zH \$ 0 В b h E W zL e W shift view beginning beginning beginning previous shift view end previous previous next end end non-blank 1/2 screen of next of next of WORD of line WORD word character character of word 1/2 screen of line character left WORD word right

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

Substitute

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

:s/foo/bar/q

:%s/foo/bar/q

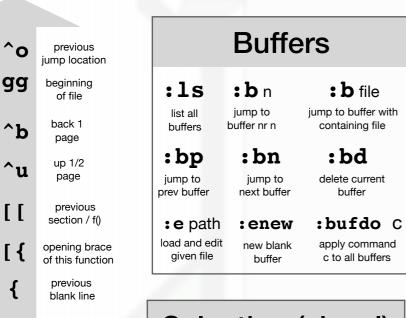
In all lines replace all 'foo' with 'bar

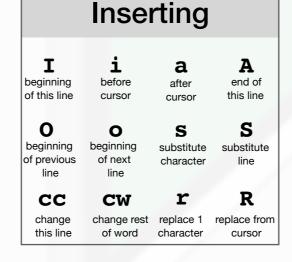
:%s/foo/bar/gic

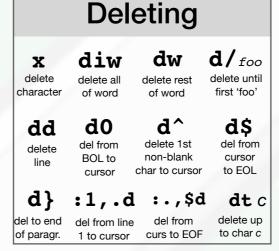
:3,9s/foo/bar/q

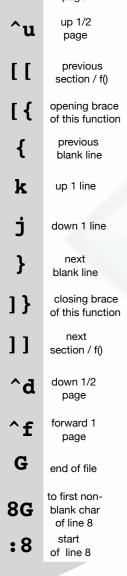
insensitive, and ask for confirmation

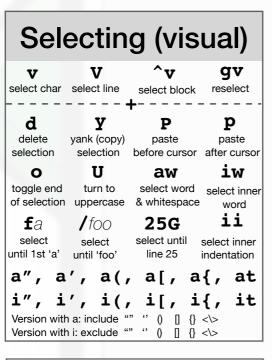
In all lines replace all 'foo' with 'bar', case-











: m. +2

my this line

2 lines below

:'<,'>m'>-5

Move selection

to 5 lines

before end of

:**m** n

mv this line

to after line n

:2,4m 9

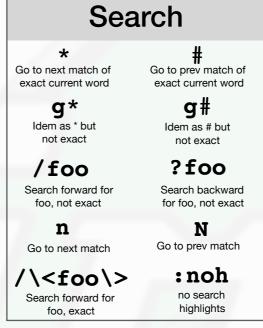
mv lines 2,3,4

to after

line 9

Copying				
YY yank this	yiw yank inner	YW yank rest		
line y 0	word	of word		
yank from BOL to cursor	yank from 1st non-blank char to cursor	yank from cursor to EOL		
Undo/Redo				

:UndotreeToggle



Search		
Go to next match of exact current word	Go to prev match of exact current word	
g* Idem as * but not exact	g# Idem as # but not exact	
/foo	?foo	
Search forward for foo, not exact	Search backward for foo, not exact	
n	N	
Go to next match	Go to prev match	
/\ <foo\></foo\>	:noh	
Search forward for foo, exact	no search highlights	

Search		:.,.+5/^/ /g
		From current line to 5 line further, prepend 3 spaces at the beginning of each line
mext match of current word g* em as * but not exact foo ch forward for on, not exact	# Go to prev match of exact current word g# Idem as # but not exact ? foo Search backward for foo, not exact	<pre>:s/\<foo\>/bar/g 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/ /g In all lines, replace tab with 4 spaces :%s/red\ pink/blue/g</foo\></pre>
n to next match < foo \ > rch forward for foo, exact	N Go to prev match no search highlights	In all lines, replace all instances of "red" and "pink" with "blue" : '<, '>s/foo/bar/g In the visually pre-selected lines, replace foo with bar.
0	la /D l	!

