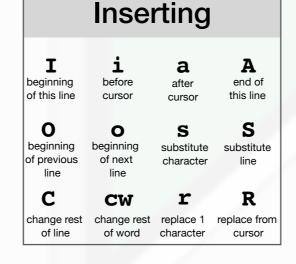
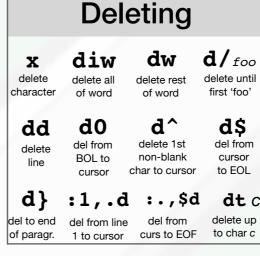
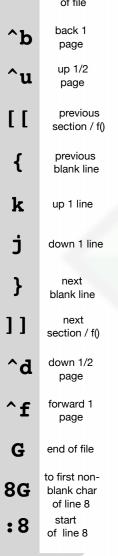
0 В b h E W \$ e W first beginning beginning beginning previous end end end previous previous next non-blank of next of next of line WORD word character character of word of WORD of line character WORD

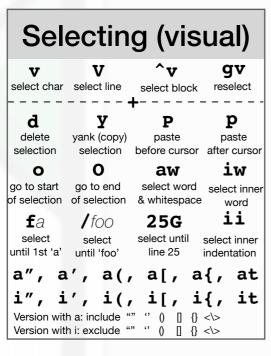
e! :e path :view path :r file **:** q :q! load file in quit, discard return to load and edit insert conunmodified file read-only mode changes aiven file tents of file : pwd : cd :saveas path :browse old echo current List recent files, allow to change dir working dir save with new name select one by number.

**Buffers** ^0 previous jump location :1s **: b** n : **b** file jump to beginning jump 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 apply command new blank section / f() given file c to all buffers buffer previous blank line Selecting (visual) up 1 line V v down 1 line



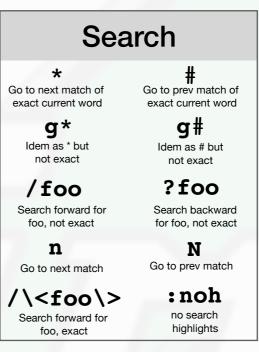


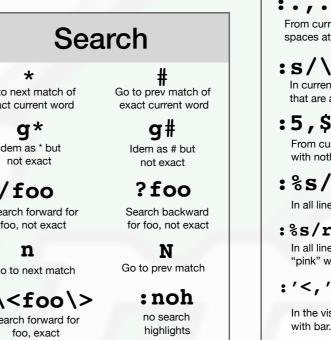


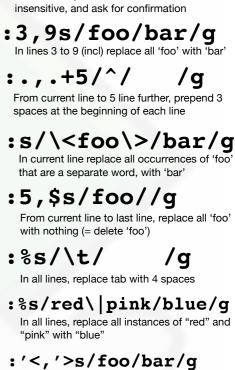


Copying		
УУ	yiw	yw
yank this line	yank inner word	yank rest of word
y0	У^	у\$
yank from BOL to cursor	yank from 1st non-blank char to cursor	yank from cursor to EOL

### Undo/Redo :earlier 4m **u** undo :later 50s :UndotreeToggle







Substitute

In current line replace all (/g) 'foo' with 'bar'

:s/foo/bar/q

:%s/foo/bar/q

In all lines replace all 'foo' with 'bar

:%s/foo/bar/gic

In all lines replace all 'foo' with 'bar', case-

In the visually pre-selected lines, replace foo

#### Move Search & Replace

:2,4m \$ :m '}-1 mv this line my this line to my lines 2,3,4 to to after line n end of paragr. after last line :2,4m 9 :m S :...+2m 9my this line mv lines 2,3,4 mv 3 lines from to after to after this line to last line after line 9

/foo Search & highlight all 'foo', go to 1st match cgn Change current search pattern (gn) Get out of edit mode <esc> Repeat for next match Go to next/prev match if need to skip one

# 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 Create arglist with C++ files and load them into buffers :argadd \*.h Add header files to arglist, and load into buffers

:arqdo %s/foo/bar/qe 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

toggle between file browser view options/pathmove cursor to nearest match in file browser

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

between files, buffers, and <u>most recently</u> used. Can be used for "Open Recent..."

### Global command

: [range]g/pattern/command

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

Move all lines from cursor to EOF, that do not contain 'foo', to EOF.

For all lines move them to the beginning of the file = reverse file.

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

:g/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

**g~\$** Toggle case from cursor to EOL

**gUw** 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

\1v Jump to PDF viewer at the same point of the cursor

\1c Clean auxiliary files

[ Jump to the previous (sub)section

] ] Jump to the next (sub)section

% Move between matching delimiters

### Tags and YCM

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

**:ts** name List all tag matches, and lets you select one. If name is absent, use list of last :tag command.

Jump to tag List all matches of Go back In insert mode: show under cursor tag under cursor one jump window with f() def.

:Tagbar Toggle show/hide tagbar

Double-click or press enter on a tag to jump

• Press space on a function to display arg list.

:!ctags -R Generate tags file, recursively, in current dir

#### **Windows**

:split :vsplit :only :close

split topbottom

split left-right

keep only current window

close current

# Indenting

#### **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!

:%ArrangeColumn! Prettify the columns by aligning them

:NewDelimiter; Change the delimiter to;

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

:SearchInColumn 1 /foo/

Search for patter foo in column 1

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

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!

**:Glog** -- % Show the git log of current file per commit, and lets you select a commit. Press enter on parent commits, to go further back.

:Gstatus Show the git index

**:Gdiff** Shows differences between working copy of the current file and the index, in a split window. **:q** to exit.

In case of merge conflicts: open 3-way differences.

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).

:diffget //2 include left version in working copy
:diffget //3 include right version in working copy

:diffupdate update the diff highlighting

#### 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

**: q** Quit the commit browser

# **Special Text Objects**

**gn** Current search pattern

ii Inner indentation level, no line above

**aa** Function <u>arg</u> under cursor, with comma

**a**<u>i</u>ndentation level + one line above (*Python*)

**ia** Function **a**rg under cursor, without comma (*inner*)

a<u>I</u> Indentation level + line above + line below (C++)

Example in C++ : caI : change current indentation level incl. { } above/below

### **Macros**

qa Start recording to register 'a'Ga Execute macroFancy commands5@a Execute 5 times

**q** Stop recording

Execute 5 times

req List all macros

:reg Lis

### Outliner (.otl files)

\cb \cx \cd \n \Insert Toggle Delete Set fold Draw checkbox 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/ZC**Open/close
Open/close all
fold under

Open/close all

cursor

cursor

zO/zC zm/zr
Open/close all folds under lncrease /reduce fold level

zr zj/zk
Go to start o

Go to start of next/previous fold

# Clipboard

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

:r !pbpaste Insert contents of clipboard at current position

## (Un)commenting

 ↑▼
 Block select

 select
 Select the block

 I
 Insert mode at beginning

 //
 Write comment symbols

<esc> Exit & apply to all lines

**^v** Bloselect Se

Block select

Select comment symbols

**d** delete

### **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 Jump to marker 'a' (back quote) `a Show a list of markers :marks :delm ab /! Delete marker 'a' and 'b' / all markers :c `a Change from cursor to marker 'a' :]' / [' Jump to next/prev line with marker Special marker: last change in buffer

### Miscellaneous

# **Easymotion**

\\w Word motion, then select appropriate letter. Forward.
\\b Idem but backward
\\s char Trigger character motion, then select letter.