

Linux BASH Scripting

contents :

This section is about BASH Scripting for Beginners.

BASH

Bash is a Unix shell and command language written by Brian Fox for the GNU Project as a free software replacement for the Bourne shell. **BASH** stands for **B**ourne **A**gain **S**hell .

BASH uses for Writing Software installation Script for Unix Based Operating System . Bash is a command processor that typically runs in a text window, where the user types commands that cause actions . Bash can also read and execute commands from a file, called a script.

How to write BASH Script

We need text editor “ Sublime Text ” or “ gedit ” . If we don't have any text editor , we can use only terminal . Linux distros have come with ‘nano’ text editor default. ‘Nano’ has no graphical view. It is used from terminal .

Here, I have shown how to write script by gedit / Atom / terminal .

If your Linux machine hasn't gedit install gedit .

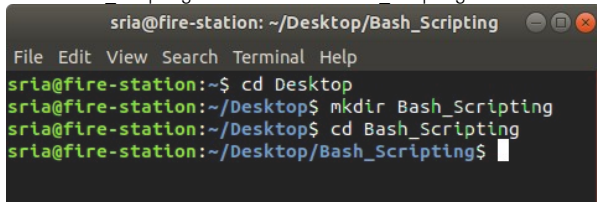
To install gedit open terminal by “ **ctrl + alt + t** ” from your keyboard . Then type “ **sudo apt-get install gedit** ”

Or For installing Atom : click [Here](#) to download .deb file then install by ubuntu software manager or terminal .

The previous section we have seen about some Linux Commands. We will use some of them here.

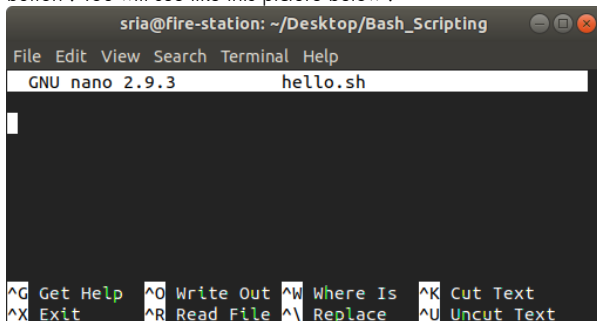
Print a String

By terminal : Open your terminal. Create a folder where you want to put your script files . I have chosen Desktop and create a folder name Bash_Scripting . Go inside the Bash_Scripting Folder .

A terminal window titled 'sria@fire-station: ~/Desktop/Bash_Scripting'. The menu bar includes File, Edit, View, Search, Terminal, and Help. The command history shows: 'sria@fire-station:~\$ cd Desktop', 'sria@fire-station:~/Desktop\$ mkdir Bash_Scripting', 'sria@fire-station:~/Desktop\$ cd Bash_Scripting', and 'sria@fire-station:~/Desktop/Bash_Scripting\$' with a cursor at the end.

```
sria@fire-station: ~/Desktop/Bash_Scripting
File Edit View Search Terminal Help
sria@fire-station:~$ cd Desktop
sria@fire-station:~/Desktop$ mkdir Bash_Scripting
sria@fire-station:~/Desktop$ cd Bash_Scripting
sria@fire-station:~/Desktop/Bash_Scripting$
```

Now type " nano hello.sh " . we are going to create a script file name hello.sh (.sh is extension of bash script file) then hit enter button . You will see like this picture below .

A terminal window titled 'sria@fire-station: ~/Desktop/Bash_Scripting' showing the nano text editor. The title bar says 'GNU nano 2.9.3' and the filename is 'hello.sh'. The editor area is empty with a cursor at the top left. At the bottom, there is a status bar with keyboard shortcuts: ^G Get Help, ^O Write Out, ^W Where Is, ^K Cut Text, ^X Exit, ^R Read File, ^\ Replace, and ^U Uncut Text.

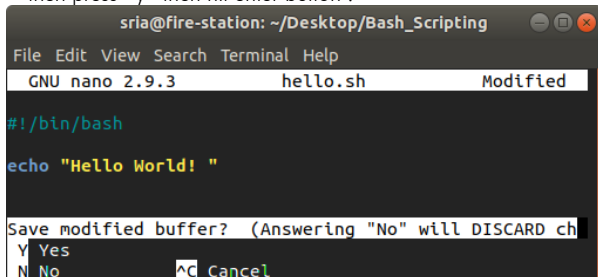
```
sria@fire-station: ~/Desktop/Bash_Scripting
File Edit View Search Terminal Help
GNU nano 2.9.3 hello.sh
^G Get Help ^O Write Out ^W Where Is ^K Cut Text
^X Exit ^R Read File ^\ Replace ^U Uncut Text
```

Now we start our journey to write script.

```
#!/bin/bash
```

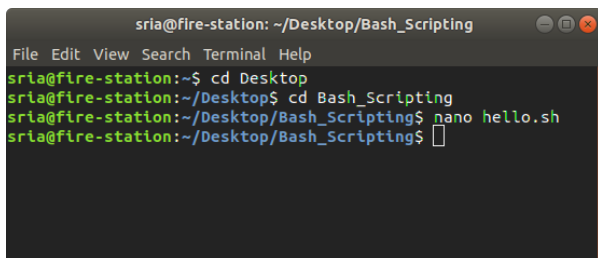
echo "Hello World ! "echo is = print . It helps to print String. **Bash** is case sensitive .

