

BASH PROJECTS

1. BASH CASE EXAMPLE 1:

```
juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch bash_case.sh  
juhi@05d861eca28c5c4:~$ chmod 777 bash_case.sh  
juhi@05d861eca28c5c4:~$ gedit bash_case.sh  
MESA: error: ZINK: failed to choose pdev  
glx: failed to create drisw screen  
juhi@05d861eca28c5c4:~$ ./bash_case.sh  
Do you know Java Programming?  
Yes/No? :Yes  
That's amazing.  
  
juhi@05d861eca28c5c4:~$ ./bash_case.sh  
Do you know Java Programming?  
Yes/No? :No  
It's easy. Let's start learning from javatpoint.  
juhi@05d861eca28c5c4:~$ _
```

```
bash_case.sh  
~  
1 #!/bin/bash  
2  
3 echo "Do you know Java Programming?"  
4 read -p "Yes/No? :" Answer  
5 case $Answer in  
6     Yes|yes|y|Y)  
7         echo "That's amazing."  
8         echo  
9         ;;  
10    No|no|N|n)  
11        echo "It's easy. Let's start learning from javatpoint."  
12        ;;  
13 esac
```

2. BASH CASE EXAMPLE 2:

```
juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch example2_bashCash.sh  
juhi@05d861eca28c5c4:~$ chmod 777 example2_bashCash.sh  
juhi@05d861eca28c5c4:~$ gedit example2_bashCash.sh  
juhi@05d861eca28c5c4:~$ ./example2_bashCash.sh  
Which Operating System are you using?  
Windows, Android, Chrome, Linux, Others?  
Type your OS Name:windows  
That's common. You should try something new.  
  
juhi@05d861eca28c5c4:~$ ./example2_bashCash.sh  
Which Operating System are you using?  
Windows, Android, Chrome, Linux, Others?  
Type your OS Name:Android  
This is my favorite. It has lots of applications.  
  
juhi@05d861eca28c5c4:~$ ./example2_bashCash.sh  
Which Operating System are you using?  
Windows, Android, Chrome, Linux, Others?  
Type your OS Name:Chrome  
Cool!!! It's for pro users. Amazing Choice.  
  
juhi@05d861eca28c5c4:~$ ./example2_bashCash.sh  
Which Operating System are you using?  
Windows, Android, Chrome, Linux, Others?  
Type your OS Name:Linux  
You might be serious about security!!  
  
juhi@05d861eca28c5c4:~$ ./example2_bashCash.sh  
Which Operating System are you using?  
Windows, Android, Chrome, Linux, Others?  
Type your OS Name:Others  
Sounds interesting. I will try that.  
  
juhi@05d861eca28c5c4:~$ _
```

```
Open [v] [+]  
*example2_bashCash.sh  
~  
1 #!/bin/bash  
2  
3 echo "Which Operating System are you using?"  
4 echo "Windows, Android, Chrome, Linux, Others?"  
5 read -p "Type your OS Name:" OS  
6  
7 case $OS in  
8     Windows|windows)  
9         echo "That's common. You should try something new."  
10        echo  
11        ;;  
12     Android|android)  
13        echo "This is my favorite. It has lots of applications."  
14        echo  
15        ;;  
16     Chrome|chrome)  
17        echo "Cool!!! It's for pro users. Amazing Choice."  
18        echo  
19        ;;  
20     Linux|linux)  
21        echo "You might be serious about security!!"  
22        echo  
23        ;;  
24     *)  
25        echo "Sounds interesting. I will try that."  
26        echo  
27        ;;  
28 esac
```

3. For loop (EXAMPLE 1)

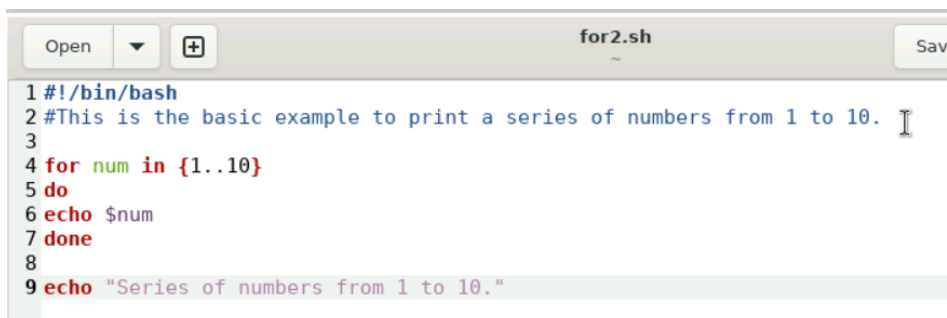
```
juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch for_example1.sh  
juhi@05d861eca28c5c4:~$ chmod 777 for_example1.sh  
juhi@05d861eca28c5c4:~$ gedit for_example1.sh  
juhi@05d861eca28c5c4:~$ ./for_example1.sh  
Start  
learning  
from  
Javatpoint.  
Thank You.  
juhi@05d861eca28c5c4:~$ _
```

```
Open ▼ + for_  
1 #!/bin/bash  
2 #This is the basic example of 'for loop'.  
3  
4 learn="Start learning from Javatpoint."  
5  
6 for learn in $learn  
7 do  
8 echo $learn  
9 done  
10  
11 echo "Thank You."
```

4. For loop example2:

juhi@05d861eca28c5c4: ~

```
juhi@05d861eca28c5c4:~$ touch for2.sh
juhi@05d861eca28c5c4:~$ chmod 777 for2.sh
Command 'chmod' not found, did you mean:
  command 'chmod' from deb coreutils (9.4-2ubuntu2)
Try: sudo apt install <deb name>
juhi@05d861eca28c5c4:~$ chmod 777 for2.sh
juhi@05d861eca28c5c4:~$ gedit for2.sh
MESA: error: ZINK: failed to choose pdev
glx: failed to create drisw screen
juhi@05d861eca28c5c4:~$ ./for2.sh
1
2
3
4
5
6
7
8
9
10
Series of numbers from 1 to 10.
juhi@05d861eca28c5c4:~$
```



The screenshot shows a text editor window titled "for2.sh" with a toolbar containing "Open", a dropdown arrow, a "+" icon, and "Save". The editor contains the following code:

```
1#!/bin/bash
2#This is the basic example to print a series of numbers from 1 to 10.
3
4for num in {1..10}
5do
6echo $num
7done
8
9echo "Series of numbers from 1 to 10."
```

5. For INCREMENT:

```
juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch for3.sh  
juhi@05d861eca28c5c4:~$ chnod 777 for3.sh  
Command 'chnod' not found, did you mean:  
  command 'chmod' from deb coreutils (9.4-2ubuntu2)  
Try: sudo apt install <deb name>  
juhi@05d861eca28c5c4:~$ chmod 777 for3.sh  
juhi@05d861eca28c5c4:~$ gedit for3.sh  
juhi@05d861eca28c5c4:~$ ./for3.sh  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
juhi@05d861eca28c5c4:~$
```

```
1 #!/bin/bash  
2  
3 #For Loop to Read a Range with Increment  
4  
5 for num in {1..10..1}  
6 do  
7 echo $num  
8 done
```

6. FOR DECREMENT:

```
juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch for4.sh  
juhi@05d861eca28c5c4:~$ chmod 777 for4.sh  
juhi@05d861eca28c5c4:~$ gedit for4.sh  
juhi@05d861eca28c5c4:~$ ./for4.sh  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
0  
juhi@05d861eca28c5c4:~$ _
```

```
1 #!/bin/bash  
2  
3 #For Loop to Read a Range with Decrement  
4  
5 for num in {10..0..1}  
6 do  
7 echo $num  
8 done
```

7. FOR LOOP FOR READ ARRAY VARIABLES


```
juhi@05d861eca28c5c4:~$ touch for_array.sh  
juhi@05d861eca28c5c4:~$ chmod 777 for_array.sh  
juhi@05d861eca28c5c4:~$ gedit for_array.sh  
^C  
juhi@05d861eca28c5c4:~$ ./for_array.sh  
WelcometoJavatpoint  
juhi@05d861eca28c5c4:~$ _
```

```
1 #!/bin/bash
2
3 #Array Declaration
4 arr=( "Welcome""to""Javatpoint" )
5
6 for i in "${arr[@]}"
7 do
8 echo $i
9 done
```

8. For Loop to Read white spaces in String as word separators

 juhi@05d861eca28c5c4: ~

```
juhi@05d861eca28c5c4:~$ touch 8_for.sh
juhi@05d861eca28c5c4:~$ chmod 777 8_for.sh
juhi@05d861eca28c5c4:~$ gedit 8_for.sh
juhi@05d861eca28c5c4:~$ ./8_for.sh
Let's
start
learning
from
Javatpoint.
juhi@05d861eca28c5c4:~$
```

Open  8_for.sh

```
1 #!/bin/bash
2 #For Loop to Read white spaces in String as word separators
3
4 str="Let's start
5 learning from Javatpoint."
6
7 for i in $str;
8 do
9 echo "$i"
10 done
```

9. For Loop to Read each line in String as a word

```
juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch 9_for.sh  
juhi@05d861eca28c5c4:~$ chmod 777 9_for.sh  
juhi@05d861eca28c5c4:~$ gedit 9_for.sh  
juhi@05d861eca28c5c4:~$ ./9_for.sh  
let's start  
learning from  
javatpoint.  
juhi@05d861eca28c5c4:~$
```

```
1 #!/bin/bash  
2 #For Loop to Read each line in String as a word  
3  
4 str="Let's start  
5 learning from  
6 Javatpoint."  
7  
8 for i in "$str";  
9 do  
10 echo "$i"  
11 done
```

