vimrc and basic vim commands.

1.) .vimrc

- most likely there wont be a `.vimrc` file in home, you have to create one
    - `mkdir -p ~/.vim ~/.vim/autoload ~/.vim/backup ~/.vim/colors ~/.vim/plugged`
    - `touch ~/.vimrc` or just `touch .vimrc` to create file.

Now open `.vimrc` with vim and add the following

```
" Disable compatibility with vi which can cause unexpected issues.
set nocompatible

" Enable type file detection. Vim will be able to try to detect the type of
file in use.
filetype on

" Enable plugins and load plugin for the detected file type.
filetype plugin on

" Load an indent file for the detected file type.
filetype indent on
```

- To highlight syntax.

```
" Turn syntax highlighting on.
syntax on
```

(`set` command to set - maybe)

- to show line number

```
" Add numbers to each line on the left-hand side.
set number
```

- to incrementally highlights the matching character you type.

```
" While searching though a file incrementally highlight matching characters
as you type.
set incsearch
```

Reference to Learn VimScript The Hard Way

Vim Editor And vim commands.

- when you enter vim you are actully in command mode

- type `i` to enter `insert` mode.
- `:w <filensme>` to save the file
- `:q` - to get out of that interface
- `:q!` - to get out by overriding (without saving the changes).
- `A` : enters `insert` mode but cursor would be at the end of the line.

4.

- `gg` - to go to the top.
- `G` - to get to the bottom.

5. open file.

- `vim filename` : to open that file on vim.

5.1. open multiple files

- `vim -o fistfile secondfile.` - opens the two files in a horizontal split.
- `vim -O firstfile secondfile` - opens tthe two files in a vertical split.

5.2. to open files in different tabs.

```
vim -p file1 file2
```

6. delete

- after exiting insert mode

  - `x` : to delete. (deletes by character)

- `dd` : to delete the whole line.
  - `u` : to undo.

7. read file

- `:r <filename> to read the file`. Then save it an quit, by `:wq`.

```
most boring tie for work, and Mrs. Dursley gossiped away happily
as she wrestled a screaming Dudley into his high chair.
None of them noticed a large, tawny owl fl utter past the
 window.
At half past eight, Mr. Dursley picked up his briefcase, pecked
Mrs. Dursley on the cheek, and tried to kiss Dudley good-bye but
missed, because Dudley was now having a tantrum and throwing
his cereal at the walls. "Little tyke," chortled Mr. Dursley as he left
the house. He got into his car and backed out of number four's
drive.
It was on the corner of the street that he noticed the fi rst sign
of something peculiar — a cat reading a map. For a second, Mr.
Dursley didn't realize what he had seen — then he jerked his head
around to look again.  ere was a tabby cat standing on the corner.
total 20
-rwxrwxrwx 1 gowthamunni gowthamunni 1702 Nov  3 11:43 harrypotter.txt
-rwxrwxrwx 1 gowthamunni gowthamunni 9801 Nov  3 11:45 lordofmysteries.txt
-rwxrwxrwx 1 gowthamunni gowthamunni 1608 Nov  3 11:44 lordofrings.txt
-rwxrwxrwx 1 gowthamunni gowthamunni    0 Nov  3 13:56 noon.txt
Thu Nov  3 13:58:51 IST 2022
:r noon.txt
```

8. most efficient way to go back to terminal.

one of the most efficient way to back to terminal without quitting is by `ctrl + z`. This way the file you were updating or modifying would be in a suspended state and you can back to it by `fg` (foreground).

```
[1]+  Stopped                 vim harrypotter.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221103$
```

- it says stopped. Like this you can edit multiple files in parallel

```
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221103$ jobs
[1]-  Stopped                 vim harrypotter.txt
[2]+  Stopped                 vim lordofrings.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221103$
```

- `fg 1` - edit first file
- `fg -2` - edit 2nd file. (`fg <number>`)

9. to run linux commands in background.

- with in command mode
  - `:! <linux command>` and `enter` to get back .

10.

- to open the file at a particular line number : `vim +50 harrypotter.txt` - cursor would be at that line.

```
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221104$ vim +20
lordofrings.txt
```

11. buffer

We already know how to read a file within vim (using `:r <filename>`). Now when we run that command it's actually not written or added to the previously opened file. Suppose I opened the `harrypotter.txt` file and then `read` the `lordofrings.txt` file, Now you can see the `two` files but this is in a suspended state - an intermediate state - like a placeholder - or a `buffer` (by pressing `u` for undo - you can undo that). Now after writing (`:w`) and saving it that's no more the case.

```
17 this was too much of a good thing; it seemed unfair that
18 anyone should possess (apparently) perpetual youth as well
19 as (reputedly) inexhaustible wealth.
20 'It will have to be paid for,' they said. 'It isn't natural, and
21 trouble will come of it!'
22 But so far trouble had not come; and as Mr. Baggins was
23 generous with his money, most people were willing to forgive him his oddities and his good fortune. He remained
24 on visiting terms with his relatives (except, of course, the
25 Sackville-Bagginses), and he had many devoted admirers
26 among the hobbits of poor and unimportant families. But he
27 had no close friends, until some of his younger cousins began
28 to grow up.
~
:r harrypotter.txt
```

11.1. To open and edit multiple files

- open a file in vim like `vim harrypotter.txt`. Then open a new one by `:e lordofring.txt` , that way you have new file opened (or a new buffer opened) and can go back and forth by `:bp` (buffer previous) and `:bn` (buffer next).

- `:enew` - open new file.

- `:bd` - *buffer delete*.

11.2. To add additional buffers without switching to that.

syntax

```
:badd <filename>
```

- `badd` : Buffer add.

13. Visual mode

- `v` - to enter visual mode. Now if you want to copy something it's easier to do so by entering visual mode. Put the cursor at the start and and then enter visual mode. then by using `arrow keys` select the ones you want to copy and enter `y` to `copy`. After that use `p` to paste.

```
19 as (reputedly) inexhaustible wealth.
20
21
22
23 generous with his money, most people were willing to forgive him his oddities and his good fortune. He remained
24 on visiting terms with his relatives (except, of course, the
25 Sackville-Bagginses), and he had many devoted admirers
26 among the hobbits of poor and unimportant families. But he
27 had no close friends, until some of his younger cousins began
28 to grow up.
~
~
~
~
~
~
~
~
~
~
~
-- VISUAL --
```

14. sort

- select the texts in visual mode ---> press `:` -----> enter `sort ui` to sort alphabetically.

```
 20
 21
 22
 23
 24 on visiting terms with his relatives (except, of course, the
 25 Sackville-Bagginses), and he had many devoted admirers
 26 among the hobbits of poor and unimportant families. But he
 27 had no close friends, until some of his younger cousins began
 28 to grow up.



:'<,'>sort ui
```

15. Find and replace in vim

Syntax

```
:%s/thing to replace/thing to replace with/g
```

- `s` - may be to substitute.
- `g` - replace everywhere.

eg:

```
vim harrypotter.txt
:%s/Mr/master/g
```

replace Mr with master.

16. split

- Split screen. That can be done in two ways, horizontal split and vertical split
- syntax for horixontal mode : `:split harrypotter.txt` or just `:S harrypotter.txt`
- for vertical split : `:vsplit harrypotter.txt` or just `:vs harrypotter.txt`

17. to show difference between two files.

syntax

```
vim -d harrypotter.txt harrypotterchanged.txt
```

Note:

- dd is actually copying it (the line) to the paste buffer, so it's possible to paste it somewhere else using p .

Reference

- [Reference to tutorial](#)
- [Reference to docs on vim commands](#)