

CASE

Example 1

In this example, we have defined a simple scenario to demonstrate the use of the case statement.

```
pravalika18@d73d6822db255ef:~$ touch case_ex1.sh
pravalika18@d73d6822db255ef:~$ chmod +x case_ex1.sh
pravalika18@d73d6822db255ef:~$ gedit case_ex1.sh

(gedit:840): IBUS-WARNING **: 09:46:14.090: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory
^C
pravalika18@d73d6822db255ef:~$ ./case_ex1.sh
Do you know Java Programming?
Yes/No? :yes
That's amazing.

pravalika18@d73d6822db255ef:~$ cat case_ex1.sh
#!/bin/bash

echo "Do you know Java Programming?"
read -p "Yes/No? :" Answer
case $Answer in
    Yes|yes|y|Y)
        echo "That's amazing."
        ;;
    No|no|N|n)
        echo "It's easy. Let's start learning from javatpoint."
        ;;
    *)
        ;;
esac
pravalika18@d73d6822db255ef:~$
```

Example 2

In this example, we have defined a combined scenario where there is also a default case when no previous matched case is found.

```
pravalika18@d73d6822db255ef: ~  
;;  
esac  
pravalika18@d73d6822db255ef:~$ touch case_ex2.sh  
pravalika18@d73d6822db255ef:~$ chmod +x case_ex2.sh  
pravalika18@d73d6822db255ef:~$ gedit case_ex2.sh  
(gedit:981): IBUS-WARNING **: 09:49:08.133: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./case_ex2.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.  
pravalika18@d73d6822db255ef:~$ cat case_ex2.sh  
#!/bin/bash  
  
echo "Which Operating System are you using?"  
echo "Windows, Android, Chrome, Linux, Others?"  
read -p "Type your OS Name:" OS  
  
case $OS in  
    windows|windows)  
        echo "That's common. You should try something new."  
        echo  
        ;;  
    Android|android)  
        echo "This is my favorite. It has lots of applications."  
        echo  
        ;;  
    Chrome|chrome)  
        echo "Cool!!! It's for pro users. Amazing Choice."  
        echo  
        ;;  
    Linux|linux)  
        echo "You might be serious about security!!"  
        echo  
        ;;  
    *)  
        echo "Sounds interesting. I will try that."  
        echo  
        ;;  
esac
```

```
pravalika18@d73d6822db255ef: ~  
Cool!!! It's for pro users. Amazing Choice.  
pravalika18@d73d6822db255ef:~$ cat case_ex2.sh  
#!/bin/bash  
  
echo "Which Operating System are you using?"  
echo "Windows, Android, Chrome, Linux, Others?"  
read -p "Type your OS Name:" OS  
  
case $OS in  
    windows|windows)  
        echo "That's common. You should try something new."  
        echo  
        ;;  
    Android|android)  
        echo "This is my favorite. It has lots of applications."  
        echo  
        ;;  
    Chrome|chrome)  
        echo "Cool!!! It's for pro users. Amazing Choice."  
        echo  
        ;;  
    Linux|linux)  
        echo "You might be serious about security!!"  
        echo  
        ;;  
    *)  
        echo "Sounds interesting. I will try that."  
        echo  
        ;;  
esac  
pravalika18@d73d6822db255ef:~$
```

FOR LOOP

1.Basic 'For Loop' Example

```
pravalika18@d73d6822db255ef: ~  
pravalika18@d73d6822db255ef:~$ touch ex1.sh  
pravalika18@d73d6822db255ef:~$ chmod +x ex1.sh  
pravalika18@d73d6822db255ef:~$ gedit ex1.sh  
(gedit:545): IBUS-WARNING **: 09:58:33.634: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./ex1.sh  
Start  
learn  
from  
Javatpoint.  
Thank You.  
pravalika18@d73d6822db255ef:~$ cat ex1.sh  
#!/bin/bash  
#This is the basic example of 'for loop'.  
  
learn="Start learning from Javatpoint."  
  
for learn in $learn  
do  
echo $learn  
done  
  
echo "Thank You."  
pravalika18@d73d6822db255ef:~$
```

2. For Loop to Read a Range

```
pravalika18@d73d6822db255ef: ~  
pravalika18@d73d6822db255ef:~$ touch ex2.sh  
pravalika18@d73d6822db255ef:~$ chmod +x ex2.sh  
pravalika18@d73d6822db255ef:~$ gedit ex2.sh  
(gedit:693): IBUS-WARNING **: 11:44:36.782: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./ex2.sh  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
Series of numbers from 1 to 10.  
pravalika18@d73d6822db255ef:~$ cat ex2.sh  
#!/bin/bash  
#This is the basic example to print a series of numbers from 1 to 10.  
  
for num in {1..10}  
do  
echo $num  
done  
  
echo "Series of numbers from 1 to 10."  
pravalika18@d73d6822db255ef:~$
```

3. For Loop to Read a Range with Increment.

```
pravalika18@d73d6822db255ef:~$ touch ex3.sh
pravalika18@d73d6822db255ef:~$ chmod +x ex3.sh
chmod: cannot access 'ex3.sh': No such file or directory
pravalika18@d73d6822db255ef:~$ touch ex3.sh
pravalika18@d73d6822db255ef:~$ chmod +x ex3.sh
pravalika18@d73d6822db255ef:~$ gedit ex3.sh

(gedit:806): IBUS-WARNING **: 11:46:50.159: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory
^C
pravalika18@d73d6822db255ef:~$ ./ex3.sh
1
2
3
4
5
6
7
8
9
10
pravalika18@d73d6822db255ef:~$ cat ex3.sh
#!/bin/bash

#For Loop to Read a Range with Increment

for num in {1..10..1}
do
echo $num
done
pravalika18@d73d6822db255ef:~$
```

4. For Loop to Read a Range with Decrement.

```
pravalika18@d73d6822db255ef:~$ touch ex4.sh
pravalika18@d73d6822db255ef:~$ chmod +x ex4.sh
pravalika18@d73d6822db255ef:~$ gedit ex4.sh

(gedit:922): IBUS-WARNING **: 11:47:59.427: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory
^C
pravalika18@d73d6822db255ef:~$ ./ex4.sh
10
9
8
7
6
5
4
3
2
1
0
pravalika18@d73d6822db255ef:~$ cat ex4.sh
#!/bin/bash

#For Loop to Read a Range with Decrement

for num in {10..0..1}
do
echo $num
done
pravalika18@d73d6822db255ef:~$
```

5. For Loop to Read Array Variables.

```
pravalika18@d73d6822db255ef: ~  
echo $1  
done  
pravalika18@d73d6822db255ef:~$ touch ex6.sh  
pravalika18@d73d6822db255ef:~$ chmod +x ex6.sh  
pravalika18@d73d6822db255ef:~$ gedit ex6.sh  
(gedit:1154): IBUS-WARNING **: 11:51:51.982: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./ex6.sh  
WelcometoJavatpoint  
pravalika18@d73d6822db255ef:~$ cat ex6.sh  
#!/bin/bash  
  
#Array Declaration  
arr=( "Welcome""to""Javatpoint" )  
  
for i in "${arr[@]}"  
do  
echo $i  
done  
pravalika18@d73d6822db255ef:~$
```

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

```
pravalika18@d73d6822db255ef: ~  
pravalika18@d73d6822db255ef:~$ chmod +x ex7.sh  
pravalika18@d73d6822db255ef:~$ gedit ex7.sh  
(gedit:1377): IBUS-WARNING **: 11:53:35.548: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./ex7.sh  
Let's  
start  
learning  
from  
Javatpoint.  
pravalika18@d73d6822db255ef:~$ cat ex7.sh  
#!/bin/bash  
#For Loop to Read white spaces in String as word separators  
  
str="Let's stant  
learning from Javatpoint."  
  
for i in $str;  
do  
echo "$i"  
done  
pravalika18@d73d6822db255ef:~$
```

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

```
pravalika18@d73d6822db255ef: ~  
done  
pravalika18@d73d6822db255ef:~$ touch ex8.sh  
pravalika18@d73d6822db255ef:~$ chmod +x ex8.sh  
pravalika18@d73d6822db255ef:~$ gedit ex8.sh  
(gedit:1522): IBUS-WARNING **: 11:55:03.225: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./ex8.sh  
Let's start  
learning from  
Davatpoint.  
pravalika18@d73d6822db255ef:~$ cat ex8.sh  
#!/bin/bash  
#For Loop to Read each line in String as a word  
  
str="Let's start  
learning from  
Davatpoint."  
  
for i in "$str";  
do  
echo "$i"  
done  
pravalika18@d73d6822db255ef:~$
```

8. For Loop to Read Three-expression

```
pravalika18@d73d6822db255ef: ~  
pravalika18@d73d6822db255ef:~$ chmod +x ex9.sh  
pravalika18@d73d6822db255ef:~$ gedit ex9.sh  
(gedit:1634): IBUS-WARNING **: 11:56:11.912: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./ex9.sh  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
pravalika18@d73d6822db255ef:~$ cat ex9.sh  
#!/bin/bash  
#For Loop to Read Three-expression  
  
for ((i=1; i<=10; i++))  
do  
echo "$i"  
done  
pravalika18@d73d6822db255ef:~$
```

9. For Loop with a Break Statement

```
pravalika18@d73d6822db255ef:~  
pravalika18@d73d6822db255ef:~$ touch ex10.sh  
pravalika18@d73d6822db255ef:~$ chmod +x ex10.sh  
pravalika18@d73d6822db255ef:~$ gedit ex10.sh  
(gedit:2168): IBUS-WARNING **: 11:57:43.996: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./ex10.sh  
2  
4  
6  
8  
10  
12  
14  
16  
18  
20  
pravalika18@d73d6822db255ef:~$ cat ex10.sh  
#!/bin/bash  
#Table of 2  
  
for table in {2..100..2}  
do  
    echo $table  
    if [ $table == 20 ]; then  
        break  
    fi  
done  
pravalika18@d73d6822db255ef:~$
```

10. For Loop with a Continue Statement

```
pravalika18@d73d6822db255ef:~  
pravalika18@d73d6822db255ef:~$ gedit ex11.sh  
(gedit:2288): IBUS-WARNING **: 11:59:10.234: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./ex11.sh  
1  
2  
3  
4  
5  
16  
17  
18  
19  
20  
pravalika18@d73d6822db255ef:~$ cat ex11.sh  
#!/bin/bash  
#Numbers from 1 to 20, ignoring from 6 to 15 using continue statement"  
  
for ((i=1; i<=20; i++));  
do  
    if [[ $i -gt 5 && $i -lt 16 ]];  
    then  
        continue  
    fi  
    echo $i  
done  
pravalika18@d73d6822db255ef:~$
```

