

vi / vim keyboard shortcuts cheat sheet

ZZ	:x save and exit (:q! quit no save)
G	go to last line
gg	go to line 1 (5gg or :5 = go to line 5) 80 go to 80 th column
O	beginning of line (^ = 1 st non-blank)
\$	end of line (EOL)
dd	delete a line (5dd = delete 5 lines); yy = copy a line
dw	delete a word; de = delete to end of word
8x	delete 8 characters – vim way: {count}{command}{motion}
.	repeat last <i>change</i>
u	undo (ctrl-r = redo) or :e! to reload file
d delete	D delete to end of line; d0 = delete to beginning of line
y yank (copy)	Y yank lines {command}
p put (paste)	P put before
x delete a char	X delete a char to the left (4X = delete 4 chars to the left)
i insert	I insert 1 st non-blank o open a line + insert (O above)
a append	A append at EOL A; then j. or 78A-<Esc>
c change	C change to EOL ce change word (n. next & repeat)
r replace a char	R replace/overwrite mode
s substitute	S substitute lines J join next line
5>>	shift/indent 5 lines (<< to shift left)
w next word	W next word (blank-delimited) {motion}
b back a word	B back a word (blank-delim)
e word end	E word end (blank-delim) { } paragraph () sentence
fx find x	; next occurrence , previous occurrence (tx till x)
% jump between brackets	0 [] {} d% = delete to bracket
ma	set mark 'a' , mb set mark 'b'
:a,'bd	delete lines from mark 'a' to 'b' <i>range command</i> range: 'a','b = lines from 'a' to 'b' % = entire file; 2,9y = line 2 to 9; .,+2y = current line plus 2
'a	jump to mark 'a'
/foo	search 'foo' (\<foo\> only word 'foo') /foo\C \c case-sensitive
n	next match (N = previous match) ?foo = search backward
:%s/old/new/gci	% = entire file as range, s = substitute g = globally, c = confirm, i = ignore case, I = case-sensitive
:a,'bs/old/new/g	substitute all occurrences from mark a to mark b
"a10yy	select register [a] and yank 10 lines (" +y = system clipboard)
:w new.txt	write to <i>new.txt</i> (:w = save) only marked :a,'bw tmp.txt

:e new.txt	close and edit new file; :n next file; :p previous file
!ls -l	run shell command (!cmd = read output of <i>cmd</i>)
hjkl	left/down/up/right; +/- = down/up at 1 st non-blank
ctrl-f, ctrl-b	page forward / backward (ctrl-u, ctrl-d) HML High/Med/Low
*	search word under cursor (# backward)
dib or diB or di"	del inner block () or {} or "" or tag; dit <p>text-object</p>
gUw	uppercase word (gUU a line) gu = lowercase, g~ invert case
qa	start recording to macro 'a' q = stop recording
@a	run macro 'a' ; nnoremap <t_k4> @a = maps termcap F4 key
v	visual mode (o jump to other end; V = whole line)
U	uppercase marked text (u lowercase, ~ invert case)
ctrl-v	<i>visual block</i> mode (column mode). Insert multiple lines: shift-i enter text (change only visible 1st line), then press esc -- inserted text will apply to all marked lines
ctrl-w-s	window split horiz. (ctrl-w-v = vertical :vsplit)
ctrl-w-w	switch window =/+/<> resize; c close; q quit; o ; HJKL move
ctrl-6	toggle last buffers, see :ls (<nn> ctrl-6 same as :e #nn)
gt or gT	next/prev. tab page; :tabedit :tabclose
=ip	autoindent inner paragraph (beautify) gqip format (textwidth)
:retab	converts TABs to spaces; :set noet then :retab!
:set or :opt	current settings; :reg :marks :map :vert term netrw :Ex [/ <i>etc/</i>]
:set ve=all	virtual edit (:h 've') useful to put rectangle block at EOL
:set ft=python	set filetype and syntax (<i>messages, logtalk, dosbatch...</i>)
:<,>normal .	run normal dot command for all marked lines
:a,'b!sort -k2.1	sort from mark 'a' to 'b' (! as <i>filter</i>) using 2 nd field, pos 1
:g!/xyz/d	globally (all lines that) not (!) containing xyz -- delete
:h :g	help 'global' command (:h i_ctrl-r)
ctrl-]	jump to topic under cursor. ctrl-o go back old (ctrl-i in/out)
:ctrl-r	" copied text; % filename (command/insert mode)
ctrl-n, ctrl-p	autocompletion; ctrl-v<tab> real tab; ctrl-w del word; ctrl-y
ctrl-x, ctrl-o	autocompletion (omnifunc)

~/vimrc

```
set number " se nu! (on/off) se nu? (current) se nu& (default)
set nowrap autoindent incsearch ignorecase ruler ttyfast visualbell
set backspace=indent,eol,start
set laststatus=2 mouse=a bg=dark t_Co=256 colorcolumn=80
hi StatusLine ctermfg=7 ctermf=4
hi LineNr ctermfg=DarkGrey
set tabstop=8 expandtab shiftwidth=4 softtabstop=4 smarttab
set listchars=eol:~,tab:|,trail:~,extends:~,precedes:<
set clipboard=unnamedplus fileformat=dos fileencoding=utf-8
let mapleader=" " " or <Space> instead of \
nnoremap <F3> :set list! hlsearch!<CR>
```