

**Vim Cheat Sheet**



|  |  |  |  |
| --- | --- | --- | --- |
| **Global** | | **Cursor Movement** | |
| :h[elp] keyword | Display the details of the keyword followed by the help command. | h | Bring the cursor to left |
| :sav[eas] file | Save the current buffer with a new filename | j | Bring the cursor to down |
| :clo[se] | Close current pane | k | Bring the cursor to up |
| :ter[minal] | Open a terminal window | l | Bring the cursor to right |
| K | Open man page for the word under the cursor | gj | Bring the cursor to down (multi-line text) |
| **Insert Mode - Inserting/Appending Text** | | gk | Bring the cursor to up (multi-line text) |
| i | Moves the cursor to the beginning of the current line and switches to Insert mode. | H | Bring the cursor to the top of screen |
| I | Insert at the beginning of the line | M | Bring the cursor to the middle of screen |
| a | Moves the cursor one character to the right | L | Move to bottom of screen |
| A | Moves the cursor to the end of the current line. | w | Jump forwards to the start of a word |
| o | Creates a new line below the current line. | W | Bring the cursor to the start of the next word. |
| O | Creates a new line above the current line. | e | Jump forwards to the end of a word |
| ea | Insert (append) at the end of the word | E | Jump forwards to the end of a word (words can contain punctuation) |
| CTRL + H | Deletes the character before the cursor | b | Jump backward to the start of a word |
| CTRL + W | Delete word before the cursor | B | Jump backward to the start of a word (words can contain punctuation) |
| CTRL + J | Begin new line | ge | Jump backward to the end of a word |
| CTRL + T | Indent (move right) line one shiftwidth | gE | Jump backward to the end of a word (words can contain punctuation) |
| CTRL + D | De-Indent (move left) line one shiftwidth | % | Move the cursor between matching opening and closing brackets (parentheses, square brackets, or curly braces). |
| CTRL + N | Insert (auto-complete) next match before the cursor | 0 | Jump to the start of the line |



**Vim Cheat Sheet**



|  |  |  |  |
| --- | --- | --- | --- |
| CTRL + P | Insert (auto-complete) previous match before the cursor | ^ | Jump to the first non-blank character of the line |
| CTRL + RX | Insert the contents of register x | $ | Jump to the end of the line |
| CTRL + OX | Temporarily enter normal mode to issue one normal-mode command x. | g\_ | Jump to the last non-blank character of the line |
| ESC OR CTRL + C | Exit insert mode | gg | Go to the first line of the document |
| **Cut and Paste** | | G | Pressing 'G' (in normal mode) moves the cursor to the last line of the file. It is equivalent to typing 'gg' followed by 'G'. |
| yy | Yank (copy) a line | gd | Move to local declaration |
| 2yy | Yank (copy) 2 lines | gD | Move to global declaration |
| yw | Yank (copy) the characters of the word from the cursor position to the start of the next word | fx | Jump to next occurrence of character x |
| yiw | Yank (copy) word under the cursor | tx | Jump to before next occurrence of character x |
| yaw | Yank (copy) word under the cursor and the space after or before it | Fx | Jump to the previous occurrence of character x |
| y$ or Y | Yank (copy) to end of line | Tx | Jump to after previous occurrence of character x |
| p | Put (paste) the clipboard after cursor | ; | Repeat previous f, t, F or T movement |
| P | Put (paste) before cursor | , | Repeat previous f, t, F or T movement, backwards |
| gp | Put (paste) the clipboard after cursor and leave cursor after the new text | } | Jump to next paragraph (or function/block, when editing code) |
| gP | Put (paste) before cursor and leave cursor after the new text | { | Jump to previous paragraph (or function/block, when editing code) |
| dd | Delete (cut) a line | Ctrl + u - move back 1/2 a screen | |
| 2dd | Delete (cut) 2 lines | zz | Centre cursor on screen |
| dw | Delete (cut) the characters of the word from the cursor position to the start of the next word | zt | Position cursor on top of the screen |
| diw | Delete (cut) word under the cursor | zb | Position cursor on bottom of the screen |
| daw | Delete (cut) word under the cursor and the space after or before it | CTRL + E | Move screen down one line (without moving cursor) |
| :3,5d | Delete lines starting from 3 to 5 | CTRL + Y | Move screen up one line (without moving cursor) |

# Vim Cheat Sheet

**Search and Replace** CTRL + B Move back one full screen

/pattern Search for pattern CTRL + F Move forward one full screen

?pattern Search backward for pattern CTRL + D Move forward 1/2 a screen

\vpattern

very magic' pattern: Non-Alphanumeric characters are interpreted as special regex symbols (no escaping needed)

## Editing

n Repeat search in the same direction

1. Repeat search in opposite direction

:%s/old/new/g Replace all old with new throughout file

:noh[lsearch] Remove highlighting of search matches

## Marking Text (Visual Mode)

v Start visual mode, mark lines, then do a command (like y-yank)

V Start line wise visual mode

o Move to other end of marked area CTRL + V Start visual block mode

1. Move to other corner of block

aw Mark a word

ab A block with ()

aB A block with {}

at A block with <> tags

ib Inner block with ()

1. Replace a single character.

R Replace more than one character until ESC is pressed.

