

Vim Cheat Sheet

Global	Editing	Macros	Working with multiple files
:h[elp] keyword - open help for keyword	r - replace a single character.	qa - record macro a	:e[dit] file - edit a file in a new buffer
:sav[eas] file - save file as	R - replace more than one character, until ESC is pressed.	q - stop recording macro	:bn[ext] - go to the next buffer
:clo[se] - close current pane	J - join line below to the current one with one space in between	@a - run macro a	:bp[revious] - go to the previous buffer
:ter[минаl] - open a terminal window	gJ - join line below to the current one without space in between	@@ - rerun last run macro	:bd[elete] - delete a buffer (close a file)
K - open man page for word under the cursor	gwp - reflow paragraph	Cut and paste	:b[uffer]# - go to a buffer by index #
Cursor movement	g~ - switch case up to motion	yy - yank (copy) a line	:b[uffer] file - go to a buffer by file
h - move cursor left	gu - change to lowercase up to motion	2yy - yank (copy) 2 lines	:ls or :buffers - list all open buffers
j - move cursor down	gU - change to uppercase up to motion	yw - yank (copy) the characters of the word from the cursor position to the start of the next word	:sp[lit] file - open a file in a new buffer and split window
k - move cursor up	cc - change (replace) entire line	yiw - yank (copy) word under the cursor	:vs[plit] file - open a file in a new buffer and vertically split window
l - move cursor right	c\$ - change (replace) to the end of the line	yaw - yank (copy) word under the cursor and the space after or before it	:vert[ical] ba[ll] - edit all buffers as vertical windows
H - move to top of screen	c\$ - change (replace) to the end of the line	y\$ - yank (copy) to end of line	:tab ba[ll] - edit all buffers as tabs
M - move to middle of screen	cw or ce - change (replace) to the end of the word	p - put (paste) the clipboard after cursor	Ctrl + ws - split window
L - move to bottom of screen	ciw - change (replace) entire word	P - put (paste) before cursor	Ctrl + wv - split window vertically
w - jump forwards to the start of a word	s - delete character and substitute text	dd - delete (cut) a line	Ctrl + ww - switch windows
W - jump forwards to the start of a word (words can contain punctuation)	S - delete line and substitute text (same as cc)	2dd - delete (cut) 2 lines	Ctrl + wq - quit a window
e - jump forwards to the end of a word	xp - transpose two letters (delete and paste)	dw - delete (cut) the characters of the word from the cursor position to the start of the next word	Ctrl + wx - exchange current window with next one
E - jump forwards to the end of a word (words can contain punctuation)	u - undo	diw - delete (cut) word under the cursor	Ctrl + w= - make all windows equal height & width
b - jump backwards to the start of a word	U - restore (undo) last changed line	daw - delete (cut) word under the cursor and the space after or before it	Ctrl + wh - move cursor to the left window (vertical split)
B - jump backwards to the start of a word (words can contain punctuation)	Ctrl + r - redo	D - delete (cut) to the end of the line	Ctrl + wl - move cursor to the right window (vertical split)
ge - jump backwards to the end of a word	. - repeat last command	d\$ - delete (cut) to the end of the line	Ctrl + wj - move cursor to the window below (horizontal split)
gE - jump backwards to the end of a word (words can contain punctuation)	Marking text (visual mode)	x - delete (cut) character	Ctrl + wk - move cursor to the window above (horizontal split)
% - move to matching character (default supported pairs: '()', '{}', '[]' - use :h matchpairs in vim for more info)	v - start visual mode, mark lines, then do a command (like y-yank)	Indent text	Diff
0 - jump to the start of the line	V - start linewise visual mode	>> - indent (move right) line one shiftwidth	zf - manually define a fold up to motion
^ - jump to the first non-blank character of the line	o - move to other end of marked area	<< - de-indent (move left) line one shiftwidth	zd - delete fold under the cursor
\$ - jump to the end of the line	Ctrl + v - start visual block mode	>% - indent a block with () or {} (cursor on brace)	za - toggle fold under the cursor
g_ - jump to the last non-blank character of the line	O - move to other corner of block	>ib - indent inner block with ()	zo - open fold under the cursor
gg - go to the first line of the document	aw - mark a word	>at - indent a block with <> tags	zc - close fold under the cursor
G - go to the last line of the document	ab - a block with ()	3== - re-indent 3 lines	zr - reduce (open) all folds by one level
	aB - a block with {}	=% - re-indent a block with () or {} (cursor on brace)	zm - fold more (close) all folds by one level
	at - a block with <> tags		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)
			dp or :diffpu[t] - put difference (to other buffer)
			:diffthis - make current window part of diff
			:dif[fupdate] - update differences

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

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 **: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)