

BASH PROJECT

Case statement 1

```
samim@9819fd877cfb527:~$ sudo nano case1.sh
[sudo] password for samim:
samim@9819fd877cfb527:~$ sudo chmod +x case1.sh
samim@9819fd877cfb527:~$ ./case1.sh
Do you know Java Programming?
Yes/No? :yes
That's amazing.
```

```

samim@9819fd877cfb527: ~
GNU nano 7.2
#!/bin/bash

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

Bash case 2

```
samim@9819fd877cfb527:~$ sudo nano case2.sh
samim@9819fd877cfb527:~$ sudo chmod +x case2.sh
samim@9819fd877cfb527:~$ ./case2.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.

samim@9819fd877cfb527:~$ ./case2.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.

samim@9819fd877cfb527:~$ ./case2.sh
Which Operating System are you using?
Windows, Android, Chrome, Linux, Others?
Type your OS Name:boom baam
Sounds interesting. I will try that.
```

BASH PROJECT

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/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
```

For loop 1

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/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."
```

```
samim@9819fd877cfb527:~$ ./for1.sh  
Start  
learning  
from  
Javatpoint.  
Thank You.
```

BASH PROJECT

For loop 2

```
samim@9819fd877cfb527:~$ sudo nano for2.sh
[sudo] password for samim:
samim@9819fd877cfb527:~$ sudo chmod +x for2.sh
samim@9819fd877cfb527:~$ ./for2.sh
1
2
3
4
5
6
7
8
9
10
```

```

samim@9819fd877cfb527: ~
GNU nano 7.2
#!/bin/bash
#For Loop to Read a Range with Increment
for num in {1..10..1}
do
echo $num
done
```

For loop 3

```
samim@9819fd877cfb527:~$ sudo nano for3.sh
samim@9819fd877cfb527:~$ sudo chmod +x for3.sh
samim@9819fd877cfb527:~$ ./for3.sh
10
9
8
7
6
5
4
3
2
1
0
```

```

samim@9819fd877cfb527: ~
GNU nano 7.2
#!/bin/bash
#For Loop to Read a Range with Decrement.
for num in {10..0..1}
do
echo $num
done
```

BASH PROJECT

For loop 4

```
samim@9819fd877cfb527:~$ ./for4.sh
Welcome
to
Javatpoint
```

```
samim@9819fd877cfb527: ~
GNU nano 7.2
#!/bin/bash
#Array Declaration
arr=( "Welcome" "to" "Javatpoint" )
for i in "${arr[@]}"
do
echo $i
done
```

For loop 5

```
samim@9819fd877cfb527:~$ sudo nano for5.sh
samim@9819fd877cfb527:~$ sudo chmod +x for5.sh
samim@9819fd877cfb527:~$ ./for5.sh
Let's
start
learning
from
Javatpoint.
```

```
samim@9819fd877cfb527: ~
GNU nano 7.2
#!/bin/bash
#For Loop to Read white spaces in String as word separators
str="Let's start
learning from Javatpoint."
for i in $str;
do
echo "$i"
done
```

For loop 6

```
samim@9819fd877cfb527:~$ sudo nano for6.sh
samim@9819fd877cfb527:~$ sudo chmod +x for6.sh
samim@9819fd877cfb527:~$ ./for6.sh
Let's start
learning from
Javatpoint.
```

BASH PROJECT

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/bin/bash  
#For Loop to Read each line in String as a word  
str="Let's start  
learning from  
Javatpoint."  
for i in "$str";  
do  
echo "$i"  
done
```

For loop 7

```
samim@9819fd877cfb527:~$ sudo nano for7.sh  
samim@9819fd877cfb527:~$ sudo chmod +x for7.sh  
samim@9819fd877cfb527:~$ ./for7.sh  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/bin/bash  
#For Loop to Read Three-expression  
for ((i=1; i<=10; i++))  
do  
echo "$i"  
done
```

