🐚 BASICS OF SHELL SCRIPTING

📝 Assignment 2 – Script Execution & Explanation

Objective: Understand how existing scripts in the `Scripts/` folder work.
We'll analyze two sample Bash scripts, explain them line by line, and show example runs.

📌 Script 1: `print_numbers.sh`

🎯 Purpose

Prints numbers from 1 to a specified limit using a loop.

📜 Script Code

```
#!/bin/bash
# print_numbers.sh

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

🔎 Line-by-Line Explanation

- `#!/bin/bash` → tells the system to use the **Bash shell** to run this script.
- `for i in {1..5}` → loops through numbers **1** to **5**.
- `do` → starts the loop block.
- `echo "Number: $i"` → prints each number with the label **Number:**.
- `done` → ends the loop.

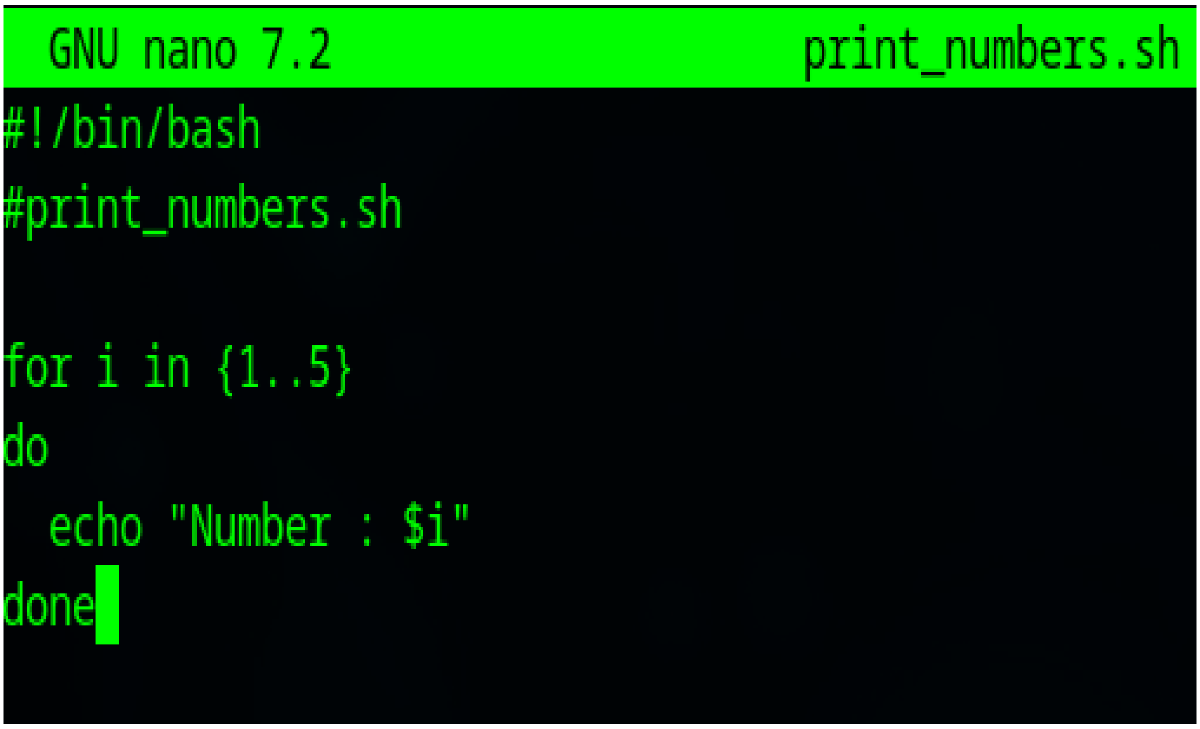
▶️ Example Run

Command:

