

0 ^ B b h l e w E W \$

beginning of line first non-blank character previous WORD previous word previous character next character end of word beginning of next word end of WORD beginning of next WORD end of line

:e path :e! :view path :r file :q :q!
load and edit return to load file in insert con- quit quit, discard
given file unmodified file read-only mode tents of file changes

:w :saveas path :browse old :cd :pwd
write file save with new name List recent files, allow to change dir echo current
select one by number. working dir

^o previous jump location
gg beginning of file
^b back 1 page
^u up 1/2 page
[[previous section / f()
{ previous blank line
k up 1 line
j down 1 line
} next blank line
]] next section / f()
^d down 1/2 page
^f forward 1 page
G end of file
8G to first non-blank char of line 8
:8 start of line 8

Buffers

:ls :b n :b file
list all jump to jump to buffer with
buffers buffer nr n containing file

:bp :bn :bd
jump to jump to delete current
prev buffer next buffer buffer

:e path :enew :bufdo C
load and edit new blank apply command
given file buffer c to all buffers

Inserting

I i a A
beginning before after end
of this line cursor cursor of this line

O o s S
beginning beginning substitute substitute
of previous of next character line
line

C cw r R
change rest change rest replace 1 replace from
of line of word character cursor

Deleting

x diw dw d/ foo
delete delete all delete rest delete until
character of word of word first 'foo'

dd d0 d^ d\$
delete del from delete 1st del from
line BOL to cursor non-blank cursor
char to cursor to EOL

d} :1,.d :.,\$d dt c
del to end del from line del from delete up
of paragr. 1 to cursor curs to EOF to char c

Substitute

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

:%s/foo/bar/g
In all lines replace all 'foo' with 'bar'

:%s/foo/bar/gic
In all lines replace all 'foo' with 'bar', case-
insensitive, and ask for confirmation

:3,9s/foo/bar/g
In lines 3 to 9 (incl) replace all 'foo' with 'bar'

:.,.+5/^/ /g
From current line to 5 line further, prepend 3
spaces at the beginning of each line

: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
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.

Selecting (visual)

v V ^v gv
select char select line select block reselect

d y P p
delete yank (copy) paste paste
selection selection before cursor after cursor

o O aw iw
go to start go to end select word select inner
of selection of selection & whitespace word

fa /foo 25G ii
select select select until select inner
until 1st 'a' until 'foo' line 25 indentation

a", a', a(, a[, a{, at
i", i', i(, i[, i{, it

Version with a: include "" " 0 [] {} <>
Version with i: exclude "" " 0 [] {} <>

Copying

yy yiw yw
yank this yank inner yank rest
line word of word

y0 y^ y\$
yank from yank from 1st yank from
BOL to non-blank char to cursor cursor
cursor to EOL

Undo/Redo

u undo :earlier 4m
^r redo :later 50s

:UndotreeToggle

Search

* #
Go to next match of Go to prev match of
exact current word exact current word

g* g#
Idem as * but Idem as # but
not exact not exact

/foo ?foo
Search forward for Search backward
foo, not exact for foo, not exact

n N
Go to next match Go to prev match

/\<foo\> :noh
Search forward for no search
foo, exact highlights

Move

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

:m \$:2,4m 9 :.,.+2m 9
mv this line mv lines 2,3,4 mv 3 lines from
to after to after this line to
last line line 9 after line 9

Search & Replace

/foo Search & highlight all 'foo', go to 1st match

cgn Change current search pattern (gn)

<esc> Get out of edit mode

. Repeat for next match

n / N 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

: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 <u>most recently used</u> . Can be used for “ Open Recent... ”

Global command	
: [range] g/pattern/command	
:g/foo/d	Delete all lines that contain ‘foo’
:.,\$g!/foo/m\$	Move all lines from cursor to EOF, that do not contain ‘foo’, to EOF.
:g/^/m0	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	
\ll	Start/stop continuous compiling mode
\lv	Jump to PDF viewer at the same point of the cursor
\lc	Clean auxiliary files
[[Jump to the previous (sub)section
]]	Jump to the next (sub)section
%	Move between matching delimiters

Tags and YCM

:tag <i>name</i>	Jump to (1st) match of the tag. :stag shows it in a new window. <i>Name</i> can be a regex (e.g. <i>/Detect*</i>).		
:stag			
:ts <i>name</i>	List all tag matches, and lets you select one. If name is absent, use list of last :tag command.		
^]	g]	^o	^Space Tab
Jump to tag under cursor	List all matches of tag under cursor	Go back one jump	In insert mode: show window with f() def.
:Tagbar	Toggle show/hide tagbar <ul style="list-style-type: none">• 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 <i>current</i> dir		

Windows			
:split	:vsplit	:only	:close
split top-bottom	split left-right	keep only current window	close current

Indenting					
>>	<<	>	<	>ap	>aB
indent current line	un-indent current line	indent selected lines	un-indent selected lines	indent current parag.	indent current 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: <pre>(feature) \$ git checkout master (master) \$ git merge feature</pre> 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). <pre>:diffget //2 include left version in working copy :diffget //3 include right version in working copy :diffupdate update the diff highlighting</pre>	
: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	
m a	Set marker named ‘a’ under cursor
m a!	Same but override previous marker ‘a’
’a	Go to start of line containing marker ‘a’
`a	Jump to marker ‘a’ (back quote)
:marks	Show a list of markers
: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

Outliner (.otl files)				
\cb	\cx	\cd	\n	\-
Insert checkbox	Toggle checkbox	Delete checkbox	Set fold level to <i>n</i>	Draw dashed line
\s	\S	\t	\d	
Sort sub-tree under cursor ascending	Sort sub-tree under cursor descending	Append time stamp HH:MM:SS	Append date stamp YYYY:MM:DD	

Folds			
zo/zc	zO/zC	zm/zr	zj/zk
Open/close 1 fold under cursor	Open/close all folds under cursor	Increase /reduce fold level	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

Macros			
 qa	Start recording to register ‘a’	@a	Execute macro
 ...	Fancy commands	5@a	Execute 5 times
 q	Stop recording	:reg	List all macros

Special Text Objects			
gn	Current search pattern	i i	Inner indentation level, no line above
aa	Function arg under cursor, with comma	a i	Indentation level + one line above (<i>Python</i>)
ia	Function arg under cursor, without comma (<i>inner</i>)	a I	Indentation level + line above + line below (<i>C++</i>)
Example in C++ : caI : change current indentation level incl. { } above/below			

(Un)commenting			
 ^v	Block select	 ^v	Block select
 select	Select the block	 select	Select comment symbols
 I	Insert mode at beginning	 d	delete
 //	Write comment symbols		
 <esc>	Exit & apply to all lines		

Miscellaneous	
^a / ^x	Incr/decr the nr under cursor. Useful to automate.
f / F char	Jump to next/prev occurrence of <i>char</i> on this line
J	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