For loop 8

BASH PROJECT

```
samim@9819fd877cfb527:~$ sudo nano for8.sh
samim@9819fd877cfb527:~$ sudo chmod +x for8.sh
samim@9819fd877cfb527:~$ ./for8.sh
2
4
6
8
10
12
14
16
18
20
```

```
GNU nano 7.2
#!/bin/bash
#Table of 2.
for table in {2..100..2}
do
echo $table
if [ $table == 20 ]; then
break
fi
done
```

For loop 9

```
samim@9819fd877cfb527:~$ sudo nano for9.sh
samim@9819fd877cfb527:~$ sudo chmod +x for9.sh
samim@9819fd877cfb527:~$ ./for9.sh
1
2
3
4
5
16
17
18
19
20
```

BASH PROJECT

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/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
```

For loop 10

```
samim@9819fd877cfb527:~$ sudo nano for10.sh  
samim@9819fd877cfb527:~$ sudo chmod +x for10.sh  
samim@9819fd877cfb527:~$ ./for10.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
```

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/bin/bash  
i=1;  
for (( ; ; ))  
do  
sleep 1s  
echo "Current Number: $((i++))"  
done
```

BASH PROJECT

While loop 1

```
samim@9819fd877cfb527:~$ sudo nano while1.sh
[sudo] password for samim:
samim@9819fd877cfb527:~$ sudo chmod +x while1.sh
samim@9819fd877cfb527:~$ ./while1.sh
Enter starting number: 5
Enter ending number: 10
5
6
7
8
9
10
This is the sequence that you wanted.
samim@9819fd877cfb527:~$
```

```
samim@9819fd877cfb527: ~
GNU nano 7.2
#!/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."
```

While loop 2

```
samim@9819fd877cfb527:~$ sudo nano while2.sh
samim@9819fd877cfb527:~$ ./while2.sh
Enter starting number: 1
Enter ending number: 5
1
2
3
4
5
This is the sequence that you wanted.
```

```
samim@9819fd877cfb527: ~
GNU nano 7.2
#!/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."
```


BASH PROJECT

While loop 3

```
44047 Welcome to Javatpoint.
44048 Welcome to Javatpoint.
44049 Welcome to Javatpoint.^C
samim@9819fd877cfb527:~$ sudo nano while3.sh
samim@9819fd877cfb527:~$ sudo chmod +x while3.sh
samim@9819fd877cfb527:~$ ./while3.sh
```

```
samim@9819fd877cfb527: ~
GNU nano 7.2
#!/bin/bash
#An infinite while loop
i=0
while ;; do echo "$((i++)) Welcome to Javatpoint."; done
```

While loop 4

```
samim@9819fd877cfb527:~$ sudo nano while4.sh
samim@9819fd877cfb527:~$ sudo chmod +x while4.sh
samim@9819fd877cfb527:~$ ./while4.sh
Countdown for Website Launching...
10
9
8
7
6
5
4
3
Mission Aborted, Some Technical Error Found.
```

```
samim@9819fd877cfb527: ~
GNU nano 7.2
#!/bin/bash
#While Loop Example with a Break Statement
echo "Countdown for Website Launching..."
i=10
while [ $i -ge 1 ]
do
if [ $i == 2 ]
then
echo "Mission Aborted, Some Technical Error Found."
break
fi
echo "$i"
(( i-- ))
done
```

BASH PROJECT

While loop 5

```
samim@9819fd877cfb527:~$ sudo nano while5.sh
samim@9819fd877cfb527:~$ sudo chmod +x while5.sh
samim@9819fd877cfb527:~$ ./while5.sh
1
2
3
4
5
6
7
8
9
10
```

```
samim@9819fd877cfb527: ~
GNU nano 7.2
#!/bin/bash
#While loop example in C style
i=1
while((i <= 10))
do
echo $i
let i++
done
```