```
chmod 777 print_numbers.sh
./print_numbers.sh
```

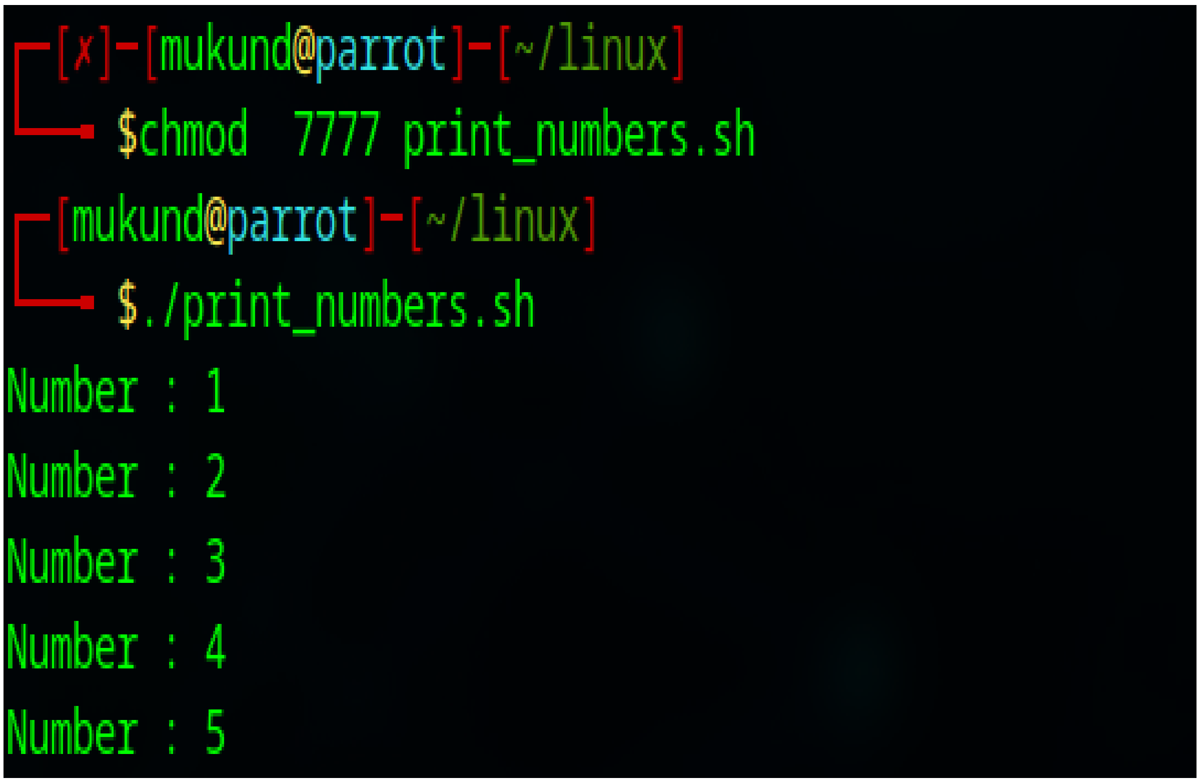
Output:

```
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5
```



```
GNU nano 7.2                                    print_numbers.sh
#!/bin/bash
#print_numbers.sh

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



```
┌─[✗]─[mukund@parrot]─[~/linux]
└──╼ $chmod  7777 print_numbers.sh
┌─[mukund@parrot]─[~/linux]
└──╼ $./print_numbers.sh
Number : 1
Number : 2
Number : 3
Number : 4
Number : 5
```

📌 Script 2: `array_loop.sh`

🎯 Purpose

Demonstrates looping through an array of items in Bash.