So you should beware of this matter. After writing , press " ctrl + c " then press ' y ' then hit enter button .



```
sria@fire-station: ~/Desktop/Bash_Scripting
File Edit View Search Terminal Help
GNU nano 2.9.3 hello.sh Modified
#!/bin/bash
echo "Hello World! "
Save modified buffer? (Answering "No" will DISCARD changes)
Y Yes
N No ^C Cancel
```

Now you are in your default terminal



```
sria@fire-station: ~/Desktop/Bash_Scripting
File Edit View Search Terminal Help
sria@fire-station:~$ cd Desktop
sria@fire-station:~/Desktop$ cd Bash_Scripting
sria@fire-station:~/Desktop/Bash_Scripting$ nano hello.sh
sria@fire-station:~/Desktop/Bash_Scripting$
```

This time we are going to execute the script . For executing the script file " hello.sh " need execute permission . To give permission type " chmod +x hello.sh " then hit enter

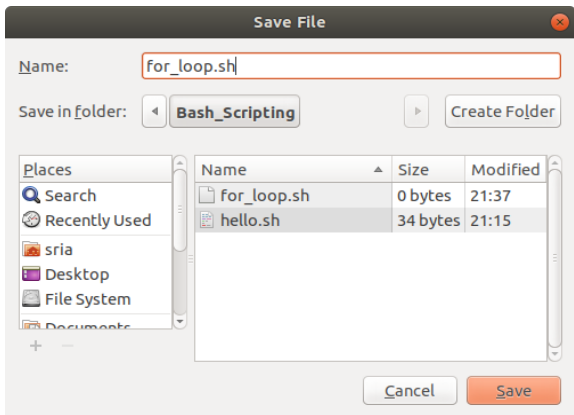
now We are able to execute the script. Type " ./hello.sh " you will see like that

```
sria@fire-station: ~/Desktop/Bash_Scripting
File Edit View Search Terminal Help
sria@fire-station:~$ cd Desktop
sria@fire-station:~/Desktop/Bash_Scripting$ chmod +x hello.sh
sria@fire-station:~/Desktop/Bash_Scripting$ ./hello.sh
Hello World!
sria@fire-station:~/Desktop/Bash_Scripting$
```

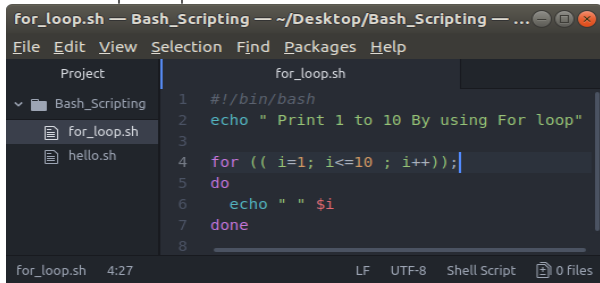
From we try another option .

This time we write script by using a graphical text editor and use terminal only for executing the script.

I have tried cover – For loop, If / Else , Switch case and how to write function and call a function .Open gedit and create a new file . I have used Atom here . Save file as for_loop.sh

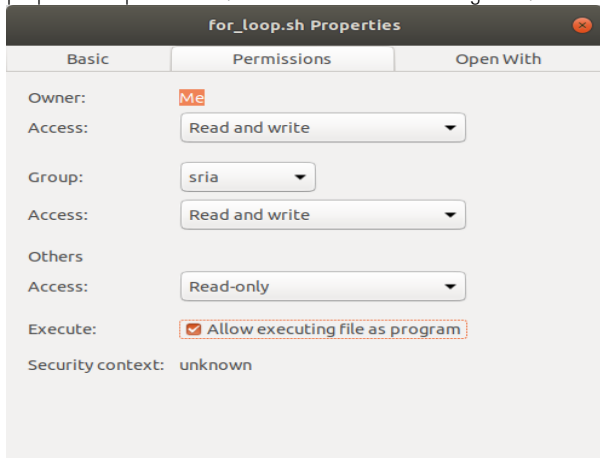


write the script and press `ctrl + s` to save file .



```
for_loop.sh — Bash_Scripting — ~/Desktop/Bash_Scripting — ...  
File Edit View Selection Find Packages Help  
Project  
  Bash_Scripting  
    for_loop.sh  
    hello.sh  
1  #!/bin/bash  
2  echo " Print 1 to 10 By using For loop"  
3  
4  for (( i=1; i<=10 ; i++));  
5  do  
6      echo " " $i  
7  done  
8  
for_loop.sh 4:27 LF UTF-8 Shell Script 0 files
```