Until loop 1

```
samim@9819fd877cfb527:~$ sudo nano until1.sh
[sudo] password for samim:
samim@9819fd877cfb527:~$ sudo chmod +x until1.sh
samim@9819fd877cfb527:~$ ./until1.sh
1
2
3
4
5
6
7
8
9
10
```

```
samim@9819fd877cfb527: ~
GNU nano 7.2
#!/bin/bash
#Bash Until Loop example with a single condition
i=1
until [ $i -gt 10 ]
do
echo $i
((i++))
done
```

BASH PROJECT

Until loop 2

```
samim@9819fd877cfb527:~$ sudo nano until2.sh
samim@9819fd877cfb527:~$ sudo chmod +x until2.sh
samim@9819fd877cfb527:~$ ./until2.sh
a = 1 & b = 0.
a = 2 & b = 1.
a = 3 & b = 2.
a = 4 & b = 3.
a = 5 & b = 4.
```

```
samim@9819fd877cfb527: ~
GNU nano 7.2
#!/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
```

Program to check strings are equal or not

```
samim@9819fd877cfb527:~$ sudo nano str1.sh
[sudo] password for samim:
Sorry, try again.
[sudo] password for samim:
samim@9819fd877cfb527:~$ sudo chmod +x str1.sh
samim@9819fd877cfb527:~$ ./str
str.sh  str1.sh
samim@9819fd877cfb527:~$ ./str1.sh
Strings are not equal.
```

```
samim@9819fd877cfb527: ~
GNU nano 7.2
#!/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
```

BASH PROJECT

Program to check if strings are not equal

```
samim@9819fd877cfb527:~$ sudo nano str2.sh
samim@9819fd877cfb527:~$ sudo chmod +x str2.sh
samim@9819fd877cfb527:~$ ./str2.sh
Strings are not equal.
```

```

samim@9819fd877cfb527: ~
GNU nano 7.2
#!/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
```

Less than operator

```
samim@9819fd877cfb527:~$ sudo nano str3.sh
samim@9819fd877cfb527:~$ sudo chmod +x str3.sh
samim@9819fd877cfb527:~$ ./str3.sh
WelcometoJavatpoint is not less then Javatpoint
```

```

samim@9819fd877cfb527: ~
GNU nano 7.2
#!/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
```

Greater than operator

BASH PROJECT

```
samim@9819fd877cfb527:~$ sudo nano str4.sh
[sudo] password for samim:
samim@9819fd877cfb527:~$ sudo chmod +x str4.sh
samim@9819fd877cfb527:~$ ./str4.sh
WelcometoJavatpoint is greater then Javatpoint
```

```
🔄 samim@9819fd877cfb527: ~
GNU nano 7.2
#!/bin/sh
str1="WelcometoJavatpoint"
str2="Javatpoint"
if [ $str1 \> $str2 ];
then
echo "$str1 is greater then $str2"
else
echo "$str1 is less then $str2"
fi
```

To check is string length is greater than 0

```
samim@9819fd877cfb527:~$ sudo nano str5.sh
[sudo] password for samim:
samim@9819fd877cfb527:~$ sudo chmod +x str5.sh
samim@9819fd877cfb527:~$ ./str5.sh
String is not empty
```

```
🔄 samim@9819fd877cfb527: ~
GNU nano 7.2
#!/bin/sh
str="WelcometoJavatpoint"
if [ -n $str ];
then
echo "String is not empty"
else
echo "String is empty"
fi
```

To check if the string is empty

BASH PROJECT

```
samim@9819fd877cfb527:~$ sudo nano string6.sh
[sudo] password for samim:
samim@9819fd877cfb527:~$ sudo chmod +x string6.sh
samim@9819fd877cfb527:~$ ./string6.sh
String is not empty
```

```
samim@9819fd877cfb527: ~
GNU nano 7.2
#!/bin/sh
str="WelcometoJavatpoint"
if [ -n $str ];
then
echo "String is not empty"
else
echo "String is empty"
fi
```

Program to calculate length of a string using #

```
samim@9819fd877cfb527:~$ sudo nano string7.sh
samim@9819fd877cfb527:~$ sudo chmod +x string7.sh
samim@9819fd877cfb527:~$ ./string7.sh
Length of 'Welcome to Javatpoint' is 21
```

```
samim@9819fd877cfb527: ~
GNU nano 7.2
#!/bin/bash
#Bash program to find the length of a string


str="Welcome to Javatpoint"
length=${#str}

echo "Length of '$str' is $length"
```

