Navigation (top bar)

`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

Left column (vertical):

`^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`	start of line 8

Buffers

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

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

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

Inserting

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

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

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

Deleting

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

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

`d}`	`:1,.d`	`:.,$d`	`dt` c
del to end of paragr.	del from line 1 to cursor	del from curs to EOF	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/\<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 per character	select per line	select rectangular block	reselect prev selection

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

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

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

Version with a: include "" '' () [] {} <>
Version with i: exclude "" '' () [] {} <>

Copying

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

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

Undo/Redo

| `u` | undo |
| `^r` | redo |

`:earlier 4m`
`:later 50s`

`:UndotreeToggle`

Search

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

`g*`	`g#`
Idem as * but not exact	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\>`	`:noh`
Search forward for foo, exact	no search highlights

Move

`:m 5`	`:m 0`	`:m $`	`:m '}-1`
mv current line to after line 5	mv current line to before line 1	mv current line to after last line	mv current line to end of paragraph

`:2,4m $`	`:2,4m 9`	`:.,.+2m 9`
mv lines 2,3,4 to after last line	mv lines 2,3,4 to after line 9	mv 3 lines starting from the current one 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

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 **m**ost **r**ecently **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

Ctags & TagBar

`:tag` *name*	Jump to (1st) match of the tag. Name can be a regular expression (e.g. */Detect**).
`:stag` *name*	Jump to (1st) match of the tag in a new window. Name can be a regular expression (e.g. */Detect**).
`:ts` *name*	List all tag matches, and lets you select one. If name is absent, use list of last :tag command.

`^]`	**`^w}`**	**`g]`**	**`^o`**
Jump to tag under the cursor	Show tag match in sep window	List all matches of tag under cursor	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`**
ident 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	**\cx**	**\cd**	*****n***	**\‑**
Insert checkbox	Toggle checkbox	Delete checkbox	Set fold level to *n*	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	**ii**	Inner indentation level, no line above
aa	Function arg under cursor, with comma	**ai**	Indentation level + one line above (*Python*)
ia	Function arg under cursor, without comma (*inner*)	**aI**	Indentation level + line above + line below (*C++*)

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

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.
: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