#Vimtutorial

1. NORMAL MODE

- First Thing First in VIM
  - :q quit vim
  - :q! quit wihtout changes
  - ZQ quit wihtout changes
  - :wq save and quit vim
  - :x save and quit vim
  - ZZ save and quit vim

A. Move in vim

- k^, h<, l>, jv
- w Forward to the beginning of next word
- W Forward to the beginning of the next WORD
- b Backward to the next beginning of a word
- B Backward to the next beginning of a WORD
- e Forward to the next end of word
- E Forward to the next end of WORD
- [n]f Forward until (nth) (o) (Inclusive)
- [n]F Backward until (nth) (o) (Inclusive)
- [n]t Forward until (nth) (o) (Exclusive)
- [n]T Backward until (nth) (o) (Exclusive)
- h move one character left
- j move one row down
- k move one row up
- l move one character right
- w move to beginning of next word
- b move to previous beginning of word
- e move to end of word
- W move to beginning of next word after a whitespace
- B move to beginning of previous word before a whitespace
- E move to end of word before a whitespace
- 0 move to beginning of line

- $ move to end of line
- _ move to first non-blank character of the line
- g_ move to last non-blank character of the line
- gg move to first line
- G move to last line
- ngg move to n'th line of file (n is a number; 12gg moves to line 12)
- nG move to n'th line of file (n is a number; 12G moves to line 12)
- H move to top of screen
- M move to middle of screen
- L move to bottom of screen
- zz scroll the line with the cursor to the center of the screen
- zt scroll the line with the cursor to the top
- zb scroll the line with the cursor to the bottom
- Ctrl-D move half-page down
- Ctrl-U move half-page up
- Ctrl-B page up
- Ctrl-F page down
- Ctrl-O jump to last (older) cursor position
- Ctrl-I jump to next cursor position (after Ctrl-O)
- Ctrl-Y move view pane up
- Ctrl-E move view pane down
- n next matching search pattern
- N previous matching search pattern
- * next whole word under cursor
- # previous whole word under cursor
- g* next matching search (not whole word) pattern under cursor
- g# previous matching search (not whole word) pattern under cursor
- gd go to definition/first occurrence of the word under cursor
- % jump to matching bracket { } [ ] ( )
- fX to next 'X' after cursor, in the same line (X is any character)
- FX to previous 'X' before cursor (f and F put the cursor on X)

- tX til next ‘X’ (similar to above, but cursor is before X)

- TX til previous ‘X’

- ; repeat above, in same direction

- , repeat above, in reverse direction

2. INSERT MODE

- i Enter insert mode at cursor
- I Enter insert mode at first non-blank character
- s Delete character under cursor and enter insert mode
- S Delete line and begin insert at beginning of same line
- a Enter insert mode *after* cursor
- A Enter insert mode at the end of the line
- o Enter insert mode on the next line
- O enter insert mode on the above line
- C Delete from cursor to end of line and begin insert

3. COPY/PASTE

- y Yank. Example: yw (yank word)
- p paste after cursor
- P paste before cursor

4. SEARCHING

- / Forward
- ? Backward
- * Word under cursor - forward (bounded)
- g* Word under cursor - forward (unbounded)
- # Word under cursor - backward (bounded)
- g# Word under cursor - backward (unbounded)
- n Next result, forward
- N Next result, backward

5. EXPERT EDITING

3dw delete 3 words 3 d w | | `word ---. |` delete } 3 words ‘ 3 ———-/

ci(c i (| | `Parenthesis text-object |` Inside text-object ‘ change

d Delete: [range]d - dd delete current line - dj delete current and next line (j = down) - 2dj delete current and 2 lines downward

c Change (Same as d but put me in insert mode) cw change word cc = S Delete current line and enter insert mode
2cw Delete 2 words and enter insert mode

~ Toggle the case of character under cursor g~ Toggle case of [motion] g~w Toggle case of cursor -> end of word tr|ue -> g~w -> trUE g~iw Toggle case of entire word under cursor tr|ue -> g~iw -> TRUE p Paste

6. REGISTERS

“ Example:”ayy Yank current line into ‘a’ register ”ap Paste ‘a’ register

Paste contents of Example: [INS] Hello [ESC] byw [INS]” -> HelloHello

7. FIND AND REPLACE

- :s/search/replace/
- :8,10 s/search/replace/g
- :%s/search/replace/g

8. Number

- :set nu
- :set nonu
- :set rnu
- :set nornu

9. Macro

- q+key to record a task
- e.g q1 to record a task @1
- press q to quit record
- to run the macro #@record_key
- e.g 100@1 will repeat the macro 100 times

10. To save a file, simply type the following command:

- :w !sudo tee %

#For More https://github.com/shawncplus/vim-classes