

Emacs support for Elixir Programming Language🚧

Description	Keystroke	Function	Note
Elixir Editing	Elixir support is very basic, it is not yet fully implemented nor fully documented. 🔧 PEL supports Elixir when pel-use-elixir user options is turned on. - 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 and 🔧 Tree-sitter 📦 PEL supports the following external packages for Elixir support, activated by the following PEL user-options: 📦 The elixir-mode external package 🔧 PEL activates it when pel-use-elixir is turned on (set to t). 📦 The alchemist external package 🔧 PEL activates it when pel-use-alchemist is turned on (set to t). 📦 The exunit external package 🔧 PEL activates it when pel-use-elixir-exunit is turned on (set to t). 📦 The lsp-elixir external package 🔧 PEL activates it when pel-use-elixir-lsp is turned on (set to t).		
Last updated on: 2025-10-15			
Open this PDF file. See also: 🔧 Help/Info	<f11> SPC x <f1> <f12> <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.
🔧 Customize PEL Elixir support	<f11> SPC x <f2> <f12> <f2>	(pel-customize-pel &optional OTHER-WINDOW)	Customize PEL Elixir support: alchemist, alchemist-iex. If OTHER-WINDOW is non-nil (use C-u), display in another window.
🔧 Customize Emacs Elixir support	<f11> SPC x <f3> <f12> <f3>	(pel-customize-library &optional OTHER-WINDOW)	Customize Emacs Elixir support: alchemist, alchemist-iex. If OTHER-WINDOW is non-nil (use C-u), display in another window.
Show PEL setup for Elixir	<f11> SPC g ? <f12> ?	(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: - 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.
Search Support	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.		
Toggle superword-mode See also: 🔧 Text Modes 🔧 Search/Replace	<f11> t m p <f12> 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. - 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...)
Elixir Shell	The following commands are used to explicitly launch an Elixir shell inside Emacs.		
Start Elixir Shell	<f11> z r x <f12> z	(alchemist-iex-run &optional ARG)	- Start an IEx process. - Show the IEx buffer if an IEx process is already run. 📦 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).

Emacs & Elixir— References

Document	Notes
Elixir	Elixir @ Wikipedia Elixir @ home
company-mode : Modular in-buffer completion framework for Emacs	