










🚧 Emacs support for the Pike Programming Language 🚧

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

Emacs & Pike — References

Document	Notes
The Pike Programming Language	<ul style="list-style-type: none"> Pike @ Wikipedia Pike Home Pike @ PLEAC - for code examples