

CHEAT SHEET

In the tables below, triple chords always accept a variant with the third key not modified by `Control`; e.g., `geiser-autodoc-show` is bound both to `C-c C-d C-s` and `C-c C-d s`.

SCHEME BUFFERS

Key	Command	Description
C-c C-z	<code>geiser-mode-switch-to-repl</code>	Switch to REPL
C-c C-a	<code>geiser-mode-switch-to-repl-and-enter</code>	Switch to REPL and current module (also <code>C-u C-c C-z</code>)
C-c C-s	<code>geiser-set-scheme</code>	Specify Scheme implementation for buffer
M-.	<code>geiser-edit-symbol-at-point</code>	Go to definition of identifier at point
M-,	<code>geiser-pop-symbol-stack</code>	Go back to where M-. was last invoked
C-c C-e C-m	<code>geiser-edit-module</code>	Ask for a module and open its file
C-c C-e C-l	<code>geiser-add-to-load-path</code>	Ask for a directory and add to Scheme load path
C-c C-e C-[<code>geiser-squarify</code>	Toggle between <code>()</code> and <code>[]</code> for current form
C-c C-\	<code>geiser-insert-lambda</code>	Insert greek lambda or, with prefix, a lambda form
C-c C-i	<code>geiser-eval-interrupt</code>	Interrupt ongoing evaluation
C-M-x	<code>geiser-eval-definition</code>	Eval definition around point
C-c C-c	<code>geiser-eval-definition</code>	Eval definition around point
C-c M-e	<code>geiser-eval-definition-and-go</code>	Eval definition around point and switch to REPL
C-c M-c	<code>geiser-eval-definition-and-go</code>	Eval definition around point and switch to REPL
C-x C-e	<code>geiser-eval-last-sexp</code>	Eval sexp before point
C-c C-r	<code>geiser-eval-region</code>	Eval region
C-c M-r	<code>geiser-eval-region-and-go</code>	Eval region and switch to REPL
C-c C-b	<code>geiser-eval-buffer</code>	Eval buffer

Key	Command	Description
C-c M-b	<code>geiser-eval-buffer-and-go</code>	Eval buffer and switch to REPL
C-c C-m C-x	<code>geiser-expand-definition</code>	Macro-expand definition around point
C-c C-m C-e	<code>geiser-expand-last-sexp</code>	Macro-expand sexp before point
C-c C-m C-r	<code>geiser-expand-region</code>	Macro-expand region
C-c C-k	<code>geiser-compile-current-buffer</code>	Compile and load current file; with prefix, restart REPL before
C-c C-l	<code>geiser-load-file</code>	Load scheme file
M-g n, C-x ‘	<code>next-error</code>	Jump to the location of next error
M-g p	<code>previous-error</code>	Jump to the location of previous error
C-c C-d C-d	<code>geiser-doc-symbol-at-point</code>	See documentation for identifier at point
C-c C-d C-s	<code>geiser-autodoc-show</code>	Show signature or value for identifier at point in echo area
C-c C-d C-m	<code>geiser-doc-module</code>	See a list of a module’s exported identifiers
C-c C-d C-i	<code>geiser-doc-look-up-manual</code>	Look up manual for symbol at point
C-c C-d C-a	<code>geiser-autodoc-mode</code>	Toggle autodoc mode
C-c <	<code>geiser-xref-callers</code>	Show callers of procedure at point
C-c >	<code>geiser-xref-callees</code>	Show callees of procedure at point
M-TAB	<code>completion-at-point</code>	Complete identifier at point
M-‘, C-.	<code>geiser-capf-complete-module</code>	Complete module name at point

REPL

Key	Command	Description
C-c C-z	<code>switch-to-geiser</code>	Start Scheme REPL, or jump to previous buffer
C-c M-o	<code>geiser-repl-clear-buffer</code>	Clear REPL buffer
C-c C-k	<code>geiser-repl-interrupt</code>	Interrupt REPL evaluation (signalling inferior scheme)
C-c C-q	<code>geiser-repl-exit</code>	Kill Scheme process
M-.	<code>geiser-edit-symbol-at-point</code>	Edit identifier at point
C-c C-l	<code>geiser-load-file</code>	Load scheme file

Key	Command	Description
TAB	<code>geiser-repl-tab-dwim</code>	Complete, indent, or go to next error
S-TAB (backtab)	<code>geiser-repl--previous-error</code>	Go to previous error in the REPL buffer
M-TAB	<code>completion-at-point</code>	Complete identifier at point
M-‘, C-.	<code>geiser-capf-complete-module</code>	Complete module name at point
C-c [, C-c C-[<code>geiser-squarify</code>	Toggle between () and [] for current form
C-c \, C-c C-\	<code>geiser-insert-lambda</code>	Insert greek lambda or, with prefix, a lambda form
C-c C-r	<code>geiser-add-to-load-path</code>	Ask for a directory and add to Scheme load path
M-p, M-n	(comint commands)	Prompt history, matching current prefix
C-c M-p, C-c M-n	(comint commands)	Previous/next prompt inputs
C-c C-m	<code>switch-to-geiser-module</code>	Set current module
C-c C-i	<code>geiser-repl-import-module</code>	Import module into current namespace
C-c C-d C-d	<code>geiser-doc-symbol-at-point</code>	See documentation for symbol at point
C-c C-d C-i	<code>geiser-doc-look-up-manual</code>	Look up manual for symbol at point
C-c C-d C-m	<code>geiser-repl--doc-module</code>	See documentation for module
C-c C-d C-a	<code>geiser-autodoc-mode</code>	Toggle autodoc mode

DOCUMENTATION BROWSER

Key	Command	Description
TAB, n	<code>forward-button</code>	Next link
S-TAB, p	<code>backward-button</code>	Previous link
N	<code>geiser-doc-next-section</code>	Next section
P	<code>geiser-doc-previous-section</code>	Previous section
f	<code>geiser-doc-next</code>	Next page
b	<code>geiser-doc-previous</code>	Previous page
k	<code>geiser-doc-kill-page</code>	Kill current page and go to previous or next
g, r	<code>geiser-doc-refresh</code>	Refresh page

Key	Command	Description
c	<code>geiser-doc-clean-history</code>	Clear browsing history
., M-.	<code>geiser-doc-edit-symbol-at-point</code>	Edit identifier at point
z	<code>geiser-doc-switch-to-repl</code>	Switch to REPL
q	<code>View-quit</code>	Bury buffer

Next: [No hacker is an island](#), Previous: [Between the parens](#), Up: [Geiser](#) [\[Contents\]](#)
[\[Index\]](#)