

Emacs Support for Clojure

Description	Keystroke	Function	Note
Clojure Programming Language Support	 This page is not written yet, except for some basic information. More information will be provided soon.  PEL activates Clojure support when the pel-use-clojure user-option is turned on (t). PEL provide extra support for the Clojure programming language and its various implementations by providing access to the following external packages:  The cider external package.  PEL activates it when the pel-use-cider user-option is turned on (t).  The clj-refactor.el external package.  PEL activates it when the pel-use-clj-refactor user-option is turned on (t).  The clojure-snippets external package.  PEL activates it when the pel-use-clojure-snippets user-option is turned on (t). • PEL provides Lispy support for Clojure when the clojure-mode is added to the list specified by pel-modes-activating-lispy user-option. See ℥l - Lispy		
Open this PDF file. See also: ℥ Help/Info	<f11> SPC C-j <f1> <f12> <f1>	(pel-help-pdf &optional OPEN-WEB-PAGE)	Open the ℥l - Clojure 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 Clojure support	<f11> SPC C-j <f2> <f12> <f2>	(pel-customize-pel &optional OTHER-WINDOW)	Customize PEL Clojure support. If OTHER-WINDOW is non-nil (use C-u), display in another window.
℥ Customize Emacs Clojure support	<f11> SPC C-j <f3> <f12> <f3>	(pel-customize-library &optional OTHER-WINDOW)	Customize Emacs Clojure support: clojure, cider, cljr. If OTHER-WINDOW is non-nil (use C-u), display in another window.