VIM Guide

Meher Krishna Patel

Created on : Octorber, 2017

Last updated : October, 2018

Table of contents

Chapter 1

Vim Commands

- First install Vim.
- To install the packages, first install 'vundle' using following command. Use 'git-shell' to run this command in Windows.

```
git clone https://github.com/VundleVim/Vundle.vim.git ~/.vim/bundle/Vundle.vim

(in Windows : use 'git-shell' to run below command)
git clone https://github.com/VundleVim/Vundle.vim.git C:/Users/<Username>/.vim/bundle/
↪Vundle.vim
```

- Copy and paste the .vimrc file in the home directory.
- In Windows, paste in the .vimrc file at C:/Users/<Username>/; and change the line "set rtp+= . . . " in .vimrc with correct location of "Vundle.vim" i.e. "set rtp+=C:/Users/<Username>/.vim/bundle/Vundle.vim".
- Use double quote to comment/uncomment the packages in .vimrc file e.g. " Plugin 'mattn/emmet-vim' will not install the package "emmet-vim". Or add some more packages in .vimrc file as required.
- After selecting the packages, open vim and run the command -> :PluginInstall and it will install the all the plugins.
- Use :help <plugin name> to see the list of operations.

1.1 Starting Vim

Commands	Descriptions
:w	save file
:w <filename>	save file as <filename> and keep the current file open
:sav <filename>	save file as <filename> and open <filename>
:q	quit (if already saved)
:q!	quit without saving
:e <filename>	open new/existing <filename> in new buffer
:wq	save and quit
:bn	go to next buffer i.e. file
:b <filename>	go to buffer with <filename>
:bd	close current file without exiting vim
:bd!	close current file without exiting vim and 'no modification'
:vim *	open all files in the directory (all in same buffer)
:vim file1 file2 file3	open file1, file2 and file3 in vim
:n	go to next file
:n <filename>	go to file name
:prev	go to previous file
ctrl-Z	suspend vim
fg	bring forground vim

1.2 Undo/Redo

Commands	Descriptions
u	undo
crtl-r	redo

1.3 Insert

Commands	Descriptions
i	insert mode at cursor
I	insert mode at beginning of the line (i.e. first character of line)
s	delete character under the cursor and enter into insert mode
S	delete the line and go to insert mode from the beginning of same line
a	insert mode after the cursor
A	insert mode at the end of line
o	insert mode on below line
O	insert mode at bottom line
C	delete from cursor to end of line and go to insert mode
r	replace current character
R	replace characters until Esc is pressed (i.e. same as insert button in keyboard)

1.4 Copy/Paste/Delete

Commands	Descriptions
y	yank (copy)
yiw	copy the word
yw	copy the word after the cursor
yy or Y	copy the line
y$	copy till end of the line from current location of cursor
“+y	copy to clipboard e.g. “+yiw will copy the word in clipboard
<F3>y	same as above (use <F3> for clipboard, remapped in .vimrc)
<F3>p	paste from clipboard (see above line as well)
p	paste after cursor (remapped as]p in .vimrc)
]p	paste with indentation
P	paste before cursor
ctrl-P	paste from clipboard (remapped in .vimrc)
shift insert	paste from clipboard (remapped in .vimrc)
d<command>	delete<command>
diw	delete word (and stay at normal mode)
ciw	delete word (and go to insert mode)
dw or cw	delete word after the cursor
dd or cc	delete line
D or C	delete till end of line
x	delete character (delete)
X	delete character (backspace)
.	repeat previous operation

1.5 Paste in search or colon commands

- Following commands can be used at command mode during search or color commands e.g. :w ctrl r ctrl w etc.

Commands	Descriptions
ctrl r “	paste the last copied data
ctrl r ctrl w	paste the word under cursor
ctrl r %	print the naem of current file
shift insert	paste the data from clipboard

1.6 Search

Commands	Descriptions
/	forward then use n/N for next/previous match
?	backward
* (asterisk)	word under cursor forward (exact match)
g*	word under the cursor (partial match)
#	word under cursor backward (exact match)
g#	word under cursor backward (partial match)
/\\<word1\\>	search for exact match for “word1”
:set ignorecase	use this option for avoiding case-matches
:set noignorecase	use this option for case-matches

1.7 Replace

- Use ‘c’, ‘g’, ‘gc’ and other combination to perform the desired replacement.