Program to calculate length of a string using expr

BASH PROJECT

```
samim@9819fd877cfb527:~$ sudo nano string8.sh
samim@9819fd877cfb527:~$ sudo chmod +x string8.sh
samim@9819fd877cfb527:~$ ./string8.sh
Length of 'Welcome to Javatpoint' is 21
```

 samim@9819fd877cfb527: ~

```
GNU nano 7.2
#!/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"
```

Using expr \$str

```
samim@9819fd877cfb527:~$ sudo nano string9.sh
samim@9819fd877cfb527:~$ sudo chmod +x string9.sh
samim@9819fd877cfb527:~$ ./string9.sh
Length of 'Welcome to Javatpoint' is 21
```

 samim@9819fd877cfb527: ~

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

#Bash script to find the length of a string

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

echo "Length of '$str' is $length"
```

Using wc command

```
samim@9819fd877cfb527:~$ sudo nano string10.sh
samim@9819fd877cfb527:~$ sudo chmod +x string10.sh
samim@9819fd877cfb527:~$ ./string10.sh
Length of 'Welcome to Javatpoint' is 22
```

BASH PROJECT

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/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"
```

Using awk command

```
samim@9819fd877cfb527:~$ sudo nano string11.sh  
samim@9819fd877cfb527:~$ sudo chmod +x string11.sh  
samim@9819fd877cfb527:~$ ./string11.sh  
Length of 'Welcome to Javatpoint' is 21
```

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/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"
```

Splitting a string using delimiter

```
samim@9819fd877cfb527:~$ sudo nano string12.sh  
samim@9819fd877cfb527:~$ ./string12.sh  
Enter any string separated by space: hi hello  
hi  
hello
```

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/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
```


BASH PROJECT

Splitting a string using symbol

```
samim@9819fd877cfb527:~$ ./string13.sh
Enter Name, State and Age separated by a comma: Samim, Karnataka, 23
Name : Samim
State : Karnataka
Age : 23
```

```
samim@9819fd877cfb527: ~
GNU nano 7.2 string13.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]}"
```

Separating string using \$IFS

```
samim@9819fd877cfb527:~$ sudo nano string14.sh
samim@9819fd877cfb527:~$ sudo chmod +x string14.sh
samim@9819fd877cfb527:~$ ./string14.sh
Enter any string separated by colon(:) Abdul:Samim:Mondal

Abdul
Samim
Mondal
```

```
samim@9819fd877cfb527: ~
GNU nano 7.2 str
#!/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
```

Splitting a string with a string

```
samim@9819fd877cfb527:~$ sudo nano string14.sh
[sudo] password for samim:
samim@9819fd877cfb527:~$ sudo nano string15.sh
samim@9819fd877cfb527:~$ sudo chmod +x string15.sh
samim@9819fd877cfb527:~$ ./string15.sh
declare -a array=( [0]="We" [1]="Welcome" [2]="You" [3]="On" [4]="Javatpoint")
```

BASH PROJECT

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/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
```

Splitting a string using trim command

```
samim@9819fd877cfb527:~$ sudo nano string16.sh  
samim@9819fd877cfb527:~$ ./string16.sh  
We  
welcome  
you  
on  
javatpoint.
```

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/bin/bash  
# Example to split a string using trim (tr) command  
my_str="We;welcome;you;on;javatpoint."  
my_arr=( $(echo "$my_str" | tr ';' '\n') ) # Replacing ';' with newlines  
  
for i in "${my_arr[@]}"  
do  
    echo "$i"  
done
```

To extract a string till a specific character from starting

```
samim@9819fd877cfb527:~$ sudo nano string17.sh  
samim@9819fd877cfb527:~$ sudo chmod +x string17.sh  
samim@9819fd877cfb527:~$ ./string17.sh  
String: We welcome you on Javatpoint.  
Total characters in a String: 29  
Substring: We welcome  
Total characters in Substring: 10
```

