

Emacs support for Elixir Programming Language🚧

Description	Keystroke	Function	Note
<h2>Elixir Editing</h2> <p>Elixir support is very basic, it is not yet fully implemented nor fully documented.</p> <p> PEL supports Elixir when pel-use-elixir user options is turned on.</p> <ul style="list-style-type: none">On Emacs >= 30, PEL supports tree-sitter if pel-use-tree-sitter is set to t.You can activate tree-sitter for Elixir by setting pel-use-elixir to 'with-tree-sitter' (as long as pel-use-tree-sitter is t and Emacs >= 30). See 🔗 Tree Sitter <p> PEL supports the following external packages for Elixir support, activated by the following PEL user-options:</p> <p> The elixir-mode external package PEL activates it when pel-use-elixir is turned on (set to t).</p> <p> The alchemist external package PEL activates it when pel-use-alchemist is turned on (set to t).</p> <p> The exunit external package PEL activates it when pel-use-elixir-exunit is turned on (set to t).</p> <p> The lsp-elixir external package PEL activates it when pel-use-elixir-lsp is turned on (set to t).</p> <p>Last updated on: 2025-10-14</p>			
Open this PDF file. See also: 🔗 Help/Info	<f11> SPC x <f1>	(pel-help-pdf &optional OPEN-WEB-PAGE)	Open the 🔗 - Elixir 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.
	<f12> <f1>		
🔗 Customize PEL Elixir support	<f11> SPC x <f2>	(pel-customize-pel &optional OTHER-WINDOW)	Customize PEL Elixir support: alchemist, alchemist-iex. <ul style="list-style-type: none">If OTHER-WINDOW is non-nil (use C-u), display in another window.
	<f12> <f2>		
🔗 Customize Emacs Elixir support	<f11> SPC x <f3>	(pel-customize-library &optional OTHER-WINDOW)	Customize Emacs Elixir support: alchemist, alchemist-iex. <ul style="list-style-type: none">If OTHER-WINDOW is non-nil (use C-u), display in another window.
	<f12> <f3>		
Show PEL setup for Elixir	<f11> SPC g ?	(pel-elixir-setup-info &optional APPEND)	Display Elixir setup information inside a "pel-elixir-info" buffer with buttons providing quick access to the customization buffer of each variable shown. The information shown includes the value and interpretation of: <ul style="list-style-type: none">pel-use-elixir (whether the classic or tree-sitter based major mode is used).the user options controlling indentation and hard tab width rendering. To append information in the buffer instead of clearing the previous content type any prefix argument (such as C-u) before the command keystroke.
	<f12> ?		
<h2>Search Support</h2> <p>In Elixir mode, the superword mode can be useful since snake_case is often used. Using superword-mode helps searching. PEL activates the superword mode by default in Elixir mode. To change this use the <f11> t <f2> to access the customize buffer.</p>			
Toggle superword-mode See also: <ul style="list-style-type: none">🔗 Text Modes🔗 Search/Replace	<f11> t m p	(superword-mode &optional ARG)	Toggle superword-mode: a minor mode that treats snake_case as one word. In Elixir ' _ ' are treated as part of words. <ul style="list-style-type: none">With a prefix argument ARG, enable superword mode if ARG is positive, and disable it otherwise.PEL provides the <f12> M-p key for the programming language modes where snake_case is popular (Emacs Lisp, C, C++, Erlang, Python, etc...)
	<f12> M-p		
<h2>Elixir Shell</h2> <p>The following commands are used to explicitly launch an Elixir shell inside Emacs.</p>			
Start Elixir Shell	<f11> z r x	(alchemist-iex-run &optional ARG)	<ul style="list-style-type: none">Start an IEx process.Show the IEx buffer if an IEx process is already run. <p> Requires elixir-mode and alchemist external packages. PEL activates these when pel-use-elixir and pel-use-alchemist user-options are turned on (set to t).</p>
	<f12> z		

Emacs & Elixir— References

Document	Notes
Elixir	<ul style="list-style-type: none">Elixir @ WikipediaElixir @ home
company-mode : Modular in-buffer completion framework for Emacs	