10.For Loop to Read Three-expression

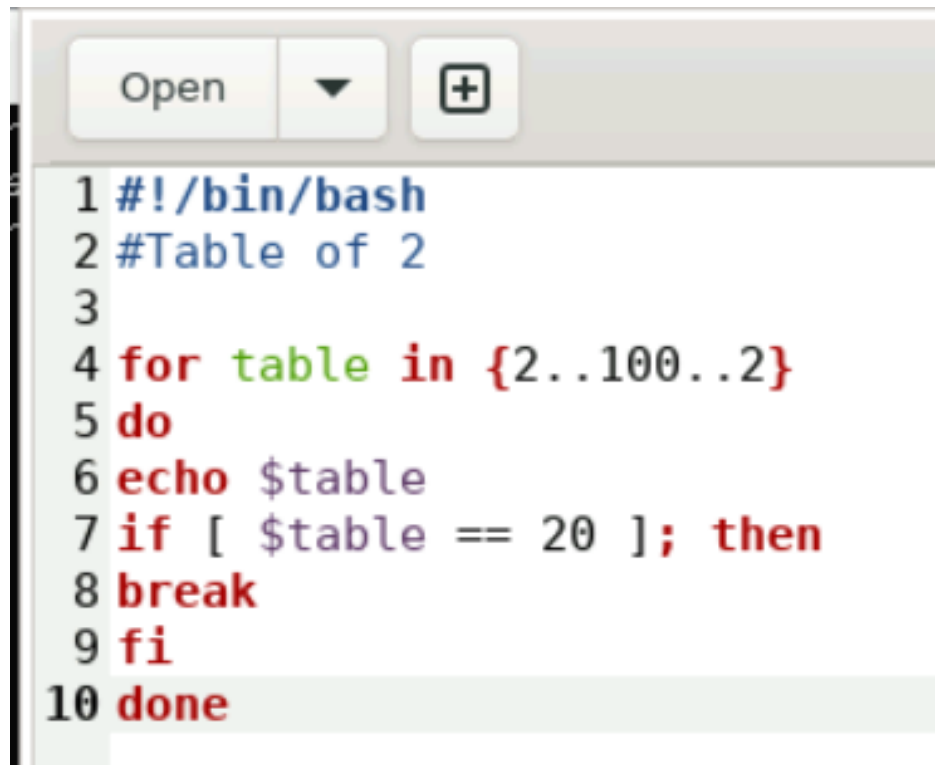
```
juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch 10_for.sh  
juhi@05d861eca28c5c4:~$ chmod 777 10_for.sh  
juhi@05d861eca28c5c4:~$ gedit 10_for.sh  
juhi@05d861eca28c5c4:~$ ./10_for.sh  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
juhi@05d861eca28c5c4:~$
```



```
Open ▼ +
1 #!/bin/bash
2 #For Loop to Read Three-expression
3
4 for ((i=1; i<=10; i++))
5 do
6 echo "$i"
7 done
```

11.For Loop with a Break Statement

```
juhi@05d861eca28c5c4: ~
juhi@05d861eca28c5c4:~$ touch for_break.sh
juhi@05d861eca28c5c4:~$ chmod 777 for_break.sh
juhi@05d861eca28c5c4:~$ gedit for_break.sh
juhi@05d861eca28c5c4:~$ ./for_break.sh
2
4
6
8
10
12
14
16
18
20
juhi@05d861eca28c5c4:~$
```

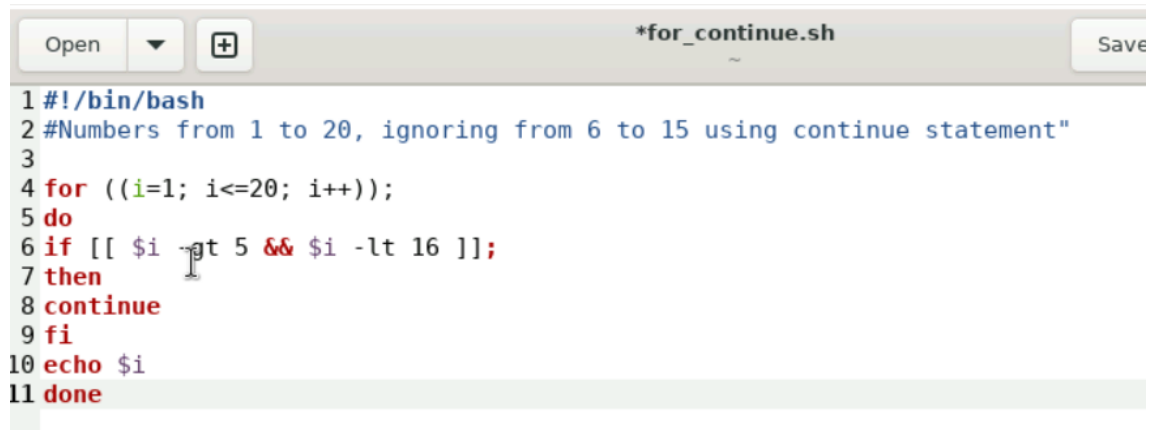


```
1 #!/bin/bash
2 #Table of 2
3
4 for table in {2..100..2}
5 do
6 echo $table
7 if [ $table == 20 ]; then
8 break
9 fi
10 done
```

12.For Loop with a Continue Statement

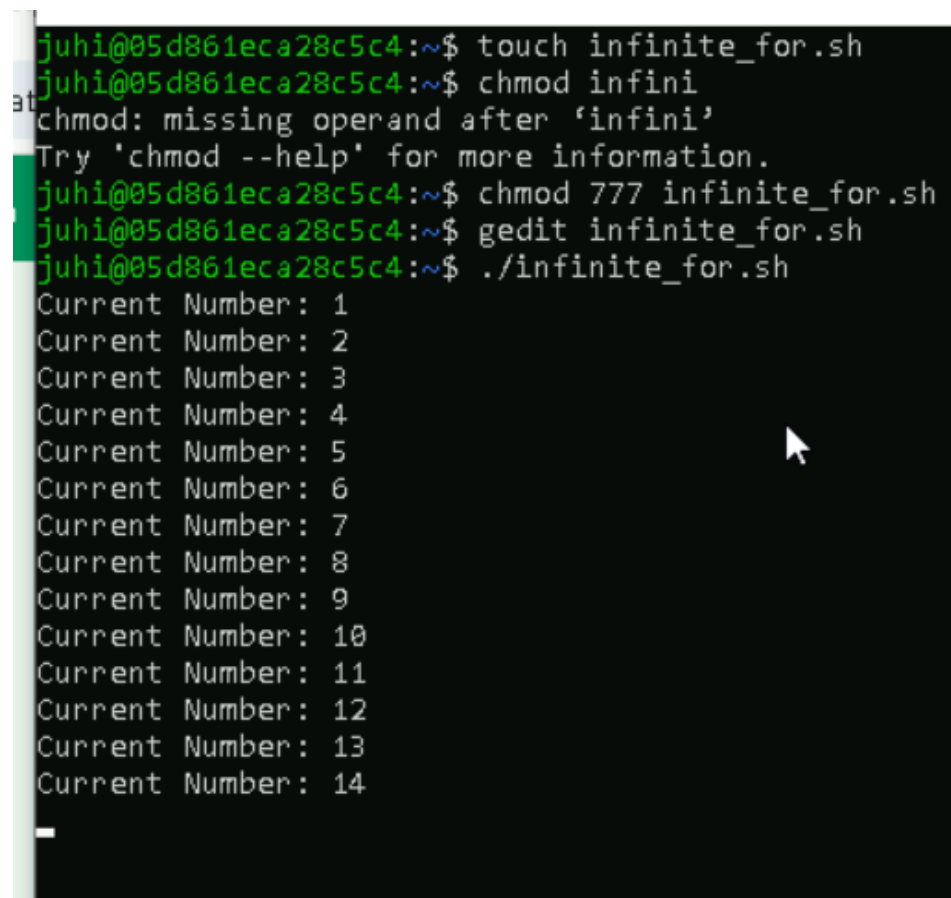


```
juhi@05d861eca28c5c4: ~
juhi@05d861eca28c5c4:~$ touch for_continue.sh
juhi@05d861eca28c5c4:~$ chmod 777 for_continue.sh
juhi@05d861eca28c5c4:~$ gedit for_continue.sh
juhi@05d861eca28c5c4:~$ ./for_continue.sh
1
2
3
4
5
16
17
18
19
20
juhi@05d861eca28c5c4:~$
```



```
1#!/bin/bash
2#Numbers from 1 to 20, ignoring from 6 to 15 using continue statement"
3
4for ((i=1; i<=20; i++));
5do
6if [[ $i -gt 5 && $i -lt 16 ]];
7then
8continue
9fi
10echo $i
11done
```