BASH PROJECT

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/bin/bash  
#Script to extract first 10 characters of a string  
echo "String: We welcome you on Javatpoint."  
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} "
```

To extract from specific character onwards

```
samim@9819fd877cfb527:~$ sudo nano string18.sh  
samim@9819fd877cfb527:~$ sudo chmod +x string18.sh  
samim@9819fd877cfb527:~$ ./string18.sh  
you on Javatpoint.
```

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/bin/bash  
#Script to print from 11th character onwards  
str="We welcome you on Javatpoint."  
substr="${str:11}"  
echo "$substr"
```

To extract a single character

```
samim@9819fd877cfb527:~$ sudo nano string19.sh  
samim@9819fd877cfb527:~$ sudo chmod +x string19.sh  
samim@9819fd877cfb527:~$ ./string19.sh  
y
```

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/bin/bash  
#Script to print 11th character of a String  
str="We welcome you on Javatpoint."  
substr="${str:11:1}"  
echo "$substr"
```

To extract the specific characters from last

BASH PROJECT

```
samim@9819fd877cfb527:~$ sudo nano string20.sh
samim@9819fd877cfb527:~$ sudo chmod +x string20.sh
samim@9819fd877cfb527:~$ ./string20.sh
Javatpoint.
```

```

samim@9819fd877cfb527: ~
GNU nano 7.2
#!/bin/bash
#Script to print 11 characters from last
str="We welcome you on Javatpoint."
substr="${str:(-11)}"
echo "$substr"
```

Basic string concatenation

```
samim@9819fd877cfb527:~$ sudo nano string21.sh
samim@9819fd877cfb527:~$ sudo chmod +x string21.sh
samim@9819fd877cfb527:~$ ./string21.sh
We welcome you on Javatpoint.
```

```

samim@9819fd877cfb527: ~
GNU nano 7.2
#!/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
```

Using variables inside string

```
samim@9819fd877cfb527:~$ sudo nano string22.sh
samim@9819fd877cfb527:~$ sudo chmod +x string22.sh
samim@9819fd877cfb527:~$ ./string22.sh
We welcome you on Javatpoint.
```

```

samim@9819fd877cfb527: ~
GNU nano 7.2
#!/bin/bash
#Script to Concatenate Strings
#Declaring String Variable
str="We welcome you"
#Add the variable within the string
echo "$str on Javatpoint."
```

BASH PROJECT

Using append with loop

```
samim@9819fd877cfb527:~$ sudo nano string23.sh
samim@9819fd877cfb527:~$ sudo chmod +x string23.sh
samim@9819fd877cfb527:~$ ./string23.sh
Printing the name of the programming languages
javapythonCC++
```

```

samim@9819fd877cfb527: ~
GNU nano 7.2
#!/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"
```

Using printf function

```
samim@9819fd877cfb527:~$ sudo nano string24.sh
samim@9819fd877cfb527:~$ sudo chmod +x string24.sh
samim@9819fd877cfb527:~$ ./string24.sh
Welcome to Javatpoint.
```

```

samim@9819fd877cfb527: ~
GNU nano 7.2
#!/bin/bash
str="Welcome"
printf -v new_str "$str to Javatpoint."
echo $new_str
```

Using literal strings

```
samim@9819fd877cfb527:~$ sudo nano string25.sh
samim@9819fd877cfb527:~$ sudo chmod +x string25.sh
samim@9819fd877cfb527:~$ ./string25.sh
Welcome to Javatpoint.
```

```

samim@9819fd877cfb527: ~
GNU nano 7.2
#!/bin/bash
str="Welcome to"
newstr="${str} Javatpoint."
echo "$newstr"
```

BASH PROJECT

Using undrescores

```
samim@9819fd877cfb527:~$ sudo nano string26.sh
samim@9819fd877cfb527:~$ sudo chmod +x string26.sh
samim@9819fd877cfb527:~$ ./string26.sh
Hello_World!
```

