Introduction to vim and terminal (CL)

By Abdoulaye Balde
@abdoulayegk

Thank you for coming!

- Hello! **I’am Abdoulaye**, I am a student at Sharda University I can call myself as data … person, I work on different project using python.
-
- **Plan for today:**

1- Introduction to the command line (Terminal)

2- Basic file navigation

3-Introduction to vim

4- minimal configuration of vim

5- Operations on files

Workshop Structure

- I will be showing less slides but more demo-ing in terminal
- There will be stops after each section for questions
- We will have time for an FAQ at the end
- There will be some questions during the workshop you are encourage to participate.

What are all these words

- Shell: User interface(command-line or graphic
- Bash: A type of shell there are other shell like:

  1-Zsh

  2- Fish shell

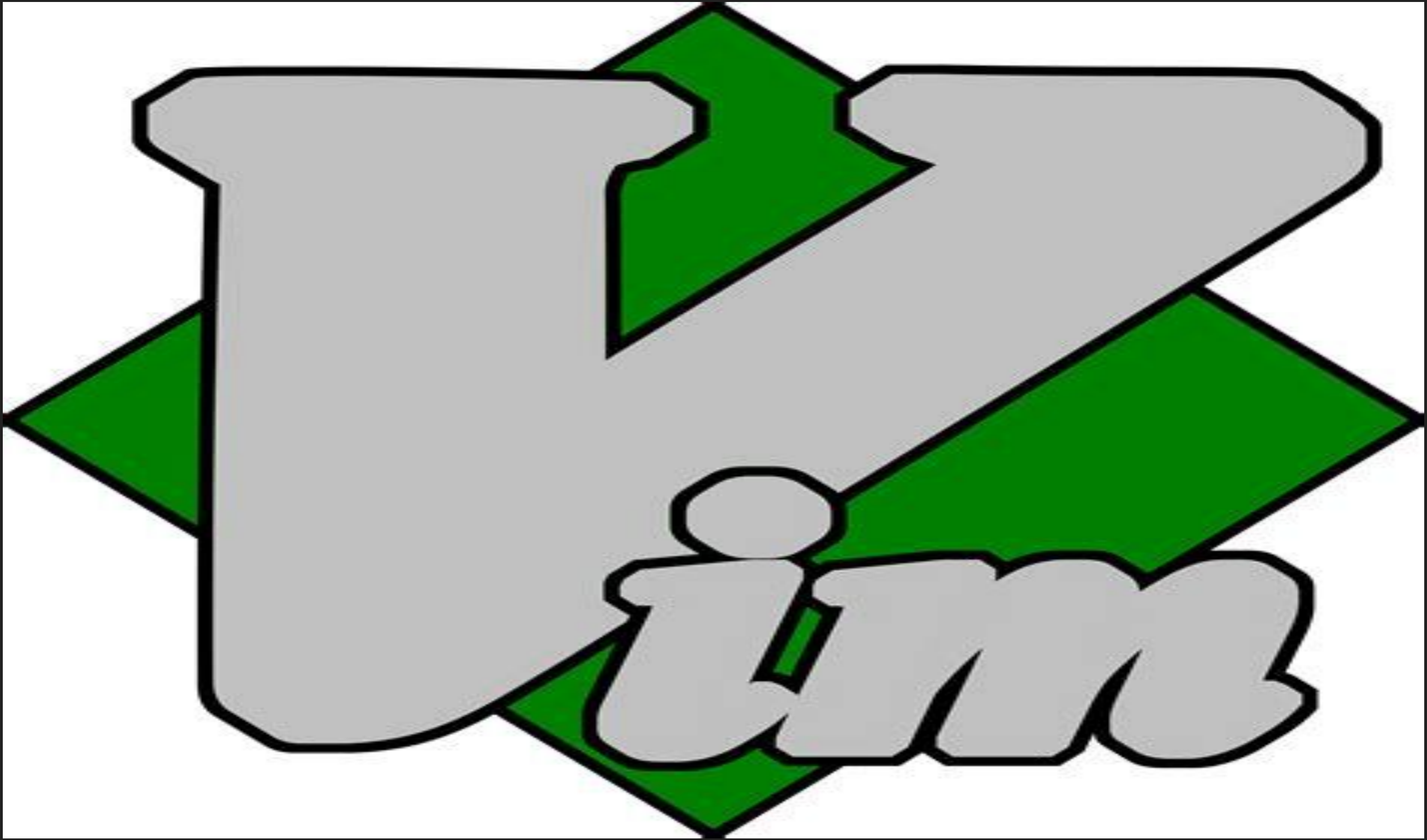
- Terminal: the program that runs a shell

