

# My Vim Cookbook

Nikhil Cherian Kurian, May 2019

Vim is very good and professional text editor. We need editors to create source for C/C++/Python programming.

## 1 Frequently used commands

### 1.1 Help

|                     |   |
|---------------------|---|
| home/user> vimtutor | at the linux command line to get tutor file |
| :help               | inside vim to get general help              |
| :help set           | get help on the command set                 |

Note: These commands are cryptitic and for a beginner refer some online help [3](#).

### 1.2 Very basic

|              |                                |
|--------------|--------------------------------|
| vim file.txt | open the file with file.txt    |
| i            | start insert text mode command |
| my text      | type some test                 |
| Esc          | escape from the insert mode    |
| :wq          | write to the file and quit     |

Note: These commands are cryptitic and for a beginner refer some online help [3](#).

### 1.3 invisible commands

| command | description   |
|---------|---|
| yy      | copy one line   |
| 4yy     | copy 4 line   |
| dd      | delete one line   |
| 4dd     | delete 4 line   |
| p       | paste after the current cursor location   |
| P       | paste before the current cursor location  |
| g*      | keep the cursor in a word and them issue this command to search for that word forward |

|     |  |
|-----|--|
| g#  | keep the cursor in a word and then issue this command to search for that word backward         |
| gq} | keep the cursor at the beginning of a paragraph and issue the command to format that paragraph |
| gq) | keep the cursor at the beginning of a line and issue the command to format that line           |
| .   | repeat the previous command  |

## 1.4 search commands

| command     | description   |
|-------------|---|
| /car        | To search for word car  |
| n           | Repeat the previous search forward  |
| N           | Repeat the previous search backward   |
| :%s/xx/yy/c | search for the word xx and replace with yy. It asks for a confirmation every time         |
| :%s/xx/yy/g | search for the word xx and replace with yy. It is global replacement without confirmation |
| :g/xx       | list all the lines that contain the word xx   |

## 1.5 visible commands

| command      | description                                       |
|--------------|---|
| :77          | cursor goes to the line 77                        |
| :\$          | cursor goes to the last line                      |
| :r file.txt  | read the contents of file.txt to the current file |
| :split       | split the file into two                           |
| :edit f1.c   | open f1.c for editing                             |
| :next        | show the next file if it is opened                |
| :previous    | show the previous file if it is opened            |
| :vsplit      | split vertically                                  |
| :vsplit f2.c | split vertically v2.c                             |
| :            |   |

## 1.6 control commands

| command     | description                           |
|-------------|---------------------------------------|
| [ctrl+] y   | copy from the above line              |
| [ctrl+] e   | copy from the below line              |
| [ctrl+] w w | switch to next window                 |
| [ctrl+] w + | increase the size of the window       |
| [ctrl+] w = | set the size equal to all the windows |

## 1.7 command line options

| command       | description            |
|---------------|------------------------|
| vim f1.c f2.c | open f1.c and f2.c     |
| :next         | show the next file     |
| :n            | show the next file     |
| :previous     | show the previous file |
| :pre          | show the previous file |

```
vim -o f1.c f2.c
```

open f1.c and f2.c in each window

## 2 Advanced commands

### 2.1 Copy buffers

```
"a3yy  
"ap  
"A3yy
```

copy three lines to buffer a  
paste buffer a (We can have maximum 26 buffers)  
append three lines to buffer a/A

### 2.2 More search options

```
:g/\ <mon\>
```

list all the lines that contains the word mon, but does not show monday and summon. That is full word only.

```
:set ignorecase
```

Ignore case in search

```
/xx\|yy
```

To search for word xx OR yy in a line

```
/. *xx\&. *yy
```

To search for word xx AND yy in a line

```
/xx.*yy
```

To search for all characters from xx till yy

```
:10,+5 s/xx/yy/g
```

search for the word xx from lines 10 to 15 and replace with yy

```
:1,. s/xx/yy/g
```

search for the word xx from lines 1 to the current line and replace with yy

```
::,$ s/xx/yy/g
```

search for the word xx from lines from the current line till end of file and replace with yy

### 2.3 More search options

```
vim -s sfile file.tex
```

open file.tex and execute the script in sfile The script file could be any file that contains vim commands.

```
:s/apple/orange/g
```

This could be the content of a script file. This can be save in file named say sfile

```
:wq
```

To search in multiple file (say all txt files) and do the changes

```
vim *.txt
```

```
qq
```

```
:%s/orange/apple/g
```

```
:wnext
```

```
q
```

```
@q
```

```
99@
```

### 2.4 C program

```
>
```

Indent the code

```
:set ai
```

Set auto indent while coding

```
%
```

issue this at { or ( and this command will find the closing match

```
*
```

issue this at a word and this command will search similar words