Commands	Descriptions
:s /word1/word2	substitute word1 with word2 in current line (only first occurrence)
:s /word1/word2/c	substitute word1 with word2 in current line after confirmation (only first occurrence)
:s /word1/word2/g	substitute word1 with word2 in current line (all occurrence)
:s /word1/word2/gc	substitute word1 with word2 in current line ater confirmation (all occurrence)
:1,4 s /word1/word2	substitute word1 with word2 in lines 1 to 4 (only first occurrence in each line)
:%s /word1/word2/g	replace all occurrences
:%s /word1/word2/gc	replace all occurrence after confirmation
:%s /ctrl-r ctrl-w/word2/gc	replace all occurance of the word under cursor after confirmation (exact match)
:s /\\<word1\\>/word2	substitute exactly matched “word1” with word2 in current line

1.8 Indentation

Commands	Descriptions
>>	Right indent the current line
5>>	Right indent 5 lines
<<	De-indent line
5==	Re-indent 5 lines
>%	Increase indent of a braced or bracketed block (place cursor on brace first)
=%	Reindent a braced or bracketed block (cursor on brace)
<%	Decrease indent of a braced or bracketed block (cursor on brace)
]p	Paste text, aligning indentation with surroundings
=i{	Re-indent the ‘inner block’, i.e. the contents of the block
=a{	Re-indent ‘a block’, i.e. block and containing braces
>i{	Increase inner block indent
<i{	Decrease inner block indent

Commands	Descriptions
:retab	convert existing tabs to spaces

1.9 Mouse settings

Commands	Descriptions
:behave mswin	mouse functionality will work as windows
:behave xterm	mouse functionality will word as x-windows

1.10 Command/Visual Mode

Commands	Descriptions
Esc or crtl-[	command mode
v	visual mode
crtl-v	visual block mode

1.11 Cursor movement

Note: To run the ‘ctrl-]’ and ‘ctrl-T’ command, we need to create the tags first. In the below command, the ‘tags’ will be created for all the python files in the directory, which will be stored in a file ‘tags’,

```
(run the following command in shell)
ctags -R *.py
```

Commands	Descriptions
h	left
j	down
k	up
l	down

Continued on next page

Table 1.1 – continued from previous page

Commands	Descriptions
w	forward to the beginning of next word
3w	forward 3 words
W	forward to the beginning of next word (only spaces are the end of word)
b	backward to the beginning of next word
B	backward to the beginning of next word (only spaces are the end of word)
e	forward to end of next word
E	forward to end of next word (only spaces are the end of the word)
gg	go to first line of page
G	go to end line of page
10G	go to 10th line
10gg	go to 10th line
0	go to first character of line
$	go to end character of line
ˆ	go to first non-black character of line
M	go to middle of the screen
fa	go to next a in current line
Fa	go to previous a in current line
ta	go to end of next a in current line
Ta	to to end of previous a in current line
ctrl-o	go to previous location e.g. we went to line 10 from line 18, ctrl-o will go to line 18 again
ctrl-i or Tab	go to next location i.e. go to line 10 again from line 18.
gd	go to the local declaration of the word under cursor
gD	go to the global declaration of the word under cursor
g*	search for the word under the cursor
g#	same as g* but in backward direction.
gf	go to the filename under the cursor, use ctrl-o to go back
ctrl-]	go to tag definition (a tag can be a function or variable name etc.); use ctrl-o to go back
ctrl-T	go back to previous loation from where ctrl-] was exectued

1.12 Screen movements

Commands	Descriptions
ctrl-d	move half screen down
ctrl-f	page down
ctrl-e	move one line down
ctrl-u	move half screen up
ctrl-b	page up
ctrl-y	move one line up
z<Enter>	move current line to the top of screen (with cursor at the beginning)
zt	move current line to the top with without changing the location of cursor
26.	move current line to the center of screen (with cursor at the beginning)
zz	move current line to the center of screen (without moving cursor)
z-	move current line to the center of screen (with cursor at the beginning)
zb	move current line to the center of screen (without moving cursor)

1.13 Unix Shell

Commands	Descriptions
:shell	go to unix shell
exit	type exit in unix shell to come back in Vim

