

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	J - join line below to the current one with one space in between	q - stop recording macro	:bn[ext] - go to the next buffer
:clo[se] - close current pane	gJ - join line below to the current one without space in between	@a - run macro a	:bp[revious] - go to the previous buffer
:ter[minal] - open a terminal window	gwip - reflow paragraph	@@ - rerun last run macro	:bd[elete] - delete a buffer (close a file)
K - open man page for word under the cursor	g~ - switch case up to motion	Cut and paste	:b[uffer]# - go to a buffer by index #
Cursor movement	gu - change to lowercase up to motion	yy - yank (copy) a line	:b[uffer] file - go to a buffer by file
h - move cursor left	gU - change to uppercase 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	y\$ - yank (copy) to end of line	: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	p - put (paste) the clipboard after cursor	: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	P - put (paste) before cursor	:tab ba[ll] - edit all buffers as tabs
M - move to middle of screen	cw - change (replace) to the end of the word	dd - delete (cut) a line	Ctrl + ws - split window
L - move to bottom of screen	ciw - change (replace) entire word	2dd - delete (cut) 2 lines	Ctrl + wv - split window vertically
w - jump forwards to the start of a word	cw - change (replace) to the end of the word	dw - delete (cut) the characters of the word from the cursor position to the start of the next word	Ctrl + ww - switch windows
W - jump forwards to the start of a word (words can contain punctuation)	S - delete character and substitute text	D - delete (cut) to the end of the line	Ctrl + wq - quit a window
e - jump forwards to the end of a word	S - delete line and substitute text (same as cc)	d\$ - delete (cut) to the end of the line	Ctrl + wx - exchange current window with next one
E - jump forwards to the end of a word (words can contain punctuation)	xp - transpose two letters (delete and paste)	x - delete (cut) character	Ctrl + w= - make all windows equal height & width
b - jump backwards to the start of a word	u - undo	Indent text	Ctrl + wh - move cursor to the left window (vertical split)
B - jump backwards to the start of a word (words can contain punctuation)	U - restore (undo) last changed line	>> - indent (move right) line one shiftwidth	Ctrl + wl - move cursor to the right window (vertical split)
% - move to matching character (default supported pairs: '()', '{}', '[]' - use :h matchpairs in vim for more info)	Ctrl + r - redo	<< - de-indent (move left) line one shiftwidth	Ctrl + wj - move cursor to the window below (horizontal split)
0 - jump to the start of the line	. - repeat last command	>% - indent a block with () or {} (cursor on brace)	Ctrl + wk - move cursor to the window above (horizontal split)
^ - jump to the first non-blank character of the line	Marking text (visual mode)	>ib - indent inner block with ()	Diff
\$ - jump to the end of the line	v - start visual mode, mark lines, then do a command (like y-yank)	>at - indent a block with <> tags	zf - manually define a fold up to motion
g_ - jump to the last non-blank character of the line	V - start linewise visual mode	3== - re-indent 3 lines	zd - delete fold under the cursor
gg - go to the first line of the document	o - move to other end of marked area	=% - re-indent a block with () or {} (cursor on brace)	za - toggle fold under the cursor
G - go to the last line of the document	Ctrl + v - start visual block mode	=ib - re-indent inner block with {}	zo - open fold under the cursor
5gg or 5G - go to line 5	O - move to other corner of block	gg=G - re-indent entire buffer	zc - close fold under the cursor
fx - jump to next occurrence of character x	aw - mark a word]p - paste and adjust indent to current line	zr - reduce (open) all folds by one level
tx - jump to before next occurrence of character x	ab - a block with ()	Exiting	zm - fold more (close) all folds by one level
Fx - jump to previous occurrence of character x	aB - a block with {}	:w - write (save) the file, but don't exit	zi - toggle folding functionality
Tx - jump to after previous occurrence of character x	at - a block with <> tags	:w !sudo tee % - write out the current file using sudo]c - jump to start of next change
	ib - inner block with ()	:wq or :x or ZZ - write (save) and quit	[c - jump to start of previous change
	iB - inner block with {}	:q - quit (fails if there are unsaved changes)	do or :diffg[et] - obtain (get) difference (from other buffer)
	it - inner block with <> tags		dp or :diffpu[t] - put difference (to other buffer)
	Esc - exit visual mode		:diffthis - make current window part of diff
			:dif[fupdate] - update differences
			:diffo[ff] - switch off diff mode for current window

