OS LAB 2

MUHAMMAD ABDULLAH | 221546 | BSCYS-IVA

Step 1:

First, create an empty file by name **hello.sh**

```
┌──(kali㉿kali)-[~/Documents]
└─$ touch hello.sh

┌──(kali㉿kali)-[~/Documents]
└─$ ls
hello.sh
```

Step 2:

Then open the file in the file editor using the **nano** command

```
┌──(kali㉿kali)-[~/Documents]
└─$ sudo nano hello.sh
[sudo] password for kali:
```

Step 3:

Now first write in the **shebang** and then type in the command as follow.

```
File  Actions  Edit  View  Help
  GNU nano 7.2
#!/bin/bash
echo "Hello Maam"
```

Step 4:

Now use the command **bash <file-name>** to run the file.

```
┌──(kali㉿kali)-[~/Documents]
└─$ bash hello.sh
Hello Maam
```

Step 5:

Now to create a variable you have name the variable and just simply assign a value as shown following.

```
File  Actions  Edit  View  Help
  GNU nano 7.2
#!/bin/bash
echo "Hello Maam"
var="Its raining"
echo $var
```

Step 6:

The command **echo $0 $1 $2** in Bash is used to display the values of the special shell parameters:

$0: This represents the name of the script or shell itself.

$1: This represents the first command-line argument passed to the script or function.

$2: Similarly, this represents the second command-line argument.

```
  GNU nano 7.2
#!/bin/bash
echo "Hello Maam"
var="Its raining"
echo $var
echo $0 $1 $2
```

Step 7:

Now to trigger it type the command in the following way.

```
┌──(kali㉿kali)-[~/Documents]
└─$ bash hello.sh Abdullah
Hello Maam
Its raining
hello.sh Abdullah

┌──(kali㉿kali)-[~/Documents]
└─$ bash hello.sh Muhammad Abdullah
Hello Maam
Its raining
hello.sh Muhammad Abdullah
```

Step 8:

Now in order to form the array use the following method and write the following code.

```
  GNU nano 7.2
#!/bin/bash
echo "Hello Maam"
var="Its raining"
echo $var
arg=($@)
echo "What you want to buy: "
echo "1: " $1
echo "2: " $2
echo "Your list"
echo ${arg[0]}
echo ${arg[1]}
```

Step 9:

No to trigger it write the command **bash hello.sh mango banana**

```
┌──(kali㉿kali)-[~/Documents]
└─$ bash hello.sh Mango Banana
Hello Maam
Its raining
What you want to buy:
1:  Mango
2:  Banana
Your list
Mango
Banana
```

Step 10:

To take in input make the following modifications in the cod. Use the command **read** to take inputs.

```
┌──(kali㉿kali)-[~/Documents]
└─$ bash hello.sh
Hello Maam
What you want to buy:
1: Mango
2: Banana
Your list
Mango
Banana
```

```
┌──(kali㉿kali)-[~/Documents]
└─$ bash hello.sh
Hello Maam
What you want to buy:
1:
2:
```

Step 11:

To hide the input use **–sp**

Step 12:

To form an array use the following code

```
  GNU nano 7.2
#!/bin/bash

echo "Hello Maam"
var="It's raining"

declare -a array

 read -p "Enter value at index: " num
 array[0]=$num
echo "Element: ${array[0]}"
```

```
┌──(kali㉿kali)-[~/Documents]
└─$ bash hello.sh
Hello Maam
Enter value at index: Hehe
Element: Hehe
```

Step 13:

Conditional statement

```
  GNU nano 7.2
#!/bin/bash

echo "Hello Maam"
var="It's raining"
echo $var

read -p "Enter value: " num

if [ "$num" -eq 100 ]
then
  echo "Equals to 100"
else
  echo "Not Equals to 100"
fi
```

```
┌──(kali㉿kali)-[~/Documents]
└─$ bash hello.sh
Hello Maam
It's raining
Enter value: 4
Not Equals to 100

┌──(kali㉿kali)-[~/Documents]
└─$ bash hello.sh
Hello Maam
It's raining
Enter value: 100
Equals to 100
```