11. Infinite Bash For Loop

```
pravalika18@d73d6822db255ef:~$ touch ex12.sh
pravalika18@d73d6822db255ef:~$ chmod +x ex12.sh
pravalika18@d73d6822db255ef:~$ gedit ex12.sh

(gedit:2404): IBUS-WARNING **: 12:00:35.354: Failed to mkdir /home/pravalika18/snap/gedit/604/.config/ibus/bus: Not a directory
^C
pravalika18@d73d6822db255ef:~$ ./ex12.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
Current Number: 15
Current Number: 16
Current Number: 17
Current Number: 18
Current Number: 19
Current Number: 20
Current Number: 21
Current Number: 22
^C
pravalika18@d73d6822db255ef:~$ cat ex12.sh
#!/bin/bash

i=1;
for (( ; ; ))
do
sleep 1s
echo "Current Number: $((i++))"
done
pravalika18@d73d6822db255ef:~$
```

BASH WHILE

1.While Loop with Single Condition:

In this example, the while loop is used with a single condition in expression. It is the basic example of while loop which will print series of numbers as per user input:

```
pravalika18@d73d6822db255ef:~$ touch ex1.sh
pravalika18@d73d6822db255ef:~$ chmod +x .sh
chmod: cannot access '.sh': No such file or directory
pravalika18@d73d6822db255ef:~$ gedit ex1.sh

(gedit:2701): IBUS-WARNING **: 04:10:56.535: Failed to mkdir /home/pravalika18/snap/gedit/604/.config/ibus/bus: Not a directory
^C
pravalika18@d73d6822db255ef:~$ ./ex1.sh
Enter starting number: 1
Enter ending number: 4
1
2
3
4
This is the sequence that you wanted.
pravalika18@d73d6822db255ef:~$ cat ex1.sh
#!/bin/bash
#Script to get specified numbers

read -p "Enter starting number: " snum
read -p "Enter ending number: " enum

while [[ $snum -le $enum ]];
do
echo $snum
((snum++))
done

echo "This is the sequence that you wanted."
pravalika18@d73d6822db255ef:~$
```


2. While Loop with Multiple Conditions

```
pravalika18@d73d6822db255ef:~  
pravalika18@d73d6822db255ef:~$ chmod +x ex2.sh  
pravalika18@d73d6822db255ef:~$ gedit ex2.sh  
(gedit:2814): IBUS-WARNING **: 04:12:35.071: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./ex2.sh  
Enter starting number: 1  
Enter ending number: 4  
1  
2  
3  
4  
This is the sequence that you wanted.  
pravalika18@d73d6822db255ef:~$ cat ex2.sh  
#!/bin/bash  
#Script to get specified numbers  
  
read -p "Enter starting number: " snum  
read -p "Enter ending number: " enum  
  
while [[ $snum -lt $enum || $snum == $enum ]];  
do  
echo $snum  
((snum++))  
done  
  
echo "This is the sequence that you wanted."  
pravalika18@d73d6822db255ef:~$
```

3. Infinite While Loop

An infinite loop is a loop that has no ending or termination. If the condition always evaluates to true, it creates an infinite loop. The loop will execute continuously until it is forcefully stopped using CTRL+C :

```
pravalika18@d73d6822db255ef:~  
pravalika18@d73d6822db255ef:~$ touch ex3.sh  
pravalika18@d73d6822db255ef:~$ chmod +x ex3.sh  
pravalika18@d73d6822db255ef:~$ gedit ex3.sh  
(gedit:3307): IBUS-WARNING **: 04:21:38.488: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ cat ex3.sh  
#!/bin/bash  
#An infinite while loop  
  
while :: do echo "Welcome to Javatpoint."; done  
pravalika18@d73d6822db255ef:~$
```


5. While Loop with a Continue Statement

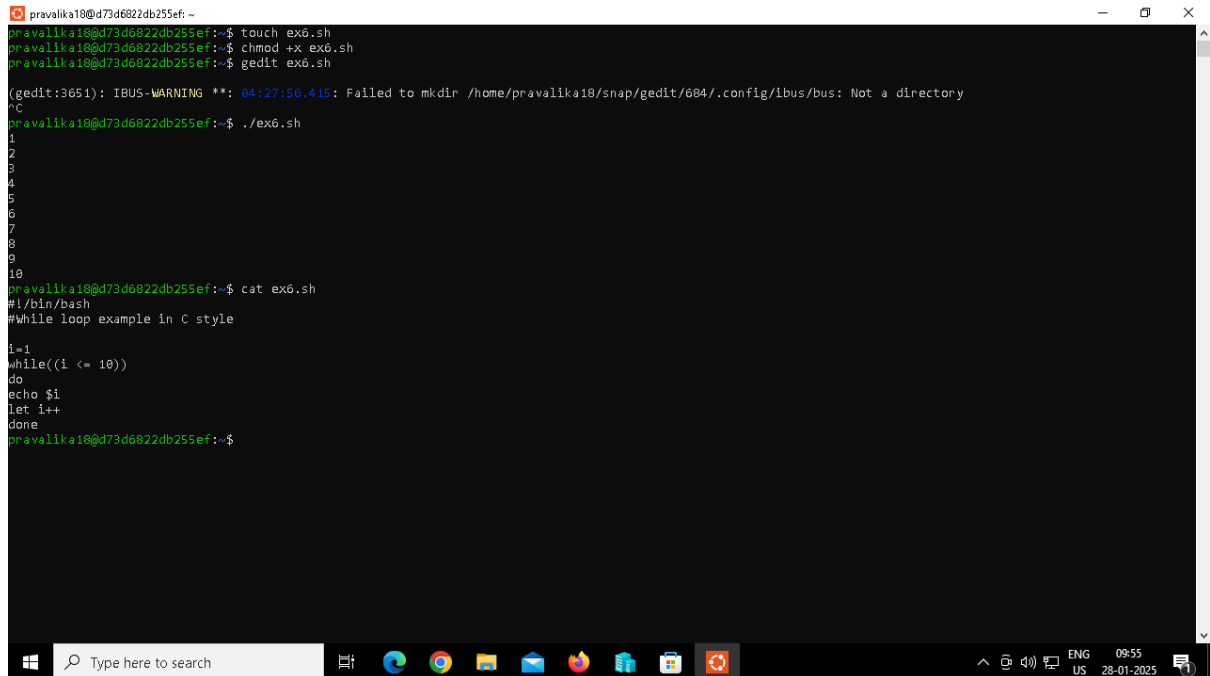
A continue statement can be used to skip the iteration for a specific condition inside the while loop.

```
pravalika18@d73d6822db255ef:~  
pravalika18@d73d6822db255ef:~$ touch ex5.sh  
pravalika18@d73d6822db255ef:~$ chmod +x ex5.sh  
pravalika18@d73d6822db255ef:~$ gedit ex5.sh  
(gedit:3531): IBUS-WARNING **: 04:25:05.752: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
** (gedit:3531): WARNING **: 04:25:35.469: atk-bridge: get_device_events_reply: unknown signature  
^C  
pravalika18@d73d6822db255ef:~$ ./ex5.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.  
pravalika18@d73d6822db255ef:~$ cat ex5.sh  
#!/bin/bash  
#While Loop Example with a Continue Statement  
  
i=0  
while [ $i -le 10 ]  
do  
    ((i++))  
    if [[ "$i" == 5 ]];  
    then  
        continue  
    fi  
    echo "Current Number : $i"  
done  
  
echo "Skipped number 5 using Continue Statement."  
pravalika18@d73d6822db255ef:~$
```

```
pravalika18@d73d6822db255ef:~  
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.  
pravalika18@d73d6822db255ef:~$ cat ex5.sh  
#!/bin/bash  
#While Loop Example with a Continue Statement  
  
i=0  
while [ $i -le 10 ]  
do  
    ((i++))  
    if [[ "$i" == 5 ]];  
    then  
        continue  
    fi  
    echo "Current Number : $i"  
done  
  
echo "Skipped number 5 using Continue Statement."  
pravalika18@d73d6822db255ef:~$
```

6. While Loop with C-Style

We can also write while loop in bash script as similar as a while loop in C programming language.

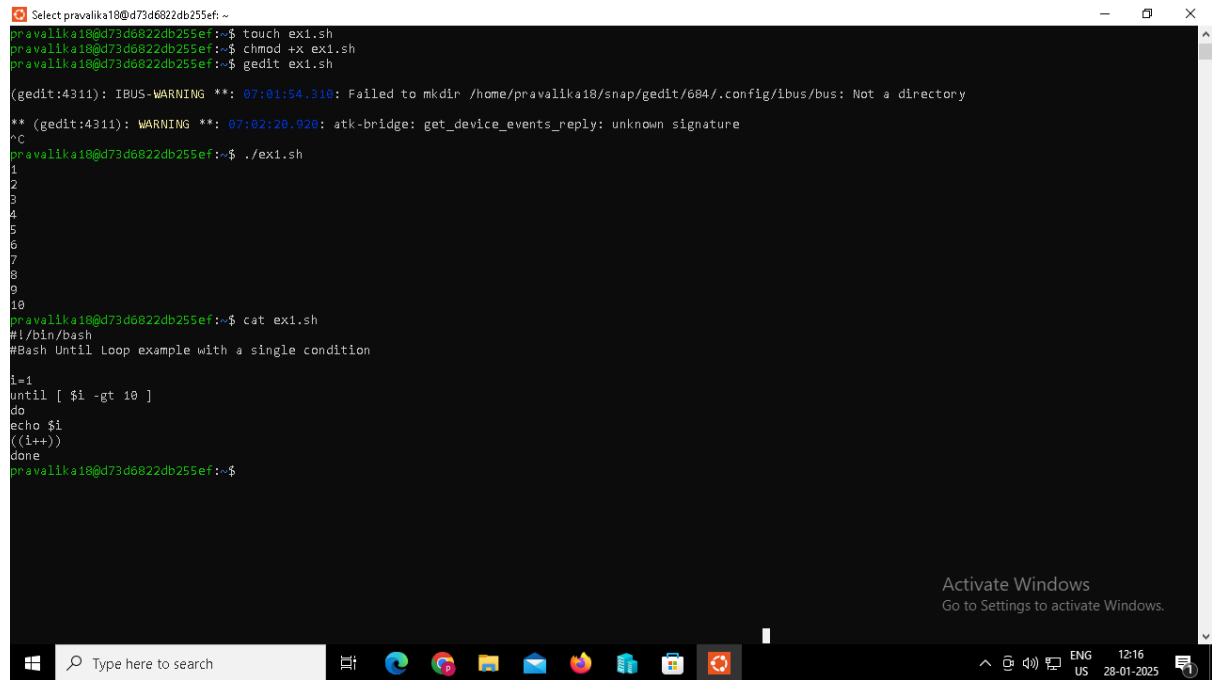


```
pravalika18@d73d6822db255ef:~  
pravalika18@d73d6822db255ef:~$ touch ex6.sh  
pravalika18@d73d6822db255ef:~$ chmod +x ex6.sh  
pravalika18@d73d6822db255ef:~$ gedit ex6.sh  
(gedit:3651): IBUS-WARNING **: 04:27:56.415: Failed to mkdir /home/pravalika18/snap/gedit/604/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./ex6.sh  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
pravalika18@d73d6822db255ef:~$ cat ex6.sh  
#!/bin/bash  
#While loop example in C style  
  
i=1  
while((i <= 10))  
do  
echo $i  
let i++  
done  
pravalika18@d73d6822db255ef:~$
```

