

Contents

1	_____general_____	1.1
1.1	key modifier	1.1
1.2	commands	1.1
1.3	help	1.1
2	_____motion_____	1.1
2.1	motions and operators	1.1
2.2	left-right motions	1.1
2.3	up-down motions	1.1
2.4	word motion	1.1
2.5	text object motion	1.2
2.6	text object selection	1.2
2.7	marks	1.2
2.8	jumps	1.2
3	_____editing_____	1.2
3.1	:: general ::	1.2
3.2	:: basic: delete ::	1.2
3.3	:: autopairs ::	1.2
3.4	:: surround.vim ::	1.2
3.4.1	:: targets ::	1.2
3.4.2	:: commands ::	1.3
3.5	:: multiple cursors ::	1.3
4	_____movement_____	1.3
4.1	:: general ::	1.3
4.2	:: visual split ::	1.3
4.3	:: easymotion (emotion) ::	1.3
5	_____split screen_____	1.3
5.1	:: general ::	1.3
5.2	:: moving cursor ::	1.3
5.3	:: moving windows ::	1.3
5.4	:: resizing ::	1.3
6	_____searching_____	1.3
6.1	:: plugins ::	1.3
6.2	:: general ::	1.3
6.3	:: incsearch ::	1.4
6.4	:: asterisk ::	1.4
7	_____NERDTree_____	1.4
7.1	:: NERDTree - general ::	1.4
7.2	:: NERDTree - file node ::	1.4
7.3	:: NERDTree - dir node ::	1.4
7.4	:: NERDTree - bookmarks ::	1.4
7.5	:: NERDTree - tree nav ::	1.4
7.6	:: NERDTree - filesystem ::	1.4
7.7	:: NERDTree - filtering ::	1.4
8	_____FZF_____	1.4
8.1	:: fzf - git ::	1.4
8.2	:: fzf - search ::	1.4
9	_____IDE_____	1.4
9.1	:: neomake ::	1.4
10	_____vcs_____	1.4
10.1	:: signify ::	1.4

1 _____general_____

1.1 key modifier

M- ALT key
M- CTRL key

1.2 commands

:viusage show all key maps
:PlugClean uninstall plugs
:PlugInstall install plugs
:PlugUpdate update plugs
,0..9 goto buf #
,<Space> distraction free view
:map show mapped keys
:map! show lmapped keys
:h key-notation help for ...
:reg displays reg content

1.3 help

:h show help
<C-J follow link
<C-T back to prev topic

2 _____motion_____

2.1 motions and operators

c change
d delete
y yank into register
~ swap case
g~~ swap case (entire ll.)
gu guU make lower-/uppercase
> < shift right/left

2.2 left-right motions

h l [cnt] left, right
0 \$ begin/end of ll.
~ 1st non-blank of ll.
g_ last non-blank of ll.
g0 gm g\$ beg,mid,end of scrn ll.
l to scrn. col [cnt]
F(.) f(.) to [cnt]'th (char) (←/→)
T(.) t(.) till [cnt]'th (char) (←/→)
, , [cnt] rpt last F,f,T,t (←/→)

2.3 up-down motions

j k [cnt] down, up
gj gk [cnt] disp ll. down, up
- + [cnt] ll. up/down 1st chr
<C-j> <C-k> 20 ll. down/up
<C-h> <C-l> prev/next empty ll.
gg G goto ll. [cnt] def. 1st/last
% goto [count] perc of file

2.4 word motion

w W word/WORD (→)
e E end of word/WORD (→)
b B word/WORD (←)
ge gE end of word/WORD (←)

A word consists of a sequence of letters, digits and underscores, or a sequence of other non-blank characters, separated with white space (spaces, tabs, <EOL>).

A WORD consists of a sequence of non-blank characters, separated with white

space. An empty line is also considered to be a WORD.

2.5 text object motion

()	[count]	sentences (←/→)
{ }	[count]	paragraphs (←/→)
[] []	[count]	to the next '{'/'}'
[[]]	[count]	to the prev '{'/'}'

A sentence is defined as ending at a '.', '!' or '?' followed by either the end of a line, or by a space or tab.
A paragraph begins after each empty line.
A section begins after a form-feed (<C-L>) in the first column.

