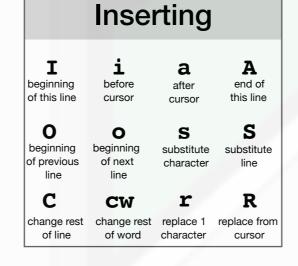
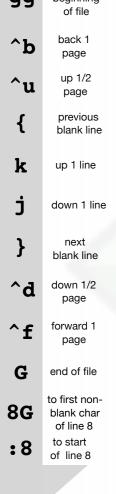
0 B b h E W \$ e W first beginning beginning beginning previous end end end previous previous next non-blank of next of next of line WORD word character character of word of WORD of line character WORD

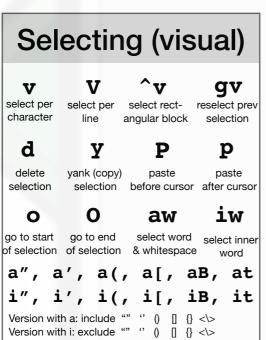
e! :e path :view path :r file **:** q :q! load file in quit, discard return to load and edit insert conunmodified file read-only mode changes aiven file tents of file : pwd : cd :saveas path :browse old echo current List recent files, allow to change dir working dir save with new name select one by number.

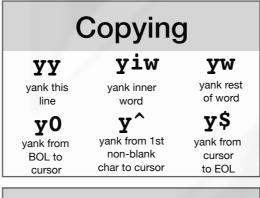
Buffers previous jump location :1s **: b** n **: b** file beginning jump to iump to buffer with list all of file buffer nr n containing file buffers back 1 :bp :bn : bd page jump to delete current buffer up 1/2 prev buffer next buffer page :bufdo C :e path :enew previous load and edit blank line apply command new blank given file c to all buffers buffer up 1 line down 1 line Selecting (visual)











Undo/Redo :earlier 4m u take document to state of 4 min ago undo ^r :later 50s redo 50 sec later in document state





In current line replace all (/q) 'foo' with 'bar' :%s/foo/bar/q

In all lines replace all 'foo' with 'bar :%s/foo/bar/gic In all lines replace all 'foo' with 'bar', caseinsensitive, and ask for confirmation :3,9s/foo/bar/q In lines 3 to 9 (incl) replace all 'foo' with 'bar :.,.+5/^/ From current line to 5 line further, prepend 3 spaces at the beginning of each line :s/\<foo\>/bar/q In current line replace all occurrences of 'foo

that are a separate word, with 'bar' :5,\$s/foo//g From current line to last line, replace all 'foo' with nothing (= delete 'foo') :%s/\t/ /q In all lines, replace tab with 4 spaces :%s/red\|pink/blue/q In all lines, replace all instances of "red" and "pink" with "blue" :'<,'>s/foo/bar/q In the visually pre-selected lines, replace foo

/q

Replace

:s/foo/bar/q

Move

5 : m my current line to after line 5

mv current line

to before line 1

mv current line

to after last line

 $: m ' \} - 1$

mv current line to end of paragraph

:2,4m \$

mv lines 2,3,4 to

after last line

:2,4m 9 my lines 2.3.4

to after line 9

:.,.+2m 9

my 3 lines starting from the current one to after line 9

vim-multiple-cursors

Selects current word in visual mode, and puts ^n cursor at end of it. Pressing it again find next occurrence and puts a cursor at the end.

Remove last virtual cursor and go back to last one.

Skip last virtual cursor and go to next occurrence of the word.

After all cursors are placed, you may opt to go out of visual mode (press v) and use the normal commands

Search/Replace in multiple files

:vimgrep /foo/g *.py Search in all files. Use :copen to get a list of matches.

:bufdo %s/foo/bar/ge | update

Apply to all buffers. Ignore pattern not found errors (e). Save file if changes were made (update).

:arg *.cpp :argadd *.h Create arglist with C++ files and load them into buffers

Add header files to arglist, and load into buffers

:argdo %s/foo/bar/ge

update

Apply to every file in arglist

netrw & Ctrl-p

:edit|read|write sftp://user@hostname/path

:edit folder (folder can be .)

:Vexplore split left-right and show file browser

i toggle between file browser view options /path move cursor to nearest match in file browser

:^p filename Fuzzy search file. Use ^f and ^b to toggle

between files, buffers, and most recently used. Can be used for "Open Recent..."

Global command

:[range]q/pattern/command

Delete all lines that contain 'foo' :q/foo/d

Move all lines from cursor to EOF. :.,\$q!/foo/m\$ that do not contain 'foo', to EOF.

For all lines move them to the $:q/^/m0$ beginning of the file = reverse file.

