



Shell | Bash  
Scripting

# Key Points

For this series, I will be using below configurations -

**OS** - Ubuntu OS 22.04

**Editors/IDEs** - Vi Editor

**Notes -**

- To create python program use below command, remember extension for **python program** is **.py** –

**vi script\_name.py**

- To create Shell | Bash script use below command, remember extension for Shell script could be different like **Bash, Zsh, Csh, Ksh**, and more but for **Bash script** is **.bash** –

**vi script\_name.bash**

- Remember to run these program, execute permission should be there (if you are using Linux). So to give permissions use below command –

**chmod 755 script\_name**

**OR to only give execute permissions**

**chmod +x script\_name**



Problem Statement :- Print “Hello World” message



## Shell

### Program



/tmp/print\_msg.sh

```
echo "Hello World"
```

```
printf "Hello World"
```

### Command Line



/tmp

```
mahesh@maheshm-ubuntu:/tmp$ chmod 755 print_msg.sh
```

```
mahesh@maheshm-ubuntu:/tmp$ ./print_msg.sh
```

```
Hello World
```

```
Hello Worldmahesh@maheshm-ubuntu:/tmp$
```

## Python

### Program



/tmp/print\_msg.py

```
Print("Hello World")
```

### Command Line



/tmp

```
mahesh@maheshm-ubuntu:/tmp$ chmod 755 print_msg.py
```

```
mahesh@maheshm-ubuntu:/tmp$ python3 print_msg.py
```

```
Hello World
```

```
mahesh@maheshm-ubuntu:/tmp$
```

## - Special Characters -

All special characters mentioned below are supported in Python and Shell | Bash.

(Note – When you will use these special characters in bash/shell script then the syntax would a bit different for echo command and printf command)

1) For New Line - **\n**

Used to print a newline.

2) For Tab - **\t**

Used in representing certain whitespace characters.

3) For Carriage return - **\r**

\r will just work as you have shifted your cursor to the beginning of the string or line.

Problem Statement :- Print below message using Special Characters

Description :-

**\n** - For New Line

**\t** - For Tab



## Bash

### Program



/tmp/print\_msg.sh

```
#!/bin/bash  
  
printf "Hello there,"  
  
printf "\nHow are you\t?"  
  
printf "\n\nI'm good :)"
```

### Command Line



/tmp

```
maresh@mareshm-ubuntu:/tmp$ chmod 755 print_msg.sh  
  
maresh@mareshm-ubuntu:/tmp$ ./print_msg.sh  
Hello there,  
How are you?  
  
I'm good :)  
mareshm-ubuntu:/tmp$
```

## Python

### Program



/tmp/print\_msg.py

```
print("Hello there,")  
  
print("How are you\t?")  
  
print("\nI'm good :)")
```

### Command Line



/tmp

```
maresh@mareshm-ubuntu:/tmp$ chmod 755 print_msg.py  
  
maresh@mareshm-ubuntu:/tmp$ python3 print_msg.py  
Hello there,  
How are you?  
  
I'm good :)  
maresh@mareshm-ubuntu:/tmp$
```

## - Variables -

- A variable is a named storage location that holds a value or a set of values.
- Variables are used to store data that can be accessed and manipulated by the script.
- Data could be of different types, such as **numbers, strings, lists, array, or more complex objects**.
- A variable as a container that stores a value and can be referenced by its name throughout your program.
- In Python, variables are dynamically typed, which means you don't need to explicitly declare the type of a variable when creating it.
- In Bash, variables are dynamically typed. This means that you don't need to explicitly declare the type of a variable when creating it, and the variable can hold values of different types throughout its lifetime.
- For example,