BASH UNTIL

Until Loop with Single Condition

In this example, the until loop contains a single condition in expression. It is the basic example of until loop which will print series of numbers from 1 to 10:



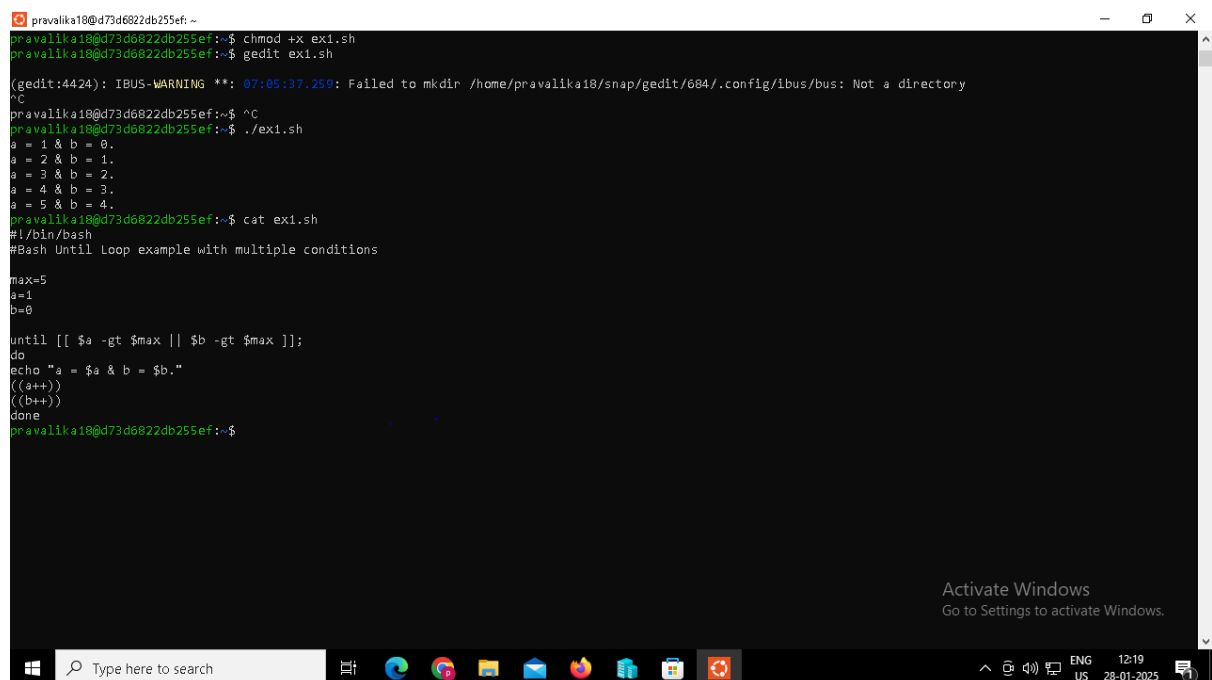
```
Select pravalika18@d73d6822db255ef: ~
pravalika18@d73d6822db255ef:~$ touch ex1.sh
pravalika18@d73d6822db255ef:~$ chmod +x ex1.sh
pravalika18@d73d6822db255ef:~$ gedit ex1.sh

(gedit:4311): IBUS-WARNING **: 07:01:54.310: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory
** (gedit:4311): WARNING **: 07:02:20.920: atk-bridge: get_device_events_reply: unknown signature
^C
pravalika18@d73d6822db255ef:~$ ./ex1.sh
1
2
3
4
5
6
7
8
9
10
pravalika18@d73d6822db255ef:~$ cat ex1.sh
#!/bin/bash
#Bash Until Loop example with a single condition

i=1
until [ $i -gt 10 ]
do
echo $i
((i++))
done
pravalika18@d73d6822db255ef:~$
```

2. Until Loop with Multiple Conditions

Following is an example with multiple conditions in an expression:



```
pravalika18@d73d6822db255ef: ~
pravalika18@d73d6822db255ef:~$ chmod +x ex1.sh
pravalika18@d73d6822db255ef:~$ gedit ex1.sh

(gedit:4424): IBUS-WARNING **: 07:05:37.259: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory
^C
pravalika18@d73d6822db255ef:~$ ./ex1.sh
a = 1 & b = 0.
a = 2 & b = 1.
a = 3 & b = 2.
a = 4 & b = 3.
a = 5 & b = 4.
pravalika18@d73d6822db255ef:~$ cat ex1.sh
#!/bin/bash
#Bash Until Loop example with multiple conditions

max=5
a=1
b=0

until [[ $a -gt $max || $b -gt $max ]];
do
echo "a = $a & b = $b."
((a++))
((b++))
done
pravalika18@d73d6822db255ef:~$
```

Bash String

Equal Operator

An equal operator (=`)` is used to check whether two strings are equal.

```
pravalika18@d73d6822db255ef: ~  
pravalika18@d73d6822db255ef:~$ touch eq.sh  
pravalika18@d73d6822db255ef:~$ chmod +x eq.sh  
pravalika18@d73d6822db255ef:~$ gedit eq.sh  
(gedit:629): IBUS-WARNING **: 10:13:59.426: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./eq.sh  
Strings are not equal.  
pravalika18@d73d6822db255ef:~$ cat eq.sh  
#!/bin/bash  
#Script to check whether two strings are equal.  
  
str1="WelcometoJavatpoint."  
str2="javatpoint"  
  
if [ $str1 = $str2 ];  
then  
echo "Both the strings are equal."  
else  
echo "Strings are not equal."  
fi  
pravalika18@d73d6822db255ef:~$
```

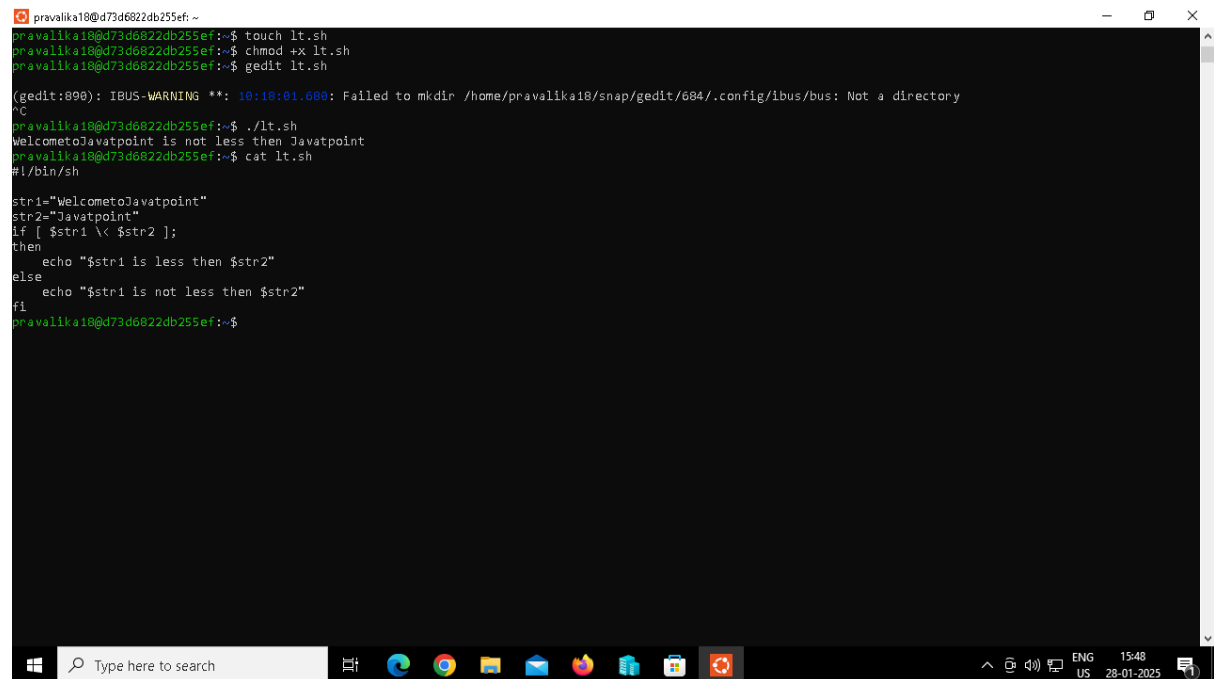
Not Equal Operator

Not equal operator (`!=`) is used to define that strings are not equal.

```
pravalika18@d73d6822db255ef: ~  
pravalika18@d73d6822db255ef:~$ touch ne.sh  
pravalika18@d73d6822db255ef:~$ chmod +x ne.sh  
pravalika18@d73d6822db255ef:~$ gedit ne.sh  
(gedit:771): IBUS-WARNING **: 10:16:53.592: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./ne.sh  
Strings are not equal.  
pravalika18@d73d6822db255ef:~$ cat ne.sh  
#!/bin/bash  
#Script to check whether two strings are equal.  
  
str1="WelcometoJavatpoint."  
str2="javatpoint"  
  
if [[ $str1 != $str2 ]];  
then  
echo "Strings are not equal."  
else  
echo "Strings are equal."  
fi  
pravalika18@d73d6822db255ef:~$
```