For all lines starting with 'foo', :q/^foo/s/\$/bar substitute/append 'bar' to EOL.

:q/foo/# Show with line nrs. Then go to a line.

Case switching

Toggle case of character under cursor

g~iw Toggle case of the inner word

q~\$ Toggle case from cursor to EOL

qUw Convert rest of word to large caps

gUU Convert current line to upper case

gu0 Convert from BOL to cursor to small caps.

Vimtex

\11 Start/stop continuous compiling mode

Jump to PDF viewer at the same point of the cursor \lv

\lc Clean auxiliary files

[[Jump to the previous (sub)section

]] Jump to the next (sub)section

Move between matching delimiters

Ctags & TagBar

Jump to (1st) match of the tag. Name can be a :tag name regular expression (e.g. /Detect*).

Jump to (1st) match of the tag in a new window. :stag name Name can be a regular expression (e.g. /Detect*).

List all tag matches, and lets you select one. If :ts name

name is absent, use list of last :tag command.

^w} g]

Jump to tag Show tag match List all matches of Go back under the cursor one jump in sep window tag under cursor

:Tagbar Toggle show/hide tagbar

:!ctaqs -R Generate tags file, recursively

Windows

:split :vsplit :close :only

split topbottom

^1

split left-right

keep only current window close current

Indenting

>> << > >ap >aB un-indent indent un-indent current line selected lines selected lines current parag. block

CSV

:WhatColumn Show current column nr

:nrColumns Show total numbers of columns

:DeleteColumn 2 Delete column nr 2

:MoveColumn 2 4 Move col 2 to right next to col 4

:DupColumn Duplicate current column

:Header 2 Open separate fixed window with first 2 header lines. Close with :Header!

Prettify the columns by aligning them :%ArrangeColumn!

:NewDelimiter ; Change the delimiter to:

Sort lines 1-10 of column 3 :1,10Sort 3

:SearchInColumn 1 /foo/

Search for patter foo in column 1

:%Substitute 1,4/foo/bar/qc

Subst in cols 1-4 foo with bar, globally + ask for confirmation

Fugitive

:Gblame Show git blame info for an indexed file.

:Gedit SHA Show info for blob, tree, commits, and tags

:Gedit master:python/transits.py Read-only!

:Gdiff Open 3-way diff of current file, to solve conflicts

E.g. if you have conflicts after:

(feature) \$ git checkout master (master) \$ git merge feature

Left window: version of the master branch Right window: version of the feature branch

Center window: current working copy

Keep your cursor in the center window (working copy).

include left version in working copy :diffget //2 include right version in working copy :diffget //3 :diffupdate update the diff highlighting

:Glog -- % Show the git log of this file per commit, and let's you select a commit

:Gstatus Show the git index

gv

:GV Open commit browser. Browse, press enter to get more info on the commit. In visual mode: show commit info on the selected lines.

:GV! Only show commit info of the current file

Quit the commit browser :q

Easymotion

\\w Trigger word motion, then select appropriate letter. Forward.

\\b Idem but backward

\\s char Trigger character motion, then select letter.

Markers

Set marker named 'a' under cursor m a Same but override previous marker 'a' m a! Go to start of line containing marker 'a' 'a `a Jump to marker 'a' (back quote) :marks Show a list of markers :delm ab /! Delete marker 'a' and 'b' / all markers

Change from cursor to marker 'a'

:c `a :]' / [' Jump to next/prev line with marker

Special marker: last change in buffer

Outliner (.otl files)

\cb \cx \cd \n \Insert Toggle Delete Set fold Draw checkbox checkbox level to n dashed line

\s \S \t \d

Sort sub-tree Sort sub-tree Append time Append date under cursor under cursor stamp stamp ascending descending HH:MM:SS YYYY:MM:DD

Folds

zo/zczo/zczm/zrzj/zkOpen/closeOpen/close all folds under cursorIncrease /reduce fold levelGo to start of next/previous fold next/previous fold

Clipboard

:w !pbcopy Write selected lines to clipboard (visual mode)

:r !pbpaste Insert contents of clipboard at current position

Macros

qaStart recording to register 'a'@aExecute macro...Fancy commands5@aExecute 5 timesqStop recording: regList all macros

Miscellaneous

^a / ^x Incr/decr the nr under cursor. Useful to automate.
f / F char Jump to next/prev occurrence of char on this line
Join this line with the next one
set scrollbind Scroll synchronously. To be set in all windows that need to be in-sync. Add! to command to undo.
ab asap as soon as possible Replace abbreviation on the fly

:set listchars=eol:¬,tab:>·,trail:~,extends:>,precedes:<,space:_

:set list Show all the invisible characters

:set nolist Stop showing all the invisible characters