

Vim



Lerning Vim and Nvim

remember you can use vim keybindings in any other editors (e.g in pycharm by installing ideavim)

Level.1

Modes:

1. Normal mode
2. Visual mode
3. insert mode
4. command mode (:)

🔗 Changing mode

to insert : i - a - I - A

insert before cursor: i

insert after cursor: a

append line: A

initial of line: I

insert in a new line : o

insert in a line above: O

to visual : v

shift + v = to go to the visual line mode

ctrl + v = go to the visual mode block mode

to go back to normal mode: Esc

Moving around

	k	
h		l
	j	

put cursor in the middle high and low in screen

shift + m

shift + h

shift + l

to move word by word forward

w

e

backwards

b

4w moving 4words

dw to delete a word

in word method (no need to select)

diw to delete a word when the cursor is in the word

ciw to delete a word and go to insert mode when cursor is in the word

yiw to copy the word that your cursor is in

dt *delete to the* char

Level.2

in normal mode

#cheat_sheet

key bindings	Action
dd	delete line
dw	delete the word
d2w	delete two words
d3j	delete 3 lines current and three below
u	undo
ctrl + r	redo
x	delete a character
y	copy (yanks)

key bindings	Action
yy	copy whole line
p	paste
d	cut
D	delete from cursor to end
di"	deletes content inside the quotation
yi"	yanks content inside the quotation
ci"	change content inside the quotation
gg	begining of the document
G	end of the document
dG	delete all the way to the end
dgg	delete all the way to the begining
10G	go to the 10th line
0	to get to the first character of the line
\$	to go to the end of a line
r	for replace and you can replace a letter
c	is for change
ce	change a word (e was for the end of the word)
cc	change the whole line (delete and go to insert)
%	if you have a for example { selected-jump---> }
t(char)	jumps to the next mentioned character
F(char)	to find char
t and F	F to find and t to jump one char before
T and F	are same thing just backward

 in the insert mode

- `ctl + w` = delete previous word
- `ctl + u` = delete all the way to the first of the line
- `ctl + o` and run normal mode commands in insert mode

Level.3

:h v_d help to know what d does in visual mode

Note: \$ vimtutor

set relativenumber

set norelativenumber

:colorscheme

~/.vimrc

-> configuration of .vimrc file

level.4

>>	indenting to right
<<	indenting to left
==	auto indent
gg=G	indenting whole file

to enter search mode /

to go to next similars press n

to go to previous ones press N

go to the first of a word and press # to find previous similar and * to find next

zz to center the current line

- full file replacing
:%s/string/replacement/g
- selected replacing
after selecting
:s/string/string/replacement/g
- repeating command
press . to execute the last command
- :reg
you can use the vim reg as a clipboard
- paste from reg 7
:7p
- yank the line to reg 6
:6yy

how to paste from sys clipboard to vim?

- macro (a sequence of vim bindings)
 1. press q to start recording
 2. write a letter as the name of the macro (for example qa)
 3. do sth (for example delete 3 words append and write 100 :
d3wA100)
 4. press q to stop recording
 5. you can check to macro stored in register by :reg
 6. @a to execute the macro a
 7. you can combine the macros

managing splits in vim

```
:split :sp
      to make a horizontal split
:vsplit :vs
      to make a vertical split
```

sources

Free code camp (Vim Tutorial for Beginners)

<https://youtu.be/RZ4p-saaQkc?si=YxkwAN0kmCjPvc0K>

The Primeagen (Vim AS Your Editor)

https://www.youtube.com/playlist?list=PLm323Lc7iSW_wuxqmKx_xxNtJC_hJbQ7R

Typecraft (Neovim for Newbs. FREE NEOVIM COURSE)

<https://www.youtube.com/playlist?list=PLsz00TDipIffreIaUNk64KxTIkQaGguqn>

dt (The Vim Lifestyle)

<https://www.youtube.com/watch?v=tExTz7GnpdQ&list=PL5--8gKSku15tivUyt0D-mERePLEzrWUz&index=13>

Nvim plugins

<https://github.com/junegunn/vim-plug>

How to install new plugins?

First: copy the git repo address to /home/amin/.config/nvim/vim.init file

Second: use vim plug command (PlugInstall) to install the plugins

When you no longer need a plugin

First: delete the git repo address from the init file
Second: use vim plug command (PlugClean) to rearrange the plugins

my vim config:

```
:set number
:set relativenumber
:set autoindent
:set tabstop=4
:set shiftwidth=4
:set smarttab
:set softtabstop=4
:set mouse=a

call plug#begin()

Plug 'http://github.com/tpope/vim-surround' " Surrounding ysw)
Plug 'https://github.com/preservim/nerdtree' " NerdTree
Plug 'https://github.com/tpope/vim-commentary' " For Commenting gcc & gc
Plug 'https://github.com/vim-airline/vim-airline' " Status bar
Plug 'https://github.com/lifepillar/pgsql.vim' " PSQL Plugging needs
:SQLSetType pgsql.vim
Plug 'https://github.com/ap/vim-css-color' " CSS Color Preview
Plug 'https://github.com/rafi/awesome-vim-colorschemes' " Retro Scheme
Plug 'https://github.com/neoclide/coc.nvim' " Auto Completion
Plug 'https://github.com/ryanoasis/vim-devicons' " Developer Icons
Plug 'https://github.com/tc50cal/vim-terminal' " Vim Terminal
Plug 'https://github.com/preservim/tagbar' " Tagbar for code navigation
Plug 'https://github.com/terryma/vim-multiple-cursors' " CTRL + N for
multiple cursors

set encoding=UTF-8

call plug#end()

nnoremap <C-f> :NERDTreeFocus<CR>
nnoremap <C-n> :NERDTree<CR>
nnoremap <C-t> :NERDTreeToggle<CR>
nnoremap <C-l> :call CocActionAsync('jumpDefinition')<CR>

nmap <F8> :TagbarToggle<CR>
```

```

:set completeopt-=preview " For No Previews

:colorscheme jellybeans

let g:NERDTreeDirArrowExpandable="+"
let g:NERDTreeDirArrowCollapsible="~"

" --- Just Some Notes ---
" :PlugClean :PlugInstall :UpdateRemotePlugins
"
" :CocInstall coc-python
" :CocInstall coc-clangd
" :CocInstall coc-snippets
" :CocCommand snippets.edit... FOR EACH FILE TYPE

" air-line
let g:airline_powerline_fonts = 1

if !exists('g:airline_symbols')
    let g:airline_symbols = {}
endif

" airline symbols
let g:airline_left_sep = '▶'
let g:airline_left_alt_sep = '⟩'
let g:airline_right_sep = '◀'
let g:airline_right_alt_sep = '⟨'
let g:airline_symbols.branch = '⌵'
let g:airline_symbols.readonly = '🔒'
let g:airline_symbols.linenr = 'ℒ'

inoremap <expr> <Tab> pumvisible() ? coc#_select_confirm() : "<Tab>"

```

vim plugins key bindings

gcc	comment the line
ysiw)	surrounding the word with)
ctl + n	file tree (nvim tree for more)
space f f	file search

space f b	search in current file
space c h	for nvchad cheat sheet