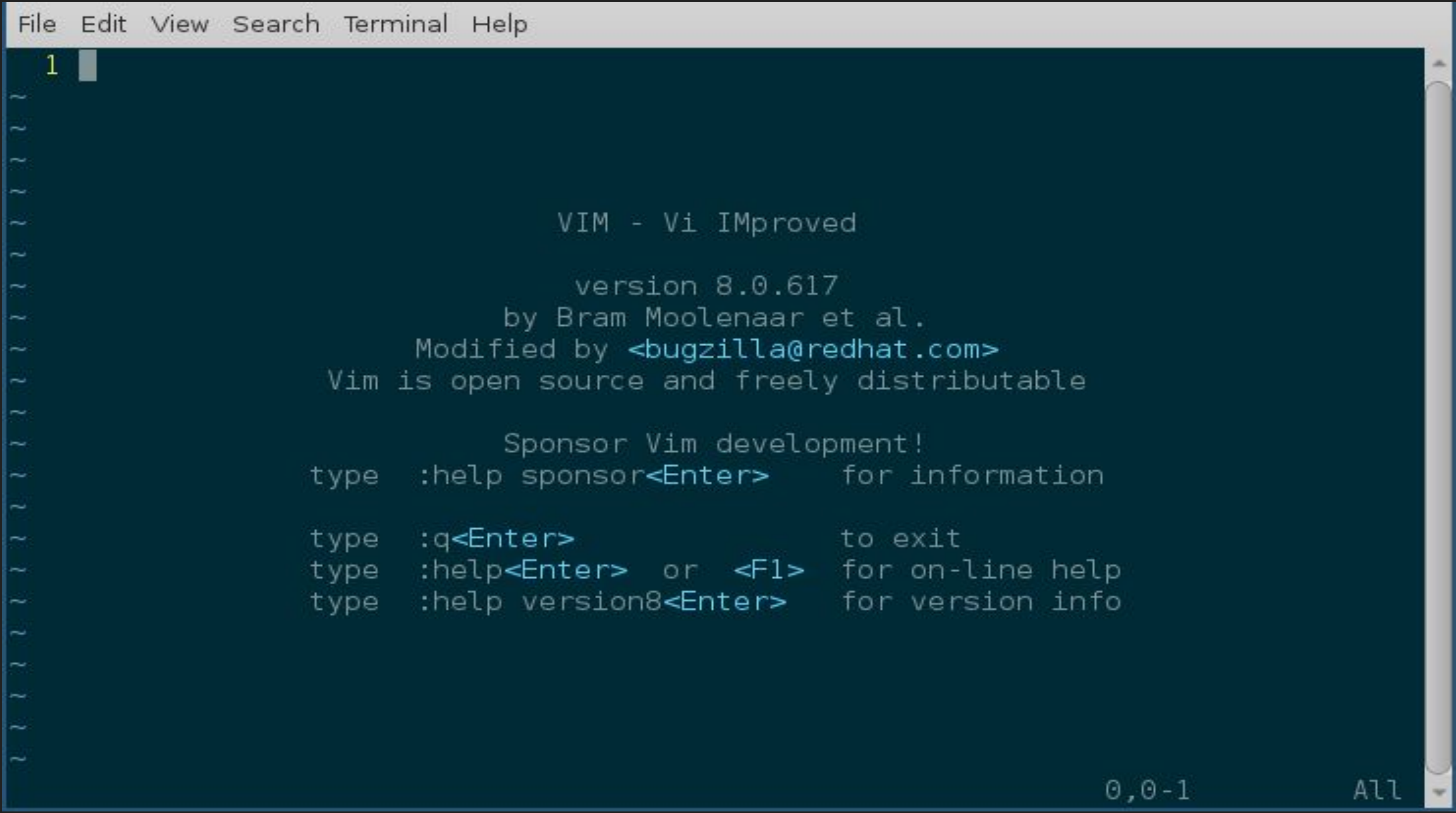
Basic Commands

- **whoami command**: whoami command will tell you which user account you are using in this system.
- **id command:** id prints real user id, and various other details related to the account.
- **Pwd:** pwd command, short for print working directory, will help you to find out the absolute path of the current directory.
- **cd command**: This command will help you to change your current directory.
- Others

