Shell Scripting:

A **shell** is simply a **macro processor** that **executes commands**.

**Bash** is the **shell**, or command language interpreter, for the GNU operating system.

Prompt:

Before change prompts properties we will save our current prompt in another environment variable:

🡪MYPROMPT=$PS1

🡪echo $MYPROMPT

🡪export PS1="[\t \j] "

Displays time of day and number of running jobs

🡪export PS1="[\d][\u@\h \w] : "

Displays date, user name, host name and current working directory. Note that \W displays only base names of the present working directory.

**EC2 instances futures can be change (color, name):**

export PS1="\[\033[1;31m\]\u@my-linux \[\033[1;37m\]\W: \$ "  
export PS1="\[\033[1;32m\]\u@my-ubuntu \[\033[1;37m\]\W: \$ "

### Shell Scripts:

Bash scripts often begin with:

#! /bin/bash

cat /etc/shells

Gives an overview of known shells on a Linux system

Default shell is set in the **/etc/passwd** file.

$ which bash

find ./ name\_of\_your\_bash

### Script File Execution

Create a new script file named hello-world.sh containing the below code:

#!/bin/bash

echo “Hello World”

Make your script executable with the chmod command and execute it

chmod +x hello-world.sh

./hello-world.sh

String Codes:

NAME="John"

echo "Hi $NAME" 🡪 Hi John

echo 'Hi $NAME' 🡪 Hi $NAME

Console Input:

read -p "Enter your name: " NAME

echo "Welcome $NAME"

read -s -p "Enter your password: " PASSWORD

echo -e "\nYour password is $PASSWORD"

**read -p:** To specify a prompt string and prompt is printed before the read is executed and doesn’t include a newline.

**read -s:** When entering sensitive information we do not want to display input coming.

Command Line Arguments ./file.sh Yasin Esma Zehra Hasan

echo "File Name is $0" 🡪 file.sh

echo "First Parameter is $1" 🡪 Yasin

echo "Second Parameter is $2" 🡪 Esma

echo "Third Parameter is $3" 🡪 Zehra

echo "All the Parameters are $@" 🡪 Yasin Esma Zehra Hasan

echo "Total Number of Parameters : $#" 🡪 4

echo "The current line number is $LINENO" 🡪 7

Arrays:

```bash

DISTROS[0]="ubuntu"

DISTROS[1]="fedora"

DISTROS[2]="debian"

DISTROS[3]="centos"

DISTROS[4]="alpine"

echo ${DISTROS[1]} 🡪 fedora

echo ${DISTROS[@]} 🡪 ubuntu fedora Debian centos alpine

echo ${#DISTROS[3]} 🡪 Debian

Simple Arithmetic:

```bash

expr 3 + 5 expr 9 / 3

expr 6 – 2 expr 7 % 2

expr 7 \\* 3

### let:

### ```bash

### let "sum = 3 + 5"

### echo $sum

x=5

let x++

echo $x

Calculator Script:

#!/bin/bash

read -p "Input first number: " first\_number

read -p "Input second number: " second\_number

let "sum = $first\_number + $second\_number"

let "sub = $first\_number - $second\_number"

let "mul = $first\_number \* $second\_number"

let "div = $first\_number / $second\_number"

echo "SUM=$sum"

echo "SUB=$sub"

echo "MUL=$mul"

echo "DIV=$div"

Double Parentheses: Evaluate arithmetic expression with double parentheses

```bash

sum=$((3 + 5))

echo $sum