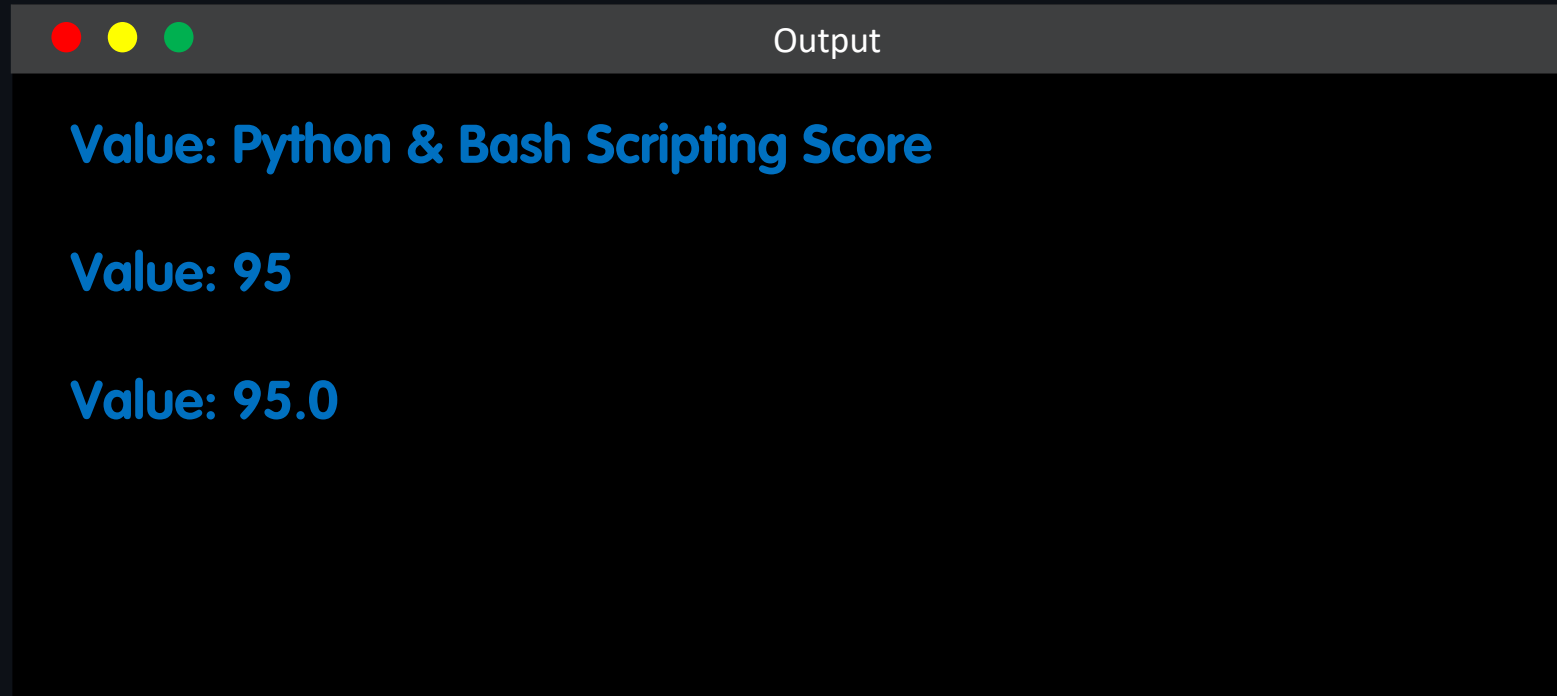
Value="Hello, world!"

Value=50

Value=50.0



Problem Statement :- Declare variables and print values of those variables



```
Value: Python & Bash Scripting Score  
Value: 95  
Value: 95.0
```

## Bash

### Program



/tmp/variables.sh

```
#!/bin/bash
Value="Python & Bash Scripting Score"
printf "Value: ${Value}\n\n"
Value=95
printf "Value: ${Value}\n\n"
Value=95.0
printf "Value: ${Value}\n\n"
```

### Command Line



/tmp

```
mahesh@maheshm-ubuntu:/tmp$ chmod 755 variables.sh

mahesh@maheshm-ubuntu:/tmp$ ./variables.sh
Value: Python & Bash Scripting Score

Value: 95

Value: 95.0
maheshm-ubuntu:/tmp$
```

## Python

### Program



/tmp/variables.py

```
Value="Python & Bash Scripting Score"
print("Value: {}\n".format(Value))

Value=95
print("Value: {}\n".format(Value))

Value=95.0
print("Value: {}".format(Value))
```

