

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 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 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

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/\
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 char	V select line	^v select block	gv reselect
-----+-----			
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
fa select until 1st 'a'	/foo select until 'foo'	25G select until line 25	ii select inner indentation
a", a', a(, a[, a{, at Version with a: include "" "" 0 [] {} <>			
i", i', i(, i[, i{, it Version with i: exclude "" "" 0 [] {} <>			

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

^r redo

:earlier 4m

:later 50s

:UndotreeToggle

Move

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

% jump to matching bracket	f c jump to next occur of c	F c jump to prev occur of c
--------------------------------------	---------------------------------------	---------------------------------------

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>m</u> ost <u>r</u> ecently <u>u</u> sed. 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!
: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: <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). :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	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			

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

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

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

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

Miscellaneous	
^a / ^x	Incr/decr the nr under cursor.
J	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
^oo	After startup: load last edited file
^n	In insert mode: word completion

Repeating	
n	Repeat search - forward
N	Repeat search - backward
@:	Rpt last ‘:’ command
.	Rpt last change in Normal mode
;	Rpt last <i>f c</i> - forward
,	Rpt last <i>f c</i> - backward
&	Rpt last :s/x/y/ subst

Easymotion	
\\w	Word motion, then select appropriate letter. Forward.
\\b	Idem but backward
\\s <i>char</i>	Trigger character motion, then select letter.