1.14 Registers

Commands	Descriptions
“ayy	copy line in register ‘a’
“ap	paste content of register ‘a’
Capital letters append the new value to previously stored values	
“Ayy	copy line and append to previous value in register “a” Then use “a to paste the value in register ‘a’
“=3*2<Enter>p	paste the result i.e. 6 at the current cursor location
:registers	display the values in all the registers
:registers abc	display the values of registers a, b and c

1.15 Multiple Files

Commands	Descriptions
:arg *grep -l ‘import’ *.py*	open all files in current folder which contains word ‘import’

1.16 Mark

Commands	Descriptions
ma	mark the line with name ‘a’ (use a-z or 0-9)
	Next go to some other line and excute following command
d’a	delete till line which is marked as ‘a’
:marks	show the list of marks
‘a	go to mark a

1.17 Sorting

Commands	Descriptions
!10G (press enter) sort (press enter)	it will sort first ten lines according to name
!G (press enter) sort (press enter)	it will sort all the lines according to names
!!ls	go to terminal, run ls command and print the output on the file (i.e. print list of file in current directory)
!!dates	go to terminal, run date command and print the output on the file

1.18 Printing the code

Commands	Descriptions
:hardcopy <filename.pdf>	open the printing-instructions-window
:TOhtml and then save the file	Save in html format, use :colorscheme default for white background

1.19 Mapping

- Please read the comments in .vimrc file for more commands and details
- comment/uncomment the command using double quote as per requirement.

Commands	Descriptions
:map	display the key-mappings,

1.19.1 Copy/paste from clip board

Use <F3> and then normal copy/paste command from clipboard.

Copy/paste to clip board	
<F3> is remapped	nnoremap <F3> "+
<ctrl-P> is remapped	nnoremap <C-P> "+]p (paste with indentation)
<F3>yiw or <F3>yy etc.	copy the word or line etc. in clipboard
<F3>p	paste the data from clipboard

1.19.2 Disable arraow keys

Below code can be used in .vimrc file

"Key mappings : disable arrow keys
no <left> <Nop>
no <down> <Nop>
no <up> <Nop>
no <right> <Nop>
ino <down> <Nop>
ino <left> <Nop>
ino <right> <Nop>
ino <up> <Nop>

1.19.3 Code execution

"python commands
" Execute : F9 (Below code is used in .vimrc file)
:autocmd FileType python :nmap <F9> :! clear <CR> :! python % <Enter>

"C/C++ commands
"Compile : F9 (Below code is used in .vimrc file)
:autocmd FileType c,cpp :nmap <F9> :! rm -r out <CR> :! clear <CR> :! g++ % -o out <Enter>
"Run : Ctrl+F9 (Below code is used in .vimrc file)
:autocmd FileType c,cpp :nmap <C-F9> :! clear <CR> :! ./out <CR>

1.20 Buffer

Commands	Descriptions
:bn	go to next buffer i.e. file
:b <filename>	go to buffer with <filename>
:bd	close current file without exiting vim
:bd!	close current file without exiting vim and 'no modification'

1.21 Split windows

Commands	Descriptions
:split	split window in two part and display current file in both window
:split <filename>	open <filename> in split window
:5 split <filename>	open <filename> in new split window with width of 5 line
:new	split window in two part with second window as blank
crtl-w j	go to below split window
crtl-w k	go to above split window
crtl-ww, crtl-w w	go to next split window
crtl-w +	increase the width of split window by one line
5 crtl-w -	decrease the width of split window by 5 line
crtl-w =	make all split window of equal size
crtl-w _	maximize the current split window

1.22 Auto completion

Commands	Descriptions
ctrl-p,ctrl-n	auto complete by looking previous/next words (use ctrl-p or ctrl-n to change the words from list)

1.23 Text files

Commands	Descriptions
:set textwidth=50	change the line after 50 character
:1,5 center 50	textwidth = 50 and center the lines 1 to 5
:1,5 right 50	textwidth = 50 and right justify the text on lines 1 to 5
:1,5 left 4	left margin = 4 for lines 1 to 5
Use $ for end of the line as shown below,	
:1,$ center 50	textwidth=50 and cneter all the line
or use % sign for the file (results is same as above,	
:% center 50	
:set wrap	turn the wrap words on
:set nowrap	turn off the wrap words

1.24 Macros

Commands	Descriptions
qa	start recording and store in reg 'a'. Then perform certain operations. press 'q' again to stop recording.
@a	execute macro
3@a	repeat macro 3 times

1.25 More commands

Commands	Descriptions
ctrl + g	name of current file
ctrl + u	move half screen up
ctrl + d	move half screen down
J	join line below with current line
3J	join below two line with this line (not 3)
z=	spell suggestion
~	change case of letter
:digraphs	to see the list of symbols e.g. copyright etc.