```
GNU nano 7.2
#!/bin/bash
str1="Hello"
str2="World!"
echo "${str1}_${str2}"
```

Using any character

```
samim@9819fd877cfb527:~$ sudo nano string27.sh
samim@9819fd877cfb527:~$ sudo chmod +x string27.sh
samim@9819fd877cfb527:~$ ./string27.sh
Enter First Name: Abdul
Enter State: Karnataka
Enter Age: 22
Name, State, Age: Abdul,Karnataka,22
```

```
GNU nano 7.2
#!/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"
```

Print element of an array with index 2

```
samim@9819fd877cfb527:~$ sudo nano arr1.sh
samim@9819fd877cfb527:~$ ./arr1.sh
Javatpoint
```

```
GNU nano 7.2
#!/bin/bash
#Script to print an element of an array with an index of 2
#declaring the array
declare -a example_array=( "Welcome" "To" "Javatpoint" )
#printing the element with index of 2
echo ${example_array[2]}
```

BASH PROJECT

To print all elements of an array

```
samim@9819fd877cfb527:~$ sudo nano arr2.sh
samim@9819fd877cfb527:~$ sudo chmod +x arr2.sh
samim@9819fd877cfb527:~$ ./arr2.sh
Welcome To Javatpoint
```

```

samim@9819fd877cfb527: ~
GNU nano 7.2
#!/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[@]}"
```

Printing keys

```
samim@9819fd877cfb527:~$ sudo nano arr3.sh
samim@9819fd877cfb527:~$ sudo chmod +x arr3.sh
samim@9819fd877cfb527:~$ ./arr3.sh
0 1 2
```

```

samim@9819fd877cfb527: ~
GNU nano 7.2
#!/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[@]}"
```

To print array length

```
samim@9819fd877cfb527:~$ sudo nano arr4.sh
samim@9819fd877cfb527:~$ sudo chmod +x arr4.sh
samim@9819fd877cfb527:~$ ./arr4.sh
The array contains 3 elements
```

BASH PROJECT

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/bin/bash  
#Declaring the Array  
declare -a example_array=( "Welcome" "To" "Javatpoint" )  
#Printing Array Length  
echo "The array contains ${#example_array[@]} elements"
```

To loop through an array

```
samim@9819fd877cfb527: ~$ sudo nano arr5.sh  
samim@9819fd877cfb527:~$ sudo chmod +x arr5.sh  
samim@9819fd877cfb527:~$ ./arr5.sh  
The key value of element Welcome is 0  
The key value of element To is 1  
The key value of element Javatpoint is 2
```

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/bin/bash  
#Script to print all keys and values using loop through the array  
declare -a example_array=( "Welcome" "To" "Javatpoint" )  
#Array Loop  
for i in "${!example_array[@]}"  
do  
echo The key value of element "${example_array[$i]}" is "$i"  
done
```

To retrieve length of an array and use c style loop

```
samim@9819fd877cfb527:~$ sudo nano arr6.sh  
samim@9819fd877cfb527:~$ sudo chmod +x arr6.sh  
samim@9819fd877cfb527:~$ ./arr6.sh  
0 Welcome  
1 To  
2 Javatpoint
```


BASH PROJECT

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/bin/bash  
#Script to loop through an array in C-style  
declare -a example_array=( "Welcome" "To" "Javatpoint" )  
#Length of the Array  
length=${#example_array[@]}  
#Array Loop  
for (( i=0; i < ${length}; i++ ))  
  
do  
echo $i ${example_array[$i]}  
done
```

Adding elements to an array

```
samim@9819fd877cfb527:~$ sudo nano arr7.sh  
samim@9819fd877cfb527:~$ sudo chmod +x arr7.sh  
samim@9819fd877cfb527:~$ ./arr7.sh  
Java Python PHP HTML JavaScript
```

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/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[@]}"
```

Adding elements using +=