13.Infinite Bash For Loop



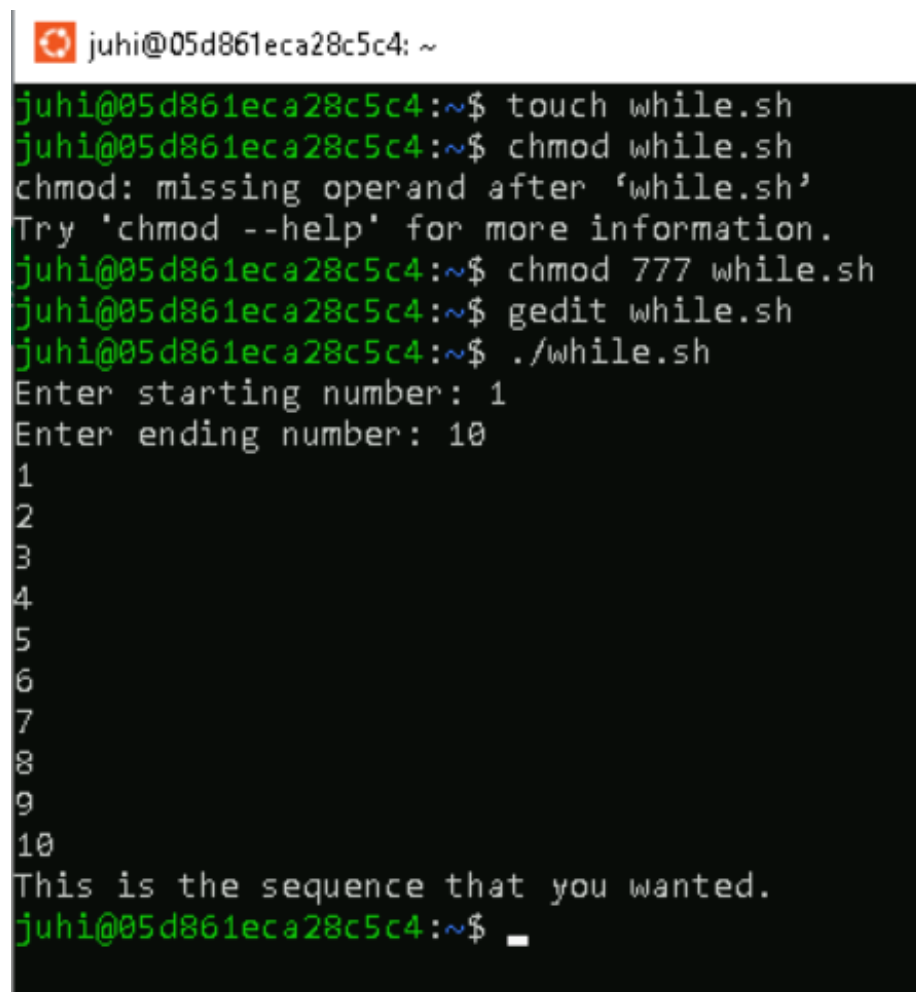
```
juhi@05d861eca28c5c4:~$ touch infinite_for.sh
juhi@05d861eca28c5c4:~$ chmod infini
chmod: missing operand after 'infini'
Try 'chmod --help' for more information.
juhi@05d861eca28c5c4:~$ chmod 777 infinite_for.sh
juhi@05d861eca28c5c4:~$ gedit infinite_for.sh
juhi@05d861eca28c5c4:~$ ./infinite_for.sh
Current Number: 1
Current Number: 2
Current Number: 3
Current Number: 4
Current Number: 5
Current Number: 6
Current Number: 7
Current Number: 8
Current Number: 9
Current Number: 10
Current Number: 11
Current Number: 12
Current Number: 13
Current Number: 14
```



The screenshot shows a code editor window with a toolbar at the top containing an 'Open' button, a dropdown arrow, and a '+' icon. The editor contains a bash script with the following lines:

```
1 #!/bin/bash
2
3 i=1;
4 for (( ; ; ))
5 do
6 sleep 1s
7 echo "Current Number: $((i++))"
8 done
```

14. WHILE LOOP



The screenshot shows a terminal window with the following commands and output:

```
juhi@05d861eca28c5c4: ~
juhi@05d861eca28c5c4:~$ touch while.sh
juhi@05d861eca28c5c4:~$ chmod while.sh
chmod: missing operand after 'while.sh'
Try 'chmod --help' for more information.
juhi@05d861eca28c5c4:~$ chmod 777 while.sh
juhi@05d861eca28c5c4:~$ gedit while.sh
juhi@05d861eca28c5c4:~$ ./while.sh
Enter starting number: 1
Enter ending number: 10
1
2
3
4
5
6
7
8
9
10
This is the sequence that you wanted.
juhi@05d861eca28c5c4:~$
```

```
1 #!/bin/bash
2 #Script to get specified numbers
3
4 read -p "Enter starting number: " snum
5 read -p "Enter ending number: " enum
6
7 while [[ $snum -le $enum ]];
8 do
9 echo $snum
10 ((snum++))
11 done
12
13 echo "This is the sequence that you wanted."
```

15. MULTIPLE WHILE LOOP

```
juhi@05d861eca28c5c4: ~
juhi@05d861eca28c5c4:~$ touch multi_while.sh
juhi@05d861eca28c5c4:~$ chmod +x multi_while.sh
juhi@05d861eca28c5c4:~$ gedit multi_while.sh
juhi@05d861eca28c5c4:~$ ./multi_while.sh
Enter starting number: 1
Enter ending number: 10
1
2
3
4
5
6
7
8
9
10
This is the sequence that you wanted.
juhi@05d861eca28c5c4:~$ _
```



```
1 #!/bin/bash
2 while :
3 do
4 echo "Welcome to Bash Scripting."
5 done
```

17. WHILE LOOP WITH BREAK STATEMENT

```
juhi@05d861eca28c5c4: ~
juhi@05d861eca28c5c4:~$ touch break_while.sh
juhi@05d861eca28c5c4:~$ chmod +x break_while.sh
juhi@05d861eca28c5c4:~$ gedit break_while.sh
juhi@05d861eca28c5c4:~$ ./break_while.sh
Countdown for Website Launching...
10
9
8
7
6
5
4
3
Mission Aborted, Some Technical Error Found.
juhi@05d861eca28c5c4:~$
```

```
1 #!/bin/bash
2 #While Loop Example with a Break Statement
3
4 echo "Countdown for Website Launching..."
5 i=10
6 while [ $i -ge 1 ]
7 do
8 if [ $i == 2 ]
9 then
10     echo "Mission Aborted, Some Technical Error Found."
11     break
12 fi
13 echo "$i"
14 (( i-- ))
15 done
```

18. WHILE LOOP WITH CONTINUE STATEMENT

```
juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch while_cont.sh  
juhi@05d861eca28c5c4:~$ chmod +x while_cont.sh  
juhi@05d861eca28c5c4:~$ gedit while_cont.sh  
juhi@05d861eca28c5c4:~$ ./while_cont.sh  
Current Number : 1  
Current Number : 2  
Current Number : 3  
Current Number : 4  
Current Number : 6  
Current Number : 7  
Current Number : 8  
Current Number : 9  
Current Number : 10  
Current Number : 11  
Skipped number 5 using Continue Statement.  
juhi@05d861eca28c5c4:~$
```

```
1 #!/bin/bash  
2 #While Loop Example with a Continue Statement  
3  
4 i=0  
5 while [ $i -le 10 ]  
6 do  
7 ((i++))  
8 if [[ "$i" == 5 ]];  
9 then  
10     continue  
11 fi  
12 echo "Current Number : $i"  
13 done  
14  
15 echo "Skipped number 5 using Continue Statement."
```


19. WHILE LOOP WITH C-STYLE

```
Select juhi@05d861eca28c5c4: ~
juhi@05d861eca28c5c4:~$ touch while_Cstyle.sh
juhi@05d861eca28c5c4:~$ chmod +x while_Cstyle.sh
juhi@05d861eca28c5c4:~$ gedit while_Cstyle.sh
juhi@05d861eca28c5c4:~$ ./while_Cstyle.sh
1
2
3
4
5
6
7
8
9
10
juhi@05d861eca28c5c4:~$ _
```

```
Open ▼ +
1 #!/bin/bash
2 i=1
3 while((i <= 10))
4 do
5 echo $i
6 let i++
7 done
```

20. UNTIL LOOP

```
juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch until.sh  
juhi@05d861eca28c5c4:~$ chmod +x until.sh  
juhi@05d861eca28c5c4:~$ gedit until.sh  
juhi@05d861eca28c5c4:~$ ./until.sh  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
juhi@05d861eca28c5c4:~$
```

```
Open ▼ + until.s  
1 #!/bin/bash  
2 #Bash Until Loop example with a single condition  
3  
4 i=1  
5 until [ $i -gt 10 ]  
6 do  
7 echo $i  
8 ((i++))  
9 done
```

21. MULTIPLE UNTIL BASH PROGRAM

```
juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch multiple_until.sh  
juhi@05d861eca28c5c4:~$ chmod multiple_until.sh  
chmod: missing operand after 'multiple_until.sh'  
Try 'chmod --help' for more information.  
juhi@05d861eca28c5c4:~$ chmod +x multiple_until.sh  
juhi@05d861eca28c5c4:~$ gedit multiple_until.sh  
^C  
juhi@05d861eca28c5c4:~$ ./multiple_until.sh  
a = 1 & b = 0.  
a = 2 & b = 1.  
a = 3 & b = 2.  
a = 4 & b = 3.  
a = 5 & b = 4.  
juhi@05d861eca28c5c4:~$ _
```

```
Open ▼ + multiple_unt  
1 #!/bin/bash  
2 #Bash Until Loop example with multiple conditions  
3  
4 max=5  
5 a=1  
6 b=0  
7  
8 until [[ $a -gt $max || $b -gt $max ]];  
9 do  
10 echo "a = $a & b = $b."  
11 ((a++))  
12 ((b++))  
13 done
```

