

Vim Cheat Sheet

Global :h[elp] keyword - open help for keyword :sav[eas] file - save file as :clo[se] - close current pane :ter[minal] - open a terminal window K - open man page for word under the cursor	Editing r - 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 gJ - join line below to the current one without 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 s - delete character and substitute text S - delete line and substitute text (same as cc) xp - transpose two letters (delete and paste) u - undo U - restore (undo) last changed line Ctrl + r - redo . - repeat last command	Macros qa - record macro a q - stop recording macro @a - run macro a @@ - rerun last run macro Cut and paste yy - yank (copy) a line 2yy - yank (copy) 2 lines yw - yank (copy) the characters of the word from the cursor position to the start of the next word yiw - yank (copy) word under the cursor yaw - yank (copy) word under the cursor and the space after or before it y\$ or Y - yank (copy) to end of line p - put (paste) the clipboard after cursor P - put (paste) before cursor gp - put (paste) the clipboard after cursor and leave cursor after the new text gP - put (paste) before cursor and leave cursor after the new text dd - delete (cut) a line 2dd - delete (cut) 2 lines dw - delete (cut) the characters of the word from the cursor position to the start of the next word diw - delete (cut) word under the cursor daw - delete (cut) word under the cursor and the space after or before it d\$ or D - delete (cut) to the end of the line x - delete (cut) character	Working with multiple files :e[dit] file - edit a file in a new buffer :bn[ext] - go to the next buffer :bp[revious] - go to the previous buffer :bd[elete] - delete a buffer (close a file) :b[uffer] # - go to a buffer by index # :b[uffer] file - go to a buffer by file :ls or :buffers - list all open buffers :sp[lit] file - open a file in a new buffer and split window :vs[plit] file - open a file in a new buffer and vertically split 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 + ww - switch windows 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 + wk - move cursor to the window above (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) Ctrl + wJ - make current window full width at the very bottom (bottommost horizontal window) Ctrl + wK - make current window full width at the very top (topmost horizontal window)
Cursor movement h - move cursor left j - move cursor down k - move cursor up l - move cursor right gj - move cursor down (multi-line text) gk - move cursor up (multi-line text) H - move to top of screen M - move to middle of screen L - move to bottom of screen w - jump forwards to the start of a word W - jump forwards to the start of a word (words can contain punctuation) e - jump forwards to the end of a word E - jump forwards to the end of a word (words can contain punctuation) b - jump backwards to the start of a word B - jump backwards to the start of a word (words can contain punctuation) ge - jump backwards to the end of a word gE - jump backwards to the end of a word (words can contain punctuation) % - move to matching character (default supported pairs: '()', '{}', '[]' - use :h matchpairs in vim for more info) O - jump to the start of the line ^ - jump to the first non-blank character of the line \$ - jump to the end of the line g_ - jump to the last non-blank character of the line gg - go to the first line of the document G - go to the last line of the document	Marking text (visual mode) v - start visual mode, mark lines, then do a command (like y-yank) V - start linewise visual mode o - move to other end of marked area Ctrl + v - start visual block mode O - 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 () iB - inner block with {} it - inner block with <> tags	Indent text >> - indent (move right) line one shiftwidth << - de-indent (move left) line one shiftwidth >% - indent a block with () or {} (cursor on brace) >ib - indent inner block with () >at - indent a block with <> tags 3== - re-indent 3 lines =% - re-indent a block with () or {} (cursor on brace)	Diff zf - manually define a fold up to motion zd - delete fold under the cursor za - toggle fold under the cursor zo - open fold under the cursor zc - close fold under the cursor zr - reduce (open) all folds by one level zm - fold more (close) all folds by one level zi - toggle folding functionality]c - jump to start of next change [c - jump to start of previous change do or :diffg[et] - obtain (get) difference (from other buffer)