  .. command

- **ls command:** We use ls command to list the files and directories inside any given directory. If you use ls command without any argument, then it will work on the current directory.
- **mkdir command**: make a directory
- **rm command:** rm command is used to remove a file, or directory. The -r option is being used to remove in a recursive way. With -f
- you force the removal, ignoring errors and never prompt. You can chain the flags, so instead of rm -r -f you can as well type rm -rf. But, always double check before you use rm -rf command, if you by mistake give this command in your home directory, or any other important directory, it will not ask to confirm, but it will delete everything there. So, please be careful and read twice before pressing enter key.


Vim



```
  1 
~
~
~
~
~                              VIM - Vi IMproved
~
~                               version 8.0.617
~                           by Bram Moolenaar et al.
~                      Modified by <bugzilla@redhat.com>
~                 Vim is open source and freely distributable
~
~                           Sponsor Vim development!
~                type  :help sponsor<Enter>    for information
~
~                type  :q<Enter>               to exit
~                type  :help<Enter>  or  <F1>  for on-line help
~                type  :help version8<Enter>   for version info
~
~
~
~
~
                                                              0,0-1         All
```

Introduction to vim

Vi/Vim is one of the most popular editors in the programming community. Vim has been around for decades, and offers many advantages and efficiencies, provided you are willing to learn it. Today, I hope to help you on the path of learning Vim. We will explore the basics of Vim in this workshop.

A Brief History of Vi and Vim

Before the appearance of the UNIX, the text editor “ed” appeared, written by Ken

Thompson in 1971. ed is regarded as one of the oldest text editors ever, as well

as the first to implement the concept of Regular Expressions. After a short period, the “ex” text editor appeared. ex is short for “EXtend”,

as it adds some features to ed and to make it simpler.

Vi appeared in 1976, written by Bill Joy which for the BSD operating system.

Licensed, of course, under the license with the same name as the operating

Vim installation

Most of the unix-like systems contain VIM editor. But if you don’t haven’t got it, use the below command to get wim editor.

- Mac OS X ====> brew install vim
- Linux =======> sudo apt install vim
- Windows ====> Gvim

Minimal Configuration

All the vim configuration is done inside the .vimrc file, vim will come by default with not configuration you are the one who will do all the configuration according to your need.

