>If a partial completion is found and ‘tempo-show-completion-buffer’ is non-nil, a buffer containing possible completions is displayed.</li> </ul>

## Emacs & Pike — References

Document	Notes
<b>The Lua Programming Language</b>	<ul style="list-style-type: none"> <li><a href="#">Lua @ Wikipedia</a></li> <li><a href="#">Lua Home</a></li> </ul>