

Bash Shell Scripts

Think of a **shell script** as a "to-do list" for your computer. Instead of you typing commands one by one into a terminal and waiting for each to finish, you bundle them into a single file that the system executes automatically.

What is a Shell Script? In Linux, a shell script is a text file containing a sequence of commands. It's written in a scripting language that the **shell** (the command-line interpreter) can understand.

Key Characteristics:

- **Automation:** They excel at repetitive tasks like backups, system monitoring, or software installation.
 - **Interpreted:** Unlike C++ or Java, shell scripts aren't "compiled" into binary code; the shell reads them line-by-line.
 - **Power:** Scripts can use loops, variables, and "if/then" logic, making them much more than just a list of commands.
-

The Bash Shell: The Linux Standard

Bash (short for **B**ourne **A**gain **S**hell) is the most common shell found in Linux distributions. It's an enhanced version of the original Unix "Bourne shell" (sh). It is the standard shell because of

1. **Universality:** It is the default shell for almost every major Linux distro (Ubuntu, Fedora, Debian) and was the long-time default for macOS.
2. **Command History:** It remembers what you typed previously (just hit the up arrow), saving you from re-typing long strings.
3. **Tab Completion:** You can type the first few letters of a file or command, hit Tab, and Bash fills in the rest.
4. **Scripting Features:** It supports complex functions, arrays, and arithmetic, making it robust enough for professional system administration.

The "Shebang"

Every good Bash script starts with a special first line called the **Shebang**:

```
#!/bin/bash
```

This tells the Linux kernel: *"Hey, don't just guess how to run this—use the Bash interpreter located at /bin/bash."*

Comparison: Shell vs. Terminal vs. Script

Term	Analogy	Function
Terminal	The Physical Computer	The window/interface where you type.
Shell (Bash)	The Operating System	The "brain" that interprets your commands.
Shell Script	A Recorded Macro	A saved file that runs a sequence of those commands.

Running Bash Scripts on Linux, MacOS, and Windows Machines

On Linux, a Bash shell script stored in a file *myscript.sh* should run natively in a terminal window. It will run on a Mac with one minor tweak, and will run on Windows, but Windows requires a specific environment to understand Bash commands.

1. Running on Linux

Linux handles Bash scripts natively. Just execute in a terminal window (assuming *myscript.sh* to be in your path)

```
./myscript.sh
```

2. Running on macOS

macOS is Unix-based, so it handles Bash scripts natively. However, modern Macs (since macOS Catalina) use **Zsh** as the default shell rather than Bash.

- **The Tweak:** The "shebang" at the top of the script (`#!/bin/bash`) is still valid because macOS keeps a version of Bash at that location for compatibility.
- **Execution:** You just need to give the file permission to run:

```
Bash
chmod +x myscript.sh
./myscript.sh
```

3. Running on Windows

Windows does not run .sh files natively in the Command Prompt or PowerShell. You have three main options:

5. **Git Bash (Recommended):** If you have Git installed on Windows, you already have "Git Bash." You can right-click in your folder and select "Git Bash Here" to run the script exactly like you would on Linux.
WSL (Windows Subsystem for Linux): This is a full Linux kernel (usually Ubuntu) running inside Windows. This is the most robust method for scientific computing.
6. **Cygwin:** A large collection of GNU and Open Source tools which provide functionality similar to a Linux distribution on Windows.