

ShellScripting

The **ShellScripting Course** is designed to introduce beginners to learn technical skills which are necessary for **DevOps, Linux and Automation**.

Who It's For

- Beginners who want to start with **Linux and automation**
- Students preparing for **DevOps, Cloud, or System Admin** roles
- IT professionals looking to **improve productivity** using scripting
- Anyone who wants to automate repetitive tasks and save time

What You'll Learn

- Basics of shell scripting and command-line usage
- Writing and executing your own shell scripts
- Working with variables, loops, and conditions
- Automating real-world tasks like Connectivity Check, and RAM Monitoring

Benefits You'll Gain

- Save time by **automating** daily manual tasks
- Build a strong foundation for **DevOps and Cloud careers**
- Increase your efficiency and **problem-solving** skills
- Add a **valuable skill** to your resume
- Gain **confidence** in working with Linux environments

Course Details

- **Duration:** 5 days (1 hour per day)
- **Mode:** Fully online – attend live
- **Start Date:** Flexible enrollment – begin anytime!

Topics that will be covered

Shell and Types

- ◆ What is shell
- ◆ Shell Commands
- ◆ Types of Shell
- ◆ Shell Scripting
- ◆ Shebang in Shell Scripting

Comments:

Learn to add single-line and multi-line comments for better code readability.

Multiline Comments:

Understand how to document scripts using multi-line comments.

Variables:

Learn how to declare and use variables in shell scripts.

Constant Variables:

Understand how to create read-only variables for fixed values.

Arrays:

Work with arrays to store and manage multiple values.

Array Key-Value:

Learn associative arrays to store data in key-value pairs.

String Operations:

Perform string manipulation like length, substring, and replacement.

User Interaction:

Take input from users and display output dynamically.

Arithmetic Operations:

Perform basic mathematical calculations in scripts.

Conditional Statements:

Use if-else conditions to control script flow.

Logical Operations:

Apply logical operators to combine multiple conditions.

Loops:

Automate repetitive tasks using for, while, and until loops.

Functions:

Create reusable blocks of code using functions.

Arguments:

Pass and use command-line arguments in scripts.

Exit Status:

Handle success and failure using exit codes.

Monitoring RAM Free Space:

Build scripts to monitor and report system memory usage.