Step 14:

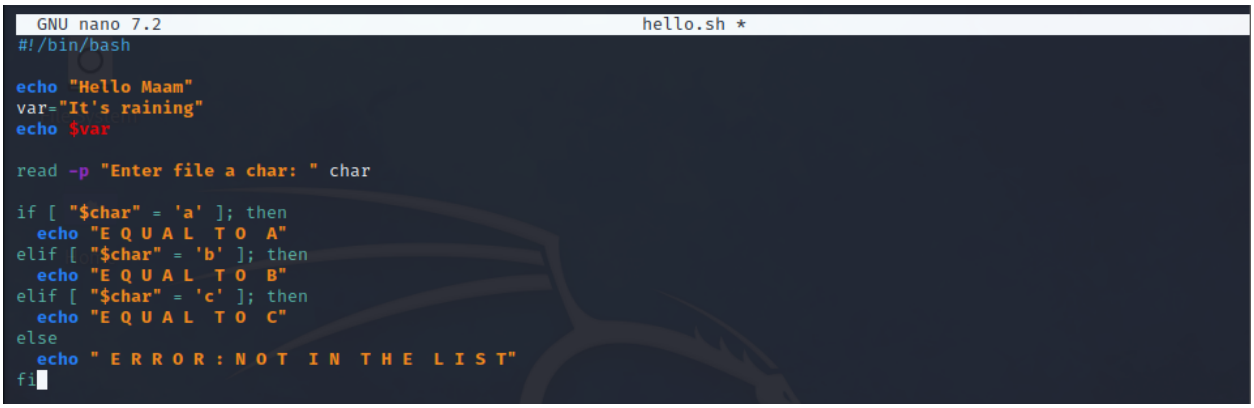
File finder

```
  GNU nano 7.2
#!/bin/bash

echo "Hello Maam"
var="It's raining"
echo $var

read -p "Enter file name: " file

if [ -e "$file" ]; then
  echo "F O U N D"
else
  echo "N O T  F O U N D"
fi
```

```
┌──(kali㉿kali)-[~/Documents]
└─$ bash hello.sh
Hello Maam
It's raining
Enter file name: hello.sh
F O U N D

┌──(kali㉿kali)-[~/Documents]
└─$ bash hello.sh
Hello Maam
It's raining
Enter file name: hello.txt
N O T  F O U N D
```

LAB TASK

Now use a code that takes a char input of a character and then checks whether it is A, B or C and then prints a message accordingly.



```
 GNU nano 7.2                                    hello.sh *
#!/bin/bash

echo "Hello Maam"
var="It's raining"
echo $var

read -p "Enter file a char: " char

if [ "$char" = 'a' ]; then
  echo "E Q U A L  T O  A"
elif [ "$char" = 'b' ]; then
  echo "E Q U A L  T O  B"
elif [ "$char" = 'c' ]; then
  echo "E Q U A L  T O  C"
else
  echo " E R R O R : N O T  I N  T H E  L I S T"
fi
```

```
┌──(kali㉿kali)-[~/Documents]
└─$ bash hello.sh
Hello Maam
It's raining
Enter file a char: a
E Q U A L  T O  A

┌──(kali㉿kali)-[~/Documents]
└─$ bash hello.sh
Hello Maam
It's raining
Enter file a char: b
E Q U A L  T O  B

┌──(kali㉿kali)-[~/Documents]
└─$ bash hello.sh
Hello Maam
It's raining
Enter file a char: c
E Q U A L  T O  C

┌──(kali㉿kali)-[~/Documents]
└─$ bash hello.sh
Hello Maam
It's raining
Enter file a char: d
 E R R O R : N O T  I N  T H E  L I S T
```

****