Less than Operator

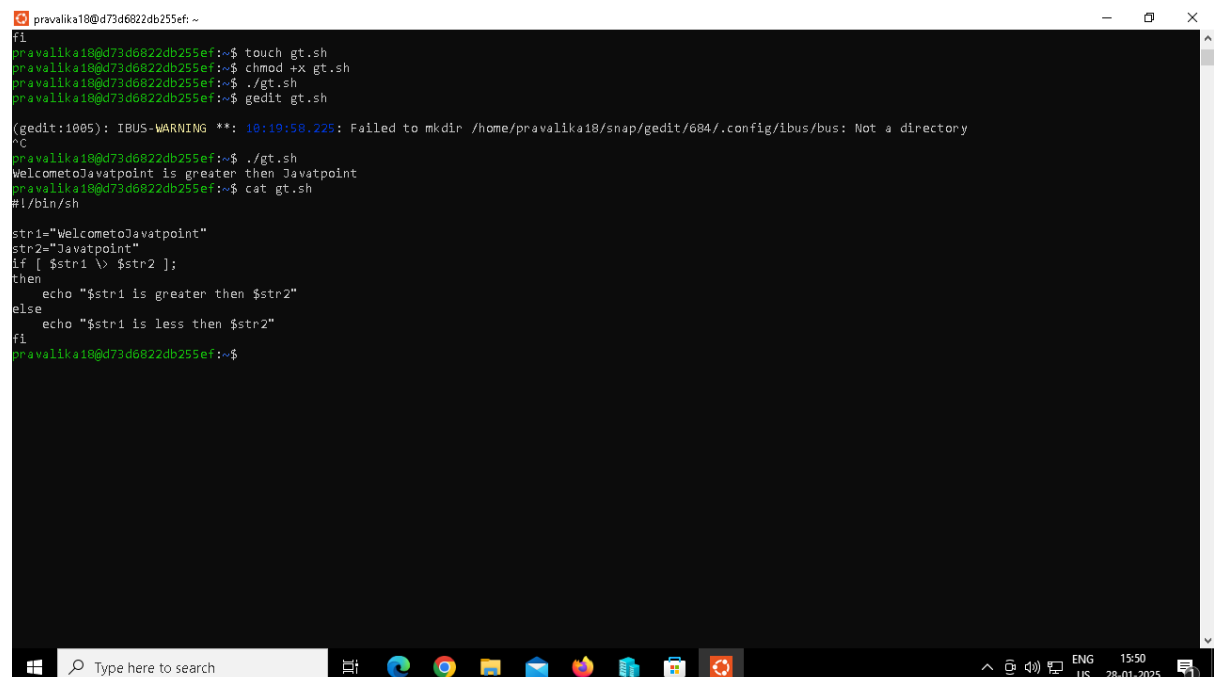
The 'less than operator (<)' is a conditional operator which is used to check if string1 is less than string2.



```
pravalika18@d73d6822db255ef: ~  
pravalika18@d73d6822db255ef:~$ touch lt.sh  
pravalika18@d73d6822db255ef:~$ chmod +x lt.sh  
pravalika18@d73d6822db255ef:~$ gedit lt.sh  
(gedit:898): IBUS-WARNING **: 10:18:01.688: Failed to mkdir /home/pravalika18/snap/gedit/604/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./lt.sh  
WelcometoJavatpoint is not less then Javatpoint  
pravalika18@d73d6822db255ef:~$ cat lt.sh  
#!/bin/sh  
  
str1="WelcometoJavatpoint"  
str2="Javatpoint"  
if [ $str1 \< $str2 ];  
then  
    echo "$str1 is less then $str2"  
else  
    echo "$str1 is not less then $str2"  
fi  
pravalika18@d73d6822db255ef:~$
```

Greater than Operator

The 'greater than operator (>)' is used to check if string1 is greater than string2.



```
pravalika18@d73d6822db255ef: ~  
pravalika18@d73d6822db255ef:~$ touch gt.sh  
pravalika18@d73d6822db255ef:~$ chmod +x gt.sh  
pravalika18@d73d6822db255ef:~$ ./gt.sh  
pravalika18@d73d6822db255ef:~$ gedit gt.sh  
(gedit:1005): IBUS-WARNING **: 10:19:58.225: Failed to mkdir /home/pravalika18/snap/gedit/604/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./gt.sh  
WelcometoJavatpoint is greater then Javatpoint  
pravalika18@d73d6822db255ef:~$ cat gt.sh  
#!/bin/sh  
  
str1="WelcometoJavatpoint"  
str2="Javatpoint"  
if [ $str1 \> $str2 ];  
then  
    echo "$str1 is greater then $str2"  
else  
    echo "$str1 is less then $str2"  
fi  
pravalika18@d73d6822db255ef:~$
```

To check if the string length is greater than Zero:

This operator is used to check if the string is zero or greater than zero.

```
pravalika18@d73d6822db255ef: ~  
pravalika18@d73d6822db255ef:~$ touch gt1.sh  
pravalika18@d73d6822db255ef:~$ chmod +x gt1.sh  
pravalika18@d73d6822db255ef:~$ gedit gt1.sh  
(gedit:1119): IBUS-WARNING **: 10:21:15.383: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./gt1.sh  
String is not empty  
pravalika18@d73d6822db255ef:~$ cat gt1.sh  
#!/bin/sh  
  
str="WelcometoJavatpoint"  
  
if [ -n $str ];  
then  
    echo "String is not empty"  
else  
    echo "String is empty"  
fi  
pravalika18@d73d6822db255ef:~$
```

To check if the string length is equal to Zero

This operator is used to check if the string is empty or equal to zero.

```
pravalika18@d73d6822db255ef: ~  
pravalika18@d73d6822db255ef:~$ touch eq0.sh  
pravalika18@d73d6822db255ef:~$ chmod +x eq0.sh  
pravalika18@d73d6822db255ef:~$ gedit eq0.sh  
(gedit:1232): IBUS-WARNING **: 10:22:28.345: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./eq0.sh  
String is empty.  
pravalika18@d73d6822db255ef:~$ cat eq0.sh  
#!/bin/sh  
  
str=""  
  
if [ -z $str ];  
then  
    echo "String is empty."  
else  
    echo "String is non-empty."  
fi  
pravalika18@d73d6822db255ef:~$
```


BASH FIND STRING

Example 1

The simplest way to calculate the length of a string is to use '#' symbol. In this example, we have used `${#string_variable_name}` to find the length of a string.

```
pravalika18@d73d6822db255ef: ~  
pravalika18@d73d6822db255ef:~$ chmod+x lg.sh  
gedit lg.sh  
pravalika18@d73d6822db255ef:~$ chmod+x lg.sh  
pravalika18@d73d6822db255ef:~$ gedit lg.sh  
(gedit:1381): IBUS-WARNING **: 10:24:06.959: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./lg.sh  
Length of 'Welcome to Javatpoint' is 21  
pravalika18@d73d6822db255ef:~$ cat lg.sh  
#!/bin/bash  
#Bash program to find the length of a string  
  
str="Welcome to Javatpoint"  
length=${#str}  
  
echo "Length of '$str' is $length"  
pravalika18@d73d6822db255ef:~$
```

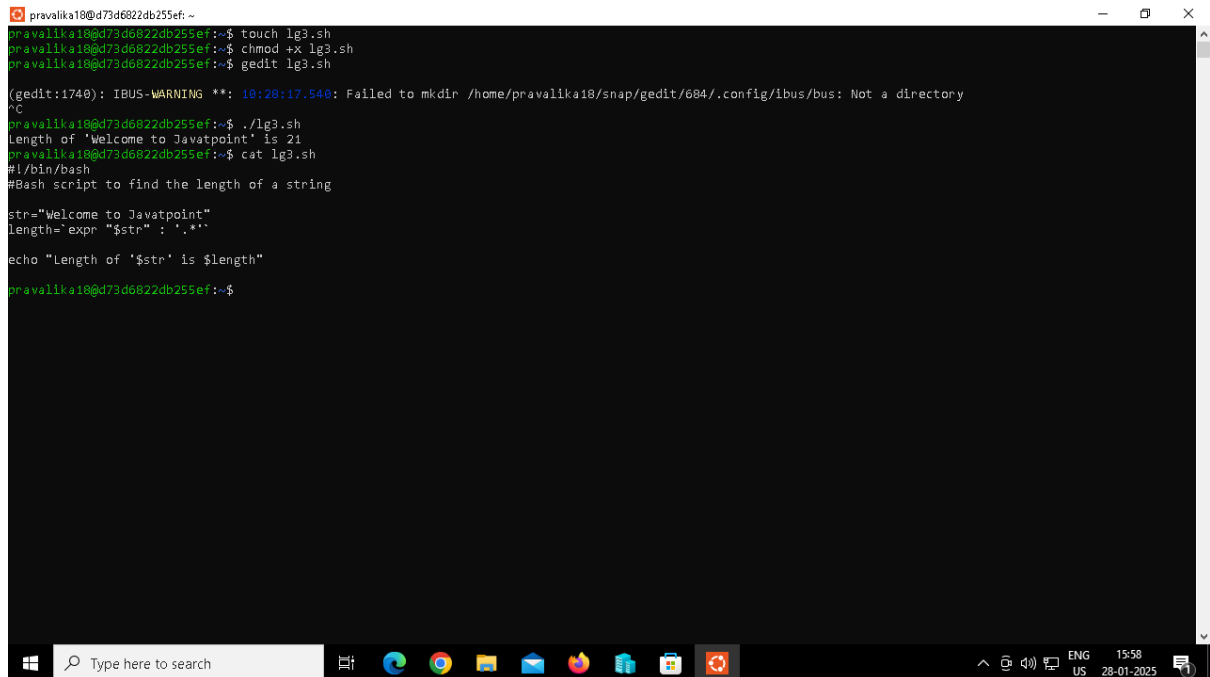
Example 2

In this example, we have used ``expr length "$str"`` to find the length of a string.

```
pravalika18@d73d6822db255ef: ~  
echo "Length of '$str' is $length"  
pravalika18@d73d6822db255ef:~$ touch lg1.sh  
pravalika18@d73d6822db255ef:~$ chmod+x lg1.sh  
pravalika18@d73d6822db255ef:~$ gedit lg1.sh  
(gedit:1505): IBUS-WARNING **: 10:25:49.955: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./lg1.sh  
Length of 'Welcome to Javatpoint' is 21  
pravalika18@d73d6822db255ef:~$ cat lg1.sh  
#!/bin/bash  
#Bash script to find the length of a string  
  
str="Welcome to Javatpoint"  
length=`expr length "$str"`  
  
echo "Length of '$str' is $length"  
pravalika18@d73d6822db255ef:~$
```

