

Linux (UNIX) Shell Scripting for Automation

Learn how to automate tasks using shell scripts with practical syntax and examples.

What is Shell Scripting?

- A shell script is a text file with a sequence of shell commands
- Common shells: `bash` , `sh` , `zsh`
- Used for **automation**, system tasks, backups, batch jobs

Writing a Script

1. Create a file: `script.sh`
2. Add shebang: `#!/bin/bash`
3. Make it executable: `chmod +x script.sh`
4. Run: `./script.sh`

Or, you can skip the step 2 and 3:

1. Run: `bash ./script.sh`



Variables

```
name="Alice"  
echo "Hello, $name"
```

- No space around `=`
- Use `$` to access

Conditionals

```
if [ "$a" -eq 5 ]; then
    echo "a is 5"
elif [ "$a" -gt 5 ]; then
    echo "a is greater than 5"
else
    echo "a is less than 5"
fi
```

Loops

For Loop

```
for i in 1 2 3; do
    echo "Number $i"
done
```

While Loop

```
count=1
while [ $count -le 5 ]; do
    echo $count
    ((count++))
done
```

Functions

```
greet() {  
  echo "Hello, $1"  
}  
  
greet "Bob"
```

- `$1` , `$2` , ... are positional parameters

File and Directory

```
mkdir myfolder  
cd myfolder  
touch file.txt  
rm file.txt
```


Redirects and Pipes

```
echo "Log entry" >> log.txt  
cat file.txt | grep "error"
```

- `>` overwrite, `>>` append
- `|` passes output to next command



Automation Example: Backup

```
#!/bin/bash
src="/home/user/docs"
dest="/backup/docs_$(date +%F).tar.gz"
tar -czf $dest $src
echo "Backup saved to $dest"
```

Cron Jobs

Run scripts at scheduled times

```
crontab -e
```

Example: run daily at 2 AM

```
0 2 * * * /path/to/script.sh
```

Useful Built-ins

- `$#` – number of arguments
- `$@` – all arguments
- `$$` – PID of script
- `$?` – exit code of last command

Best Practices

- Quote variables: `"$var"`
- Use `set -e` to stop on errors
- Use logging
- Use meaningful variable names
- Test before automating

Summary

- Shell scripting is great for automating Unix tasks
- Learn variables, loops, conditionals, and functions
- Use cron for scheduled automation

 **You're Ready**

Automate with shell scripting!