### Command Line



/tmp

```
mahesh@maheshm-ubuntu:/tmp$ chmod 755 variables.py

mahesh@maheshm-ubuntu:/tmp$ python3 variables.py
Value: Python & Bash Scripting Score

Value: 95

Value: 95.0
mahesh@maheshm-ubuntu:/tmp$
```

## - User Input -

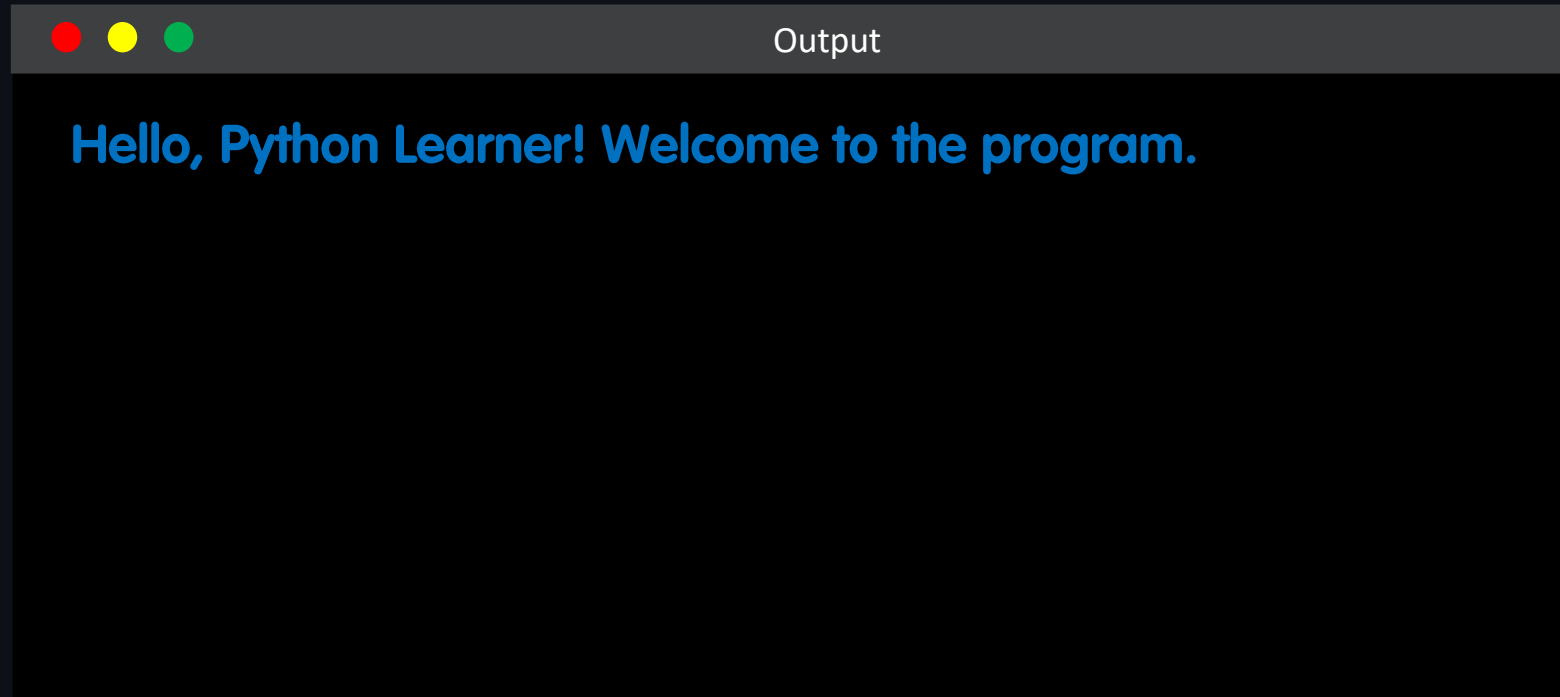
### Bash

- To take input from user, we use read function in Bash. Use below lines of code,  
`echo "Enter your name:"`  
`read name`
- To take input from user with message on same line use below lines of code,  
`echo -n "Enter your name: "`  
`read name`