Example 3

In this example, we have used ``expr "$str": '.*'`` to find the length of a string. Here, `str` is a string variable.



```
pravalika18@d73d6822db255ef:~$ touch lg3.sh
pravalika18@d73d6822db255ef:~$ chmod +x lg3.sh
pravalika18@d73d6822db255ef:~$ gedit lg3.sh

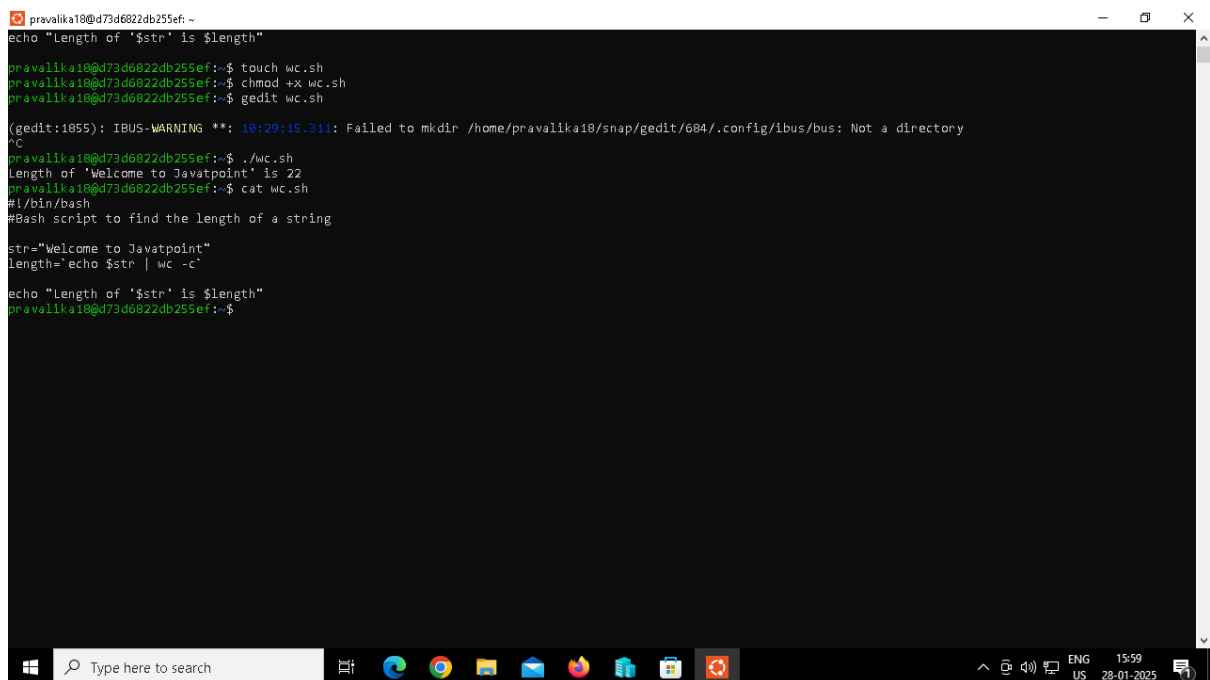
(gedit:1748): IBUS-WARNING **: 18:28:17.548: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory
^C
pravalika18@d73d6822db255ef:~$ ./lg3.sh
Length of 'Welcome to Javatpoint' is 21
pravalika18@d73d6822db255ef:~$ cat lg3.sh
#!/bin/bash
#Bash script to find the length of a string

str="Welcome to Javatpoint"
length=`expr "$str" : '.*'`

echo "Length of '$str' is $length"
pravalika18@d73d6822db255ef:~$
```

Example 4

In this example, we have used ``wc`` command to find the length of a string.



```
pravalika18@d73d6822db255ef:~$ echo "Length of '$str' is $length"
Length of 'Welcome to Javatpoint' is 22

pravalika18@d73d6822db255ef:~$ touch wc.sh
pravalika18@d73d6822db255ef:~$ chmod +x wc.sh
pravalika18@d73d6822db255ef:~$ gedit wc.sh

(gedit:1855): IBUS-WARNING **: 18:29:15.311: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory
^C
pravalika18@d73d6822db255ef:~$ ./wc.sh
Length of 'Welcome to Javatpoint' is 22
pravalika18@d73d6822db255ef:~$ cat wc.sh
#!/bin/bash
#Bash script to find the length of a string

str="Welcome to Javatpoint"
length=`echo $str | wc -c`

echo "Length of '$str' is $length"
pravalika18@d73d6822db255ef:~$
```

Example 5

In this example, we have used `awk` command to find the length of a string.

```
pravalika18@d73d6822db255ef: ~  
pravalika18@d73d6822db255ef:~$ touch awk.sh  
pravalika18@d73d6822db255ef:~$ chmod +x awk.sh  
pravalika18@d73d6822db255ef:~$ gedit awk.sh  
(gedit:1972): IBUS-WARNING **: 10:30:16.281: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./awk.sh  
Length of 'Welcome to Javatpoint' is 21  
pravalika18@d73d6822db255ef:~$ cat awk.sh  
#!/bin/bash  
#Bash script to find the length of a string  
  
str="Welcome to Javatpoint"  
length=$(echo $str |awk '{print length}')  
echo "Length of '$str' is $length"  
pravalika18@d73d6822db255ef:~$
```

Bash Split String (Split using \$IFS variable)

Example 1: Bash Split String by Space

In this example, a string is split using a space character delimiter.

```
pravalika18@d73d6822db255ef: ~  
pravalika18@d73d6822db255ef:~$ touch space.sh  
pravalika18@d73d6822db255ef:~$ chmod +x space.sh  
pravalika18@d73d6822db255ef:~$ gedit space.sh  
(gedit:543): IBUS-WARNING **: 11:54:55.839: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./space.sh  
Enter any string separated by space: hi pravalika. how are you?  
hi pravalika. how are you?  
pravalika18@d73d6822db255ef:~$ cat space.sh  
#!/bin/bash  
#Example for bash split string by space  
  
read -p "Enter any string separated by space: " str #reading string value  
  
IFS="" #setting space as delimiter  
read -ra ADDR <<<"$str" #reading str as an array as tokens separated by IFS  
  
for i in "${ADDR[@]}"; #accessing each element of array  
do  
    echo "$i"  
done  
pravalika18@d73d6822db255ef:~$
```

Example 2: Bash Split String by Symbol

In this example, a string is split using a comma (,) symbol character as a delimiter.

```
pravalika18@d73d6822db255ef: ~  
done  
pravalika18@d73d6822db255ef:~$ touch symbol.sh  
pravalika18@d73d6822db255ef:~$ chmod +x symbol.sh  
pravalika18@d73d6822db255ef:~$ gedit symbol.sh  
(gedit:689): IBUS-WARNING **: 12:15:45.220: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./symbol.sh  
Enter Name, State and Age separated by a comma: Pravalika ,TS,22  
Name : Pravalika  
State : TS  
Age : 22  
pravalika18@d73d6822db255ef:~$ cat symbol.sh  
#!/bin/bash  
#Example for bash split string by Symbol (comma)  
  
read -p "Enter Name, State and Age separated by a comma: " entry #reading string value  
  
IFS=',' #setting comma as delimiter  
read -a strarr <<<"$entry" #reading str as an array as tokens separated by IFS  
  
echo "Name : ${strarr[0]} "  
echo "State : ${strarr[1]} "  
echo "Age : ${strarr[2]}"  
pravalika18@d73d6822db255ef:~$
```

Split using \$IFS variable

Example 1: Bash Split String by Symbol

```
pravalika18@d73d6822db255ef: ~  
pravalika18@d73d6822db255ef:~$ touch wifs.sh  
pravalika18@d73d6822db255ef:~$ chmod +x wifs.sh  
pravalika18@d73d6822db255ef:~$ gedit wifs.sh  
(gedit:941): IBUS-WARNING **: 12:19:23.230: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./wifs.sh  
Enter any string separated by colon(:) Hii:Pravalika  
  
Hii  
Pravalika  
  
pravalika18@d73d6822db255ef:~$ cat wifs  
cat: wifs: No such file or directory  
pravalika18@d73d6822db255ef:~$ cat wifs.sh  
#!/bin/bash  
#Example for bash split string without $IFS  
  
read -p "Enter any string separated by colon(:) " str #reading string value  
  
readarray -d : -t strarr <<<"$str" #split a string based on the delimiter ':'  
  
printf "\n"  
  
#Print each value of Array with the help of loop  
for (( n=0; n < ${#strarr[*]}; n++ ))  
do  
echo "${strarr[n]}"  
done  
pravalika18@d73d6822db255ef:~$
```

Example 2: Bash Split String by another string

In this example, we have used idiomatic expressions where parameter expansion has completed.

```
pravalika18@d73d6822db255ef: ~
echo "${strarr[n]}"
done
pravalika18@d73d6822db255ef:~$ touch string.sh
pravalika18@d73d6822db255ef:~$ chmod +x string.sh
pravalika18@d73d6822db255ef:~$ gedit string.sh

(gedit:1057): IBUS-WARNING **: 12:21:01.557: Failed to mkdir /home/pravalika18/snap/gedit/604/.config/ibus/bus: Not a directory
^C
pravalika18@d73d6822db255ef:~$ ./string.sh
declare -a array=([0]="We" [1]="Welcome" [2]="You" [3]="On" [4]="Javatpoint")
pravalika18@d73d6822db255ef:~$ cat ./string.sh
#!/bin/bash
#Example for bash split string by another string

str="WeLearnWelcomeLearnYouLearnOnLearnJavatpoint"
delimiter=Learn
s=${str%$delimiter}
array=();
while [[ $s ]];
do
array+=("${s%%$delimiter}*") ;
s=${s%%$delimiter};;
done;
declare -p array
pravalika18@d73d6822db255ef:~$
```

Example 3: Bash Split String using Trim Command

In this example, we have used trim (tr) command to split a string. Instead of using the read command, the trim command is used to split a string on the delimiter.

```
pravalika18@d73d6822db255ef: ~
do
echo $i
done
pravalika18@d73d6822db255ef:~$ touch trim.sh
pravalika18@d73d6822db255ef:~$ chmod +x trim.sh
pravalika18@d73d6822db255ef:~$ gedit trim.sh

(gedit:1292): IBUS-WARNING **: 12:31:12.114: Failed to mkdir /home/pravalika18/snap/gedit/604/.config/ibus/bus: Not a directory
^C
pravalika18@d73d6822db255ef:~$ ./trim.sh
We
welcome
you
on
javatpoint.
pravalika18@d73d6822db255ef:~$ cat trim.sh
#!/bin/bash
# Example to split a string using tr command

my_str="We;welcome;you;on;javatpoint."
my_arr=$(echo $my_str | tr ';' '\n')

for i in "${my_arr[@]}"
do
echo $i
done
pravalika18@d73d6822db255ef:~$
```

Bash Substring

Example 1: To Extract till Specific Characters from Starting

```
pravalika18@d73d6822db255ef:~  
pravalika18@d73d6822db255ef:~$ touch sub1.sh  
pravalika18@d73d6822db255ef:~$ chmod +x sub1.sh  
pravalika18@d73d6822db255ef:~$ gedit sub1.sh  
(gedit:1256): IBUS-WARNING **: 10:30:14.272: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./sub1.sh  
String: We welcome you on Javatpoint.  
Total characters in a String: 29  
Substring: We welcome  
Total characters in Substring: 10  
pravalika18@d73d6822db255ef:~$ cat sub1.sh  
#!/bin/bash  
#Script to extract first 10 characters of a string  
  
echo "String: $str"  
str="We welcome you on Javatpoint."  
  
echo "Total characters in a String: ${#str} "  
  
substr="${str:0:10}"  
  
echo "Substring: $substr"  
echo "Total characters in Substring: ${#substr} "  
pravalika18@d73d6822db255ef:~$
```

