

# Introduction to Shell Scripting

## What is Shell Scripting?

A **Shell Script** is a text file containing a sequence of **commands** that are executed by a **Linux shell (command-line interpreter)**. It is used to automate repetitive tasks, manage system operations, and execute complex workflows efficiently.

A shell script allows users to execute multiple commands in sequence without manually typing them in the terminal each time.

---

## Why Use Shell Scripting?

Shell scripting is widely used in system administration, automation, and DevOps for tasks such as:

- ☒ **Automating repetitive tasks** (e.g., backups, log management)
  - ☒ **Managing files and directories** (e.g., creating, renaming, deleting)
  - ☒ **User management** (e.g., creating users, assigning permissions)
  - ☒ **Monitoring system performance** (e.g., checking CPU usage, memory)
  - ☒ **Scheduling tasks** (e.g., using `cron` for periodic execution)
- 

## Types of Shells in Linux

There are different types of shell interpreters available in Linux:

Shell Type	Description
<b>Bash (Bourne Again Shell)</b>	Default and most widely used shell
<b>Sh (Bourne Shell)</b>	Original Unix shell
<b>Csh (C Shell)</b>	Uses C-like syntax
<b>Ksh (Korn Shell)</b>	Advanced scripting features
<b>Zsh (Z Shell)</b>	Extended version of Bash with extra features

Most Linux distributions use **Bash** (`/bin/bash`) as the default shell.

---

## Basic Structure of a Shell Script

A shell script typically follows this structure:

- ❶ **Shebang (#!)** – Specifies the shell interpreter
- ❷ **Commands** – A series of commands to execute
- ❸ **Comments (#)** – For documentation

## Example Shell Script:

```
bash
CopyEdit
#!/bin/bash
# This is a simple shell script

echo "Hello, World!"    # Print a message
```

### 📌 Explanation:

- `#!/bin/bash` → This is called a **shebang**, and it tells the system to use the **Bash shell** to interpret the script.
- `echo "Hello, World!"` → Prints text to the terminal.

# Creating and Running a Shell Script

## ❶ Creating a Script

To create a script file, use a text editor like `nano`, `vim`, or `touch` command:

```
bash
CopyEdit
nano my_script.sh
```

## ❷ Writing the Script

Inside the file, add the following content:

```
bash
CopyEdit
#!/bin/bash
echo "Welcome to Shell Scripting!"
```

## ❸ Making the Script Executable

Before running, give execution permissions:

```
bash
CopyEdit
chmod +x my_script.sh
```

## ❹ Running the Script

Execute the script using:

```
bash
CopyEdit
./my_script.sh
```