

# Vim Cheat Sheet

Global	Editing	Macros	Working with multiple files
<b>:h[elp]</b> <b>keyword</b> - open help for keyword	<b>r</b> - replace a single character.	<b>qa</b> - record macro a	<b>:e[dit]</b> <b>file</b> - edit a file in a new buffer
<b>:sav[eas]</b> <b>file</b> - save file as	<b>R</b> - replace more than one character, until <b>ESC</b> is pressed.	<b>q</b> - stop recording macro	<b>:bn[ext]</b> - go to the next buffer
<b>:clo[se]</b> - close current pane	<b>J</b> - join line below to the current one with one space in between	<b>@a</b> - run macro a	<b>:bp[revious]</b> - go to the previous buffer
<b>:ter[minal]</b> - open a terminal window	<b>gJ</b> - join line below to the current one without space in between	<b>@@</b> - rerun last run macro	<b>:bd[elete]</b> - delete a buffer (close a file)
<b>K</b> - open man page for word under the cursor	<b>g~</b> - switch case up to motion	Cut and paste	<b>:b[uffer]#</b> - go to a buffer by index #
Cursor movement	<b>gu</b> - change to lowercase up to motion	<b>yy</b> - yank (copy) a line	<b>:b[uffer]</b> <b>file</b> - go to a buffer by file
<b>h</b> - move cursor left	<b>gU</b> - change to uppercase up to motion	<b>2yy</b> - yank (copy) 2 lines	<b>:ls</b> or <b>:buffers</b> - list all open buffers
<b>j</b> - move cursor down	<b>cc</b> - change (replace) entire line	<b>yw</b> - yank (copy) the characters of the word from the cursor position to the start of the next word	<b>:sp[lit]</b> <b>file</b> - open a file in a new buffer and split window
<b>k</b> - move cursor up	<b>c\$</b> or <b>C</b> - change (replace) to the end of the line	<b>yiw</b> - yank (copy) word under the cursor	<b>:vs[plit]</b> <b>file</b> - open a file in a new buffer and vertically split window
<b>l</b> - move cursor right	<b>ciw</b> - change (replace) entire word	<b>yaw</b> - yank (copy) word under the cursor and the space after or before it	<b>:vert[ical]</b> <b>ba[ll]</b> - edit all buffers as vertical windows
<b>gj</b> - move cursor down (multi-line text)	<b>cw</b> or <b>ce</b> - change (replace) to the end of the word	<b>y\$</b> or <b>Y</b> - yank (copy) to end of line	<b>:tab</b> <b>ba[ll]</b> - edit all buffers as tabs
<b>gk</b> - move cursor up (multi-line text)	<b>S</b> - delete character and substitute text	<b>p</b> - put (paste) the clipboard after cursor	<b>Ctrl</b> + <b>ws</b> - split window
<b>H</b> - move to top of screen	<b>S</b> - delete line and substitute text (same as cc)	<b>P</b> - put (paste) before cursor	<b>Ctrl</b> + <b>wv</b> - split window vertically
<b>M</b> - move to middle of screen	<b>xp</b> - transpose two letters (delete and paste)	<b>gp</b> - put (paste) the clipboard after cursor and leave cursor after the new text	<b>Ctrl</b> + <b>ww</b> - switch windows
<b>L</b> - move to bottom of screen	<b>u</b> - undo	<b>gP</b> - put (paste) before cursor and leave cursor after the new text	<b>Ctrl</b> + <b>wq</b> - quit a window
<b>w</b> - jump forwards to the start of a word	<b>U</b> - restore (undo) last changed line	<b>dd</b> - delete (cut) a line	<b>Ctrl</b> + <b>wx</b> - exchange current window with next one
<b>W</b> - jump forwards to the start of a word (words can contain punctuation)	<b>Ctrl</b> + <b>r</b> - redo	<b>2dd</b> - delete (cut) 2 lines	<b>Ctrl</b> + <b>w=</b> - make all windows equal height & width
<b>e</b> - jump forwards to the end of a word	<b>.</b> - repeat last command	<b>dw</b> - delete (cut) the characters of the word from the cursor position to the start of the next word	<b>Ctrl</b> + <b>wh</b> - move cursor to the left window (vertical split)
<b>E</b> - jump forwards to the end of a word (words can contain punctuation)	Marking text (visual mode)	<b>diw</b> - delete (cut) word under the cursor	<b>Ctrl</b> + <b>wl</b> - move cursor to the right window (vertical split)
<b>b</b> - jump backwards to the start of a word	<b>v</b> - start visual mode, mark lines, then do a command (like y-yank)	<b>daw</b> - delete (cut) word under the cursor and the space after or before it	<b>Ctrl</b> + <b>wj</b> - move cursor to the window below (horizontal split)
<b>B</b> - jump backwards to the start of a word (words can contain punctuation)	<b>V</b> - start linewise visual mode	<b>:3,5d</b> - delete lines starting from 3 to 5	<b>Ctrl</b> + <b>wk</b> - move cursor to the window above (horizontal split)
<b>ge</b> - jump backwards to the end of a word	<b>Ctrl</b> + <b>v</b> - start visual block mode	<b>:g/{pattern}/d</b> - delete all lines containing pattern	<b>Ctrl</b> + <b>wh</b> - make current window full height at far left (leftmost vertical window)
<b>gE</b> - jump backwards to the end of a word (words can contain punctuation)	<b>O</b> - move to other corner of block	<b>:g!/{pattern}/d</b> - delete all lines not containing pattern	<b>Ctrl</b> + <b>wL</b> - make current window full height at far right (rightmost vertical window)
<b>%</b> - move to matching character (default supported pairs: '()', '{}', '[]' - use :h matchpairs in vim for more info)	<b>aw</b> - mark a word	<b>d\$</b> or <b>D</b> - delete (cut) to the end of the line	<b>Ctrl</b> + <b>wJ</b> - make current window full width at the very bottom (bottommost horizontal window)
<b>0</b> - jump to the start of the line	<b>ab</b> - a block with ()	<b>x</b> - delete (cut) character	<b>Ctrl</b> + <b>wK</b> - make current window full width at the very top (topmost horizontal window)
<b>^</b> - jump to the first non-blank character of the line	<b>aB</b> - a block with {}	Indent text	Diff
<b>\$</b> - jump to the end of the line	<b>at</b> - a block with <> tags	<b>&gt;&gt;</b> - indent (move right) line one shiftwidth	<b>zf</b> - manually define a fold up to motion
<b>g_</b> - jump to the last non-blank character of the line	<b>ib</b> - inner block with ()	<b>&lt;&lt;</b> - de-indent (move left) line one shiftwidth	<b>zd</b> - delete fold under the cursor
<b>gg</b> - go to the first line of the document	<b>iB</b> - inner block with {}	<b>&gt;%</b> - indent a block with () or {} (cursor on brace)	<b>za</b> - toggle fold under the cursor
<b>G</b> - go to the last line of the document	<b>it</b> - inner block with <> tags	<b>&lt;%</b> - de-indent a block with () or {} (cursor on brace)	<b>zo</b> - open fold under the cursor
			<b>zc</b> - close fold under the cursor
			<b>zr</b> - reduce (open) all folds by one level
			<b>zm</b> - fold more (close) all folds by one level
			<b>zi</b> - toggle folding functionality
			<b>]c</b> - jump to start of next change
			<b>[c</b> - jump to start of previous change
			<b>do</b> or <b>:diffg[et]</b> - obtain (get) difference (from other buffer)

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

**Ctrl + j** - begin new line 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

**: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)