5gg or 5G - go to line 5	Esc - exit visual mode	=iB - re-indent inner block with {}	dp or :diffpu[t] - put difference (to other buffer)
gd - move to local declaration	Visual commands	gg=G - re-indent entire buffer	:diffthis - make current window part of diff
gD - move to global declaration	> - shift text right]p - paste and adjust indent to current line	:dif[fupdate] - update differences
fx - jump to next occurrence of character x	< - shift text left		:diff[ff] - switch off diff mode for current window
tx - jump to before next occurrence of character x	y - yank (copy) marked text	Exiting	
Fx - jump to the previous occurrence of character x	d - delete marked text	:w - write (save) the file, but don't exit	
Tx - jump to after previous occurrence of character x	~ - switch case	:w !sudo tee % - write out the current file using sudo	
; - repeat previous f, t, F or T movement	u - change marked text to lowercase	:wq or :x or ZZ - write (save) and quit	
, - repeat previous f, t, F or T movement, backwards	U - change marked text to uppercase	:q - quit (fails if there are unsaved changes)	
} - jump to next paragraph (or function/block, when editing code)	Registers	:q! or ZQ - quit and throw away unsaved changes	
{ - jump to previous paragraph (or function/block, when editing code)	:reg[isters] - show registers content	:wqa - write (save) and quit on all tabs	
zz - center cursor on screen	"xy - yank into register x	Search and replace	
Ctrl + e - move screen down one line (without moving cursor)	"xp - paste contents of register x	/pattern - search for pattern	
Ctrl + y - move screen up one line (without moving cursor)	"+y - yank into the system clipboard register	?pattern - search backward for pattern	
Ctrl + b - move back one full screen	"+p - paste from the system clipboard register	\vpattern - 'very magic' pattern: non-alphanumeric characters are interpreted as special regex symbols (no escaping needed)	
Ctrl + f - move forward one full screen	Marks and positions	n - repeat search in same direction	
Ctrl + d - move forward 1/2 a screen	:marks - list of marks	N - repeat search in opposite direction	
Ctrl + u - move back 1/2 a screen	ma - set current position for mark A	:s/old/new/g - replace all old with new throughout file	
Insert mode - inserting/appending text	`a - jump to position of mark A	:s/old/new/gc - replace all old with new throughout file with confirmations	
i - insert before the cursor	y`a - yank text to position of mark A	:noh[lsearch] - remove highlighting of search matches	
I - insert at the beginning of the line	`O - go to the position where Vim was previously exited	Search in multiple files	
a - insert (append) after the cursor	`" - go to the position when last editing this file	:vim[grep] /pattern/ {`{file}`} - search for pattern in multiple files	
A - insert (append) at the end of the line	`.` - go to the position of the last change in this file	:cn[ext] - jump to the next match	
o - append (open) a new line below the current line	`~` - go to the position before the last jump	:cp[revious] - jump to the previous match	
O - append (open) a new line above the current line	:ju[mps] - list of jumps	:cope[n] - open a window containing the list of matches	
ea - insert (append) at the end of the word	Ctrl + i - go to newer position in jump list	:ccl[ose] - close the quickfix window	
Ctrl + h - delete the character before the cursor during insert mode	Ctrl + o - go to older position in jump list	Tabs	
Ctrl + w - delete word before the cursor during insert mode	:changes - list of changes	:tabnew or :tabnew {page.words.file} - open a file in a new tab	
Ctrl + j - begin new line during insert mode	g, - go to newer position in change list	Ctrl + wT - move the current split window into its own tab	
	g; - go to older position in change list		
	Ctrl +] - jump to the tag under cursor		

Ctrl + t - indent (move right) line one shiftwidth during insert mode

Ctrl + d - de-indent (move left) line one shiftwidth during insert mode

Ctrl + n - insert (auto-complete) next match before the cursor during insert mode

Ctrl + p - insert (auto-complete) previous match before the cursor during insert mode

Ctrl + rx - insert the contents of register x

Ctrl + ox - Temporarily enter normal mode to issue one normal-mode command x.

Esc - exit insert mode

gt or **:tabn[ext]** - move to the next tab

gT or **:tabp[revious]** - move to the previous tab

#gt - move to tab number #

:tabm[ove] # - move current tab to the #th position (indexed from 0)

:tabc[lose] - close the current tab and all its windows

:tabo[nly] - close all tabs except for the current one

:tabdo command - run the command on all tabs (e.g. **:tabdo q** - closes all opened tabs)