

Vim Cheat Sheet

Global	Editing	Macros
:h[elp] keyword - open help for keyword	r - replace a single character.	qa - record macro a
:sav[eas] file - save file as	R - replace more than one character, until ESC is pressed.	q - stop recording macro
:clo[se] - close current pane	J - join line below to the current one with one space in between	@a - run macro a
:ter[minal] - open a terminal window	gJ - join line below to the current one without space in between	@@ - rerun last run macro
K - open man page for word under the cursor		Cut and paste
Cursor movement	gwip - reflow paragraph	yy - yank (copy) a line
h - move cursor left	g~ - switch case up to motion	2yy - yank (copy) 2 lines
j - move cursor down	gu - change to lowercase up to motion	yw - yank (copy) the characters of the word from the cursor position to the start of the next word
k - move cursor up	gU - change to uppercase up to motion	yiw - yank (copy) word under the cursor
l - move cursor right	cc - change (replace) entire line	yaw - yank (copy) word under the cursor and the space after or before it
gj - move cursor down (multi-line text)	c\$ or C - change (replace) to the end of the line	y\$ or Y - yank (copy) to end of line
gk - move cursor up (multi-line text)	ciw - change (replace) entire word	p - put (paste) the clipboard after cursor
H - move to top of screen	cw or ce - change (replace) to the end of the word	P - put (paste) before cursor
M - move to middle of screen	S - delete character and substitute text (same as cl)	gp - put (paste) the clipboard after cursor and leave cursor after the new text
L - move to bottom of screen	S - delete line and substitute text (same as cc)	gP - put (paste) before cursor and leave cursor after the new text
w - jump forwards to the start of a word	xp - transpose two letters (delete and paste)	dd - delete (cut) a line
W - jump forwards to the start of a word (words can contain punctuation)	u - undo	2dd - delete (cut) 2 lines
e - jump forwards to the end 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
E - jump forwards to the end of a word (words can contain punctuation)	Ctrl + r - redo	diw - delete (cut) word under the cursor
b - jump backwards to the start of a word	. - repeat last command	daw - delete (cut) word under the cursor and the space after or before it
B - jump backwards to the start of a word (words can contain punctuation)	Marking text (visual mode)	:3,5d - delete lines starting from 3 to 5
ge - jump backwards to the end of a word	v - start visual mode, mark lines, then do a command (like y-yank)	:g/{pattern}/d - delete all lines containing pattern
gE - jump backwards to the end of a word (words can contain punctuation)	V - start linewise visual mode	:g!/{pattern}/d - delete all lines not containing pattern
% - move cursor to matching character (default supported pairs: '()', '{}', '[]' - use :h matchpairs in vim for more info)	O - move to other end of marked area	d\$ or D - delete (cut) to the end of the line
0 - jump to the start of the line	Ctrl + v - start visual block mode	x - delete (cut) character
^ - jump to the first non-blank character of the line	O - move to other corner of block	Indent text
\$ - jump to the end of the line	aw - mark a word	>> - indent (move right) line one shiftwidth
	ab - a block with ()	
	aB - a block with {}	

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)
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

g_ - jump to the last non-blank character of the line	at - a block with <> tags	<< - de-indent (move left) line one shiftwidth	zC - close fold under the cursor
gg - go to the first line of the document	ib - inner block with ()	>% - indent a block with () or {} (cursor on brace)	zR - reduce (open) all folds by one level
G - go to the last line of the document	iB - inner block with {}	<% - de-indent a block with () or {} (cursor on brace)	zM - fold more (close) all folds by one level
5gg or 5G - go to line 5	it - inner block with <> tags	>ib - indent inner block with ()	zi - toggle folding functionality
gd - move to local declaration	Esc or Ctrl + c - exit visual mode	>at - indent a block with <> tags]c - jump to start of next change
gD - move to global declaration	Visual commands	3== - re-indent 3 lines	[c - jump to start of previous change
fx - jump to next occurrence of character x	> - shift text right	=% - re-indent a block with () or {} (cursor on brace)	do or :diffg[et] - obtain (get) difference (from other buffer)
tx - jump to before next occurrence of character x	< - shift text left	=iB - re-indent inner block with {}	dp or :diffpu[t] - put difference (to other buffer)
Fx - jump to the previous occurrence of character x	y - yank (copy) marked text	gg=G - re-indent entire buffer	:diffthis - make current window part of diff
Tx - jump to after previous occurrence of character x	d - delete marked text]p - paste and adjust indent to current line	:dif[fupdate] - update differences
; - repeat previous f, t, F or T movement	~ - switch case	Exiting	:diffo[ff] - switch off diff mode for current window
, - repeat previous f, t, F or T movement, backwards	u - change marked text to lowercase	:w - write (save) the file, but don't exit	
} - jump to next paragraph (or function/block, when editing code)	U - change marked text to uppercase	:w !sudo tee % - write out the current file using sudo	
{ - jump to previous paragraph (or function/block, when editing code)	Registers	:wq or :X or ZZ - write (save) and quit	
zz - center cursor on screen	:reg[isters] - show registers content	:q - quit (fails if there are unsaved changes)	
zt - position cursor on top of the screen	"xy - yank into register x	:q! or ZQ - quit and throw away unsaved changes	
zb - position cursor on bottom of the screen	"xp - paste contents of register x	:wqa - write (save) and quit on all tabs	
Ctrl + e - move screen down one line (without moving cursor)	"+y - yank into the system clipboard register	Search and replace	
Ctrl + y - move screen up one line (without moving cursor)	"+p - paste from the system clipboard register	/pattern - search for pattern	
Ctrl + b - move screen up one page (cursor to last line)	Marks and positions	?pattern - search backward for pattern	
Ctrl + f - move screen down one page (cursor to first line)	:marks - list of marks	\vpattern - 'very magic' pattern: non-alphanumeric characters are interpreted as special regex symbols (no escaping needed)	
Ctrl + d - move cursor and screen down 1/2 page	ma - set current position for mark A	n - repeat search in same direction	
Ctrl + u - move cursor and screen up 1/2 page	`a - jump to position of mark A	N - repeat search in opposite direction	
Insert mode - inserting/appending text	y`a - yank text to position of mark A	:%s/old/new/g - replace all old with new throughout file	
i - insert before the cursor	`0 - go to the position where Vim was previously exited	:%s/old/new/gc - replace all old with new throughout file with confirmations	
I - insert at the beginning of the line	`" - go to the position when last editing this file	:noh[lsearch] - remove highlighting of search matches	
a - insert (append) after the cursor	`. - go to the position of the last change in this file	Search in multiple files	
	`` - go to the position before the last jump	:vim[grep] /pattern/{ {file} ` } - search for pattern in multiple	
	:ju[mps] - list of jumps		
	Ctrl + i - go to newer position in jump list		
	Ctrl + o - go to older position in jump list		
	:changes - list of changes		

A - insert (append) at the end of the line

O - append (open) a new line below the current line

O - append (open) a new line above the current line

ea - insert (append) at the end of the word

Ctrl + **h** - delete the character before the cursor during insert mode

Ctrl + **w** - delete word before the cursor during insert mode

Ctrl + **j** - add a line break at the cursor position during insert mode

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 or **Ctrl** + **c** - exit insert mode

g, - go to newer position in change list

g; - go to older position in change list

Ctrl + **]** - jump to the tag under cursor

files

:cn[ext] - jump to the next match

:cp[revious] - jump to the previous match

:cope[n] - open a window containing the list of matches

:ccl[ose] - close the quickfix window

Tabs

: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)