

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[minal] - 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	gwip - 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	ciw - change (replace) entire word	p - put (paste) the clipboard after cursor	Ctrl + ws - split window
L - move to bottom of screen	cw or ce - change (replace) to the end of the word	P - put (paste) before cursor	Ctrl + ww - split window vertically
w - jump forwards to the start of a word	s - delete character and substitute text	gp - put (paste) the clipboard after cursor and leave cursor after the new text	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)	gP - put (paste) before cursor and leave cursor after the new text	Ctrl + wq - quit a window
e - jump forwards to the end of a word	xp - transpose two letters (delete and paste)	dd - delete (cut) a line	Ctrl + wx - exchange current window with next one
E - jump forwards to the end of a word (words can contain punctuation)	u - undo	2dd - delete (cut) 2 lines	Ctrl + w= - make all windows equal height & width
b - jump backwards to the start of a word	U - restore (undo) last changed line	dw - delete (cut) the characters of the word from the cursor position to the start of the next word	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	diw - delete (cut) word under the cursor	Ctrl + wl - move cursor to the right window (vertical split)
ge - jump backwards to the end of a word	. - repeat last command	daw - delete (cut) word under the cursor and the space after or before it	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)	D - delete (cut) to the end of the line	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)	d\$ - delete (cut) to the end of the line	Ctrl + wH - make current window full height at far left (leftmost vertical window)
O - jump to the start of the line	V - start linewise visual mode	x - delete (cut) character	Ctrl + wL - make current window full height at far right (rightmost vertical window)
^ - jump to the first non-blank character of the line	o - move to other end of marked area	Indent text	Ctrl + wJ - make current window full width at the very bottom (bottommost horizontal window)
\$ - jump to the end of the line	Ctrl + v - start visual block mode	>> - indent (move right) line one shiftwidth	Ctrl + wK - make current window full width at the very top (topmost horizontal window)
g_ - jump to the last non-blank character of the line	O - move to other corner of block	<< - de-indent (move left) line one shiftwidth	Diff
gg - go to the first line of the document	aw - mark a word	>% - indent a block with () or {} (cursor on brace)	zf - manually define a fold up to motion
G - go to the last line of the document	ab - a block with ()	>ib - indent inner block with ()	zd - delete fold under the cursor
	aB - a block with {}		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

5gg or 5G - go to line 5	at - a block with <> tags	>at - indent a block with <> tags	[c - jump to start of previous change
gd - move to local declaration	ib - inner block with ()	3== - re-indent 3 lines	do or :diffg[et] - obtain (get) difference (from other buffer)
gD - move to global declaration	iB - inner block with {}	=% - re-indent a block with () or {} (cursor on brace)	dp or :diffpu[t] - put difference (to other buffer)
fx - jump to next occurrence of character x	it - inner block with <> tags	=iB - re-indent inner block with {}	:diffthis - make current window part of diff
tx - jump to before next occurrence of character x	Esc - exit visual mode	gg=G - re-indent entire buffer	:dif[fupdate] - update differences
Fx - jump to previous occurence of character x	Visual commands]p - paste and adjust indent to current line	:diffo[ff] - switch off diff mode for current window
Tx - jump to after previous occurence of character x	> - shift text right		
; - repeat previous f, t, F or T movement	< - shift text left		
, - repeat previous f, t, F or T movement, backwards	Y - yank (copy) marked text	Exiting	
} - jump to next paragraph (or function/block, when editing code)	d - delete marked text	:w - write (save) the file, but don't exit	
{ - jump to previous paragraph (or function/block, when editing code)	~ - switch case	:w !sudo tee % - write out the current file using sudo	
zz - center cursor on screen	u - change marked text to lowercase	:wq or :x or ZZ - write (save) and quit	
Ctrl + e - move screen down one line (without moving cursor)	U - change marked text to uppercase	:q - quit (fails if there are unsaved changes)	
Ctrl + y - move screen up one line (without moving cursor)	Registers	:q! or ZQ - quit and throw away unsaved changes	
Ctrl + b - move back one full screen	:reg[isters] - show registers content	:wqa - write (save) and quit on all tabs	
Ctrl + f - move forward one full screen	"xy - yank into register x	Search and replace	
Ctrl + d - move forward 1/2 a screen	"xp - paste contents of register x	/pattern - search for pattern	
Ctrl + u - move back 1/2 a screen	" +y - yank into the system clipboard register	?pattern - search backward for pattern	
Insert mode - inserting/appending text	" +p - paste from the system clipboard register	\vpattern - 'very magic' pattern: non-alphanumeric characters are interpreted as special regex symbols (no escaping needed)	
i - insert before the cursor	Marks and positions	n - repeat search in same direction	
I - insert at the beginning of the line	:marks - list of marks	N - repeat search in opposite direction	
a - insert (append) after the cursor	ma - set current position for mark A	:%s/old/new/g - replace all old with new throughout file	
A - insert (append) at the end of the line	`a - jump to position of mark A	:%s/old/new/gc - replace all old with new throughout file with confirmations	
o - append (open) a new line below the current line	y`a - yank text to position of mark A	:noh[lsearch] - remove highlighting of search matches	
O - append (open) a new line above the current line	`O - go to the position where Vim was previously exited	Search in multiple files	
ea - insert (append) at the end of the word	`" - go to the position when last editing this file	:vim[greplpattern/]{`{file}`} - search for pattern in multiple files	
Ctrl + h - delete the character before the cursor during insert mode	`. - go to the position of the last change in this file	:cn[ext] - jump to the next match	
Ctrl + w - delete word before the cursor during insert mode	`` - go to the position before the last jump	:cp[revious] - jump to the previous match	
Ctrl + j - begin new line during insert mode	:ju[mps] - list of jumps	:cope[n] - open a window containing the list of matches	
	Ctrl + i - go to newer position in jump list	:ccl[ose] - close the quickfix window	
	Ctrl + o - go to older position in jump list		
	:changes - list of changes		
	g, - go to newer position in change list		
	g; - go to older position in change list		
	Ctrl +] - jump to the tag under cursor		

Tabs

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

:tabnew or **:tabnew {page.words.file}** - open a file in a new tab

Ctrl + wT - move the current split window into its own tab

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)