





Emacs Support for Gerbil Scheme 🚧

Description	Keystroke	Function	Note
<div>Gerbil Scheme Programming Language Support</div> <div>See also:</div> <div><ul style="list-style-type: none">🔗 - Scheme🔗 File/Directory Variables</div>	<div> PEL support for Gerbil Scheme is preliminary.</div> <div> PEL activates Gerbil 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 gerbil-mode external package.  PEL activates it when the pel-use-gerbil user-option is turned on (t). Used only for Gerbil Scheme.</div> <div><ul style="list-style-type: none">The Gerbil programming language is a specialized Scheme. The Gerbil files use the same extension as Scheme: .ss.<ul style="list-style-type: none">To activate the gerbil-mode automatically for Gerbil files, it is customary to use Emacs file variable to identify the mode: the first line of the file should have the following text: <code>;; -*- Gerbil -*-</code>The gerbil-mode package can also be installed by installing Gerbil.PEL provides Lispy support for Gerbil Scheme when the gerbil-mode is added to the list specified by pel-modes-activating-lispy user-option.<ul style="list-style-type: none">See 🔗 - Lispy</div>		
<div>Open this PDF file.</div> <div>See also: 🔗 Help/Info</div>	<div><f11> SPC C-s C-e <f1></div> <div><f12> <f1></div>	<div>(pel-help-pdf &optional OPEN-WEB-PAGE)</div>	Open the 🔗 - Gerbil 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.
<div>🔗 Customize PEL Gerbil Scheme support</div>	<div><f11> SPC C-s C-e <f2></div> <div><f12> <f2></div>	<div>(pel-customize-pel &optional OTHER-WINDOW)</div>	Customize PEL Gerbil Scheme support. <ul style="list-style-type: none">If OTHER-WINDOW is non-nil (use C-u), display in another window.
<div>🔗 Customize Emacs Gerbil Scheme support</div>	<div><f11> SPC C-s C-e <f3></div> <div><f12> <f3></div>	<div>(pel-customize-library &optional OTHER-WINDOW)</div>	Customize Emacs Scheme support: gerbil-mode, scheme, geiser, quack, lispy. <ul style="list-style-type: none">If OTHER-WINDOW is non-nil (use C-u), display in another window.
	Use the following commands to interact with the gxi Scheme Gerbil REPL		
<div>Compile current buffer</div>	<div>C-c C-f</div>	<div>(gerbil-compile-current-buffer)</div>	Compile the current buffer
<div>Import current buffer</div>	<div><ul style="list-style-type: none">C-c C-iC-c <tab></div>	<div>(gerbil-import-current-buffer)</div>	Import current buffer
<div>Reload current buffer</div>	<div>C-c C-r</div>	<div>(gerbil-reload-current-buffer)</div>	Reload current buffer
<div>Build</div>	<div>C-c C-b</div>	<div>(gerbil-build)</div>	Build
<div>Evaluate current definition</div>	<div>C-c C-e</div>	<div>(scheme-send-definition)</div>	Send the current definition to the inferior Gerbil Scheme process.
<div>Evaluate marked region</div>	<div>C-c C-c</div>	<div>(scheme-send-region START END)</div>	Send the current region to the inferior Gerbil Scheme process.
<div>Restart inferior scheme process</div>	<div>C-x 9</div>	<div>(restart-scheme)</div>	Restart the inferior Gerbil Scheme Process
<div>Open the Gerbil REPL</div>	<div><f12> z</div>	<div>(pel-gerbil-repl &optional N)</div>	Run the Gerbil REPL in window specified by N. <ul style="list-style-type: none">By default use the other window. If a numeric argument is specified, its value correspond to the direction of a numeric keypad:<div><div>8</div><div>46</div><div>2</div></div>That is:<ul style="list-style-type: none">8: up4: left6: right2: down0 and 5 identify the current window.
<div>Erase the content of REPL</div>	<div><f12> C-1</div>	<div>(pel-clear-scheme-repl-buffer)</div>	Erase content of the Gerbil Scheme REPL running under Emacs.