J Join line below to the current one with one space in between

gwip Reflow paragraph

g~ Switch case up to motion

gu Change to lowercase up to motion

gU Change to uppercase up to motion

cc Change (replace) entire line

c$ or C Change (replace) to the end of the line ciw Change (replace) entire word

cw or ce Change (replace) to the end of the word

1. Delete character and substitute text

S Delete line and substitute text (same as cc)

xp Transpose two letters (delete and paste)

u Undo

iB Inner block with {} U Restore (undo) last changed line

it Inner block with <> tags ESC or CTRL +Exit visual mode



## Visual Commands

> Shift text right

< Shift text left

y Yank (copy) marked text

d Delete marked text

CTRL + R Redo

. Repeat last command

## Marks and Positions

:marks List of marks

ma Set current position for mark A

`a Jump to position of mark A

y`a Yank text to position of mark A



**Vim Cheat Sheet**

|  |  |  |  |
| --- | --- | --- | --- |
| ~ | Switch case | `0 | Go to the position where Vim was previously exited |
| u | Change marked text to lowercase | `" | Go to the position when last editing this file |
| U | Change marked text to uppercase | `. | Go to the position of the last change in this file |
| **Registers** | | `` | Go to the position before the last jump |
| :reg[isters] | Show registers content | :ju[mps] | List of jumps |
| "xy | Yanks (copies) the selected text and stores it in the register x | CTRL + I | Go to newer position in jump list |
| "+y | Yank into the system clipboard register | :changes | List of changes |
| "+p | puts the contents of the system clipboard register after the cursor position | g, | Go to newer position in change list |
| 0 | moves the cursor to the beginning of the current line. | g; | Go to older position in change list |
| " | Unnamed register, last delete or yank | CTRL + ] | Jump to the tag under cursor |
| % | Current file name | **Indent Text** | |
| # | Alternate file name | >> | Indent (move right) line one shiftwidth |
| **Macros** | | << | De-indent (move left) line one shiftwidth |
| qa | Record macro | >% | Indent a block with () or {} (cursor on  brace) |
| q | Stop recording macro | <% | De-Indent a block with () or {} (cursor on brace) |
| @a | Run macro | >ib | Indent inner block with () |
| @@ | Rerun last run macro | >at | Indent a block with <> tags |
| Exiting | | 3== | Re-indent 3 lines |
| :w | Write (save) the file, but don't exit | =% | Re-indent a block with () or {} (cursor on  brace) |
| :w !sudo tee  % | Write out the current file using sudo | =iB | Re-indent inner block with {} |
| :wq or :x or ZZ | Write (save) and quit | gg=G | Re-indent entire buffer |
| :q | Quit (fails if there are unsaved changes) | ]p | Paste and adjust indent to current line |
| :tabnew or  :tabnew  {page.words.fil e} - | Open a file in a new tab | za | Toggle fold under the cursor |



|  |  |  |  |
| --- | --- | --- | --- |
| :q! or ZQ | Quit and throw away unsaved changes | **Diff** | |
| :wqa | Write (save) and quit on all tabs | zf | Manually define a fold up to motion |
| **Tabs** | | zd | Delete fold under the cursor |
| CTRL + wT | Move the current split window into its own tab | zo | Open fold under the cursor |
| gt or  :tabn[ext] | Move to the next tab | zc | Close fold under the cursor |
| gT or  :tabp[revious] | Move to the previous tab | zr | Reduce (open) all folds by one level |
| #gt | Move to tab number # | zm | Fold more (close) all folds by one level |
| :tabm[ove] # | Move current tab to the #th position (indexed from 0) | zi | Toggle folding functionality |
| :tabc[lose] | Close the current tab and all its windows | ]c | Jump to start of next change |
| :tabo[nly] | Close all tabs except for the current one | [c | Jump to start of previous change |
| :tabdo command | Run the command on all tabs (e.g.  :tabdo q - closes all opened tabs) | do or  :diffg[et] | Obtain (get) difference (from other buffer) |
| **Working With Multiple Files** | | dp or  :diffpu[t] | Put difference (to other buffer) |
| :e[dit] file | Edit a file in a new buffer | :diffthis | Make current window part of diff |
| :bn[ext] | Go to the next buffer | :dif[fupdate] | Update differences |
| :bp[revious] | Go to the previous buffer | :diffo[ff] | Switch off diff mode for current window |
| :bd[elete] | Delete a buffer (close a file) | **Search In Multiple Files** | |
| :b[uffer]# | Go to a buffer by index # | :vim[grep]  /pattern/  {`{file}`} | Search for pattern in multiple files |
| :b[uffer] file | Go to a buffer by file | :cn[ext] | Jump to the next match |
| :ls or :buffers | List all open buffers | :cp[revious] | Jump to the previous match |

**Prepared By:** [Mohammad Shah Miran](https://linuxsimply.com/mohammad-shah-miran/) **Copyright ©2023 linuxsimply.com| All rights reserved.**

|  |  |  |  |
| --- | --- | --- | --- |
| :sp[lit] file | Open a file in a new buffer and split window | :cope[n] | Open a window containing the list of matches |
| :vs[plit] file | Open a file in a new buffer and vertically split window | :ccl[ose] | Close the quickfix window |
| :vert[ical] ba[ll] | Edit all buffers as vertical windows |  | |
| :tab ba[ll] | Edit all buffers as tabs |
| CTRL + ws | Split window |
| CTRL + wv | Split window vertically |
| CTRL + wq | Quit a window |
| CTRL + wx | Exchange current window with next one |
| CTRL + w= | Make all windows equal height & width |
| CTRL + wh | Move cursor to the left window (vertical split) |
| CTRL + wl | Move cursor to the right window (vertical split) |
| CTRL + wj | Move cursor to the window below (horizontal split) |
| CTRL + wH | Make current window full height at far left (leftmost vertical window) |
| CTRL + wL | Make current window full height at far right (rightmost vertical window) |
| he very top (topmost horizontal window) | |
| CTRL + wJ | Make current window full width at the very bottom (bottommost horizontal window) |
| CTRL + wK | Make current window full width |

**Prepared By:** [Mohammad Shah Miran](https://linuxsimply.com/mohammad-shah-miran/) **Copyright ©2023 linuxsimply.com| All rights reserved.**