**01 Starting off with Sha-Bang!!!!**

**What is Shell?**

The shell is a command line interpreter and is the interface between the user and the kernel. It gathers input from you and executes programs based on that input. When a program finishes executing, it displays that program's output.

Linux machine boots up, it executes the shell scripts in /etc/rc.d to restore the system configuration and set up services

**Shell characteristics and details**

* It is collection of Shell commands and programming constructs, written in a single file
* Shell executes the script by spawning another (child) shell:
  + that is when we run one shell program, it runs in a child shell area
* Scripts are interpreted; not compiled:
  + interpreted: each line will be processed by the system during run time. If any syntax error exists, that line will be skipped with error
  + compiled: in compiled oriented language, code will first be checked for syntax correctness. if any syntax error then compiler will catch that during the pre-processing time instead of checking during run time

**Types of Shell?**

* Bourne again shell [BASH]
* Bourne shell [SH]
* Korn shell [KSH]
* C shell [CSH]

**BASH Shell**

A *shell program* is typically an executable **binary** that takes commands that you type and (once you hit return), translates those commands into (ultimately) system calls to the Operating System API.

**SHELL SCRIPTING**

List of commands

When not to use shell scripts

* Resource-intensive tasks, especially where speed is a factor (sorting, hashing, recursion [2] ...)
* Procedures involving heavy-duty math operations, especially floating point arithmetic, arbitrary
* precision calculations, or complex numbers (use C++ or FORTRAN instead)
* Cross-platform portability required (use C or Java instead)
* Complex applications, where structured programming is a necessity (type-checking of variables, Function prototypes, etc.
* #!/bin/sh
* #!/bin/bash
* #!/usr/bin/perl
* #!/usr/bin/tcl
* #! /bin/sed -f
* #!/bin/awk –f

**#!**  Is a 2 byte magic number that designates a file type, in this case it denote executable shell script

**/bin/bash/** is a path name. This is the path to the program that interprets the commands in the script, whether it be a shell, a programming language, or a utility.

**Invoking the Script**

* **sh <script name>** (Not recommended is using **sh <scriptname**, since this effectively disables reading from stdin within the script)
* **bash <script name>**
* **./ <script\_name>**
* **/home/venkat/<script name>**

Create script folder & place all script there & use that path as global path variable. So you can invoke script anywhere.

Sudo vim /etc/profile

export SCRIPT\_HOME="/home/ec2-user/Shell\_programing/"

export PATH=$SCRIPT\_HOME:$PATH

source /etc/profile