\mathbf{f}_{c} \mathbf{F}_{C} jump to jump to jump to prev occu matching next occur of c bracket

scroll current line to middle of the screen

Move Search & Replace

:2,4m \$

my lines 2.3.4 to

after last line

:.,.+2m 9

mv 3 lines from

this line to

after line 9

u undo

/foo Search & highlight all 'foo', go to 1st match cgn Change current search pattern (gn) - no ':' Get out of edit mode <esc> Repeat for next match Go to next/prev match if need to skip one

:earlier 4m

:later 50s

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 :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

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 **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

 $\begin{tabular}{ll} \begin{tabular}{ll} \beg$

\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 --extra=+f Create 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!

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

[c Previous hunk lc Next hunk

:Gdiffsplit! In case of merge conflicts: open 3-way diffs

E.g. if you have conflicts after: \$ git checkout master \$ git merge feature

Left/center/right window: master / current working copy / feature 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

fzf

:Files Fuzzy search files

:Lines Fuzzy grep in all open buffers

:BLines Fuzzy grep in current buffer

: Tags Fuzzy search all tags

:BTags Fuzzy search tags in current buffer

:GFiles? Git status

:Commits Fuzzy search in all commits

:BCommits Fuzzy search in commits of this buffer

Special Text Objects

gn Current search pattern

ii Inner indentation level, no line above

a<u>a</u> Function <u>a</u>rg under cursor, with comma

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

Function <u>arg</u> 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

Tabularize

:Tab/=

Align selection on the '=' char

:Tab/,/r1c110

First part right aligned, then 1 space, then comma center aligned, then 1 space, then part left aligned, no extra space, then rest

Miscellaneous

^a / ^x Incr/decr the nr under cursor. Join this line with the next one

:set scrollbind Scroll synchronously. Set in all windows that need to be in-sync. Add! to to undo.

:ab fe for example Replace abbreviation on the fly

:set list Show all the invisible characters

:set nolist Stop showing all the invisible characters

^00 After startup: load last edited file ^n In insert mode: word completion

:Matrix Follow the white rabbit

:ToggleWhitespace (Un)highlight trailing whitespace

:StripWhitespace Strip trailing whitespace

:set wrap linebreak Wrap text around

:%s/[ctrl]v[ctrl]m/\r/g Convert dos ^M to unix carriage return

gv Reselect last selection

Vr-Select line, replace all chars with `-`

Folds

zo/zc Open/close

1 fold under

cursor

z0/zC

Open/close all folds under cursor

zm/zr

Increase /reduce fold level

Go to start of next/previous fold

zj/zk

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) 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

Macros

@a qa Start recording to register 'a' Execute macro Fancy commands 5@a Execute 5 times q Stop recording :req List all macros

Repeating

; Rpt last f c - forward

& Rpt last :s/x/y/ subst

Block select

delete

Select comment symbols

Rpt last f c - backward

Repeat search - forward

Repeat search - backward

Rpt last ':' command

Rpt last change in Normal mode

Swapping

Select the first part of text to be swapped... CX {motion}

... go to the 2nd part of text. Repeat cmd: swaps

CXX Like cx for the current line CXC Clear selection and memory

(Un)commenting

Old School:

Block select

Select the block

Insert mode at beginning

Write comment symbols <esc> Exit & apply to all lines

With tcomment:

Toggle comment. E.g gc4j or gcaB gc{motion}

Toggle comment for this line. gcc

Surround

Surround motion by given delimiter **ys** {motion}{delimiter}

yss {delimiter} Surround current line by delimiter

ds {delimiter} Delete surrounding delimiter

cs {old delim}{new delim} Change surrounding delimiters

cst {new delim} Change surrounding tags to delimiters

Easymotion

\\w Word motion, then select appropriate letter. Forward.

\\b Idem but backward

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

MacVim

%= / **%**-Increase/decrease font size

Insert Mode

^0 In INSERT mode: go back to normal mode for 1 command

^r=<*expr>* In Insert mode: compute and insert result of *expression*

Clipboard

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

:r !pbpaste Insert contents of clipboard at current position