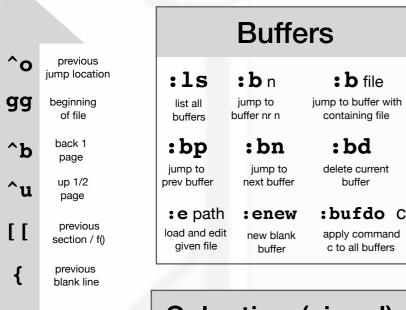
0 В b h E W \$ e W first beginning beginning beginning previous end end end previous previous next non-blank of next of next of line WORD word character character of word of WORD of line character WORD

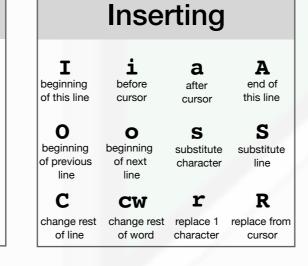
e! :e path :view path :r file **:** q :q! load file in quit, discard return to load and edit insert conunmodified file read-only mode changes aiven file tents of file : pwd : cd :saveas path :browse old echo current List recent files, allow to change dir working dir save with new name select one by number.

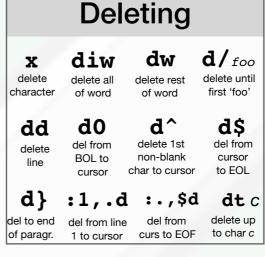


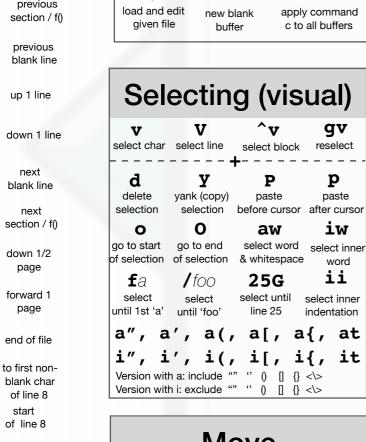
k

G

:8







:2,4m \$

my lines 2,3,4 to

after last line

:2,4m 9

mv lines 2,3,4

to after

: **m** n

mv this line

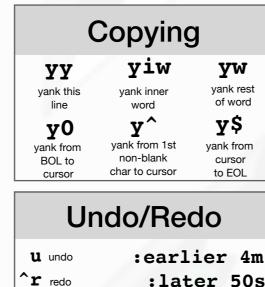
to after line n

:m S

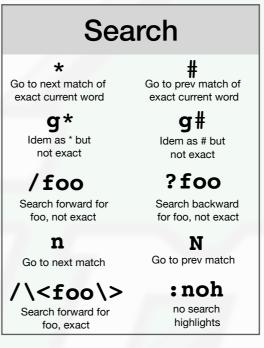
my this line

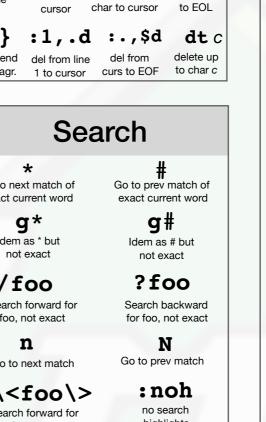
to after

last line



:UndotreeToggle





:5,\$s/foo//q From current line to last line, replace all 'foo' with nothing (= delete 'foo') :%s/\t/

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

Substitute

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"

/q

/q

:'<,'>s/foo/bar/q

In the visually pre-selected lines, replace foo

Move Search & Replace

:m '}-1 my this line to end of paragr. :...+2m 9mv 3 lines from this line to 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/path move cursor to nearest match in file browser

P Fuzzy search file. Use ^f and ^b to toggle between files, buffers, and **m**ost **r**ecently

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

>> < > > ap >aB

ident un-indent indent un-indent indent current
current line 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!

:%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!

: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 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

: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

: q Quit the commit browser

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 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 ZO/ZC
Open/close all

1 fold under

cursor

Open/close all folds under cursor

zm/zr
Increase /reduce

fold level

zj/zk
Go to start of next/previous fold

J

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

Special Text Objects

The Current search pattern The Inner indentation level, no line above

 $\underline{a}\underline{\underline{a}} \quad \begin{array}{ll} \text{Function arg under cursor,} \\ \text{with comma} \end{array} \quad \underline{a}\underline{\underline{i}} \quad \begin{array}{ll} \text{Indentation level + one line} \\ \text{above (Python)} \end{array}$

ia Function arg under cursor, without comma (inner) a Indentation level + line above + line below (C++)

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

(Un)commenting

 T
 Block select
 T
 Block select

 Select
 Select the block
 Select
 Select

 I
 Insert mode at beginning
 d
 delet

// Write comment symbols

<esc> Exit & apply to all lines

 ^▼
 Block select

 Select
 Select comment symbols

 d
 delete

Miscellaneous

^a / ^xIncr/decr the nr under cursor. Useful to automate.f / F charJump 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