22. STRING BASH (EQUAL OPERATORS):

```
juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch string_bash.sh  
juhi@05d861eca28c5c4:~$ chmod 777 string_bash.sh  
juhi@05d861eca28c5c4:~$ gedit string_bash.sh  
juhi@05d861eca28c5c4:~$ ./string_bash.sh  
Strings are not equal.  
juhi@05d861eca28c5c4:~$ _
```

```
1 #!/bin/bash  
2 #Script to check whether two strings are equal.  
3  
4 str1="WelcometoJavatpoint."  
5 str2="javatpoint"  
6  
7 if [ $str1 = $str2 ];  
8 then  
9 echo "Both the strings are equal."  
10 else  
11 echo "Strings are not equal."  
12 fi
```

23. NOT EQUAL OPERATORS:

```
juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch Not_equal.sh  
juhi@05d861eca28c5c4:~$ chmod 777 Not_equal.sh  
juhi@05d861eca28c5c4:~$ gedit Not_equal.sh  
juhi@05d861eca28c5c4:~$ ./Not_equal.sh  
Strings are not equal.  
juhi@05d861eca28c5c4:~$ _
```

```

1 #!/bin/bash
2 #Script to check whether two strings are equal.
3
4 str1="WelcometoJavatpoint."
5 str2="javatpoint"
6
7 if [ $str1 = $str2 ];
8 then
9 echo "Both the strings are equal."
10 else
11 echo "Strings are not equal."
12 fi

```

24. LESS THAN OPERATORS:

```

juhi@05d861eca28c5c4: ~
juhi@05d861eca28c5c4:~$ touch Less_Operator.sh
juhi@05d861eca28c5c4:~$ chmod 777 Less_Operator.sh
juhi@05d861eca28c5c4:~$ gedit Less_Operator.sh
juhi@05d861eca28c5c4:~$ ./Less_Operator.sh
WelcometoJavatpoint is not less then Javatpoint
juhi@05d861eca28c5c4:~$ _

```

```

1 #!/bin/sh
2
3 str1="WelcometoJavatpoint"
4 str2="Javatpoint"
5 if [ $str1 \< $str2 ];
6 then
7     echo "$str1 is less then $str2"
8 else
9     echo "$str1 is not less then $str2"
10 fi

```

25. GREATER THAN OPERATORS

juhi@05d861eca28c5c4: ~

```
juhi@05d861eca28c5c4:~$ touch gretaer_bash.sh
juhi@05d861eca28c5c4:~$ chmod 777 greater_bash.sh
chmod: cannot access 'greater_bash.sh': No such file or directory
juhi@05d861eca28c5c4:~$ chmod 777 gretaer_bash.sh
juhi@05d861eca28c5c4:~$ gedit gretaer_bash.sh
juhi@05d861eca28c5c4:~$ ./gretaer_bash.sh
WelcometoJavatpoint is greater then Javatpoint
juhi@05d861eca28c5c4:~$
```



The screenshot shows a code editor window with a toolbar at the top containing an 'Open' button, a dropdown arrow, and a '+' icon. The editor contains a shell script with the following lines:

```
1 #!/bin/sh
2
3 str1="WelcometoJavatpoint"
4 str2="Javatpoint"
5 if [ $str1 \> $str2 ];
6 then
7     echo "$str1 is greater then $str2"
8 else
9     echo "$str1 is less then $str2"
10 fi
```

26. TO CHECK IF THE STRING LENGTH IS GREATER THAN ZERO:

juhi@05d861eca28c5c4: ~

```
juhi@05d861eca28c5c4:~$ touch string_example.sh
juhi@05d861eca28c5c4:~$ chmod 777 string_example.sh
juhi@05d861eca28c5c4:~$ gedit string_example.sh
juhi@05d861eca28c5c4:~$ ./string_example.sh
String is not empty
juhi@05d861eca28c5c4:~$
```

```
1 #!/bin/sh
2
3 str="WelcometoJavatpoint"
4
5 if [ -n $str ];
6 then
7     echo "String is not empty"
8 else
9     echo "String is empty"
10 fi
```

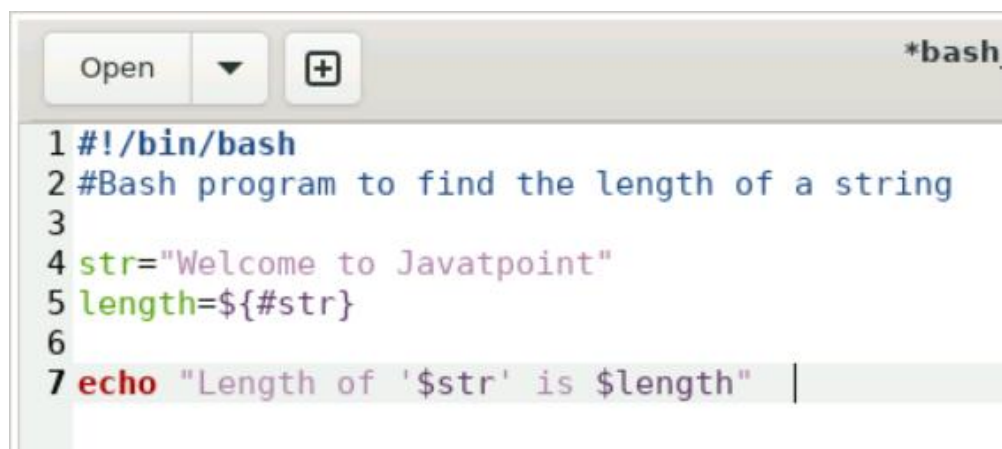
27. TO CHECK IF THE STRING LENGTH IS EQUAL TO ZERO:

```
juhi@05d861eca28c5c4: ~
juhi@05d861eca28c5c4:~$ touch string_length.sh
juhi@05d861eca28c5c4:~$ chmod 777 string_length.sh
juhi@05d861eca28c5c4:~$ geidt string_length.sh
Command 'geidt' not found, did you mean:
  command 'gedit' from snap gedit (46.1)
  command 'gedit' from deb gedit (46.1-3)
See 'snap info <snapname>' for additional versions.
juhi@05d861eca28c5c4:~$ gedit string_length.sh
juhi@05d861eca28c5c4:~$ ./string_length.sh
String is empty.
juhi@05d861eca28c5c4:~$
```

```
1 #!/bin/sh
2
3 str=""
4
5 if [ -z $str ];
6 then
7     echo "String is empty."
8 else
9     echo "String is non-empty."
10 fi
```

28. BASH FIND STRING EXAMPLE 1:

```
juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch bash_fine.sh  
juhi@05d861eca28c5c4:~$ chmod 777 bash_fine.sh  
juhi@05d861eca28c5c4:~$ gedit bash_fine.sh  
juhi@05d861eca28c5c4:~$ ./bash_file  
-bash: ./bash_file: No such file or directory  
juhi@05d861eca28c5c4:~$ ./bash_fine.sh  
Length of 'Welcome to Javatpoint' is 21  
juhi@05d861eca28c5c4:~$ _
```

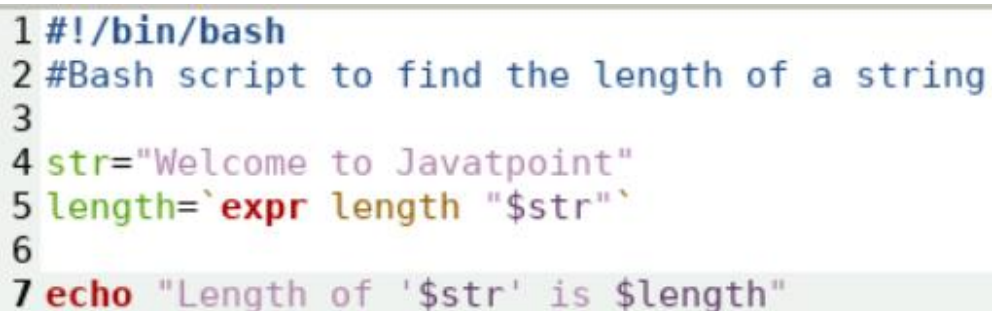


The screenshot shows a code editor window with a title bar containing 'Open', a dropdown arrow, and a '+' icon. The file name '*bash_' is visible in the top right corner. The editor contains the following code:

```
1 #!/bin/bash  
2 #Bash program to find the length of a string  
3  
4 str="Welcome to Javatpoint"  
5 length=${#str}  
6  
7 echo "Length of '$str' is $length" |
```

29. BASH FIND STRING EXAMPLE 2:


```
juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch bash_find2.sh  
juhi@05d861eca28c5c4:~$ chmod 777 bash_find2.sh  
juhi@05d861eca28c5c4:~$ gedit bash_find2.sh  
juhi@05d861eca28c5c4:~$ ./bash_find2.sh  
Length of 'Welcome to Javatpoint' is 21  
juhi@05d861eca28c5c4:~$ _
```



The screenshot shows a code editor window with the following code:

```
1 #!/bin/bash  
2 #Bash script to find the length of a string  
3  
4 str="Welcome to Javatpoint"  
5 length=`expr length "$str"`  
6  
7 echo "Length of '$str' is $length"
```



30. BASH FIND STRING EXAMPLE3:

 juhi@05d861eca28c5c4: ~

```
juhi@05d861eca28c5c4:~$ touch find3.sh
juhi@05d861eca28c5c4:~$ chmod 777 find3.sh
juhi@05d861eca28c5c4:~$ gedit find3.sh
juhi@05d861eca28c5c4:~$ ./find3.sh
Length of 'Welcome to Javatpoint' is 21
juhi@05d861eca28c5c4:~$ _
```

```
1 #!/bin/bash
2 #Bash script to find the length of a string
3
4 str="Welcome to Javatpoint"
5 length=`expr "$str" : '.*'`
6
7 echo "Length of '$str' is $length"
```

