






















Emacs Support for Scheme

Description	Keystroke	Function	Note
Scheme Programming Language Support	<div>  This page is not written yet, except for some basic information. More information will be provided soon. </div> <div>  PEL activates Scheme support when the pel-use-scheme user-option is turned on (t). </div> <div> PEL provide extra support for the Scheme programming language and its various implementations by providing access to the following external packages: </div> <div>  The geyser external package.  PEL activates it when the pel-use-geiser user-option is turned on (t). </div> <div>  The quack external package.  PEL activates it when the pel-use-quack user-option is turned on (t). </div> <div> Emacs built-in Scheme support recognizes the Scheme dialect like Gambit Scheme, Gerbil Scheme and Racket. When pel-use-scheme is turned on PEL provides the <f12> keys for the Scheme buffers for this basic support. If you need more specialized support for the Theme dialect of your choice, PEL also provides some. See the relevant PDF for those, including: </div> <div> <ul style="list-style-type: none">  Gambit  Gerbil  Racket </div> <div> <ul style="list-style-type: none"> PEL provides Lispy support for Scheme when the scheme-mode is added to the list specified by pel-modes-activating-lispy user-option. See img alt="pdf icon" data-bbox="356 218 373 228"/> Lispy </div>		
Open this PDF file. See also: img alt="pdf icon" data-bbox="81 241 98 251"/> Help/Info 	<div>  SPC C-s C-s <f1> </div> <div>  <f12> <f1> </div>	<div>  pel-help-pdf & optional OPEN-WEB-PAGE) </div>	Open the  Scheme 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 Scheme support	<div>  SPC C-s C-s <f2> </div> <div>  <f12> <f2> </div>	<div>  pel-customize-pel & optional OTHER-WINDOW) </div>	Customize PEL Scheme support. <ul style="list-style-type: none"> If OTHER-WINDOW is non-nil (use C-u), display in another window.
 Customize Emacs Scheme support	<div>  SPC C-s C-s <f3> </div> <div>  <f12> <f3> </div>	<div>  pel-customize-library & optional OTHER-WINDOW) </div>	Customize Emacs Scheme support: scheme, geiser, quack. <ul style="list-style-type: none"> If OTHER-WINDOW is non-nil (use C-u), display in another window.