Example 2: To Extract from Specific Character onwards

```
pravalika18@d73d6822db255ef:~  
touchpravalika18@d73d6822db255ef:~$ touch ex2.sh  
pravalika18@d73d6822db255ef:~$ chmod +x ex2.sh  
pravalika18@d73d6822db255ef:~$ gedit ex2.sh  
(gedit:580): IBUS-WARNING **: 11:30:53.827: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./ex2.sh  
you on Javatpoint.  
pravalika18@d73d6822db255ef:~$ cat ex2.sh  
#!/bin/bash  
#Script to print from 11th character onwards  
  
str="We welcome you on Javatpoint."  
substr="${str:11}"  
echo "$substr"  
pravalika18@d73d6822db255ef:~$
```

Example 3: To Extract a Single Character

```
pravalika18@d73d6822db255ef: ~  
pravalika18@d73d6822db255ef:~$ touch sub3.sh  
pravalika18@d73d6822db255ef:~$ chmod +x sub3.sh  
pravalika18@d73d6822db255ef:~$ gedit sub3.sh  
(gedit:737): IBUS-WARNING **: 11:52:02.802: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
** (gedit:737): WARNING **: 11:52:29.327: atk-bridge: get_device_events_reply: unknown signature  
^C  
pravalika18@d73d6822db255ef:~$ ./sub3.sh  
-bash: ./sub3.sh: No such file or directory  
pravalika18@d73d6822db255ef:~$ ./sub3.sh  
y  
pravalika18@d73d6822db255ef:~$ cat sub3.sh  
#!/bin/bash  
#Script to print 11th character of a String  
  
str="We welcome you on Javatpoint."  
substr="${str:11:1}"  
echo "$substr"  
pravalika18@d73d6822db255ef:~$
```

Example 4: To Extract the specific characters from last

```
pravalika18@d73d6822db255ef: ~  
echo "$substr"  
pravalika18@d73d6822db255ef:~$ touch sub4.sh  
pravalika18@d73d6822db255ef:~$ chmod +x sub4.sh  
pravalika18@d73d6822db255ef:~$ gedit sub4.sh  
(gedit:853): IBUS-WARNING **: 11:53:53.359: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./sub4.sh  
Javatpoint.  
pravalika18@d73d6822db255ef:~$ cat sub4.sh  
#!/bin/bash  
#Script to extract 11 characters from last  
  
str="We welcome you on Javatpoint."  
substr="${str:(-11)}"  
echo "$substr"  
pravalika18@d73d6822db255ef:~$
```

Bash Concatenate String

Example 1: Write Variables Side by Side

This is the basic example of String Concatenation, and we do not need any extra operator or function in this method.

```
pravalika18@d73d6822db255ef: ~  
echo "$substr"  
pravalika18@d73d6822db255ef:~$ touch var.sh  
pravalika18@d73d6822db255ef:~$ chmod +x var.sh  
pravalika18@d73d6822db255ef:~$ gedit var.sh  
  
(gedit:968): IBUS-WARNING **: 11:55:41.411: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./var.sh  
We welcome you on Javatpoint.  
pravalika18@d73d6822db255ef:~$ cat var.sh  
#!/bin/bash  
#Script to Concatenate Strings  
  
#Declaring the first String  
str1="We welcome you"  
  
#Declaring the Second String  
str2=" on Javatpoint."  
  
#Combining first and second string  
str3="$str1$str2"  
  
#Printing a new string by combining both  
echo $str3  
pravalika18@d73d6822db255ef:~$
```

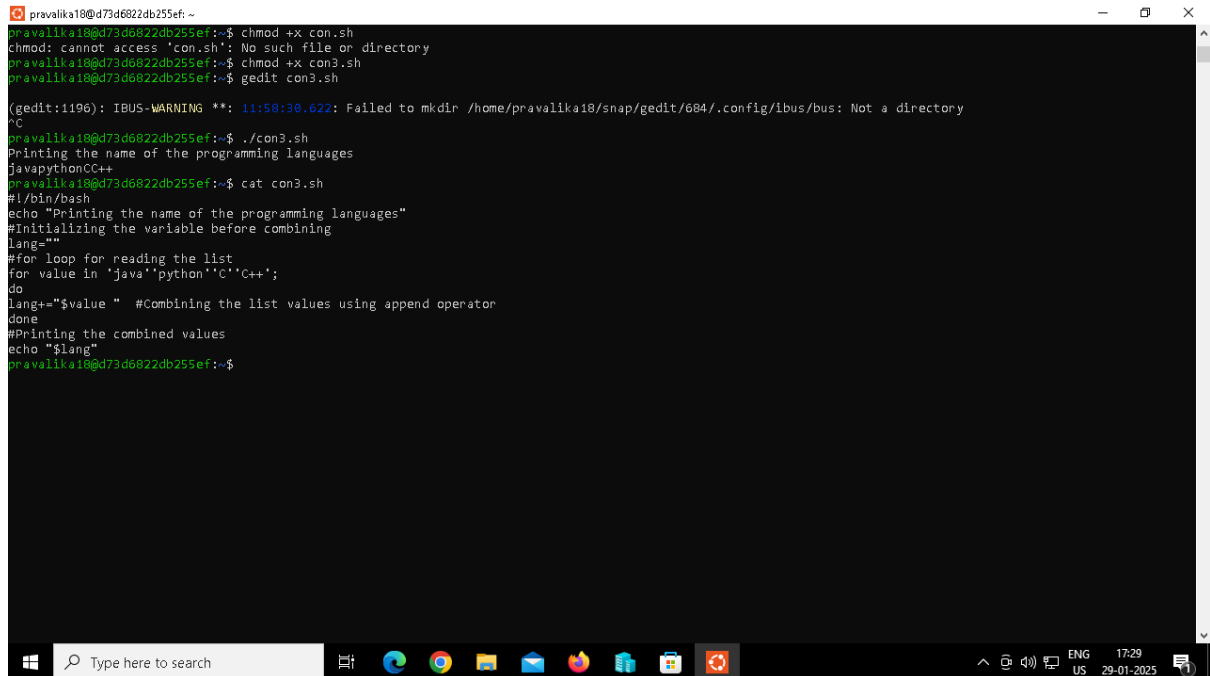
Example 2: Using Double Quotes

We can use variables inside the string, which is defined with double-quotes. The string variable can be applied in any position of the string data.

```
pravalika18@d73d6822db255ef: ~  
#Printing a new string by combining both  
echo $str3  
pravalika18@d73d6822db255ef:~$ touch con2.sh  
pravalika18@d73d6822db255ef:~$ chmod +x con2.sh  
pravalika18@d73d6822db255ef:~$ gedit con2.sh  
  
(gedit:1080): IBUS-WARNING **: 11:57:15.424: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./con2.sh  
We welcome you on Javatpoint.  
pravalika18@d73d6822db255ef:~$ cat con2.sh  
#!/bin/bash  
#Script to Concatenate Strings  
  
#Declaring String Variable  
str="We welcome you"  
  
#Add the variable within the string  
echo "$str on Javatpoint."  
pravalika18@d73d6822db255ef:~$
```


Example 3: Using Append Operator with Loop

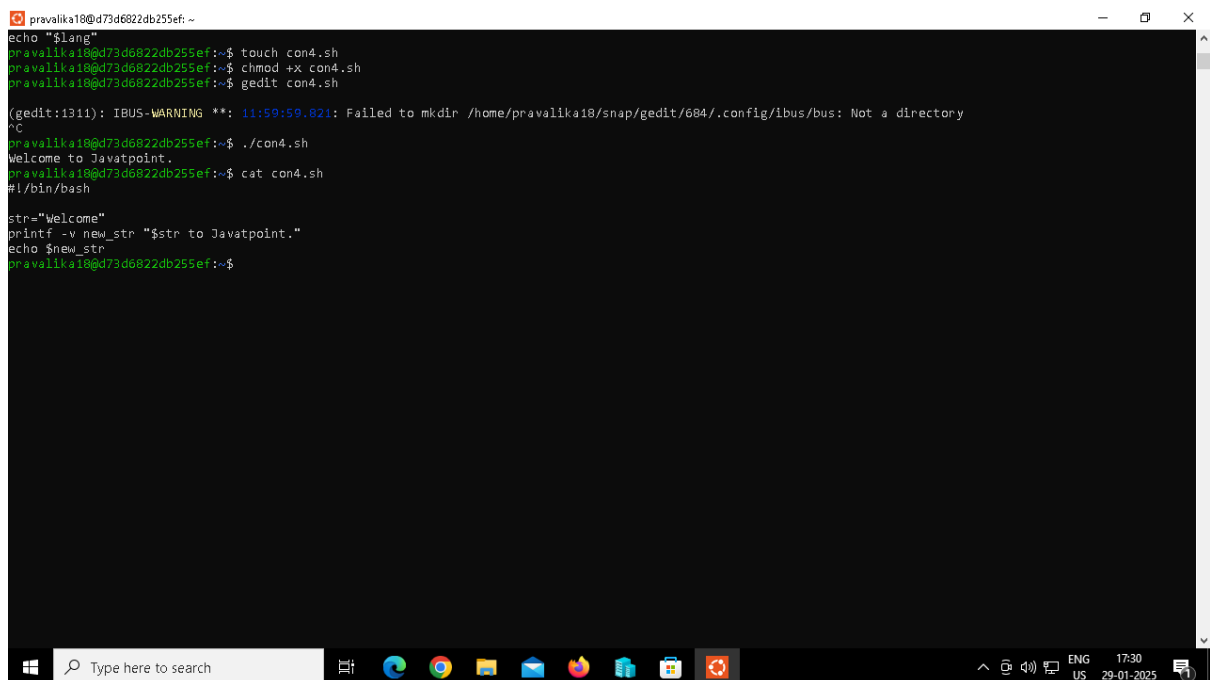
Most of the popular programming languages provide support for append operator (+=) which is the combination of the plus and equal sign. It will add new strings to the end of the string variable.