1.26 Block Visual Mode

Add same items in the beginning/end/middle of all the lines

Commands	Descriptions
crtl-v	select the block with cursor movement
press I (insert before) or A (insert after) or c (replace)	type the text -> press Esc -> block will be replaces by text

1.27 Zoom Screen

Commands	Descriptions
Zoom in	crtl-+
Zoom out	ctrl–

1.28 Save session

Commands	Descriptions
:mksession name.vim	save session
:mksession! name.vim	override session
:source name.vim	load session

1.29 Folding

Use space (remapped in .vimrc) at the line below the function definition for folding/unfolding the code.

1.30 Plugins

First install ‘vundle’ using following command. Use ‘git-shell’ to run this command in Windows.

- git clone https://github.com/VundleVim/Vundle.vim.git ~/.vim/bundle/Vundle.vim
- After that copy and paste the .vimrc file in the home directory (in Windows paste in the director C:/Users/<Username>/)
- Use double quote to comment/uncomment” the packages in .vimrc file e.g. ” Plugin ‘mattn/emmet-vim’ will not install the package “emmet-vim”. Or add some more packages in .vimrc file as required.
- After selecting the packages, open vim and run the command -> :PluginInstall and it will install the all the plugins.
- Use :help <plugin name> to see the list of operations.

1.30.1 NERDTree

Commands	Descriptions
NERDTree	turn on the directory structure
NERDTree!	disable the directory structure
m	then use various operations for delete, and create files.
e.g	
m-a meher	create a file with name meher
m-a meher/	create a folder with name meher

1.30.2 Surround

Commands	Descriptions	Results
ysiw”	add “” to the word	Meher -> “Meher”
ds”	delete surround quotes	“Meher” -> Meher
cs[(or cs])	change surround [] by ()	[2+3]/2 -> (2+3)/2
cst<h1>	change <p> tag to <h1>	<p>Meher Krishna</p> -> <h1>Meher Krishna</h1>

1.30.3 NerdCommenter

- By default <leader> is backslash key (\).
- In the current .vimrc file, the <leader> is remapped with ‘g’.

Commands	Descriptions
<leader>cc	comment line
3<leader>cc	comment 3 lines
3<leader>cu	uncomment 3 lines
3<leader>c<space>	toggle comment in 3 lines (if first line is comment, then uncomment below lines as well)
3<leader>ci	toggle comment in 3 lines individually
<leader>c$	comment from cursor to end of line
<leader>cA	comment from cursor to end of line (and go to insert mode)

1.30.4 vim-table-mode

Follow the below steps to create the 'rst-table' in Vim,

- Enable/Disable table mode using ':TableModeEnable/TableModeDisable' or using ':TableModeToggle',
- Type column-names e.g. |Col1 | Col2| and press enter
- Then press | twice to create the table. Do the same to add more lines in the table.
- Also, go above the |Col1 | Col2| and press | twice (required for correct rst-table-format)

1.30.5 vim-extline

This is used to underline the text, which is required for making Headings in .rst document.

- Type the symbols (in insert mode) e.g. = and then press ctrl-l ctrl-l to complete it.
- Or press ctrl-l and then press the symbol,
- Or press ctrl-l and then press a number e.g. 1, 2 etc. This is autocomplete the underline base on heading levels.
- Some more commands are listed below,

Commands	Descriptions
ctrl-l ctrl-l	Auto-line update
ctrl-l ctrl-h	Horizontal line update
ctrl-l ctrl-u	Change to underlined title
ctrl-l ctrl-o	Change to overlined title
ctrl-l ctrl-i	Change to underlined and overlined title
ctrl-l = or = ctrl-l	Force Section heading (with = as underline)
ctrl-l 2	Force Section heading (level 2)

1.30.6 ConqureShell

Use :ConqueTermSplit to start the terminal in the Vim.

1.30.7 Airline

Airline theme plugin for better view e.g. display name of file, line number, column number etc.