



NeoVim commands and shortcuts cheatsheet

Conventions in this cheatsheet

All Vim commands or shortcuts exposed in this cheatsheet are written in monospaced font, like `/ {pattern}` for searching for a pattern over the current file.

All Vim commands or shortcuts are accompanied by the logo of the Vim mode you need to be in order to use that command/shortcut. For example, if you see the shortcut “ `NORMAL` `gj`”, this means that to use this shortcut, you need first to enter into Vim normal mode, and then, you press the keys “`g`” and “`j`” (one after another) on your keyboard.

When you need to hold down a sequence of keys together, we will use the plus sign (+). So, for example, the shortcut `<Space> + f` means: “press and hold down the `<Space>` key, then, press the `f` key”.

When you need to provide some input to the command/shortcut, like a pattern to search for, or a number of lines to jump, then this input will be surrounded by curly brackets (`{` and `}`). So if you see `{pattern}` inside a shortcut/command, this means that you need to type a pattern in the place of this `{pattern}`.

Vim modes

- Enter visual mode: being in `NORMAL` mode, press `v`, and then, press any key that moves the cursor, such as the arrow keys, or the keys `w`, `b`, `e` and `{`
- Enter command mode: `NORMAL` `:`
- Enter insert mode: `NORMAL` `i`
- Go back into normal mode: `INSERT` `VISUAL` `COMMAND` `<Esc>`

Quit Neovim | Create, Save and Close files

- Save the current file: `COMMAND` `:w`

- Create a new text file: `COMMAND` `:e {path to new file}`, then, use the save command `COMMAND` `:w` to save this new file into your system.
- Open the file explorer. In other words, browse/navigate through the files in the current directory: `COMMAND` `:e .`
- Close everything (all active windows), and then, Quit/Exit NeoVim: `COMMAND` `:qa`
- Close the current window (this works for any kind of window in NeoVim, e.g. a plugin window, a tab or a splitted window). However, if you have only one window active, then, this command will actually Quit/Exit NeoVim: `COMMAND` `:q`

Move cursor

- Move left `NORMAL` `h` or the left arrow `<Left arrow>`
- Move down `NORMAL` `j` or the down arrow `<Down arrow>`
- Move up `NORMAL` `k` or the up arrow `<Up arrow>`
- Move right `NORMAL` `l` or the right arrow `<Right arrow>`
- Jump to the beginning of the next word `NORMAL` `w`
- Jump to the end of the next word `NORMAL` `e`
- Jump to the beginning of the current/previous word `NORMAL` `b`
- Jump to the start of the current line `NORMAL` `0`
- Jump to the first non-blank character of the current line `NORMAL` `_`
- Jump to the end of the current line `NORMAL` `$`
- Jump to previous line that is blank `NORMAL` `{`
- Jump to next line that is blank `NORMAL` `}`
- Jump to the first line of the current file: `NORMAL` `gg`



NeoVim commands and shortcuts cheatsheet

- Jump to the last line of the current file: **NORMAL** G

Search and replace

If you only type **NORMAL** /{pattern}, then NeoVim will just highlight the matches found in the current file. However, if type **NORMAL** /{pattern} and then press Enter, NeoVim will move the cursor to the first match found in the file. You can also press the n key over and over again to navigate through all matches found across the file.

- Search for a pattern over the next lines: **NORMAL** /{pattern}
- Search for a pattern over the previous lines: **NORMAL** ?{pattern}
- If you dont have Telescope installed, and want to search for a pattern across all files in your current directory: **COMMAND** :vim[grep] /{pattern}/ **/*
- Search and replace the first occurrence of a pattern over the current line: **COMMAND** :s/{pattern}/{replacement}
- Search and replace the first occurrence of a pattern over all lines: **COMMAND** :%s/{pattern}/{replacement}
- Search and replace all occurrences of a pattern over the current line: **COMMAND** :%s/{pattern}/{replacement}/g
- Search and replace all occurrences of a pattern over all lines: **COMMAND** :%s/{pattern}/{replacement}/g

Comment lines

- Comment the current line: **NORMAL** gcc or gbc
- Comment multiple lines at once: being in **NORMAL** mode, press v, then, move the cursor in any possible way to enter **VISUAL** mode. Then, if you do enter visual

mode, start to move the cursor to mark the lines you want to comment. After you marked the lines you want, you can type gc or gb to comment them.