2.6 text object selection

This cmds can only be used while in Visual mode or after an operator. The "a" cmds select an object including white space, the "i" cmds select an "inner" object without white space.
aw aW a word/WORD
iW iW inner word/WORD
as is a/inner sentence
ap ip a/inner paragraph
a] i] a/inner [] block
a) ab a () block
i) ib inner () block
a> i> a/inner <> block
at it a/inner <a>... block
a] aB a block
i] iB inner block
a" a' a` a quoted string
i" i' i` inner quoted string

Examples:

d1 del char
daw diW del a/inner word
dgn dgW del the next/prev match
dd del [cnt] line
das del a sentence
dib del inner () block

2.7 marks

Jumping:
' (backtick) specified location
' (single quote) ll. (1st non-blank)

Marks:

{a-z} valid within one file
{A-Z} valid between files
[' [prev chged/yanked (1st char)
] '] prev chged/yanked (last char)
' < ' 1st/last ll. visual sel
' .. ' before latest jmp
' .. ' where Insert mode exited
' .. ' where last chg were made
['] next ll. w/ lowercase mark
['] prev ll. w/ lowercase mark

m{a-zA-Z} set mark a-zA-Z at cursor
m' m' set ctxt mark; jmp ' ' ..
m[m] set ' ['] mark
m< m> set ' < ' > mark; chg gv
'{a-z} jump (current buf)
'{A-Z} jump (chg file)
:marks list marks
:delm {_} del marks
:delm! del current buf marks

2.8 jumps

<C-o> <C-i> jump back/forward
:ju[mps] print jump list
:cle[arjumps] clear jump list

3 _____editing_____

3.1 :: general ::

c d y change, delete, yank
s replace
v visual select
x delect char
go newline below (no imode)
g0 newline above (no imode)

3.2 :: basic: delete ::

d1 delete character (alias: "x")
diw delete inner word
daw delete a word
diW delete inner WORD (see WORD)
daW delete a WORD (see WORD)
dgn delete the next search pattern match
dd delete one line
dis delete inner sentence
das delete a sentence
dib delete inner '(' ')' block
dab delete a '(' ')' block
dip delete inner paragraph
dap delete a paragraph
diB delete inner "" "" block
daB delete a "" "" block

3.3 :: autopairs ::

<C-v> insert pair w/o plugin
<A-p> disable/enable plugin
<A-e> {}'hello' → {'hello'}

3.4 :: surround.vim ::

3.4.1 :: targets ::

w W s word, WORD, sentence

3.4.2 :: commands ::

ds delete surroundings
ds" "hell*o" → hello
ds) (1+5*6)/2 → 1+56/2
dst <p>*h</p> → h
cs change surroundings
cs"" "H *w!" → 'H w!'
cs"<q> "H *w!" → <q>H w!</q>
cs]) (1+5*6)/2 → [1+56]/2
cs)[(1+5*6)/2 → [1+56]/2
cst<p> <a>.* → <p>.</p>
ys you surroundings
ysiW Hee *wor! → 'Hee (wor)!'
ysiW Hee *wor! → 'Hee (wor)!'<C-j>
ysW add: if *x>3 (→ if (x>3) (<C-k>
cs]) chg: [123+5*6]/2 → (123+56)624
vlllls' visual: my str = *wheel; -> my str = *wheel;

3.5 :: multiple cursors ::

<C-n> start/next key
g<C-n> start/next key (w/o bounds)
<C-n> next key
<C-p> prev key
<C-x> skip key
<A-n> select all keys
g<A-n> select all keys (w/o bounds)

4 _____movement_____

4.1 :: general ::

w e b move start/end word
W E B move start/end WORD
h j k l left, down, up, right
h j k l left, down, up, right
<C-h> <C-l> prev/next empty line
<C-j> <C-k> 20 lines down/up
<C-o> <C-i> jump back/forward
0 \$ begin/end of line
gg G begin/end of file
% match block

4.2 :: visual split ::

Splitting on visual selection.
<C-w>gr resize split to selection
<C-w>gss split out vsel
<C-w>gsa split out vsel above
<C-w>gsb split out vsel below

