1 What is Shell Scripting?

A **shell script** is a text file containing Linux/Unix commands that are executed automatically by the shell.

The shell is a command interpreter that interacts with the operating system.

Common shells:

- `bash` (most commonly used)
- `sh`
- `zsh`

Instead of typing commands manually every time, you can automate them using a script.

Example:
Instead of typing:

```
cp file.txt backup/
mv file.txt old/
```

You can write a script that does everything automatically.

---

2 Basic Structure of a Shell Script

```
#!/bin/bash

echo "Hello World"
```

Explanation:

- `#!/bin/bash` → Shebang (tells the system to use bash)
- `echo` → prints output to the terminal

---

3 How to Create and Run a Script

Step 1: Create file

```
nano myscript.sh
```

Step 2: Write your script

Step 3: Give execute permission

```
chmod +x myscript.sh
```

Step 4: Run the script

```
./myscript.sh
```

Or:

```
bash myscript.sh
```

---

4 Basic Shell Programs

✅ 1. Create a File (touch)

```
#!/bin/bash

echo "Creating file..."
touch myfile.txt
echo "File created successfully"
```

✅ 2. Copy a File

```
#!/bin/bash

cp myfile.txt copy_myfile.txt
echo "File copied"
```

✅ 3. Move or Rename a File

```
#!/bin/bash

mv copy_myfile.txt renamed_file.txt
echo "File moved/renamed"
```

✅ 4. Delete a File

```
#!/bin/bash

rm renamed_file.txt
echo "File deleted"
```

✅ 5. Write Logs to a File

```
#!/bin/bash

echo "Script started at $(date)" >> mylog.log
echo "Process running..." >> mylog.log
echo "Script ended at $(date)" >> mylog.log

echo "Log written successfully"
```

>> means append to file.

✅ 6. Show Last Lines of a File (tail)

```
#!/bin/bash

echo "Last 5 lines of log:"
tail -5 mylog.log
```

✅ 7. Take User Input

```
#!/bin/bash

echo "Enter file name:"
read filename

touch $filename
echo "$filename created"
```

✅ 8. Simple Backup Script

```
#!/bin/bash

echo "Enter file to backup:"
read file

cp $file $file.bak
echo "Backup created: $file.bak"
```

✅ 9. If Condition Example

```
#!/bin/bash

if [ -f myfile.txt ]
then
    echo "File exists"
else
    echo "File does not exist"
fi
```

`-f` checks if a file exists.

✅ 10. Loop Example

```
#!/bin/bash

for i in 1 2 3 4 5
do
    echo "Number: $i"
done
```

5️⃣ Important Basic Commands

Command	Purpose
touch	Create file
cp	Copy file
mv	Move/Rename
rm	Delete file
mkdir	Create folder
rmdir	Delete folder
cat	Display file content
tail	Show last lines
head	Show first lines
echo	Print output
read	Take input

6️⃣ Variables Example

```
#!/bin/bash

name="John"
echo "Hello $name"
```

---

7 File Permissions

Check permissions:

```
ls -l
```

Give execute permission:

```
chmod +x script.sh
```

Linux permissions:

- r → read
- w → write
- x → execute