📜 Script Code

```bash
#!/bin/bash
# array_loop.sh

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

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

🔎 Line-by-Line Explanation

- `#!/bin/bash` → ensures the script runs with the **Bash interpreter**.
- `fruits=("apple" "banana" "cherry")` → defines an array named **fruits**.
- `for fruit in "${fruits[@]}"` → loops through all items in the array.
- `do` → begins the loop block.
- `echo "Fruit: $fruit"` → prints each fruit with the label **Fruit:**.
- `done` → ends the loop.

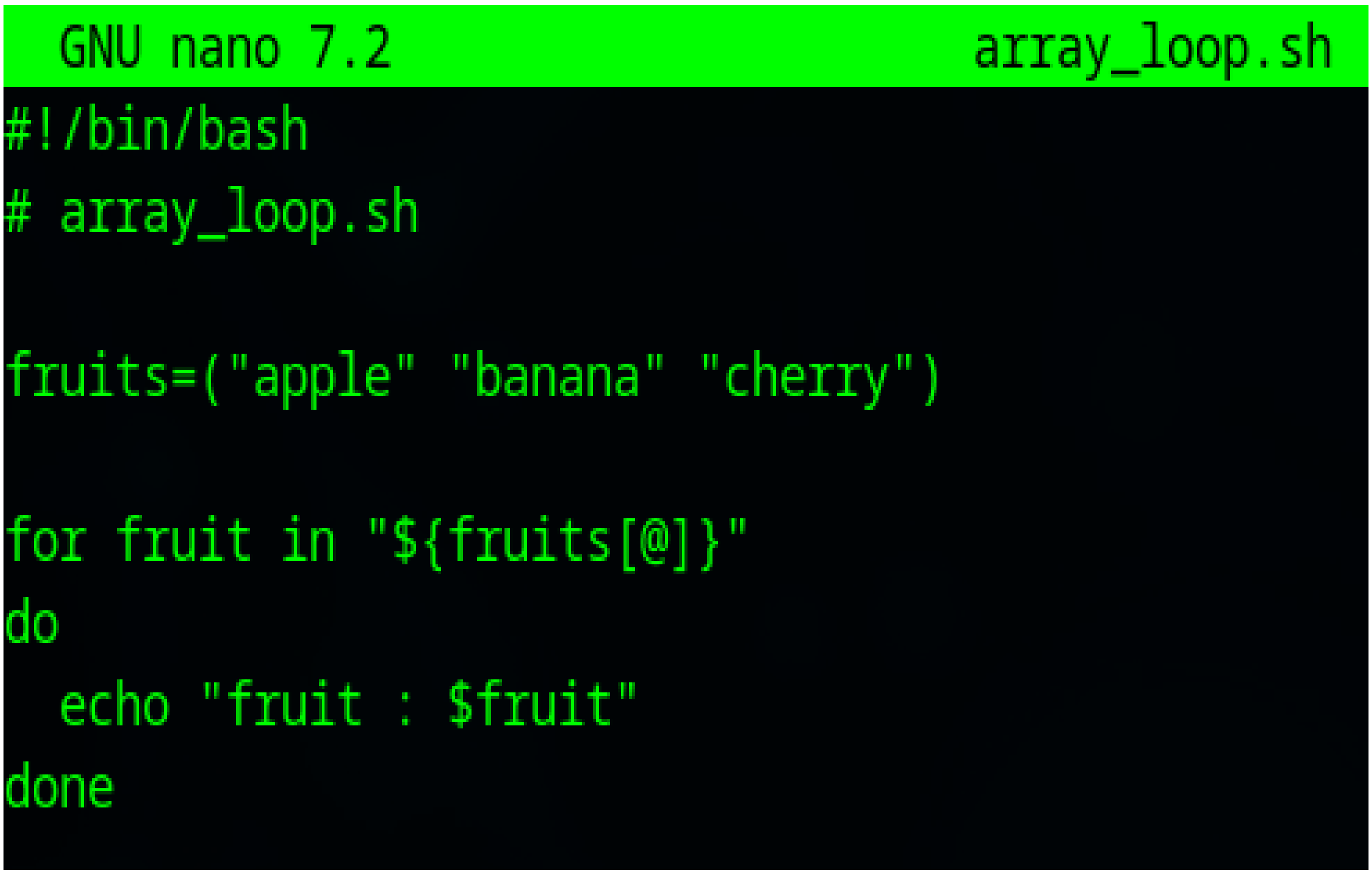
▶️ Example Run

Command:

```
./array_loop.sh
```

Output:

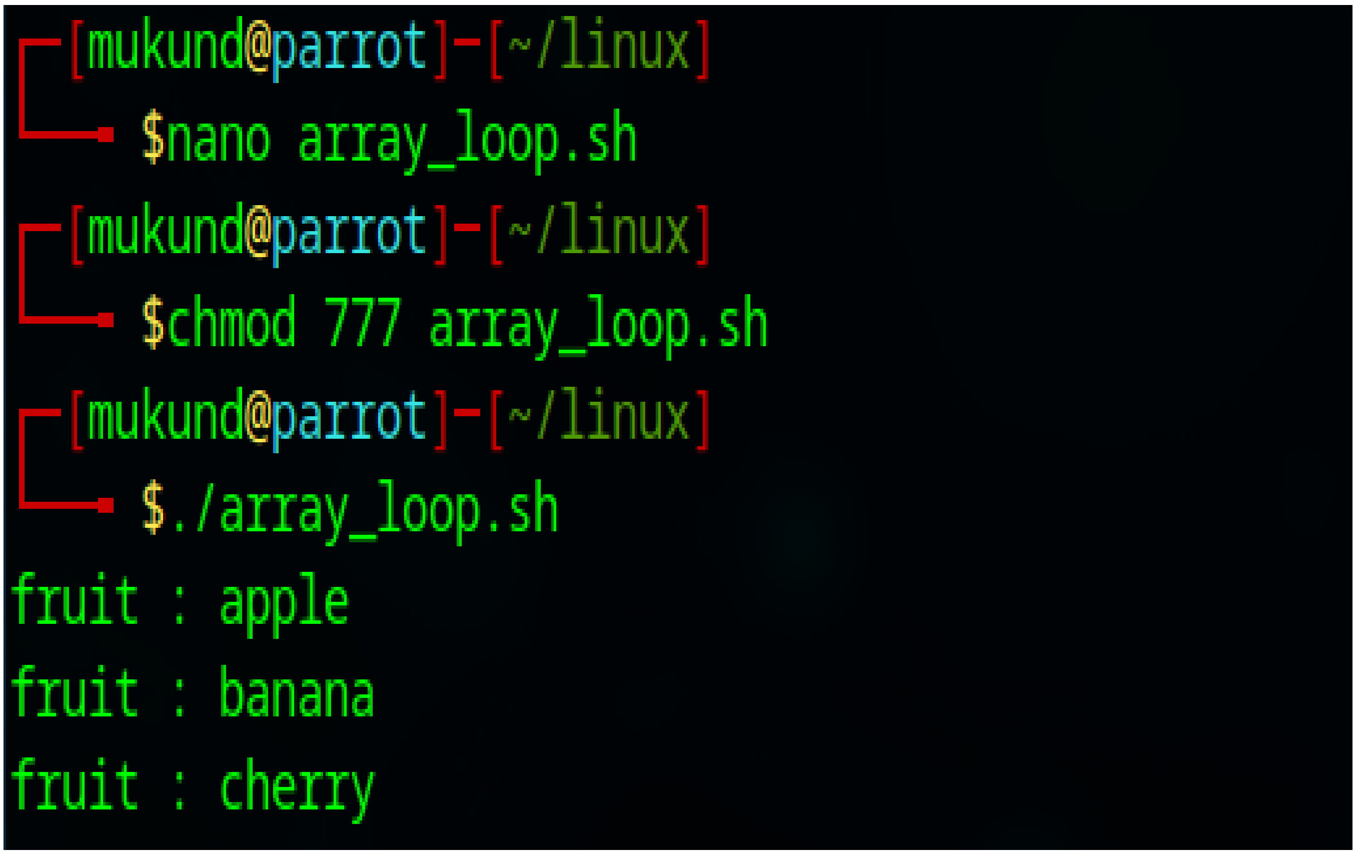
```
Fruit: apple
Fruit: banana
Fruit: cherry
```



```
  GNU nano 7.2                          array_loop.sh
#!/bin/bash
# array_loop.sh

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

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



```
┌─[mukund@parrot]─[~/linux]
└──╼ $nano array_loop.sh
┌─[mukund@parrot]─[~/linux]
└──╼ $chmod 777 array_loop.sh
┌─[mukund@parrot]─[~/linux]
└──╼ $./array_loop.sh
fruit : apple
fruit : banana
fruit : cherry
```

```
./scriptname.sh
```

❓ Extra Questions

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

It is called a **shebang**. It tells the system which interpreter to use for the script (in this case, **Bash**).
Without it, the script might not run properly or could use the wrong shell.

Q2: How do you make a script executable?

Use the `chmod` command to give it execute permissions:

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

Then run it with:

```
./scriptname.sh
```