Go to the folder where you put your script files and give permission for executing. I have kept my files in `/Desktop/Bash_Scripting` . Go to `for_loop.sh` and click right button of mouse , you will find properties → permission . And Tick to Allow executing files .



for_loop.sh Properties		
Basic	Permissions	Open With
Owner:	Me	
Access:	Read and write	
Group:	sria	
Access:	Read and write	
Others		
Access:	Read-only	
Execute:	<input checked="" type="checkbox"/> Allow executing file as program	
Security context:	unknown	

Open your terminal by clicking right button of mouse above the folder and below you will find open in terminal.

Or , you can go that folder using "cd Desktop/Bash_Scripting " then " ./for_loop.sh "

```
sria@fire-station: ~/Desktop/Bash_Scripting
File Edit View Search Terminal Help
sria@fire-station:~/Desktop/Bash_Scripting$ ./for_loop.sh
Print 1 to 10 By using For loop
1
2
3
4
5
6
7
8
9
10
sria@fire-station:~/Desktop/Bash_Scripting$
```

Here is our output of written script of for_loop.sh .

This is the way to write and execute bash script. I have added more example below .

Standard Input Output

```
input_output.sh — ~/Desktop/Bash_Scripting — Atom
File Edit View Selection Find Packages Help
Project
  Bash_Scripting
    for_loop.sh
    hello.sh
    input_output.sh
input_output.sh
1  #!/bin/bash
2
3  echo "What is Your Name ?"
4  read name;
5
6  echo " Your name is $name"
7
```

read helps you take input from keyboard. I take an input in "name" variable and call the variable by using \$ with variable name. I call the " name " variable by using \$name .

Output :

```
sria@fire-station: ~/Desktop/Bash_Scripting
File Edit View Search Terminal Help
sria@fire-station:~/Desktop/Bash_Scripting$ ./input_output.sh
What is Your Name ?
jakaria
Your name is jakaria
sria@fire-station:~/Desktop/Bash_Scripting$
```

In the second Line I give my name as input .

IF / ELSE

```
if_else.sh — ~/Desktop/Bash_Scripting — Atom
File Edit View Selection Find Packages Help
Project
  Bash_Scripting
    for_loop.sh
    hello.sh
    if_else.sh
    input_output.s
if_else.sh
1  #!/bin/bash
2  echo "Have a nice day ? "
3  read input;
4  if [[ $input == yes ]]; then
5      echo "Good Morning, Sir"
6  else
7      echo "I wish You have a nice day."
8  fi
9
if_else.sh 7:37 LF UTF-8 Shell Script 0 files
```

if condition is started by if and finished by fi (reverse of if).

Output of if / else

if input == yes then

2nd shot

[illegible]3rd shot

```
calculator.sh — ~/Others/Desktop/Bash_Scripting — Atom
File Edit View Selection Find Packages Help

32     sub|sub| )
33     sub= `expr $Number1 - $Number2`
34     echo "The Subtraction is : "$subt
35
36 ;;
37 Multiplication|Mul|mul )
38     m= `expr $Number1 \* $Number2`
39     echo "The Multiplication is : "$m
40 ;;
41 Div|div|/ )
42     d= `expr $Number1 / $Number2`
43     echo "The Division is : "$d
44 ;;
45 * )
46     echo "You Enter a wrong input . Try again !!"
47 ;;
48
49 esac
50
51 b= `date`
52
53 ex= `expr a - b`
54 echo " Execution Time : " $ex

~/Others/Desktop/Bash_Scripting/calculator.sh 50:1
LF UTF-8 Shell Sx 0 files
```

```
sria@fire-station: ~/Desktop/Bash_Scripting
File Edit View Search Terminal Help
Today is 'Thu Jun 14 22:40:40 +06 2018'
-----
Simple Calculator By using Linux bash Scripting
-----
Enter First Number :
30
Enter Second Number :
3
-----
Operation For                                     Write Command
-----
Addition                                           'Add' or 'add' or '+'
Substraction                                       'Sub' or 'sub' or '-'
Multiplication                                     'Mul' or 'mul'
Division                                           'Div' or 'div' or '/'
-----
Enter Your Command :
mul
The Multiplication is : 90
expr: non-integer argument
Execution Time :
sria@fire-station:~/Desktop/Bash_Scripting$
```

Above the picture , you have seen a simple calculator using Linux bash Scripting .

These stuffs only for beginners .

You should Try :

1. Write a bash Script which you give input as day and output a Year – month – day .
2. Find Leap year by giving input a year.
3. Using If else find the greatest or smallest number by giving input two numbers .

You will find more problems in google.

NB. Bash Script takes everything as a String .