A terminal window showing the execution of a script named con3.sh. The script prints the name of programming languages and uses a loop with the append operator to combine them into a single string. The output shows the languages 'java', 'python', 'C', and 'C++' being combined into 'java python C C++'.

```
pravalika18@d73d6822db255ef:~  
pravalika18@d73d6822db255ef:~$ chmod +x con.sh  
chmod: cannot access 'con.sh': No such file or directory  
pravalika18@d73d6822db255ef:~$ chmod +x con3.sh  
pravalika18@d73d6822db255ef:~$ gedit con3.sh  
(gedit:1196): IBUS-WARNING **: 11:58:38.622: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./con3.sh  
Printing the name of the programming languages  
javapythonCC++  
pravalika18@d73d6822db255ef:~$ cat con3.sh  
#!/bin/bash  
echo "Printing the name of the programming languages"  
#Initializing the variable before combining  
lang=""  
#for loop for reading the list  
for value in 'java' 'python' 'C' 'C++';  
do  
lang+="$value " #Combining the list values using append operator  
done  
#Printing the combined values  
echo "$lang"  
pravalika18@d73d6822db255ef:~$
```

Example 4: Using the Printf Function

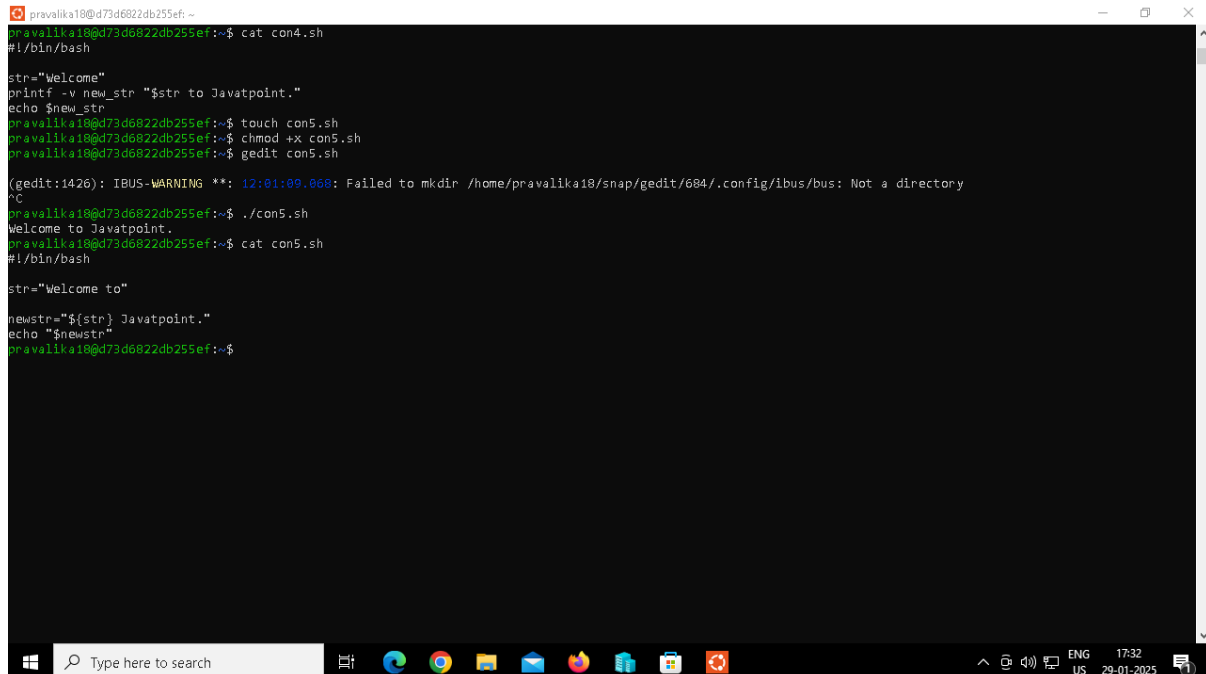
In bash, printf is a function which is used to print and concatenate the strings.

A terminal window showing the execution of a script named con4.sh. The script prints 'Welcome to Javatpoint.' and uses the printf function to concatenate a string variable with the text ' to Javatpoint.'. The output shows 'Welcome to Javatpoint.'.

```
pravalika18@d73d6822db255ef:~  
echo "$lang"  
pravalika18@d73d6822db255ef:~$ touch con4.sh  
pravalika18@d73d6822db255ef:~$ chmod +x con4.sh  
pravalika18@d73d6822db255ef:~$ gedit con4.sh  
(gedit:1311): IBUS-WARNING **: 11:59:59.821: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./con4.sh  
Welcome to Javatpoint.  
pravalika18@d73d6822db255ef:~$ cat con4.sh  
#!/bin/bash  
  
str="Welcome"  
printf -v new_str "$str to Javatpoint."  
echo $new_str  
pravalika18@d73d6822db255ef:~$
```

Example 5: Using Literal Strings

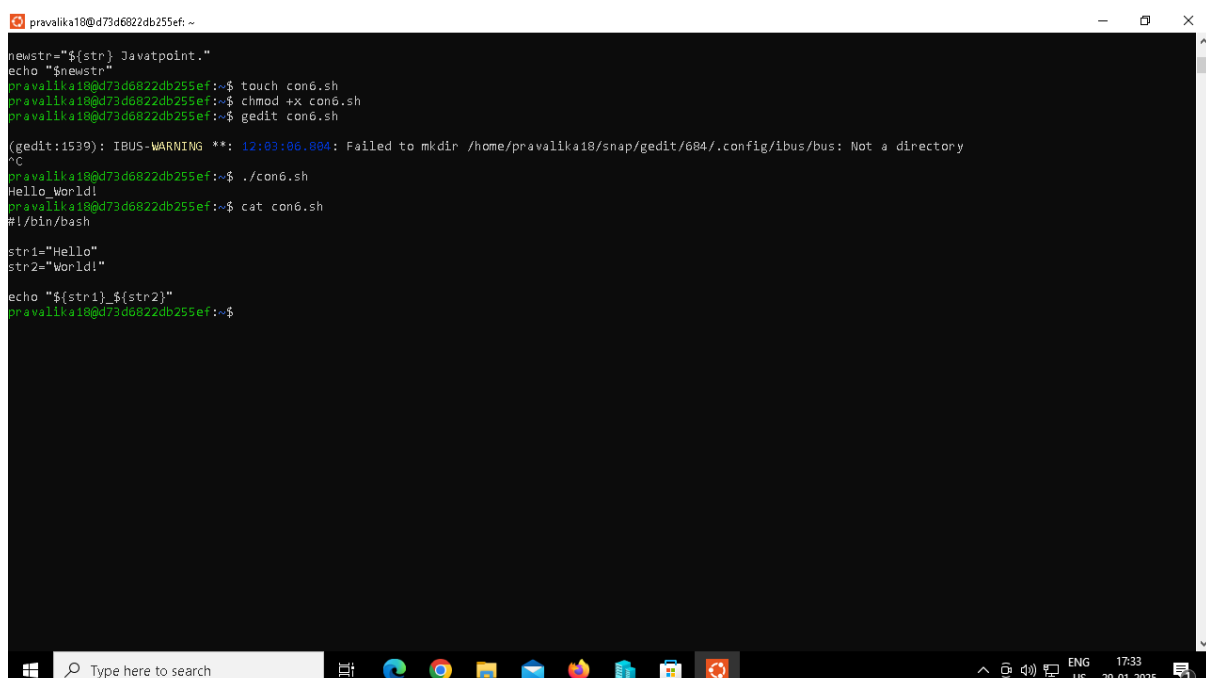
String concatenation can also be performed with a literal string by using curly braces `{}`. They should be used in such a way that the variable does not mix up with the literal string.

A terminal window showing a bash script for string concatenation using curly braces. The script defines a variable 'str' with the value 'Welcome', then uses 'printf' to print 'new_str' followed by 'str' concatenated with a literal string ' Javatpoint.'. The output is 'Welcome to Javatpoint.'. The script also includes file operations: creating 'con5.sh', setting permissions, and editing it with 'gedit'.

```
pravalika18@d73d6822db255ef:~  
pravalika18@d73d6822db255ef:~$ cat con4.sh  
#!/bin/bash  
  
str="Welcome"  
printf -v new_str "$str to Javatpoint."  
echo $new_str  
pravalika18@d73d6822db255ef:~$ touch con5.sh  
pravalika18@d73d6822db255ef:~$ chmod +x con5.sh  
pravalika18@d73d6822db255ef:~$ gedit con5.sh  
(gedit:1426): IBUS-WARNING **: 12:01:09.068: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./con5.sh  
Welcome to Javatpoint.  
pravalika18@d73d6822db255ef:~$ cat con5.sh  
#!/bin/bash  
  
str="Welcome to"  
  
newstr="${str} Javatpoint."  
echo "$newstr"  
pravalika18@d73d6822db255ef:~$
```

Example 6: Using Underscore

Using underscore for concatenating the string in bash shell is one of the common tasks. It is mostly used for assigning a name to the files.

A terminal window showing a bash script for string concatenation using an underscore. The script defines two variables, 'str1' with the value 'Hello' and 'str2' with the value 'World!'. It then uses 'echo' to print the concatenation of 'str1' and 'str2' with an underscore as a separator. The output is 'Hello_World!'. The script also includes file operations: creating 'con6.sh', setting permissions, and editing it with 'gedit'.

```
pravalika18@d73d6822db255ef:~  
newstr="${str} Javatpoint."  
echo "$newstr"  
pravalika18@d73d6822db255ef:~$ touch con6.sh  
pravalika18@d73d6822db255ef:~$ chmod +x con6.sh  
pravalika18@d73d6822db255ef:~$ gedit con6.sh  
(gedit:1539): IBUS-WARNING **: 12:03:06.884: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./con6.sh  
Hello_World!  
pravalika18@d73d6822db255ef:~$ cat con6.sh  
#!/bin/bash  
  
str1="Hello"  
str2="World!"  
  
echo "${str1}_${str2}"  
pravalika18@d73d6822db255ef:~$
```

Example 7: Using any Character

```
pravalika18@d73d6822db255ef:~  
pravalika18@d73d6822db255ef:~$ touch con7.sh  
pravalika18@d73d6822db255ef:~$ chmod +x con7.sh  
pravalika18@d73d6822db255ef:~$ gedit con7.sh  
(gedit:1652): IBUS-WARNING **: 12:04:13.977: Failed to mkdir /home/pravalika18/snap/gedit/604/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./con7.sh  
Enter First Name: Pravalika  
Enter State: TS  
Enter Age: 22  
Name, State, Age: Pravalika,TS,22  
pravalika18@d73d6822db255ef:~$ cat con7.sh  
#!/bin/bash  
#String Concatenation by Character (,) with User Input  
  
read -p "Enter First Name: " name  
read -p "Enter State: " state  
read -p "Enter Age: " age  
  
combine="$name,$state,$age"  
  
echo "Name, State, Age: $combine"  
pravalika18@d73d6822db255ef:~$
```