31. BASH FIND STRING EXAMPLE4:

 juhi@05d861eca28c5c4: ~

```
juhi@05d861eca28c5c4:~$ touch find4.sh
juhi@05d861eca28c5c4:~$ chmod 777 find4.sh
juhi@05d861eca28c5c4:~$ gedit find4.sh
juhi@05d861eca28c5c4:~$ ./find4.sh
Length of 'Welcome to Javatpoint' is 22
juhi@05d861eca28c5c4:~$ _
```

```
1 #!/bin/bash
2 #Bash script to find the length of a string
3
4 str="Welcome to Javatpoint"
5 length=`echo $str | wc -c`
6
7 echo "Length of '$str' is $length"
```

32. BASH FIND STRING EXAMPLE 5:

```
juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch example5.sh  
juhi@05d861eca28c5c4:~$ chmod +x example5.sh  
juhi@05d861eca28c5c4:~$ gedit example5.sh  
MESA: error: ZINK: failed to choose pdev  
glx: failed to create drisw screen  
juhi@05d861eca28c5c4:~$ ./example5.sh  
Length of 'Welcome to Javatpoint' is 21  
juhi@05d861eca28c5c4:~$ _
```

```
1 #!/bin/bash  
2 #Bash script to find the length of a string  
3  
4 str="Welcome to Javatpoint"  
5 length=`echo $str |awk '{print length}'`  
6  
7 echo "Length of '$str' is $length"
```

33. BASH SPLIT EXAMPLE 1:

```
juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch split.sh  
juhi@05d861eca28c5c4:~$ chmod 777 split.sh  
juhi@05d861eca28c5c4:~$ gedit split.sh  
juhi@05d861eca28c5c4:~$ ./split.sh  
Enter any string separated by space: I Love Programming  
I Love Programming  
juhi@05d861eca28c5c4:~$ _
```

```

1#!/bin/bash
2#Example for bash split string by space
3
4read -p "Enter any string separated by space: " str #reading string value
5
6IFS=' ' #setting space as delimiter
7read -ra ADDR <<<"$str" #reading str as an array as tokens separated by IFS
8
9for i in "${ADDR[@]}"; #accessing each element of array
10do
11echo "$i"
12done

```

34. BASH SPLIT EXAMPLE 2:

```

juhi@05d861eca28c5c4: ~
juhi@05d861eca28c5c4:~$ touch split2.sh
juhi@05d861eca28c5c4:~$ chmod +x split2.sh
juhi@05d861eca28c5c4:~$ gedit split2.sh
juhi@05d861eca28c5c4:~$ ./split2.sh
Enter Name, State and Age separated by a comma: Juhi, Bihar , 23
Name : Juhi
State : Bihar
Age : 23
juhi@05d861eca28c5c4:~$ _

```

```

1#!/bin/bash
2#Example for bash split string by Symbol (comma)
3
4read -p "Enter Name, State and Age separated by a comma: " entry #reading string value
5
6IFS=',' #setting comma as delimiter
7read -a strarr <<<"$entry" #reading str as an array as tokens separated by IFS
8
9echo "Name : ${strarr[0]} "
10echo "State : ${strarr[1]} "
11echo "Age : ${strarr[2]}"

```

35. BASH SPLIT EXAMPLE 3:

```

juhi@05d861eca28c5c4: ~
juhi@05d861eca28c5c4:~$ touch split3.sh
juhi@05d861eca28c5c4:~$ touch split3.sh
juhi@05d861eca28c5c4:~$ chmod +x split3.sh
juhi@05d861eca28c5c4:~$ gedit split3.sh
juhi@05d861eca28c5c4:~$ ./split3.sh
Enter any string separated by colon(:) We:Welcome:Guest

We
Welcome
Guest

juhi@05d861eca28c5c4:~$ _

```

```

1 #!/bin/bash
2 #Example for bash split string without $IFS
3
4 read -p "Enter any string separated by colon(:) " str #reading string value
5
6 readarray -d : -t strarr <<<"$str" #split a string based on the delimiter ':'
7
8 printf "\n"
9
10 #Print each value of Array with the help of loop
11 for (( n=0; n < ${#strarr[*]}; n++ ))
12 do
13 echo "${strarr[n]}"
14 done

```

36. BASH SPLIT EXAMPLE4:

```

❏ juhi@05d861eca28c5c4: ~
juhi@05d861eca28c5c4:~$ touch split4.sh
juhi@05d861eca28c5c4:~$ chmod +x split4.sh
juhi@05d861eca28c5c4:~$ gedit split4.sh
juhi@05d861eca28c5c4:~$ ./split4.sh
declare -a array=([0]="We" [1]="Welcome" [2]="You" [3]="On" [4]="Javatpoint")
juhi@05d861eca28c5c4:~$ _


```

```

1 #!/bin/bash
2 #Example for bash split string by another string
3
4 str="WeLearnWelcomeLearnYouLearnOnLearnJavatpoint"
5 delimiter=Learn
6 s=$str$delimiter
7 array=();
8 while [[ $s ]];
9 do
10 array+=( "${s%%"$delimiter"}" );
11 s=${s#"${delimiter}"};
12 done;
13 declare -p array

```

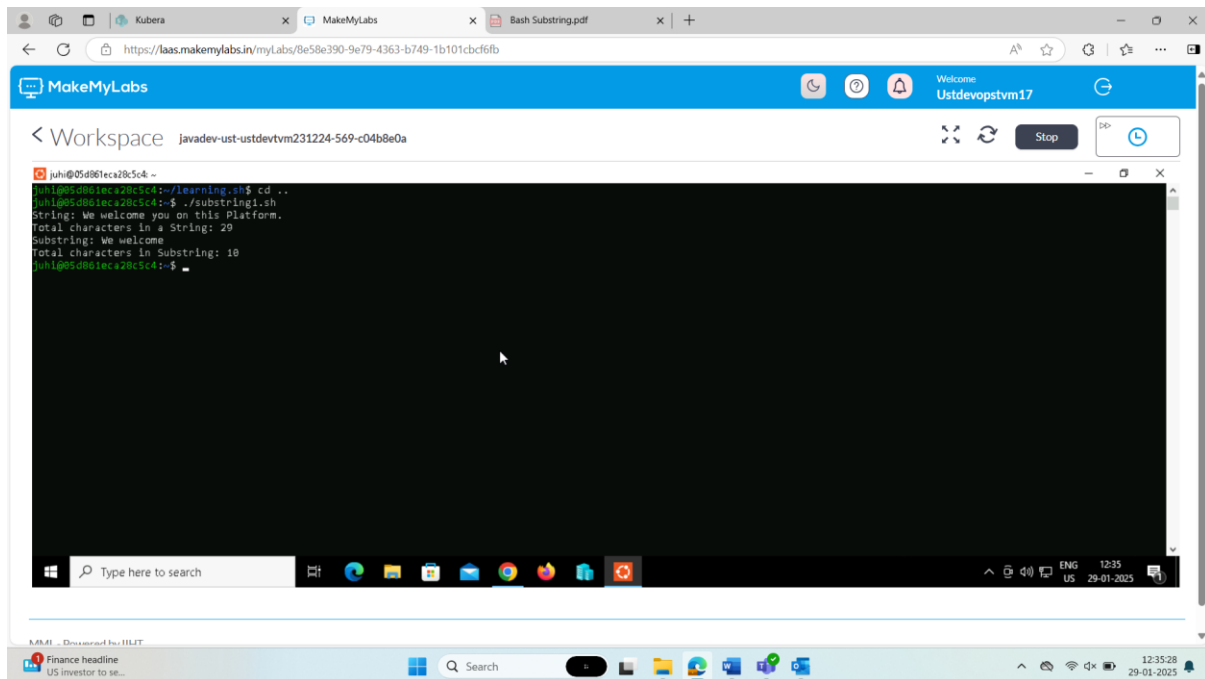
37. BASH SPLIT EXAMPLE 5:

 juhi@05d861eca28c5c4: ~

```
juhi@05d861eca28c5c4:~$ touch split5.sh
juhi@05d861eca28c5c4:~$ chmod +x split5.sh
juhi@05d861eca28c5c4:~$ gedit split5.sh
juhi@05d861eca28c5c4:~$ ./split5.sh
tr: missing operand after ';'
Two strings must be given when translating.
Try 'tr --help' for more information.
juhi@05d861eca28c5c4:~$ gedit split5.sh
juhi@05d861eca28c5c4:~$ ./split5.sh
We
welcome
you
on
javatpoint.
juhi@05d861eca28c5c4:~$ _
```

```
1 #!/bin/bash
2 # Example to split a string using tr command
3
4 my_str="We;welcome;you;on;javatpoint."
5 my_arr=($(echo $my_str | tr ';' '\n'))
6
7 for i in "${my_arr[@]}"
8 do
9     echo $i
10 done
11
```

