

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

^o previous jump location

gg beginning of file

^b back 1 page

^u up 1/2 page

{ previous blank line

k up 1 line

j down 1 line

} next blank line

^d down 1/2 page

^f forward 1 page

G end of file

8G to first non-blank char of line 8

:8 to start of line 8

Buffers

:ls list all buffers	:b n jump to buffer nr n	:b file jump to buffer with containing file
:bp jump to prev buffer	:bn jump to next buffer	:bd delete current buffer
:e path load and edit given file	:enew new blank buffer	:bufdo C apply command c to all buffers

Inserting

I beginning of this line	i before cursor	a after cursor	A end of this line
O beginning of previous line	o beginning of next line	s substitute character	S substitute line
C change rest of line	cw change rest of word	r replace 1 character	R replace from cursor

Deleting

x delete character	diw delete all of word	dw delete rest of word	d/ foo delete until first 'foo'
dd delete line	d0 del from BOL to cursor	d^ delete 1st non-blank char to cursor	d\$ del from cursor to EOL
d} del to end of paragr.	:1,.d del from line 1 to cursor	:.,\$d del from curs to EOF	dt c delete up to char c

Replace

: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/\
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 select per character	V select per line	^v select rectangular block	gv reselect prev selection
d delete selection	y yank (copy) selection	P paste before cursor	p paste after cursor
o go to start of selection	O go to end of selection	aw select word & whitespace	iw select inner word
a", a', a(, a[, aB, at i", i', i(, i[, iB, it Version with a: include "" ' [] {} <> Version with i: exclude "" ' [] {} <>			

Copying

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

Undo/Redo

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

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 Search forward for foo, not exact	?foo Search backward for foo, not exact
n Go to next match	N Go to prev match
/\<foo\>< b=""> Search forward for foo, exact</foo\><>	:noh no search highlights

Move

:m 5 mv current line to after line 5	:m 0 mv current line to before line 1	:m \$ 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 mv lines 2,3,4 to after line 9	:.+.2m 9 mv 3 lines starting from the current one to after line 9	

vim-multiple-cursors

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

^p Remove last virtual cursor and go back to last one.

^x 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 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

Ctags & TagBar	
:tag name	Jump to (1st) match of the tag. Name can be a regular expression (e.g. <i>/Detect*</i>).
:stag name	Jump to (1st) match of the tag in a new window. Name can be a regular expression (e.g. <i>/Detect*</i>).
:ts name	List all tag matches, and lets you select one. If name is absent, use list of last :tag command.
^]	Jump to tag under the cursor
^w}	Show tag match in sep window
g]	List all matches of tag under cursor
^o	Go back one jump
:Tagbar	Toggle show/hide tagbar
:!ctags -R	Generate tags file, recursively

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: (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	
: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

Insert
checkbox

\cx

Toggle
checkbox

\cd

Delete
checkbox

\n

Set fold
level to *n*

\-

Draw
dashed line

\s

Sort sub-tree
under cursor
ascending

\S

Sort sub-tree
under cursor
descending

\t

Append time
stamp
HH:MM:SS

\d

Append date
stamp
YYYY:MM:DD

Folds

zo/zc

Open/close
1 fold under
cursor

zO/zC

Open/close all
folds under
cursor

zm/zr

Increase /reduce
fold level

zj/zk

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

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

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.

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

:set list

Show all the invisible characters

:set nolist

Stop showing all the invisible characters