Bash Functions

Passing Arguments

Following is the code that illustrates the procedure on how to pass arguments to functions, and access the arguments inside the function.

```
pravalika18@d73d6822db255ef:~  
echo "Name, State, Age: $combine"  
pravalika18@d73d6822db255ef:~$ touch pass.sh  
pravalika18@d73d6822db255ef:~$ chmod +x pass.sh  
pravalika18@d73d6822db255ef:~$ gedit pass.sh  
(gedit:1766): IBUS-WARNING **: 12:14:07.325: Failed to mkdir /home/pravalika18/snap/gedit/604/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./pass.sh  
Welcome you on Javatpoint.  
  
pravalika18@d73d6822db255ef:~$ cat pass.sh  
#!/bin/bash  
#Script to pass and access arguments  
  
function_arguments()  
{  
    echo $1  
    echo $2  
    echo $3  
    echo $4  
    echo $5  
}  
  
#Calling function_arguments  
function_arguments "we""welcome""you""on""Javatpoint."  
pravalika18@d73d6822db255ef:~$
```

Variable Scope

```
pravalika18@d73d6822db255ef:~  
function arguments "We""welcome""you""on""Javatpoint."  
pravalika18@d73d6822db255ef:~$ touch scope.sh  
pravalika18@d73d6822db255ef:~$ chmod +x scope.sh  
pravalika18@d73d6822db255ef:~$ gedit scope.sh  
(gedit:1882): IBUS-WARNING **: 12:16:27.626: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ cat scope.sh  
#!/bin/bash  
  
v1='A'  
v2='B'  
  
my_var () {  
    local v1='C'  
    v2='D'  
    echo "Inside Function"  
    echo "v1 is $v1."  
    echo "v2 is $v2."  
}  
  
echo "Before Executing the Function"  
echo "v1 is $v1."  
echo "v2 is $v2."  
  
my_var  
echo "After Executing the Function"  
echo "v1 is $v1."  
echo "v2 is $v2."  
pravalika18@d73d6822db255ef:~$ ./scope.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.  
pravalika18@d73d6822db255ef:~$
```

```
pravalika18@d73d6822db255ef:~  
my_var () {  
    local v1='C'  
    v2='D'  
    echo "Inside Function"  
    echo "v1 is $v1."  
    echo "v2 is $v2."  
}  
  
echo "Before Executing the Function"  
echo "v1 is $v1."  
echo "v2 is $v2."  
  
my_var  
echo "After Executing the Function"  
echo "v1 is $v1."  
echo "v2 is $v2."  
pravalika18@d73d6822db255ef:~$ ./scope.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.  
pravalika18@d73d6822db255ef:~$
```

Return Values

```
pravalika18@d73d6822db255ef:~$ touch return.sh
pravalika18@d73d6822db255ef:~$ chmod +x return.sh
pravalika18@d73d6822db255ef:~$ gedit return.sh

(gedit:1997): IBUS-WARNING **: 12:19:10.262: Failed to mkdir /home/pravalika18/snap/gedit/604/.config/ibus/bus: Not a directory
^C
pravalika18@d73d6822db255ef:~$ ./return.sh
Hello User
Hello Reader
The previous function returned a value of 5
pravalika18@d73d6822db255ef:~$ cat return.sh
#!/bin/bash
#Setting up a return status for a function

print_it () {
    echo Hello $1
    return 5
}

print_it User
print_it Reader
echo The previous function returned a value of $?
pravalika18@d73d6822db255ef:~$
```

Overriding Commands

Example

In this example, we have overridden the 'echo' command and added the time stamp in the form of the argument to the 'echo' command.

```
pravalika18@d73d6822db255ef:~$ print_it Reader
echo The previous function returned a value of $?
pravalika18@d73d6822db255ef:~$ touch override.sh
pravalika18@d73d6822db255ef:~$ chmod +x override.sh
pravalika18@d73d6822db255ef:~$ gedit override.sh

(gedit:2111): IBUS-WARNING **: 12:20:35.339: Failed to mkdir /home/pravalika18/snap/gedit/604/.config/ibus/bus: Not a directory
** (gedit:2111): WARNING **: 12:21:01.810: atk-bridge: get_device_events_reply: unknown signature
^C
pravalika18@d73d6822db255ef:~$ ^Override
pravalika18@d73d6822db255ef:~$ ./override.sh
[01-29 12:22:57] : Welcome to Javatpoint.
pravalika18@d73d6822db255ef:~$ cat override.sh
#!/bin/bash
#Script to override command using function

echo () {
    builtin echo -n `date +"[%m-%d %H:%M:%S]"` " : "
    builtin echo $1
}

echo "Welcome to Javatpoint."
pravalika18@d73d6822db255ef:~$
```

BASH ARRAY

1.let's print an element of an array with an index of 2:

```
pravalika18@d73d6822db255ef: ~  
pravalika18@d73d6822db255ef:~$ touch index.sh  
pravalika18@d73d6822db255ef:~$ chmod +x index.sh  
pravalika18@d73d6822db255ef:~$ gedit index.sh  
(gedit:1405): IBUS-WARNING **: 06:34:46.366: Failed to mkdir /home/pravalika18/snap/gedit/604/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./index.sh  
WelcomeToJavatpoint  
pravalika18@d73d6822db255ef:~$ cat index.sh  
#!/bin/bash  
#Script to print all the elements of the array  
  
#declaring the array  
declare -a example_array=( "Welcome""To""Javatpoint" )  
  
#Printing all the elements  
echo "${example_array[@]}"  
pravalika18@d73d6822db255ef:~$
```

2. Printing the Keys of an Array

```
pravalika18@d73d6822db255ef: ~  
pravalika18@d73d6822db255ef:~$ touch keys.sh  
pravalika18@d73d6822db255ef:~$ chmod +x keys.sh  
pravalika18@d73d6822db255ef:~$ gedit keys.sh  
(gedit:1557): IBUS-WARNING **: 06:36:22.604: Failed to mkdir /home/pravalika18/snap/gedit/604/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./keys.sh  
0  
pravalika18@d73d6822db255ef:~$ cat keys.sh  
#!/bin/bash  
#Script to print the keys of the array  
  
#Declaring the Array  
declare -a example_array=( "Welcome""To""Javatpoint" )  
  
#Printing the Keys  
echo "${!example_array[@]}"  
pravalika18@d73d6822db255ef:~$
```

3. Finding Array Length

```
pravalika18@d73d6822db255ef: ~
#Printing the keys
echo "${!example_array[@]}"
pravalika18@d73d6822db255ef:~$ touch length.sh
pravalika18@d73d6822db255ef:~$ chmod +x length.sh
pravalika18@d73d6822db255ef:~$ gedit length.sh

(gedit:1701): IBUS-WARNING **: 06:53:39.483: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory
^C
pravalika18@d73d6822db255ef:~$ ./length.sh
The array contains 1 elements
pravalika18@d73d6822db255ef:~$ cat length.sh
#!/bin/bash

#Declaring the Array
declare -a example_array=( "Welcome""To""Javatpoint" )

#Printing Array Length
echo "The array contains ${#example_array[@]} elements"
pravalika18@d73d6822db255ef:~$
```

4. Adding Elements to an Array

```
pravalika18@d73d6822db255ef: ~
pravalika18@d73d6822db255ef:~$ touch add.sh
pravalika18@d73d6822db255ef:~$ chmod +x add.sh
pravalika18@d73d6822db255ef:~$ gedit add.sh

(gedit:1883): IBUS-WARNING **: 06:39:34.648: Failed to mkdir /home/pravalika18/snap/gedit/684/.config/ibus/bus: Not a directory
^C
pravalika18@d73d6822db255ef:~$ ./add.sh
JavaPythonPHPHTML JavaScript
pravalika18@d73d6822db255ef:~$ cat add.sh
#!/bin/bash

#Declaring an array
declare -a example_array=( "Java""Python""PHP""HTML" )

#Adding new element
example_array[4]="JavaScript"

#Printing all the elements
echo "${example_array[@]}"
pravalika18@d73d6822db255ef:~$
```

5. Updating Array Element

```
pravalika18@d73d6822db255ef: ~  
example_array[4]="JavaScript"  
  
#Printing all the elements  
echo ${example_array[@]}  
pravalika18@d73d6822db255ef:~$ touch update.sh  
pravalika18@d73d6822db255ef:~$ chmod +x update.sh  
pravalika18@d73d6822db255ef:~$ gedit update.sh  
  
(gedit:2040): IBUS-WARNING **: 06:41:20.468: Failed to mkdir /home/pravalika18/snap/gedit/604/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./update.sh  
WelcomeyouonSSSIT Javatpoint  
pravalika18@d73d6822db255ef:~$ cat update.sh  
#!/bin/bash  
#Script to update array element  
  
#Declaring the array  
declare -a example_array=( "We""welcome""you""on""SSSIT" )  
  
#Updating the Array Element  
example_array[4]=Javatpoint  
  
#Printig all the elements of the Array  
echo ${example_array[@]}  
pravalika18@d73d6822db255ef:~$
```

6. Deleting an Element from an Array

```
pravalika18@d73d6822db255ef: ~  
#Printig all the elements of the Array  
echo ${example_array[@]}  
pravalika18@d73d6822db255ef:~$ touch delete.sh  
pravalika18@d73d6822db255ef:~$ chmod +x delete.sh  
pravalika18@d73d6822db255ef:~$ gedit delete.sh  
  
(gedit:2205): IBUS-WARNING **: 06:43:30.373: Failed to mkdir /home/pravalika18/snap/gedit/604/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./delete.sh  
JavaPythonHTMLCSSJavaScript  
pravalika18@d73d6822db255ef:~$ cat delete.sh  
#!/bin/bash  
#Script to delete the element from the array  
  
#Declaring the array  
declare -a example_array=( "Java""Python""HTML""CSS""JavaScript" )  
  
#Removing the element  
unset example_array[1]  
  
#Printing all the elements after deletion  
echo ${example_array[@]}  
pravalika18@d73d6822db255ef:~$
```


7. Slice Array Elements

```
pravalika18@d73d6822db255ef:~  
pravalika18@d73d6822db255ef:~$ chmod +x slice.sh  
pravalika18@d73d6822db255ef:~$ gedit slice.sh  
(gedit:2350): IBUS-WARNING **: 07:01:05.794: Failed to mkdir /home/pravalika18/snap/gedit/604/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./slice.sh  
pravalika18@d73d6822db255ef:~$ gedit slice.sh  
(gedit:2496): IBUS-WARNING **: 06:46:06.768: Failed to mkdir /home/pravalika18/snap/gedit/604/.config/ibus/bus: Not a directory  
^C  
pravalika18@d73d6822db255ef:~$ ./slice.sh  
Python  
HTML  
CSS  
pravalika18@d73d6822db255ef:~$ cat slice.sh  
#!/bin/bash  
# Script to slice Array Element from index 1 to index 3  
  
# Declaring the Array  
example_array=("Java" "Python" "HTML" "CSS" "JavaScript")  
  
# Slicing the Array  
sliced_array=("${example_array[@]:1:3}")  
  
# Applying for loop to iterate over each element in the Array  
for i in "${sliced_array[@]}"  
do  
    echo $i  
done  
pravalika18@d73d6822db255ef:~$
```