38. BASH SUBSTRING

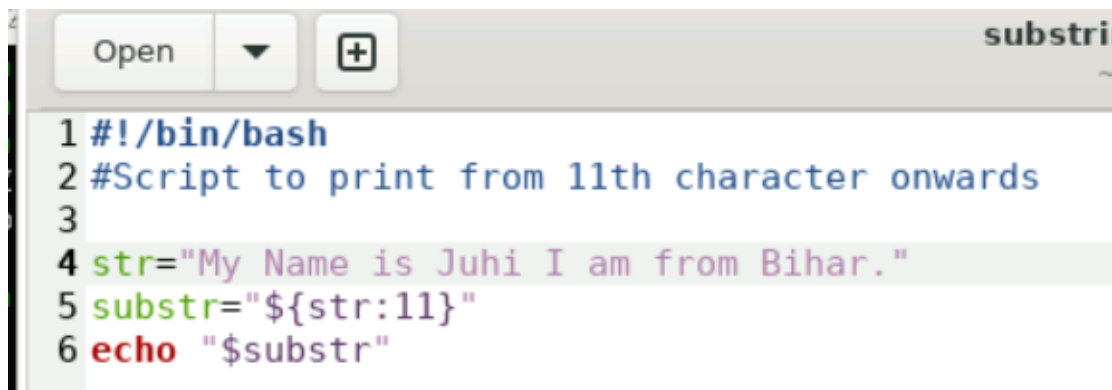


```
1 #!/bin/bash
2 #Script to extract first 10 characters of a string
3
4 echo "String: We welcome you on this Platform|"
5 str="We welcome you on Javatpoint."
6
7 echo "Total characters in a String: ${#str} "
8
9 substr="${str:0:10}"
10
11 echo "Substring: $substr"
12 echo "Total characters in Substring: ${#substr} "
```

39. TO EXTRACT FROM SPECIFIC CHARACTER ONWARD

juhi@05d861eca28c5c4: ~

```
juhi@05d861eca28c5c4:~$ touch substring2.sh
juhi@05d861eca28c5c4:~$ chmod +x substring2.sh
juhi@05d861eca28c5c4:~$ gedit substring2.sh
MESA: error: ZINK: failed to choose pdev
glx: failed to create drisw screen
^C
juhi@05d861eca28c5c4:~$ gedit substring2.sh
juhi@05d861eca28c5c4:~$ ./substring2.sh
Juhi I am from Bihar.
juhi@05d861eca28c5c4:~$ _
```



The screenshot shows a text editor window with a title bar containing 'Open', a dropdown arrow, a '+' icon, and the filename 'substring2.sh'. The editor contains the following code:

```
1 #!/bin/bash
2 #Script to print from 11th character onwards
3
4 str="My Name is Juhi I am from Bihar."
5 substr="${str:11}"
6 echo "$substr"
```

40. TO EXTRACT A STRING CHARACTER

juhi@05d861eca28c5c4: ~

```
juhi@05d861eca28c5c4:~$ touch substring3.sh
juhi@05d861eca28c5c4:~$ chmod 777 substring3.sh
juhi@05d861eca28c5c4:~$ ./substring3.sh
juhi@05d861eca28c5c4:~$ _
```

```
1 #!/bin/bash
2 #Script to print 11th character of a String
3
4 str="My name is Juhi and i am from bihar."
5 substr="${str:11:1}"
6 echo "$substr"
```

41. EXTRACT THE SPECIFIC CHARACTERS FROM LAST

```
juhi@05d861eca28c5c4: ~
juhi@05d861eca28c5c4:~$ touch substring4.sh
juhi@05d861eca28c5c4:~$ chmod +x substring4.sh
juhi@05d861eca28c5c4:~$ gedit substring4.sh
MESA: error: ZINK: failed to choose pdev
glx: failed to create drisw screen
^C
juhi@05d861eca28c5c4:~$ gedit substring4.sh
juhi@05d861eca28c5c4:~$ ./substring4.sh
from bihar
juhi@05d861eca28c5c4:~$ _
```

```
1 #!/bin/bash
2 #Script to extract 11 characters from last
3
4 str="My name is Juhi and i am from bihar"
5 substr="${str:(-11)}"
6 echo "$substr"
```

42. CONCATENATE


```
juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch concatenate.sh  
juhi@05d861eca28c5c4:~$ chmod +x concatenate.sh  
juhi@05d861eca28c5c4:~$ gedit concatenate.sh  
juhi@05d861eca28c5c4:~$ ./concatenate.sh  
We welcome you on Javatpoint.  
juhi@05d861eca28c5c4:~$
```

```
1 #!/bin/bash  
2 #Script to Concatenate Strings  
3  
4 #Declaring the first String  
5 str1="We welcome you"  
6  
7 #Declaring the Second String  
8 str2=" on Javatpoint."  
9  
10 #Combining first and second string  
11 str3="$str1$str2"  
12  
13 #Printing a new string by combining both  
14 echo $str3
```

43. USING DOUBLE QUOTE

```
Select juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch DoubleQuote.sh  
juhi@05d861eca28c5c4:~$ chmod +x DoubleQuote.sh  
juhi@05d861eca28c5c4:~$ gedit DoubleQuote.sh  
^C  
juhi@05d861eca28c5c4:~$ gedit DoubleQuote.sh  
juhi@05d861eca28c5c4:~$ ./DoubleQuote.sh  
We welcome you on Javatpoint.  
juhi@05d861eca28c5c4:~$
```

```

1 #!/bin/bash
2 #Script to Concatenate Strings
3
4 #Declaring String Variable
5 str="We welcome you"
6
7 #Add the variable within the string
8 echo "$str on Javatpoint."

```

44. USING APPEND OPERATOR WITH LOOP

```

Select juhi@05d861eca28c5c4: ~
juhi@05d861eca28c5c4:~$ touch operatorloop.sh
juhi@05d861eca28c5c4:~$ chmod +x operatorloop.sh
juhi@05d861eca28c5c4:~$ gedit operatorloop.sh
juhi@05d861eca28c5c4:~$ ./operatorloop.sh
Printing the name of the programming languages
javapythonCC++
juhi@05d861eca28c5c4:~$

```

```

1 #!/bin/bash
2 echo "Printing the name of the programming languages"
3 #Initializing the variable before combining
4 lang=""
5 #for loop for reading the list
6 for value in 'java' 'python' 'C' 'C++';
7 do
8 lang+="$value " #Combining the list values using append operator
9 done
10 #Printing the combined values
11 echo "$lang"

```

45. USING THE PRINTF FUNCTION

juhi@05d861eca28c5c4: ~

```
juhi@05d861eca28c5c4:~$ touch function1.sh
juhi@05d861eca28c5c4:~$ chmod +x function1.sh
juhi@05d861eca28c5c4:~$ gedit function1.sh
juhi@05d861eca28c5c4:~$ ./function1.sh
Welcome to Javatpoint.
juhi@05d861eca28c5c4:~$ _
```

```
2 1 #!/bin/bash
2 2
2 3 str="Welcome"
a 4 printf -v new_str "$str to Javatpoint."
2 5 echo $new_str
2
```

USING LITERAL STRING

Select juhi@05d861eca28c5c4: ~

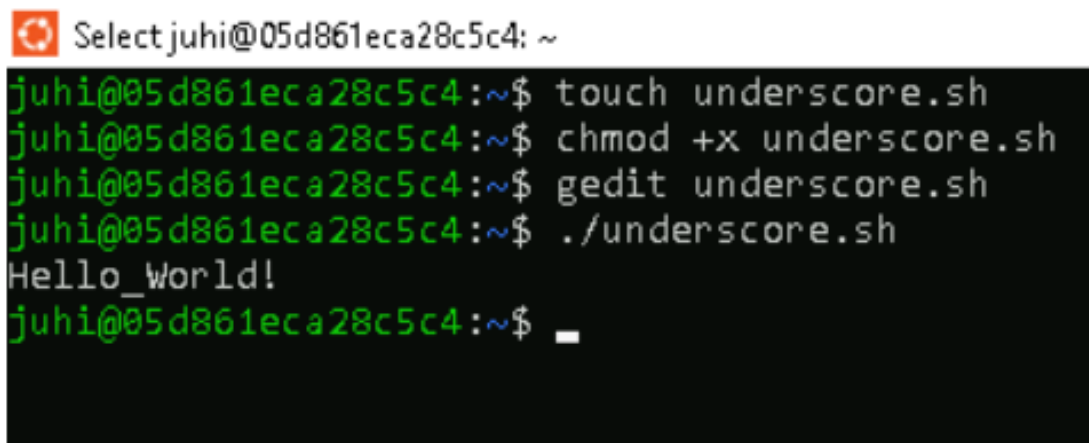
```
juhi@05d861eca28c5c4:~$ touch literal.sh
juhi@05d861eca28c5c4:~$ chmod +x literal.sh
juhi@05d861eca28c5c4:~$ gedit literal.sh
^C
juhi@05d861eca28c5c4:~$ gedit literal.sh
juhi@05d861eca28c5c4:~$ ./literal.sh
Welcome to Javatpoint.
juhi@05d861eca28c5c4:~$ _
```



A screenshot of a code editor window. At the top, there is a toolbar with an 'Open' button, a dropdown arrow, and a '+' icon. Below the toolbar, the editor displays a shell script with six lines of code, each preceded by a line number. The code is color-coded: line 1 is a shebang, line 3 is an assignment, line 5 is an assignment, and line 6 is an echo command. The script is as follows:

```
1 #!/bin/bash
2
3 str="Welcome to"
4
5 newstr="${str} Javatpoint."
6 echo "$newstr"
```

USING UNDERSCORE



A screenshot of a terminal window. The title bar shows a red icon and the text 'Select juhi@05d861eca28c5c4: ~'. The terminal content shows a series of commands and their output. The commands are: 'touch underscore.sh', 'chmod +x underscore.sh', 'gedit underscore.sh', and './underscore.sh'. The output of the last command is 'Hello_world!'. The prompt is 'juhi@05d861eca28c5c4:~\$'.

```
Select juhi@05d861eca28c5c4: ~
juhi@05d861eca28c5c4:~$ touch underscore.sh
juhi@05d861eca28c5c4:~$ chmod +x underscore.sh
juhi@05d861eca28c5c4:~$ gedit underscore.sh
juhi@05d861eca28c5c4:~$ ./underscore.sh
Hello_world!
juhi@05d861eca28c5c4:~$ _
```

```
1 #!/bin/bash
2
3 str1="Hello"
4 str2="World!"
5
6 echo "${str1}_${str2}"
```