; - repeat previous f, t, F or T movement	Visual commands > - shift text right	:q! or ZQ - quit and throw away unsaved changes
, - repeat previous f, t, F or T movement, backwards	< - shift text left	:wqa - write (save) and quit on all tabs
} - jump to next paragraph (or function/block, when editing code)	y - yank (copy) marked text	Search and replace /pattern - search for pattern
{ - jump to previous paragraph (or function/block, when editing code)	d - delete marked text	?pattern - search backward for pattern
zz - center cursor on screen	~ - switch case	\vpattern - 'very magic' pattern: non-alphanumeric characters are interpreted as special regex symbols (no escaping needed)
Ctrl + e - move screen down one line (without moving cursor)	u - change marked text to lowercase	n - repeat search in same direction
Ctrl + y - move screen up one line (without moving cursor)	Registers :reg[isters] - show registers content	N - repeat search in opposite direction
Ctrl + b - move back one full screen	"xy - yank into register x	%s/old/new/g - replace all old with new throughout file
Ctrl + f - move forward one full screen	"xp - paste contents of register x	%s/old/new/gc - replace all old with new throughout file with confirmations
Ctrl + d - move forward 1/2 a screen	"+y - yank into the system clipboard register	:noh[lsearch] - remove highlighting of search matches
Ctrl + u - move back 1/2 a screen	"+p - paste from the system clipboard register	
Insert mode - inserting/appending text	Marks and positions :marks - list of marks	Search in multiple files :vim[grep]/pattern/{ {file} } - search for pattern in multiple files
i - insert before the cursor	ma - set current position for mark A	:cn[ext] - jump to the next match
I - insert at the beginning of the line	`a - jump to position of mark A	:cp[revious] - jump to the previous match
a - insert (append) after the cursor	y`a - yank text to position of mark A	:cope[n] - open a window containing the list of matches
A - insert (append) at the end of the line	`0 - go to the position where Vim was previously exited	:ccl[ose] - close the quickfix window
o - append (open) a new line below the current line	`" - go to the position when last editing this file	
O - append (open) a new line above the current line	`. - go to the position of the last change in this file	
ea - insert (append) at the end of the word	`` - go to the position before the last jump	
Ctrl + h - delete the character before the cursor during insert mode	:ju[mps] - list of jumps	Tabs :tabnew or :tabnew {page.words.file} - open a file in a new tab
Ctrl + w - delete word before the cursor during insert mode	Ctrl + i - go to newer position in jump list	Ctrl + wT - move the current split window into its own tab
Ctrl + j - begin new line during insert mode	Ctrl + o - go to older position in jump list	gt or :tabn[ext] - move to the next tab
Ctrl + t - indent (move right) line one shiftwidth during insert mode	:changes - list of changes	gT or :tabp[revious] - move to the previous tab
Ctrl + d - de-indent (move left) line one shiftwidth during insert mode	g, - go to newer position in change list	#gt - move to tab number #
Ctrl + n - insert (auto-complete) next match before the cursor during insert mode	g; - go to older position in change list	:tabm[ove] # - move current tab to the #th position (indexed from 0)
Ctrl + p - insert (auto-complete) previous match before the cursor during insert mode	Ctrl +] - jump to the tag under cursor	:tabc[lose] - close the current tab and all its windows
Ctrl + rx - insert the contents of register x		: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)

Esc - exit insert mode
