
📄 LAB2.md – Script Execution & Explanation

💻 Assignment 2 – Script Execution & Explanation

🎯 Objective

To understand how shell scripts work by running and explaining two scripts from the *Scripts/* folder.

1. Script: *print_numbers.sh*

📌 Purpose

This script prints a sequence of numbers from 1 to 5.

📄 Code

```
#!/bin/bash
# print_numbers.sh
# Prints numbers from 1 to 5

for i in {1..5}
do
    echo "Number: $i"
done
```

❖ Line-by-Line Explanation

1. `#!/bin/bash` → tells the system to use **bash shell** to run this script.
2. `# print_numbers.sh` → a **comment**, ignored by the system.
3. `for i in {1..5}` → loop through numbers 1 to 5.
4. `do` → begins the loop body.
5. `echo "Number: $i"` → prints the current number.
6. `done` → ends the loop.

▶ Example Run

```
$ ./print_numbers.sh
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5
```

2. Script: array_loop.sh

❖ Purpose

This script demonstrates how to use arrays and loop through them.

❑ Code

```
#!/bin/bash
# array_loop.sh
# Demonstrates array usage

fruits=("apple" "banana" "cherry")

for fruit in "${fruits[@]}"
do
    echo "Fruit: $fruit"
done
```

❖ Line-by-Line Explanation

1. `#!/bin/bash` → runs the script with bash.
2. `fruits=("apple" "banana" "cherry")` → defines an **array** with 3 values.
3. `for fruit in "${fruits[@]}"` → loop through each element of the array.
4. `do` → starts loop body.
5. `echo "Fruit: $fruit"` → prints current fruit.
6. `done` → ends loop.

▶ Example Run

```
$ ./array_loop.sh
Fruit: apple
Fruit: banana
Fruit: cherry
```

❓ Extra Questions

- ◊ Q1: What is the purpose of `#!/bin/bash` at the top of a script?

☞ It is called a **shebang**. It tells the operating system to use **bash** to interpret the script. Without it, the system may not know which shell to use.

- ◊ Q2: How do you make a script executable?

⌚ Use the **chmod** command:

```
$ chmod +x scriptname.sh
```

This gives the file **execute permission**, so you can run it as **./scriptname.sh**.

☑ Conclusion

- We learned how to run and explain shell scripts.
- **print_numbers.sh** showed a simple loop with numbers.
- **array_loop.sh** demonstrated arrays and looping through them.
- Key concepts: **loops, arrays, shebang, permissions**.

---terminal screenshots

```
chayanika@aryaveer-lohia-ThinkPad-E16-Gen-2:~/Desktop/1/test$ touch print_numbers.sh
chayanika@aryaveer-lohia-ThinkPad-E16-Gen-2:~/Desktop/1/test$ nano print_numbers.sh
chayanika@aryaveer-lohia-ThinkPad-E16-Gen-2:~/Desktop/1/test$ chmod 777 print_numbers.sh
chayanika@aryaveer-lohia-ThinkPad-E16-Gen-2:~/Desktop/1/test$ bash print_numbers.sh
number : 1
number : 2
number : 3
number : 4
number : 5
chayanika@aryaveer-lohia-ThinkPad-E16-Gen-2:~/Desktop/1/test$ nano print_numbers.sh
chayanika@aryaveer-lohia-ThinkPad-E16-Gen-2:~/Desktop/1/test$ bash print_numbers.sh
print_numbers.sh: line 2: syntax error near unexpected token `('
print_numbers.sh: line 2: `colors= ("red" "green" "blue")'
chayanika@aryaveer-lohia-ThinkPad-E16-Gen-2:~/Desktop/1/test$ nano print_numbers.sh
chayanika@aryaveer-lohia-ThinkPad-E16-Gen-2:~/Desktop/1/test$ bash print_numbers.sh
Color: color
Color: color
Color: color
chayanika@aryaveer-lohia-ThinkPad-E16-Gen-2:~/Desktop/1/test$ nano print_numbers.sh
chayanika@aryaveer-lohia-ThinkPad-E16-Gen-2:~/Desktop/1/test$ bash print_numbers.sh
enter start number:2
enter end number: 5
enter step (positive number): 15
print_numbers.sh: line 7: unexpected EOF while looking for matching `''
```

```
chayanika@aryaveer-lohia-ThinkPad-E16-Gen-2:~/Desktop/1/test$ touch print_numbers.sh
chayanika@aryaveer-lohia-ThinkPad-E16-Gen-2:~/Desktop/1/test$ nano print_numbers.sh
chayanika@aryaveer-lohia-ThinkPad-E16-Gen-2:~/Desktop/1/test$ chmod 777 print_numbers.sh
chayanika@aryaveer-lohia-ThinkPad-E16-Gen-2:~/Desktop/1/test$ bash print_numbers.sh
number : 1
number : 2
number : 3
number : 4
number : 5
chayanika@aryaveer-lohia-ThinkPad-E16-Gen-2:~/Desktop/1/test$
```