USING ANY CHARACTER

```
Select juhi@05d861eca28c5c4: ~
juhi@05d861eca28c5c4:~$ touch examplenew.sh
juhi@05d861eca28c5c4:~$ chmod +x examplenew.sh
juhi@05d861eca28c5c4:~$ gedit examplenew.sh
juhi@05d861eca28c5c4:~$ ./examplenew.sh
Enter First Name: Juhi
Enter State: Bihar
Enter Age: 23
Name, State, Age: Juhi,Bihar,23
juhi@05d861eca28c5c4:~$ _
```

```

1 #!/bin/bash
2 #String Concatenation by Character (,) with User Input
3
4 read -p "Enter First Name: " name
5 read -p "Enter State: " state
6 read -p "Enter Age: " age
7
8 combine="$name,$state,$age"
9
10 echo "Name, State, Age: $combine"

```

FUNCTION METHOD1

```

juhi@05d861eca28c5c4: ~
juhi@05d861eca28c5c4:~$ touch method1.sh
juhi@05d861eca28c5c4:~$ chmod +x method1.sh
juhi@05d861eca28c5c4:~$ gedit method1.sh
juhi@05d861eca28c5c4:~$ ./method1.sh
Welcome to Javatpoint.
juhi@05d861eca28c5c4:~$

```

```

1 #!/bin/bash
2
3 JTP () {
4     echo 'Welcome to Javatpoint.'
5 }
6
7 JTP |


```

EXAMPLE: METHOD2

```
juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch method2.sh  
juhi@05d861eca28c5c4:~$ chmod +x method2.sh  
juhi@05d861eca28c5c4:~$ gedit method2.sh  
MESA: error: ZINK: failed to choose pdev  
glx: failed to create drisw screen  
^C  
juhi@05d861eca28c5c4:~$ gedit method2.sh  
juhi@05d861eca28c5c4:~$ ./method2.sh  
Welcome to Javatpoint.  
juhi@05d861eca28c5c4:~$ _
```

```
1 #!/bin/bash  
2  
3 function JTP {  
4     echo 'Welcome to Javatpoint.'  
5 }  
6  
7 JTP
```

PASSING ARGUMENTS

 juhi@05d861eca28c5c4: ~

```
juhi@05d861eca28c5c4:~$ touch passing_argu.sh
juhi@05d861eca28c5c4:~$ chmod +x passing_argu.sh
juhi@05d861eca28c5c4:~$ gedit passing_argu.sh
^C
juhi@05d861eca28c5c4:~$ ./passing_argu.sh
wewelcomeyouonJavatpoint.
```

```
1 #!/bin/bash
2 #Script to pass and access arguments
3
4 function_arguments()
5 {
6     echo $1
7     echo $2
8     echo $3
9     echo $4
10    echo $5
11 }
12
13 #Calling function_arguments
14 function_arguments "We""welcome""you""on""Javatpoint."
```

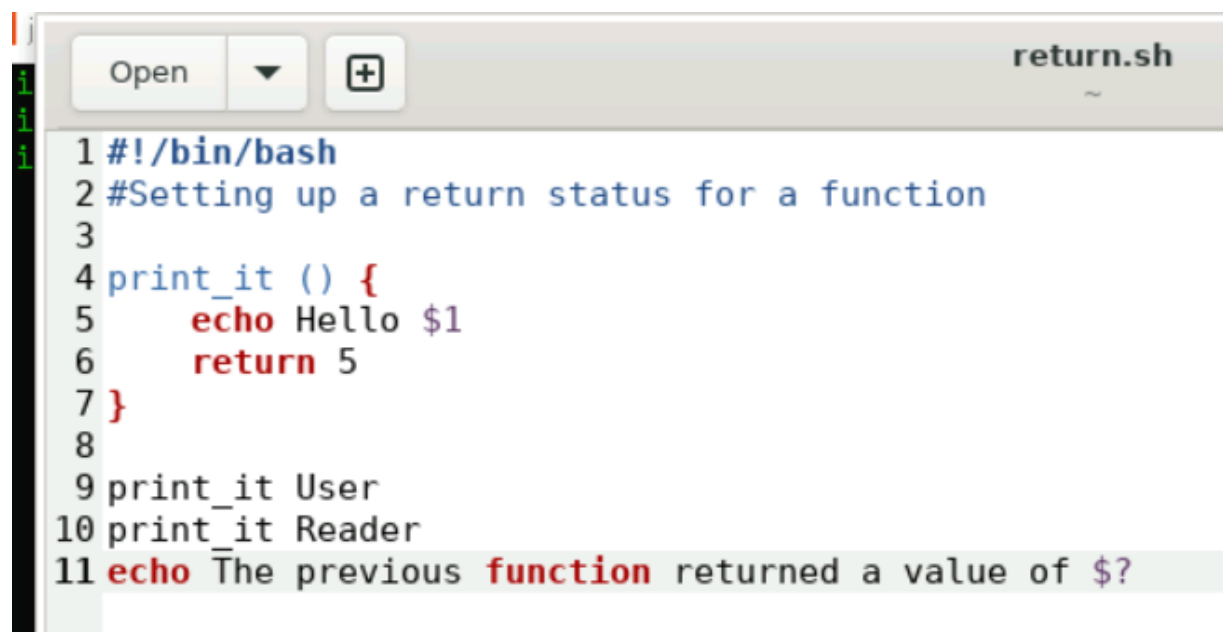

VARIABLE SCOPE

```
juhi@05d861eca28c5c4: ~  
juhi@05d861eca28c5c4:~$ touch variable.sh  
juhi@05d861eca28c5c4:~$ chmod +x variable.sh  
juhi@05d861eca28c5c4:~$ gedit variable.sh  
juhi@05d861eca28c5c4:~$ ./variable.sh  
Before Executing the Function  
v1 is A.  
v2 is B.  
Inside Function  
v1 is C.  
v2 is D.  
After Executing the Function  
v1 is A.  
v2 is D.  
juhi@05d861eca28c5c4:~$
```

```
1 #!/bin/bash  
2  
3 v1='A'  
4 v2='B'  
5  
6 my_var () {  
7     local v1='C'  
8     v2='D'  
9     echo "Inside Function"  
10    echo "v1 is $v1."  
11    echo "v2 is $v2."  
12 }  
13  
14 echo "Before Executing the Function"  
15 echo "v1 is $v1."  
16 echo "v2 is $v2."  
17  
18 my_var  
19 echo "After Executing the Function"  
20 echo "v1 is $v1."  
21 echo "v2 is $v2."
```

RETURN VALUES

```
juhi@05d861eca28c5c4:~$ touch return.sh
juhi@05d861eca28c5c4:~$ chmod +x return.sh
juhi@05d861eca28c5c4:~$ gedit return.sh
juhi@05d861eca28c5c4:~$ ./return.sh
Hello User
Hello Reader
The previous function returned a value of 5
juhi@05d861eca28c5c4:~$ _
```

A screenshot of a code editor window titled 'return.sh'. The editor has a toolbar with 'Open', a dropdown arrow, and a '+' icon. The code is as follows:

```
1#!/bin/bash
2#Setting up a return status for a function
3
4print_it () {
5    echo Hello $1
6    return 5
7}
8
9print_it User
10print_it Reader
11echo The previous function returned a value of $?
```

OVERRIDING COMMANDS

```
juhi@05d861eca28c5c4:~$ touch overriding.sh
juhi@05d861eca28c5c4:~$ chmod +x overriding.sh
juhi@05d861eca28c5c4:~$ gedit overriding.sh
juhi@05d861eca28c5c4:~$ ./overriding.sh
[01-30 05:24:02] : Welcome to Javatpoint.
juhi@05d861eca28c5c4:~$ _
```

```
1 #!/bin/bash
2 #Script to override command using function
3
4 echo () {
5     builtin echo -n `date +"[%m-%d %H:%M:%S]"` ":
6     builtin echo $1
7 }
8
9 echo "Welcome to Javatpoint." |
```

ARRAYS EXAMPLE1

```
uhi@05d861eca28c5c4:~$ touch array.sh
uhi@05d861eca28c5c4:~$ chmod +x array.sh
uhi@05d861eca28c5c4:~$ gedit array.sh
uhi@05d861eca28c5c4:~$ ./array.sh

uhi@05d861eca28c5c4:~$ ./array.sh

uhi@05d861eca28c5c4:~$ gedit array.sh
uhi@05d861eca28c5c4:~$ ./array.sh
avatpoint
uhi@05d861eca28c5c4:~$ _
```

```
1#!/bin/bash
2# Script to print an element of an array with an index of 2
3
4# declaring the array
5declare -a example_array=("Welcome" "To" "Javatpoint")
6
7# printing the element with index of 2
8echo "${example_array[2]}"
9
```