Setting Operations

- **Set [operation] show option value**
- **Etc**

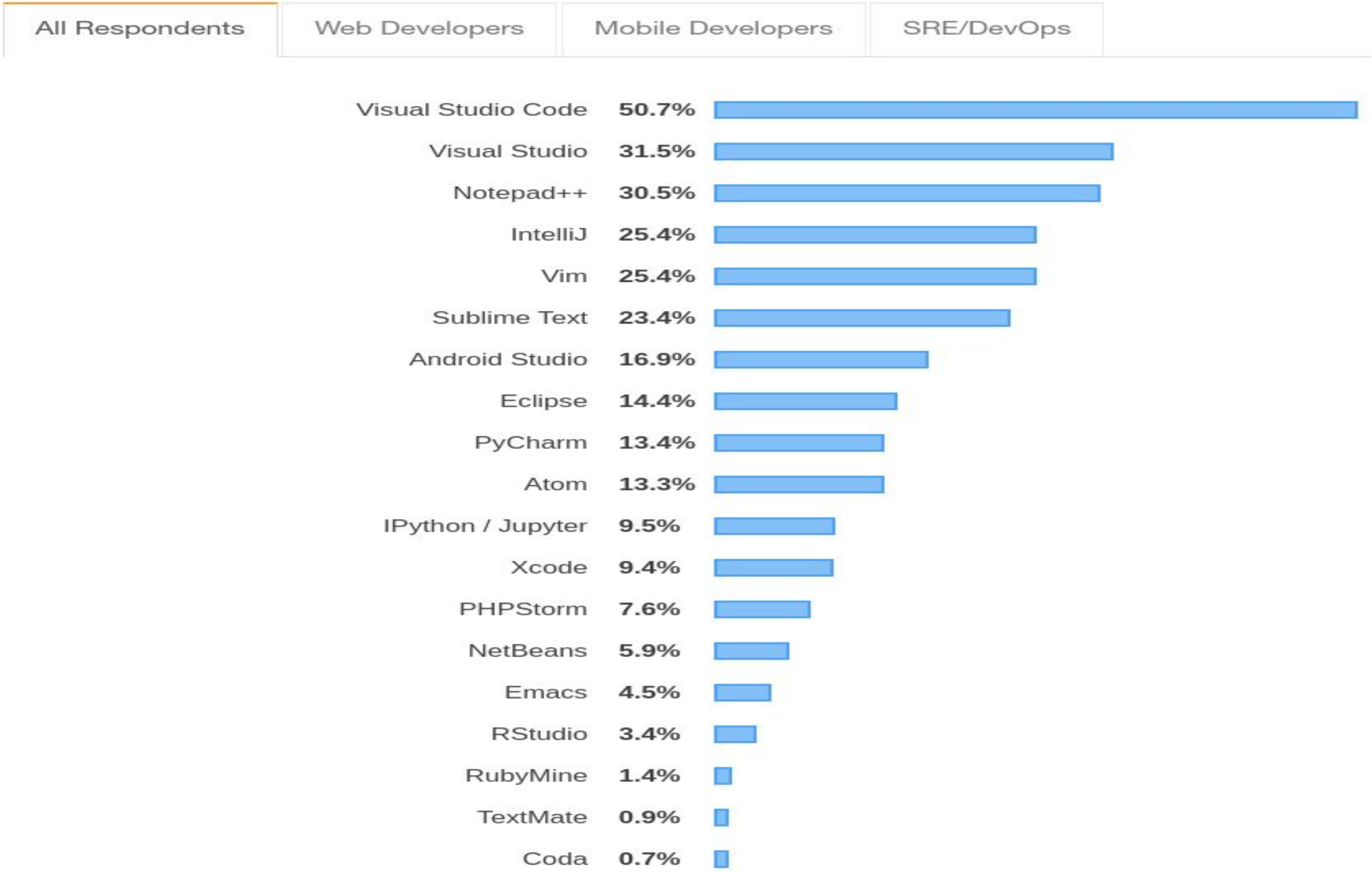
The basics of vim

There are 3 modes in the VIM editor(Given that different people say different number.

1. Insert mode – Bottom of your screen *INSERT* text will be shown
2. Command mode
3. Visual mode – Bottom of your screen *VISUAL LINE* text will be shown

If you are not sure which mode you are in press esc key, then you will be sent back to command mode. Now lets take a look at the movement of the VIM. To execute below commands you should in the command mode.

Most Popular Development Environments



- h – move the cursor one character to the left
- j – move the cursor down one line
- k – move the cursor up one line
- l – move the cursor one character to the right
- 0 – move the cursor to the beginning of the line
- ^ – move the cursor to the beginning of the line
- $ – move the cursor to the end of the line
- w – move forward one word
- b – move backward one word
- G – move to the end of the file
- gg – move to the beginning of the file
- } – move forward one paragraph
- { – move backward one paragraph

Basic of editing

- x – delete the character that the cursor is on
- X – delete the character behind the cursor
- dw – delete the word that the cursor is on
- cw – delete the word that the cursor is on and then go to insert mode
- d0 – delete to the beginning of a line
- d$ – delete to the end of a line
- dd – delete the whole line
- cc – delete the whole line and then go to insert mode
- db – delete the word behind
- dj – delete the current line and line below

- dgg – delete to the beginning of the file
- dG – delete to the end of the file
- u – undo the last operation
- r – Replace the character that the cursor is on
- R – Replace multiple characters

~

My configuration

```
git clone --depth=1 https://github.com/amix/vimrc.git ~/.vim_runtime

sh ~/.vim_runtime/install_awesome_vimrc.sh\n

 cd ~/.vim_runtime
```