09 October
Thursday

Markdown Syntax

- extension — .md

=> # ␣ → header 1
→ header 2
⋮
→ header 6

=> --- → horizontal rule

=> *ABC* → Italics
or
ABC

=> **ABC** → Bold
or
__ABC__

=>

=> [Link name](Url) → linking page

=> ![Text if link not found](link) → Image
"Logo"
→ hovertext

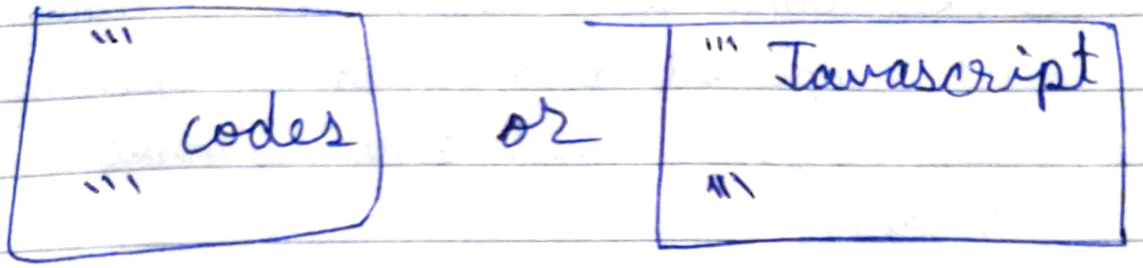
=> * one
* two
→ unordered lists

=> Tables

Notes

Appointment

=> to embed code



=> ~~strike~~

=> comments -

<!-- ABCD -->

=> striking text

<u><u>ABC</u></u>

P-T-O =>

Notes

Appointment

11 October Saturday

OCTOBER					2008
Mon		06	13	20	27
Tue		07	14	21	28
Wed	01	08	15	22	29
Thu	02	09	16	23	30
Fri	03	10	17	24	31
Sat	04	11	18	25	
Sun	05	12	19	26	

GIT Bash

=> . bashrc - is executable file which runs when bash is opened and contains specifications

=> . profile - used to store ~~info~~ environment variables

=> Creating a bash script.

① Type "$ vim testshell.sh" to open a file in vim
② Press "Insert" key to go in insert mode
③ Enter following in the file:

```
#!/bin/bash
echo "yashasvi"
```

④ Press escape to get out of insert mode.
⑤ Type :wq and press enter to save the file
⑥ Type chmod 755 testshell.sh to make the file executable
⑦ use ./testshell.sh to execute the file

12 Sunday

$ git blame <File name>
→ lists who, when changed files

$ ls
→ list all files/directories

$ cd home
→ go to home directory

$ pwd
→ show path of current directory (print working directory)

$ touch main.cpp
→ create file named "main.cpp"

$ mkdir hello
→ make a directory named hello

$ mv <filename> <dest. path>
→ moving a file to ~~to~~ destination

$ ls <folder name>
→ listing files inside a folder

$ mv <old filename> <new file name>
→ renaming

$ cat <file name>
→ showing contents of that file

$ cat > input.txt↵hello → Redirection
→ over-write hello inside input.txt file

$ cat >> input.txt↵hello
→ appending hello inside input.txt file

$ cat < hello.txt
→ taking input from file or text

$ cat < hello.txt > bye.txt
→ taking input from hello & writing to bye

LINUX

$ cp hello.txt bye.txt
→ copy from hello.txt to bye.txt.

$ rm <file name>
→ to remove a file

$ rm -f <file name>
→ remove file without confirming

$ <command> --help
→ to show command options for a command

$ man <command>
→ to open manual page of a command

$ wc <file name>
→ Gives "No. of lines", "No. of words" and "bytes used".

$ less <file name>
→ shows file part by part (large files)

$ head <file name>
→ shows top 10 lines of a file

$ tail <file name>
→ shows last 10 lines of a file

09 Sunday

$ cd ..
→ go back to the parent directory

$ touch <file name>
→ creating a new file

$ code hello.txt
→ opening a file in ~~java~~ VS code

$ clear
→ to clear the terminal screen

LINUX

Notes

Appointment

$ mv

=> Pip

=> Inp
err

$ grep

$ grep

Notes

$ mv direc1 direc2
→ moving directory direc1 inside directory direc2

=> Pipes – A coding tool that allows the output of one command to be used as the input for a different command.

$ cat file1.txt | wc -w
pipes
output works as an input of

=> Input represented by '0', output by '1' and error by '2'.

Ex use 2> to send error to a file.

$ ls -l /bin/usr > error.txt 2>&1
→ to store either output or the error in the txt file.

$ grep Sam names.txt
→ to see words in files which contain "sam"

$ grep ls -l | grep zip
→ searching in list of directories for "zip" using pipe