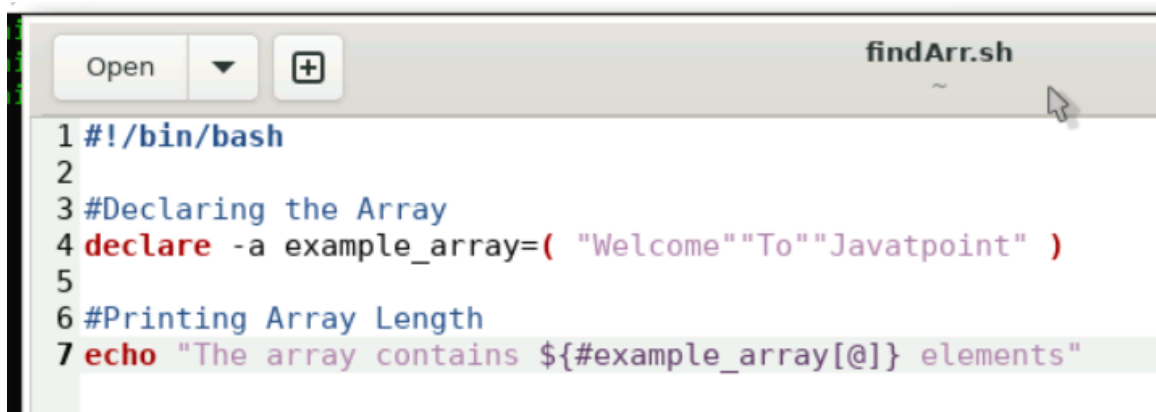
PRINTING THE KEYS OF AN ARRAY

```
juhi@05d861eca28c5c4: ~
juhi@05d861eca28c5c4:~$ touch printing.sh
juhi@05d861eca28c5c4:~$ chmod +x printing.sh
juhi@05d861eca28c5c4:~$ gedit printing.sh
juhi@05d861eca28c5c4:~$ ./printing.sh
0
juhi@05d861eca28c5c4:~$ gedit printing.sh
juhi@05d861eca28c5c4:~$ ./printing.sh
0 1 2
juhi@05d861eca28c5c4:~$
```

```
1#!/bin/bash
2# Script to print the keys of the array
3
4# Declaring the Array
5declare -a example_array=("Welcome" "To" "Javatpoint")
6|
7# Printing the keys (indices) of the array
8echo "${!example_array[@]}"
9
```

FINDING ARRAY LENGTH

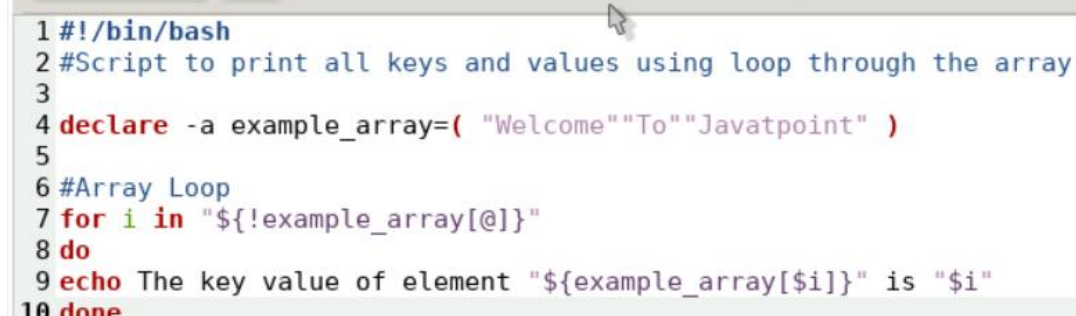
```
juhi@05d861eca28c5c4:~$ touch findArr.sh
juhi@05d861eca28c5c4:~$ chmod +x findArr.sh
juhi@05d861eca28c5c4:~$ gedit findArr.sh
juhi@05d861eca28c5c4:~$ ./findArr.sh
The array contains 1 elements
juhi@05d861eca28c5c4:~$ _
```



```
1 #!/bin/bash
2
3 #Declaring the Array
4 declare -a example_array=( "Welcome""To""Javatpoint" )
5
6 #Printing Array Length
7 echo "The array contains ${#example_array[@]} elements"
```

LOOP THROUGH THE ARRAY

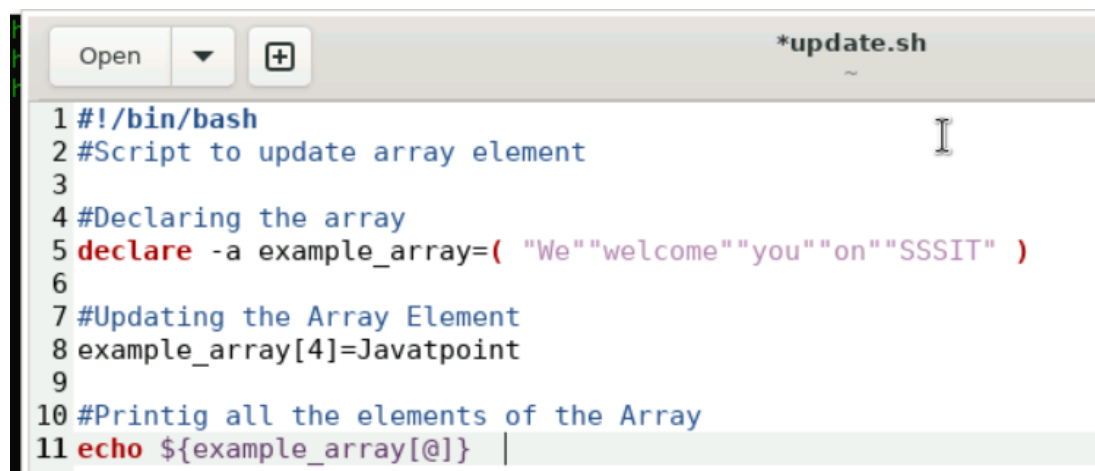
```
juhi@05d861eca28c5c4: ~
juhi@05d861eca28c5c4:~$ touch loop.sh
juhi@05d861eca28c5c4:~$ chmod +x loop.sh
juhi@05d861eca28c5c4:~$ gedit loop.sh
juhi@05d861eca28c5c4:~$ ./loop.sh
The key value of element WelcomeToJavatpoint is 0
juhi@05d861eca28c5c4:~$ _
```



```
1 #!/bin/bash
2 #Script to print all keys and values using loop through the array
3
4 declare -a example_array=( "Welcome""To""Javatpoint" )
5
6 #Array Loop
7 for i in "${!example_array[@]}"
8 do
9 echo The key value of element "${example_array[$i]}" is "$i"
10 done
```

UPDATING ARRAY ELEMENT

```
juhi@05d861eca28c5c4:~$ touch update.sh
juhi@05d861eca28c5c4:~$ chmod +x update.sh
juhi@05d861eca28c5c4:~$ gedit update.sh
juhi@05d861eca28c5c4:~$ ./update.sh
We welcome you on SSSIT Javatpoint
juhi@05d861eca28c5c4:~$ gedit update.sh
juhi@05d861eca28c5c4:~$ ./update.sh
We welcome you on Javatpoint
juhi@05d861eca28c5c4:~$ _
```

A screenshot of a code editor window titled '*update.sh'. The editor shows a shell script with 11 lines of code. The code declares an array 'example_array' with five elements: 'We', 'welcome', 'you', 'on', and 'SSSIT'. It then updates the element at index 4 to 'Javatpoint' and prints all elements of the array. The script is saved and executed, showing the output 'We welcome you on Javatpoint'.

```
1 #!/bin/bash
2 #Script to update array element
3
4 #Declaring the array
5 declare -a example_array=( "We""welcome""you""on""SSSIT" )
6
7 #Updating the Array Element
8 example_array[4]=Javatpoint
9
10 #Printig all the elements of the Array
11 echo ${example_array[@]} |
```

DELETING AN ELEMENT FROM AN ARRAY

```
juhi@05d861eca28c5c4:~$ touch delete.sh
juhi@05d861eca28c5c4:~$ chmod delete.sh
chmod: missing operand after 'delete.sh'
Try 'chmod --help' for more information.
juhi@05d861eca28c5c4:~$ chmod +x delete.sh
juhi@05d861eca28c5c4:~$ gedit delete.sh
juhi@05d861eca28c5c4:~$ ./delete.sh
Java HTML CSS JavaScript
juhi@05d861eca28c5c4:~$ _
```

```

1 #!/bin/bash
2 #Script to delete the element from the array
3
4 #Declaring the array
5 declare -a example_array=( "Java" "Python" "HTML" "CSS" "JavaScript" )
6
7 #Removing the element
8 unset example_array[1]
9
10 #Printing all the elements after deletion
11 echo "${example_array[@]}"

```

DELETING THE ENTIRE ARRAY

```

juhi@05d861eca28c5c4:~$ touch deleting.sh
juhi@05d861eca28c5c4:~$ chmod +x deleting.sh
juhi@05d861eca28c5c4:~$ gedit deleting.sh
juhi@05d861eca28c5c4:~$ ./deleting.sh

juhi@05d861eca28c5c4:~$ _

```

```

1 #!/bin/bash
2 #Script to delete the entire Array
3
4 #Declaring the Array
5 declare -a example_array=( "Java" "Python" "HTML" "CSS" "JavaScript" )
6
7 #Deleting Entire Array
8 unset example_array
9
10 #Printing the Array Elements
11 echo ${!example_array[@]}
12
13 #Printing the keys
14 echo ${!example_array[@]}

```

SLICE ARRAY ELEMENTS

```
juhi@05d861eca28c5c4:~$ touch slice.sh
juhi@05d861eca28c5c4:~$ chmod +x slice.sh
juhi@05d861eca28c5c4:~$ gedit slice.sh
juhi@05d861eca28c5c4:~$ ./slice.sh
Python
HTML
CSS
juhi@05d861eca28c5c4:~$
```

```
1#!/bin/bash
2#Script to slice Array Element from index 1 to index 3
3
4#Declaring the Array
5example_array=( "Java" "Python" "HTML" "CSS" "JavaScript" )
6
7#Slicing the Array
8sliced_array=("${example_array[@]:1:3}")
9
10#Applying for loop to iterate over each element in Array
11for i in "${sliced_array[@]}"
12do
13echo $i
14done
```