

2017/11/28 VIM Cheat-Sheet

Motions

[count] {motion}

h j k l move cursor
gj gk move cursor one screen line (good of wrapped lines)
+ beginning of next/previous line
<enter> beginning of next line
H M L top middle bottom of screen
w W beginning of next word, <shift> -->, <ctrl> -->
e E end of next word
b B backwards to start of last word, <shift> <-->, <ctrl> <-->

f F jump to next/previous occurrence of {char}
t T jump to before next/previous occurrence of {char}
; next f/F/t/T
, pervious f/F/t/T

0 move to beginning of the line
^ move to first non-blank character of the line
g^ move to first non-blank character of the line

gm move to middle of screen

\$ move to end of line
g\$ move to end of line on screen
g_ move to end last non-blank char of [count] line

| move to column [count], default is zero

gg go to the first line of the document
G go to line number <n>, default last line
go goto bytes
% goto percent

{ } jump backward/forward one paragraph
() jump backward/forward one sentence

]] beginning of next function
[[beginning of last function
][end of this function -->]
[] end of last function [<--

Editing with Motions

{count}{command}{motion}

{register}{comand}{count}{motion}

{visual mode}{motion(s)}{command}

c cc change

d dd delete

y yy yank

> >> increase indent

< >> decrease indent

! !! filter text lines through external program: !!fmt -72

= == filter text line through C-indenting program

gu gugu lowercase, also guu for whole line

gU gUgU uppercase, also gUU for whole line

gq gqgq format, also gqq for whole line

~ toggle case, characterwise only

p P paste

gp gP paste move cursor after

x delete character

X backspace delete character

. repeat command

& repeat last substitution

i I insert

o O open

a A append

s S substitute char/line

r R replace char/line

J join lines

v V visual select char/line/block

<ctrl-v> visual select char/line/block

gv reselect last visual

gq format/word wrap selected text

"x register x = a-z 0-9 A-Z " *

" unnamed register (last delete)

0 last yank

1 last multi-line delete

2-9 last 8 deletes

a-z general purpose registers text or macros

A-Z append to a-z registers

* operating system clipboard

Editing with Objects

{number}{cmd}{text object}

Objects:

aw iw word

aW iW word

as is sentence

ap ip paragraph

a" i" quote

a' i' quote

a` i` quote

a[i[block or a] i]

a(i(block or a) i) ab ib

a{ i{ block or a} i} aB iB

a< i< block or a> i>

at it tag

Commands: v c d y > < ! = gu gU gq

Search and Replace

SEARCH [count]/{pattern}/{offset}

/pattern	search for pattern
?pattern	search backward for pattern
n	repeat search in same direction
N	repeat search in opposite direction
:noh	remove highlighting of search matches
*	search next word under cursor
#	search previous word under cursor
&	repeat last substitution
%	find matching bracket {} [] ()

SUBSTITUTE {range}s/old/new/{flags}
:{range}s/old/new/{flags}

flags: g-global c-confirm i-case insensitive

range:

n,m	lines n to m, (\$ for last line)(. for current line)
%	whole buffer
'<,'>	visual marked lines

:%s/old/new/g	replace all old with new throughout file
:%s//new/g	replace last search item with new
::,+12s/old/new/g	replace all old with new on next 12 lines
:1,.s/old/new/gc	from line 1 to current line with confirmations

GLOBAL – execute Ex command for each line matching pattern

:{range}g/pattern/cmd

:g/pattern/d	Delete all lines that match pattern
:g/pattern/t\$	Copy all lines matching a pattern to end of file
:g/pattern/m\$	Move all lines matching a pattern to end of file
:g/pattern/m0	Move all lines matching a pattern to end of file
:g/pattern/normal @q	Run macro on matching lines
:g!/pattern/d	Delete all lines that do not match pattern
:v/pattern/d	Ditto

Screen Scrolling

```
z<enter>  position line on top of the screen
zt        position line on top of the screen

zz        position line on centre of the screen
z.        position line of centre of the screen

zb        poition line on bottom of the screen

<ctrl-f>  forward one full screen      <ctrl-d>  forward (down) half screen
<ctrl-b>  back one full screen         <ctrl-u>  back (up) half screen
<ctrl-e>  forward one line            <ctrl-y>  back one line
```

Marks

```
ma        set mark a at current cursor location
'a        jump to mark, first non-blank character in line
`a        jump to mark, position line and column

''        jump to last jump
'.        jump to last change
'^        jump to last insert
g;        jump to last edit, step back thru edit history
g,        jump to next edit, step forward thru the edit history

'<        jump to start of last visual block
'>        jump to end of last visual block
```

Macros

```
:{range} normal @a run macro a on range of lines... :% norm @f
```

```
qa  record macro a (use registers a-z)
q   stop recording macro
@a  run macro a
@@  rerun last run macro
:g/pattern/normal @a    run macro on matching lines
```

Windows and Buffers and Tabs

```
:e {file}      edit a file in a new buffer
:bnnext or :bn go to the next buffer
:bpnext or :bp go to the previous buffer
:bd           delete a buffer (close a file)
:ls          list all open buffers
<ctrl>6      switch to alternate file

:sp file      open a file in a new buffer and split window
:vsp file     open a file in a new buffer and vertically split window

<ctrl-w>s     split window horz
<ctrl-w>v     split window vertically
<ctrl-w>w     switch windows
<ctrl-w>c     quit a window

<ctrl-w>h <ctrl-w>j <ctrl-w>k <ctrl-w>l  move to window left/down/up/right

:tabe[dit] {file} edit file in new tab
:tabc[lose]      close tab
:tabn[ext]       next tab
:tabp[revious]   previous tab
:tabfirst        first tab
:tabl[ast]       last tab
```

File Mangement

```
:e {file}  edit a file in a new buffer
:q         quit
:q!        quit without writing (changes discarded)
:w         write file and keep working
:wq        write the current file and quit

:x         write current file, if modified, and quit
ZZ         same as :x

:qa        quit all
:qa!       quit all without writing (changes discarded)
:wa        write all changed files (buffers), and keep working
:xa        exit all (save all changes and quit)

:e .       explore files/directories
:Explore   explore file/directories
:Vex       explore with vertical split
:browse oldfiles recent opened files
:r[ead] {file} insert the file below the cursor
```

Other Stuff

u undo
<ctrl-r> redo
<ctrl-l> redraw screen

<ctrl-n> autocomplete next matching word
<ctrl-p> autocomplete previous matching word

<ctrl-a> increment number under cursor
<ctrl-x> decrement number under cursor

gf go to file under cursor

:map show key mappings
:reg show registers
:let @a="" clear register a
:retab replace tab/whitespace to fit tabstop setting
:set number turn on line numbers
:set nonumber turn off line numbers
:set relativenumber turn on relative numbers
:set norelativenumber turn off relative numbers
:set list display <tab> <space> and <EOL> chars
:l[ist] show line with control chars
:set textwidth=80 set max text width.... ggVG gq (select all format)

:set spell turn on spelling highlighting
:set nospell turn off spelling
]s next spelling error
[s previous spelling error
z= spelling suggestions

{range}g[lobal]/{pattern}/yank {register}

:%g/pattern/yank A append all lines with pattern to register A
:%v/pattern/yank A append all line without pattern to register A

SHELL COMMANDS

:! cmd execute cmd in the shell
:w !cmd write the current buffer to the stdin of an external command
:{range}!cmd write {range} to stdin of command and replaces the buffer with output
:r !{cmd} execute comand and insert(read) output below the cursor
!!{cmd} filter line(s) thru command... !!gawk '{print \$1}'