4.3 :: easymotion (emotion) ::

sd 1 char
sf 1 char (overwin)
sh ,h 1 char (line back)
sl ,l 1 char (line fore)
sj ,j 1 char (down)
sk ,k 1 char (up)
ss 2 chars
s/ x chars
sn goto next match

5 _____split screen_____

5.1 :: general ::

<C-w>v new vertical split
<C-w>s new horizontal split
<C-w>c close split
<C-w>o close other splits
<C-w>H close split

5.2 :: moving cursor ::

<A-h> <A-l> navigate left/right
<A-j> <A-k> navigate up/down

5.3 :: moving windows ::

<C-w>r rotate windows
<C-w>x swap windows
<C-w>K move top; full width
<C-w>J move bottom; full width
<C-w>H move left; full height
<C-w>L move right; full height
<C-w>T move to new tab

5.4 :: resizing ::

<C-w>= make equally h/w
<C-w>+ inc height
<C-w>- dec height
<C-PageUp> inc height
<C-PageDown> dec height
<C-w>> inc width
<C-w>< dec width
:res[ize] N set height to N
:res[ize] +N grow height by N
:res[ize] -N shrink height by N
:vertical res N set width to N
:vertical res +N grow width by N
:vertical res -N shrink width by N

6 _____searching_____

anzug

6.1 :: plugins ::

anzu cmd line status
asterisk improved * (visual mode)
incsearch search itself

6.2 :: general ::

<Esc><Esc> clear search
,* highlight word
<C-r> replace vselected word
ggn jump to first search match
GN jump to last search match

6.3 :: incsearch ::

/ search forward
? search backward
g/ search (stay)
n N next/prev match
<S-z>/ fuzzy search forward
<S-z>? fuzzy search backward
<S-z>g/ fuzzy search (stay)
z/ emotion search (fore)
z? emotion search (back)
zg/ emotion search (stay)
<Tab> move to next match
<S-Tab> move to prev match
<C-j> move to next page match
<C-k> move to prev page match

6.4 :: asterisk ::

match word (forward; stay)
match word (backward)
g* search word (forward; stay)
g# search word (backward)

7 _____NERDTree_____

7.1 :: NERDTree - general ::

,<Tab> toggle NERDtree
,<S-Tab> find file in NERDTree
q close window
A min-max window
? toggle help

7.2 :: NERDTree - file node ::

o open
go open (preview)
t open in new tab
T open in new tab silently
i open in split
gi open in split (preview)
s open in vsplit
gs open in vsplit (preview)

7.3 :: NERDTree - dir node ::

o open+close node
O recursively open node
t open in new tab
T open in new tab silently
x close parent of node
X close all child nodes
s open in vsplit
gs open in vsplit (preview)

7.4 :: NERDTree - bookmarks ::

o open bookmark
go preview file
go find file file in tree
t open file in new tab
T open file in new tab silently
D delete bookmark

:Bookmark <...> create
:ClearBookmarks <...> delete
:EditBookmark edit

7.5 :: NERDTree - tree nav ::

P go to root
p go to parent
K go to first child
J go to last child
<C-j> go to next sibling
<C-k> go to prev sibling

7.6 :: NERDTree - filesystem ::

C change tree root to selected dir
u move root up a dir
U move root up a dir (leave root)
r refresh cursor dir
R refresh root
m show menu
cd change the CWD
CD change tree root to CWD

7.7 :: NERDTree - filtering ::

I hidden files
F show/hide files
B show/hide bookmarks

8 _____FZF_____

8.1 :: fzf - git ::

,fgb blame
,fgc commits
,fgd diff
,fgf files
,fgs status
,fgl log

8.2 :: fzf - search ::

,fa ag
,fb buffers
,fû color schemes
,fc command history
,ff files
,fl lines
,fm maps
,fr marks
,fs snippets
,fo old files+buffers
,f/ search history

9 _____IDE_____

9.1 :: neomake ::

<F3> prev err/warn
<F4> next err/warn

10 _____vcs_____

10.1 :: signify ::

,n next hunk
,<S-n> prev hunk