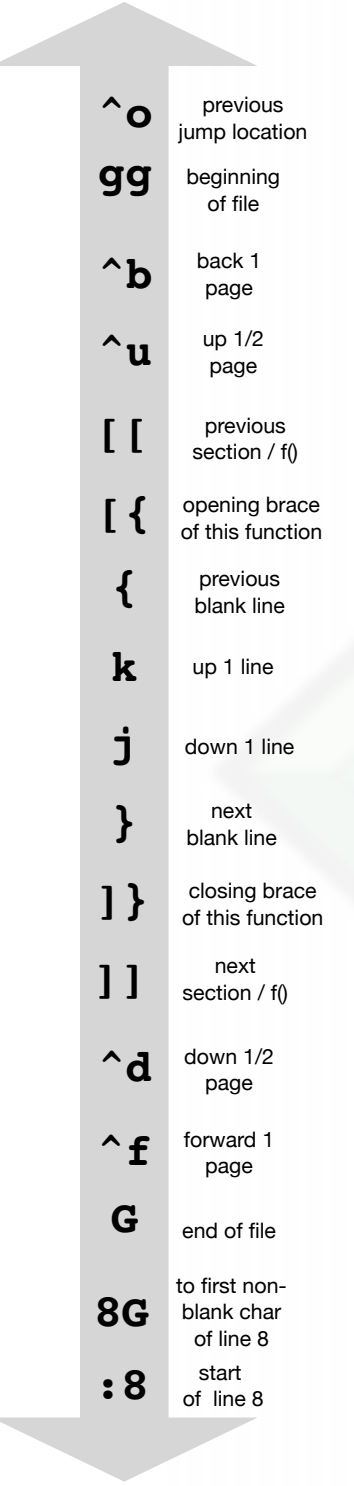


:e path load and edit given file	:e! return to unmodified file	:view path load file in read-only mode	:r file insert contents of file	:q quit	:q! quit, discard changes
:w write file	:saveas path save with new name	:browse old List recent files, allow to select one by number.	:cd change dir	:pwd echo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



% jump to matching bracket	f c jump to next occur of c	F c jump to prev occur of c
zz scroll current line to middle of the screen		

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

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 toggle end of selection	U turn to uppercase	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 i", i', i(, i[, i{, it			
Version with a: include "" " 0 [] {} <> Version with i: exclude "" " 0 [] {} <>			

Move		
:m n mv this line to after line n	:m.+2 mv this line 2 lines below	:2,4m \$ mv lines 2,3,4 to after last line
:2,4m 9 mv lines 2,3,4 to after line 9	: '<,'>m'>-5 Move selection to 5 lines before end of selection	:. ,.+2m 9 mv 3 lines from this line to after line 9

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

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
^r redo	:later 50s
:UndotreeToggle	

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

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 parag.	:1,.d del from line 1 to cursor	:. ,.d del from curs to EOF	dt c delete up to char c

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

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/^ \s\+//g In all lines, remove leading whitespace ^: start of line, \s: whitespace: \+ mult. occur.
: '<,'>s/foo/bar/g In the visually pre-selected lines, replace foo with bar.

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>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
]]		In insert mode: close environment
cse / dse		change/delete env. Cf vim-surround

Tags		
:tag	name	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	name	List all tag matches, and lets you select one. If name is absent, use list of last :tag command.
^]	g]	^o
Jump to tag under cursor	List all matches of tag under cursor	Go back one jump
:Tagbar		Toggle show/hide tagbar <ul style="list-style-type: none">Double-click or press enter on a tag to jumpPress space on a function to display arg list.
:!ctags -R --extra=+f		Create 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		Shows differences between working copy of the current file and the index, in a split window. :q to exit.
[c		Previous hunk
]c		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	i<i>i</i>	Inner indentation level, no line above
aa	Function arg under cursor, with comma	a<i>i</i>	Indentation level + one line above (<i>Python</i>)
ia	Function arg under cursor, without comma (<i>inner</i>)	a<i>I</i>	Indentation level + line above + line below (<i>C++</i>)
Example in C++ : caI : change current indentation level incl. { } above/below			

Tabularize		
:Tab/=		Align selection on the ‘=’ char
:Tab/,/r1c1l10		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.
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
:set rnu!	Toggle between relative and absolute line nrs
^oo	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	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

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

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

Repeating

n	Repeat search - forward	;	Rpt last $f c$ - forward
N	Repeat search - backward	,	Rpt last $f c$ - backward
@:	Rpt last ':' command	&	Rpt last :s/x/y/ subst
.	Rpt last change in Normal mode		

Swapping

<code>CX</code> <i>{motion}</i>	Select the first part of text to be swapped...
<code>.</code>	... select the 2nd part of text. Repeat cmd: swaps
<code>CXX</code>	Like cx for the current line
<code>CXC</code>	Clear selection and memory

(Un)commenting

Old School:

^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		

With tcomment:

gc{motion}	Toggle comment. E.g gc4j or gcaB
gcc	Toggle comment for this line.

Surround	
ys {motion}{delimiter}	Surround motion by given 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
v selection S {delim}	Surround using visual mode

Sneak	
S <i>char1 char2</i>	Jump forward to next occur. of char1 char2
S <i>char1 char2</i>	Jump backward to next occur. of char1 char2

Move block	
^h ^l	Move selection left / right
^j ^k	Move selection up / right

Insert Mode

^o 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

MacVim	
⌘= / ⌘-	Increase/decrease font size

Diff

\$ vimdiff file1 file2	Launch in diff mode
:vert diffsplit file	Vertical split, diff with current
:diffupdate	Update diff after editing
:[c /]c	Go to previous/next change
:diffget	Apply change from other to this window
:diffput	Apply change from this to other window

Multiple Cursors

^n / <esc>	Enter / exit visual-multi mode
(ctrl)↑ / (ctrl)↓	Create cursors right above / below
(shift)← / (shift)→	Add characters to your selection
n / N	Go to and select next / prev occurrence
q	Skip current selection and select next occur.
Q	Skip current selection and go to prev selec.
[/]	Navigate backward/forward through selections
Start insert mode with i / I / a / A / c / C ...	