Copy and paste text

- Copy (or yank) text: **NORMAL** y
- Paste copied text: **NORMAL** p
- Copy current line: **NORMAL** yy
- Copy multiple lines at once: being in **NORMAL** mode, press v, then, move the cursor in any possible way to enter **VISUAL** mode. Then, if you do enter visual mode, start to move the cursor to mark the lines you want to copy. After you marked the lines you want, you can type y to copy them.

Insert mode, start writing new code

- Enter insert mode before the cursor: **NORMAL** i
- Enter insert mode after the cursor (append): **NORMAL** a
- Create a new line below the current line, then, move the cursor to this new line, then, enter insert mode: **NORMAL** o
- Create a new line above the current line, then, move the cursor to this new line, then, enter insert mode: **NORMAL** O
- Enter insert mode at the beginning of the current line: **NORMAL** I
- Enter insert mode at the end of the current line (append): **NORMAL** A

Delete content

- Delete current line **NORMAL** dd



NeoVim commands and shortcuts cheatsheet

- Delete current word **NORMAL** bdw
- Delete previous word **NORMAL** db
- Delete next word **NORMAL** wdw
- Mark multiple lines and delete them at once: being in **NORMAL** mode, press v, then, move the cursor in any possible way to enter **VISUAL** mode. Then, if you do enter visual mode, start to move the cursor to mark the lines you want to delete. After you marked the lines you want, just press d to delete them.

Indentation and code style

- Reindent the entire file: **NORMAL** gg=G
- Reformat the entire file using the code style of by the LSP¹: **COMMAND** :Format
- Add more indentation (i.e. move right) to the current line **INSERT** <Ctrl> + t
- Subtract indentation (i.e. move left) of the current line **INSERT** <Ctrl> + d

Opening multiple files across different tabs

You can have multiple files open at the same time in NeoVim. You just have to open these files in different tabs.

- Open the file in a new tab: **COMMAND** :tabnew {path to file}
- Go to next tab: **COMMAND** :tabn

¹This command will use the current LSP to reformat your source code, so it conforms with the code style recommended by this LSP. For example, if you are editing a C source file, you are likely using the clang LSP. Then, this :Format command would apply the code style recommended by the clang LSP over the current file

- Go to previous tab: **COMMAND** :tabp
- Close the current tab: **COMMAND** :tabc

Useful commands from the Telescope plugin

The Telescope [NeoVim plugin](#) is an extremely useful plugin, and you should have it installed in your NeoVim. It provides many useful shortcuts for searching files and patterns across your project.

- List all shortcuts activated by Telescope (this is very useful when you don't remember the shortcut you are looking for): **COMMAND** :Telescope keymaps
- Search for a file in your current working directory: **NORMAL** <Space>sf
- Search for a file in your "recently opened files" list²: **NORMAL** <Space>?
- Search for all occurrences of a pattern across your current working directory (or project): **NORMAL** <Space>sw or <Space>sg

Package Manager (Mason)

[Mason](#) is the most popular package manager for NeoVim, and is the most effective way of installing NeoVim plugins.

- Opens the Mason package manager: **COMMAND** :Mason
- Search for a package/plugin: when you are inside Mason, just use the normal Vim search command /{pattern}.
- Install a package: if the cursor is over the package, press i to install it.
- Uninstall a package: if the cursor is over the package, press X to uninstall it.
- Update a package: if the cursor is over the package, press u to update it.

²This is an interesting shortcut because this "recently opened files" also contains files that are outside of your current working directory (or your current project).