### Python

- To take input from user, we use input() function in Python. Use below line to read input from user,  
`name = input("Enter your name: ")`

Problem Statement :- Take input from User and print message



## Bash

### Program



/tmp/take\_user\_input.sh

```
#!/bin/bash  
  
echo -n "Enter your name: "  
  
read name  
  
printf "\nHello, ${name}! Welcome to the program.\n\n"
```

### Command Line



/tmp

```
mahesh@maheshm-ubuntu:/tmp$ chmod 755 take_user_input.sh  
  
mahesh@maheshm-ubuntu:/tmp$ ./take_user_input.sh  
Enter your name: Bash Scripting Learner  
  
Hello, Bash Scripting Learner! Welcome to the program.  
  
maheshm-ubuntu:/tmp$
```

## Python

### Program



/tmp/take\_user\_input.py

```
name = input("Enter your name: ")  
  
print("\nHello, " + name + "! Welcome to the program.\n")
```

### Command Line



/tmp

```
mahesh@maheshm-ubuntu:/tmp$ chmod 755 take_user_input.py  
  
mahesh@maheshm-ubuntu:/tmp$ python3 take_user_input.py  
Enter your name: Python Learner  
  
Hello, Python Learner! Welcome to the program.  
  
mahesh@maheshm-ubuntu:/tmp$
```

## - Comments -

### Bash

- **Single-line comments** -

You can use the # symbol to indicate a single-line comment. Anything after the # symbol on the same line will be treated as a comment.

For e.g.

```
# Take Input from user
```

- **Multi-line comments** -

Bash does not have built-in support for multi-line comments, but you can achieve the same effect by enclosing your comments in a block of code that is not executed.

One common approach is to use a :, which is a null command, as the starting and ending lines of the comment block.

For e.g.

```
:'
```

```
This is a multi-line comment.
```

```
You can write multiple lines of comments here.
```

```
This block will not be executed by the interpreter.
```

```
'
```

## - Comments -

### Python

- **Single-line comments** -

You can use the # symbol to indicate a single-line comment. Anything after the # symbol on the same line will be treated as a comment.

For e.g.

```
# Take Input from user
```

- **Multi-line comments** -

Python does not have a built-in syntax for multi-line comments like some other programming languages. But you can achieve the same effect by using triple quotes (""" or ''') to create a multi-line string.

For e.g.

```
"""This is a multi-line comment.
```

```
You can write multiple lines of comments here."""
```

**OR**

```
'''This is a multi-line comment.
```

```
You can write multiple lines of comments here.'''
```

## Bash

### Program



/tmp/add\_comments.sh

```
#!/bin/bash

# Taking input from the user

: 'This is a multi-line comment.

You can write multiple lines of comments here.'

printf "\nHow to comment line or lines of code in Bash Script?\n\n"
```

### Command Line



/tmp

```
maresh@mareshm-ubuntu:/tmp$ chmod 755 add_comments.sh

maresh@mareshm-ubuntu:/tmp$ ./add_comments.sh
How to comment line or lines of code in Bash Script?

mareshm-ubuntu:/tmp$
```

## Python

### Program



/tmp/add\_comments.py

```
# Taking input from the user

"""This is a multi-line comment.

You can write multiple lines of comments here."""

print("\nHow to comment line or lines of code in Python?\n")
```

### Command Line



/tmp

```
maresh@mareshm-ubuntu:/tmp$ chmod 755 add_comments.py

maresh@mareshm-ubuntu:/tmp$ python3 add_comments.py
How to comment line or lines of code in Python?

maresh@mareshm-ubuntu:/tmp$
```





If you like this post and find this post helpful, then please share.



Like



Comment



Save



<https://www.linkedin.com/in/themr255/>

Reach me at : [maheshrm255@gmail.com](mailto:maheshrm255@gmail.com)

GitHub : <https://github.com/themr255>

TWS Community Builder : <https://www.linkedin.com/company/trainwithshubham/>

TWS YouTube Channel: <https://www.youtube.com/@TrainWithShubham>