

Commands

Command mode can be activated by pressing `:`. The built-in commands are:

Name	Description
<code>:quit</code> , <code>:q</code>	Close the current view.
<code>:quit!</code> , <code>:q!</code>	Force close the current view, ignoring unsaved changes.
<code>:open</code> , <code>:o</code>	Open a file from disk into the current view.
<code>:buffer-close</code> , <code>:bc</code> , <code>:bclose</code>	Close the current buffer.
<code>:buffer-close!</code> , <code>:bc!</code> , <code>:bclose!</code>	Close the current buffer forcefully, ignoring unsaved changes.
<code>:buffer-close-others</code> , <code>:bco</code> , <code>:bcloseother</code>	Close all buffers but the currently focused one.
<code>:buffer-close-others!</code> , <code>:bco!</code> , <code>:bcloseother!</code>	Force close all buffers but the currently focused one.
<code>:buffer-close-all</code> , <code>:bca</code> , <code>:bcloseall</code>	Close all buffers without quitting.
<code>:buffer-close-all!</code> , <code>:bca!</code> , <code>:bcloseall!</code>	Force close all buffers ignoring unsaved changes without quitting.
<code>:buffer-next</code> , <code>:bn</code> , <code>:bnext</code>	Goto next buffer.
<code>:buffer-previous</code> , <code>:bp</code> , <code>:bprev</code>	Goto previous buffer.
<code>:write</code> , <code>:w</code>	Write changes to disk. Accepts an optional path (:write some/path.txt)
<code>:write!</code> , <code>:w!</code>	Force write changes to disk creating necessary subdirectories. Accepts an optional path (:write! some/path.txt)
<code>:write-buffer-close</code> , <code>:wbc</code>	Write changes to disk and closes the buffer. Accepts an optional path (:write-buffer-close some/path.txt)
<code>:write-buffer-close!</code> , <code>:wbc!</code>	Force write changes to disk creating necessary subdirectories and closes the buffer. Accepts an optional path (:write-buffer-close! some/path.txt)

Name	Description
<code>:new</code> , <code>:n</code>	Create a new scratch buffer.
<code>:format</code> , <code>:fmt</code>	Format the file using the LSP formatter.
<code>:indent-style</code>	Set the indentation style for editing. ('t' for tabs or 1-16 for number of spaces.)
<code>:line-ending</code>	Set the document's default line ending. Options: crlf, lf.
<code>:earlier</code> , <code>:ear</code>	Jump back to an earlier point in edit history. Accepts a number of steps or a time span.
<code>:later</code> , <code>:lat</code>	Jump to a later point in edit history. Accepts a number of steps or a time span.
<code>:write-quit</code> , <code>:wq</code> , <code>:x</code>	Write changes to disk and close the current view. Accepts an optional path (:wq some/path.txt)
<code>:write-quit!</code> , <code>:wq!</code> , <code>:x!</code>	Write changes to disk and close the current view forcefully. Accepts an optional path (:wq! some/path.txt)
<code>:write-all</code> , <code>:wa</code>	Write changes from all buffers to disk.
<code>:write-all!</code> , <code>:wa!</code>	Forcefully write changes from all buffers to disk creating necessary subdirectories.
<code>:write-quit-all</code> , <code>:wqa</code> , <code>:xa</code>	Write changes from all buffers to disk and close all views.
<code>:write-quit-all!</code> , <code>:wqa!</code> , <code>:xa!</code>	Write changes from all buffers to disk and close all views forcefully (ignoring unsaved changes).
<code>:quit-all</code> , <code>:qa</code>	Close all views.
<code>:quit-all!</code> , <code>:qa!</code>	Force close all views ignoring unsaved changes.
<code>:cquit</code> , <code>:cq</code>	Quit with exit code (default 1). Accepts an optional integer exit code (:cq 2).
<code>:cquit!</code> , <code>:cq!</code>	Force quit with exit code (default 1) ignoring unsaved changes. Accepts an optional integer exit code (:cq! 2).
<code>:theme</code>	Change the editor theme (show current theme if no name specified).
<code>:yank-join</code>	Yank joined selections. A separator can be provided as first argument. Default value is newline.
<code>:clipboard-yank</code>	Yank main selection into system clipboard.
<code>:clipboard-yank-join</code>	Yank joined selections into system clipboard. A separator can be provided as first argument. Default value is newline.

Name	Description
<code>:primary-clipboard-yank</code>	Yank main selection into system primary clipboard.
<code>:primary-clipboard-yank-join</code>	Yank joined selections into system primary clipboard. A separator can be provided as first argument. Default value is newline.
<code>:clipboard-paste-after</code>	Paste system clipboard after selections.
<code>:clipboard-paste-before</code>	Paste system clipboard before selections.
<code>:clipboard-paste-replace</code>	Replace selections with content of system clipboard.
<code>:primary-clipboard-paste-after</code>	Paste primary clipboard after selections.
<code>:primary-clipboard-paste-before</code>	Paste primary clipboard before selections.
<code>:primary-clipboard-paste-replace</code>	Replace selections with content of system primary clipboard.
<code>:show-clipboard-provider</code>	Show clipboard provider name in status bar.
<code>:change-current-directory , :cd</code>	Change the current working directory.
<code>:show-directory , :pwd</code>	Show the current working directory.
<code>:encoding</code>	Set encoding. Based on https://encoding.spec.whatwg.org .
<code>:character-info , :char</code>	Get info about the character under the primary cursor.
<code>:reload , :rl</code>	Discard changes and reload from the source file.
<code>:reload-all , :rla</code>	Discard changes and reload all documents from the source files.
<code>:update , :u</code>	Write changes only if the file has been modified.
<code>:lsp-workspace-command</code>	Open workspace command picker
<code>:lsp-restart</code>	Restarts the language servers used by the current doc

Name	Description
<code>:lsp-stop</code>	Stops the language servers that are used by the current doc
<code>:tree-sitter-scopes</code>	Display tree sitter scopes, primarily for theming and development.
<code>:tree-sitter-highlight-name</code>	Display name of tree-sitter highlight scope under the cursor.
<code>:debug-start , :dbg</code>	Start a debug session from a given template with given parameters.
<code>:debug-remote , :dbg-tcp</code>	Connect to a debug adapter by TCP address and start a debugging session from a given template with given parameters.
<code>:debug-eval</code>	Evaluate expression in current debug context.
<code>:vsplit , :vs</code>	Open the file in a vertical split.
<code>:vsplit-new , :vnew</code>	Open a scratch buffer in a vertical split.
<code>:hsplit , :hs , :sp</code>	Open the file in a horizontal split.
<code>:hsplit-new , :hnew</code>	Open a scratch buffer in a horizontal split.
<code>:tutor</code>	Open the tutorial.
<code>:goto , :g</code>	Goto line number.
<code>:set-language , :lang</code>	Set the language of current buffer (show current language if no value specified).
<code>:set-option , :set</code>	Set a config option at runtime. For example to disable smart case search, use <code>:set search.smart-case false</code> .
<code>:toggle-option , :toggle</code>	Toggle a boolean config option at runtime. For example to toggle smart case search, use <code>:toggle search.smart-case</code> .
<code>:get-option , :get</code>	Get the current value of a config option.
<code>:sort</code>	Sort ranges in selection.
<code>:rsort</code>	Sort ranges in selection in reverse order.
<code>:reflow</code>	Hard-wrap the current selection of lines to a given width.
<code>:tree-sitter-subtree , :ts-subtree</code>	Display tree sitter subtree under cursor, primarily for debugging queries.
<code>:config-reload</code>	Refresh user config.