Scripting provides capability to ease/automate the manual tasks with efficiency and error-less. Shell scripting is the native scripting language of Linux/Unix based systems.

It is the backbone of automation we do day by day on Linux/Unix systems. It helps to automate the system operational and administrative works.

As it is native to Linux/Unix OS, it’s very easy to learn and automate the monotonous/repetitive activity/process.

1. **Introduction**

 ■ What is SHELL  
 ■ Types of SHELL (SH/BASH/CHS/TCSH/KSH)  
 ■ SHELL features  
 ■ Scope and advantage of SHELL Script

 ■ Shell forking  
 ■ Bash environment set up (.bashrc/.bash\_profile/.bash\_history/.bash\_login/.bash\_logout/.vimrc)

1. **Writing and debugging scripts**

■ What forms script

■ Creating and Running scripts  
 ■ Debugging scripts

1. **Building blocks of scripting**

 ■ Variables (Definition and Type: Local/Environmental)  
 ■ Initializing/Uninitializing variables

 ■ Reading variables from users interactively

 ■ Play with shell Arrays  
 ■ Operators (Arithmetic/Relational/String/Boolean/FileTest)  
 ■ Arithmetic and Logical operations

■ Deep dive in String manipulation  
 ■ Calculation using bc (Bash Calculator)  
 ■ Deep dive in Input/Output Redirection in Shell  
 ■ Aliases

■ Work flow designing

■ Various useful Linux in-built commands

■ Various useful Linux external commands

1. **Conditional statements**

 ■ Simple if/else statements  
 ■ Nested if/else statements   
 ■ Practice on conditional statements

1. **Repetitive tasks using Loops**

 ■ The FOR loop (Shell and C style)  
 ■ The WHILE loop  
 ■ The UNTIL loop  
 ■ Break and continue

 ■ GETOPS functionality  
 ■ Practice on loops

1. **The Case statements**

 ■ Usage of case statement  
 ■ How to write case statements  
 ■ Practice on case statements

1. **Functions**

 ■ What is function  
 ■ Why function needed  
 ■ How to write functions  
 ■ Calling functions in modular functionality  
 ■ Passing arguments in functions

 ■ Here and Help functions  
 ■ Practice on functions

1. **Catching signals and Traps**

 ■ What is signals and Traps

 ■ Types of signals  
 ■ Why signals/traps needed  
 ■ How to catch signal/trap

 ■ Exit and exit status  
 ■ Use cases scenarios and practices

1. **Writing interactive scripts**

 ■ When interactive scripts are needed  
 ■ How to write interactive script

■ Reading arguments in run-time and use it in script

1. **Advance Shell script coverage**

■ Deep dive in Regular Expressions

* Deep understanding of Meta characters
* Huge real time RegEx examples

■ Deep dive in AWK programming

* AWK structure
* General field processing
* Pattern processing with RegEx and output customization
* Using user defined variables inside AWK
* Inbuilt variables in AWK
* If/Else condition inside AWK
* Loops inside AWK
* Function inside AWK

■ Inclusion of SED programming

■ Writing system monitoring plug-in in Shell