**Shell Scripting**

* touch 🡪 create a file
* cat /etc/shells 🡪 Shell type supported by our OS
* which bash 🡪 Location of the bash in our system
* #! 🡪 “Hash Bang”, “Shebang”
* echo 🡪 Printing
* read 🡪 Scanning
  + -p 🡪 to read from same line as printf
  + -sp 🡪 to take input but not visible on the screen
  + -a 🡪 to read array
  + if read has no var to take input by default it is stored in REPLY variable
* chmod +x 🡪 to change the permission to x (executable)
* # 🡪 comment
* System variables 🡪 (Capital mostly) Defined by OS
* User variables 🡪 (small letter maybe) Defined by user
* $BASH 🡪 (similar to **which bash**) location of bash
* $BASH\_VERSION 🡪 Version
* $HOME 🡪 home directory
* $PWD 🡪 present working directory
* $ symbol before a variable name gives value of variable
* ${names[0]} 🡪 to read an array
* $1, $2, $3… 🡪 Command Line Arguments
* $0 🡪 run command will also be printed (./file.sh)
* $@ 🡪 array of Command line arguments ()
* $# 🡪 number of command line arguments
* -e 🡪 checks if file exists or not
* -f 🡪 file exists and if it is a regular file or not
* -d 🡪 Directory exists or not
* -b 🡪 block special file (binary file)
* -c 🡪 character special file (text file)
* -s 🡪file is empty or not (false if empty)
* -r 🡪 checks for read permission
* -w 🡪 checks for write permission
* -x 🡪 checks for execute permission
* if, then, elif, else, fi
* int
  + -eq, -ne, -gt, -ge, -lt, -le using []
  + <, >, <=, >= using (())
* string
  + =, ==, != using []
  + <=, >=, <, > using [[]]
  + -z 🡪 isNull
* cat> 🡪 erases content and writes
* cat>> 🡪 appends to the file
* -a 🡪 and operator [ exp -a exp ]
* && 🡪 [ exp ] && [ exp ] or [[ exp && exp ]]