```
samim@9819fd877cfb527:~$ sudo nano arr8.sh  
samim@9819fd877cfb527:~$ sudo chmod +x arr8.sh  
samim@9819fd877cfb527:~$ ./arr8.sh  
Java Python PHP JavaScript CSS SQL
```

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/bin/bash  
#Declaring the Array  
declare -a example_array=( "Java" "Python" "PHP" )  
#Adding new elements  
example_array+=( JavaScript CSS SQL )  
#Printing all the elements  
echo "${example_array[@]}"
```

BASH PROJECT

Updating array element

```
samim@9819fd877cfb527:~$ sudo nano arr9.sh
samim@9819fd877cfb527:~$ sudo chmod +x arr9.sh
samim@9819fd877cfb527:~$ ./arr9.sh
We welcome you on Javatpoint
```

```
samim@9819fd877cfb527: ~
GNU nano 7.2
#!/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[@]}"
```

Deleting an array element

```
samim@9819fd877cfb527:~$ sudo nano arr10.sh
samim@9819fd877cfb527:~$ sudo chmod +x arr10.sh
samim@9819fd877cfb527:~$ ./arr10.sh
Java HTML CSS JavaScript
```

```
samim@9819fd877cfb527: ~
GNU nano 7.2
#!/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[@]}"
```

Deleting an entire array

```
samim@9819fd877cfb527:~$ sudo nano arr11.sh
samim@9819fd877cfb527:~$ sudo chmod +x arr11.sh
samim@9819fd877cfb527:~$ ./arr11.sh
```

BASH PROJECT

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/bin/bash  
#Script to delete the entire Array  
#Declaring the Array  
declare -a example_array=( "Java" "Python" "HTML" "CSS" "JavaScript" )  
#Deleting Entire Array  
unset example_array  
#Printing the Array Elements  
echo ${!example_array[@]}  
#Printing the keys  
echo ${!example_array[@]}
```

Slice array elements

```
samim@9819fd877cfb527:~$ sudo nano arr12.sh  
samim@9819fd877cfb527:~$ sudo chmod +x arr12.sh  
samim@9819fd877cfb527:~$ ./arr12.sh  
Python  
HTML  
CSS
```

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/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=("${example_array[@]:1:3}")  
#Applying for loop to iterate over each element in Array  
for i in "${sliced_array[@]}"  
do  
echo $i  
done
```

Basic function

```
samim@9819fd877cfb527:~$ sudo nano fun1.sh  
samim@9819fd877cfb527:~$ sudo chmod +x fun1.sh  
samim@9819fd877cfb527:~$ ./fun1.sh  
Welcome to Javatpoint.
```

BASH PROJECT

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/bin/bash  
JTP () {  
echo 'Welcome to Javatpoint.'  
}  
JTP
```

Passing argument to a function

```
samim@9819fd877cfb527:~$ sudo nano fun2.sh  
samim@9819fd877cfb527:~$ ./fun2.sh  
We  
welcome  
you  
on  
Javatpoint.
```

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/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."
```

Variable scope

```
samim@9819fd877cfb527:~$ sudo nano fun3.sh  
samim@9819fd877cfb527:~$ sudo chmod +x fun3.sh  
samim@9819fd877cfb527:~$ ./fun3.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.
```

BASH PROJECT

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/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."
```

Return a value

```
samim@9819fd877cfb527:~$ sudo nano fun4.sh  
samim@9819fd877cfb527:~$ sudo chmod +x fun4.sh  
samim@9819fd877cfb527:~$ ./fun4.sh  
Hello User  
Hello Reader  
The previous function returned a value of 5
```

```
samim@9819fd877cfb527: ~  
GNU nano 7.2  
#!/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 $?
```

Overriding commands

BASH PROJECT

```
samim@9819fd877cfb527:~$ sudo nano fun5.sh  
samim@9819fd877cfb527:~$ sudo chmod +x fun5.sh  
samim@9819fd877cfb527:~$ ./fun5.sh  
[01-30 04:45:25] : Welcome to Javatpoint.
```