

BASH SCRIPTING PROJECT

CASE STATEMENT

1.A Shell Program to define a simple scenario to demonstrate the use of the ‘Case Statement’.

Step 1: Create an “case.sh” script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0:~# touch case.sh
root@9a4a8a5799315e0:~# nano case.sh
root@9a4a8a5799315e0:~# chmod +x case.sh
```

Step 3: Write the Code in nano case.sh script file

```
root@9a4a8a5799315e0: ~
GNU nano 7.2 case.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."
echo
;;
No|no|N|n)
echo "It's easy. Let's start learning from javatpoint."
echo
;;
esac
```

Step 4: Output

```
root@9a4a8a5799315e0:~# ./case.sh
Do you know Java Programming?
Yes/No? :yes
That's amazing.

root@9a4a8a5799315e0:~# ./case.sh
Do you know Java Programming?
Yes/No? :no
It's easy. Let's start learning from javatpoint.
```

2.A Shell Program to define a combined scenario to demonstrate the use of the ‘Case Statement’.

Step 1: Create an “case1.sh” script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0:~# touch case1.sh
root@9a4a8a5799315e0:~# nano case1.sh
root@9a4a8a5799315e0:~# chmod +x case1.sh
```

Step 3: Write the Code in nano case1.sh.sh script file

```
root@9a4a8a5799315e0: ~
GNU nano 7.2 case1.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
```

Step 4: Output

```
root@9a4a8a5799315e0:~# ./case1.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.

root@9a4a8a5799315e0:~# ./case1.sh
Which Operating System are you using?
Windows, Android, Chrome, Linux, Others?
Type your OS Name:linux
You might be serious about security!!

root@9a4a8a5799315e0:~# ./case1.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.

root@9a4a8a5799315e0:~# ./case1.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.
```

```
root@9a4a8a5799315e0:~# ./case1.sh
Which Operating System are you using?
Windows, Android, Chrome, Linux, Others?
Type your OS Name:Cent OS
Sounds interesting. I will try that.
```

FOR LOOP

3.A Shell Program to demonstrate the use of 'For Loop'.

Step 1: Create an "forloop.sh" script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0:~# touch forloop.sh
root@9a4a8a5799315e0:~# nano forloop.sh
root@9a4a8a5799315e0:~# chmod +x forloop.sh
```

Step 3: Write the Code in nano forloop.sh script file

```
root@9a4a8a5799315e0: ~
GNU nano 7.2 forloop.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."_
```

Step 4: Output

```
root@9a4a8a5799315e0:~# ./forloop.sh
Start
learning
from
Javatpoint.
Thank You.
```

4.A Shell Program to demonstrate the use of 'For Loop' to read a range.

Step 1: Create an "forloop1.sh" script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0: ~  
root@9a4a8a5799315e0:~# touch forloop1.sh  
root@9a4a8a5799315e0:~# nano forloop1.sh  
root@9a4a8a5799315e0:~# chmod +x forloop1.sh
```

Step 3: Write the Code in nano forloop1.sh script file

```
root@9a4a8a5799315e0: ~  
GNU nano 7.2 forloop1.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."_
```

Step 4: Output

```
root@9a4a8a5799315e0:~# ./forloop1.sh  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
Series of numbers from 1 to 10.
```

5.A Shell Program to demonstrate the use of ‘For Loop to read a range with Increment’.

Step 1: Create an “forloop2.sh” script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0:~# touch forloop2.sh  
root@9a4a8a5799315e0:~# nano forloop2.sh  
root@9a4a8a5799315e0:~# chmod +x forloop2.sh
```

Step 3: Write the Code in nano forloop2.sh.sh script file

```
Select root@9a4a8a5799315e0: ~  
GNU nano 7.2 forloop2.sh  
#!/bin/bash  
#For Loop to Read a Range with Increment  
for num in {1..10..1}  
do  
echo $num  
done
```

Step 4: Output

```
root@9a4a8a5799315e0:~# ./forloop2.sh  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

6. A Shell Program to demonstrate the use of ‘For Loop to read a range with Decrement’.

Step 1: Create an “forloop3.sh” script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0:~# touch forloop3.sh
root@9a4a8a5799315e0:~# nano forloop3.sh
root@9a4a8a5799315e0:~# chmod +x forloop3.sh
```

Step 3: Write the Code in nano forloop3.sh script file

```
Select root@9a4a8a5799315e0: ~
GNU nano 7.2                                forloop3.sh *
#!/bin/bash
#For Loop to Read a Range with Decrement
for num in {10..0..1}
do
echo $num
done
```

Step 4: Output

```
root@9a4a8a5799315e0:~# ./forloop3.sh
10
9
8
7
6
5
4
3
2
1
0
```

7.A Shell Program to demonstrate the use of ‘For Loop’ to iterate over elements of an array.

Step 1: Create an “forloop4.sh” script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0:~# touch forloop4.sh
root@9a4a8a5799315e0:~# nano forloop4.sh
root@9a4a8a5799315e0:~# chmod +x forloop4.sh
```

Step 3: Write the Code in nano forloop4.sh script file

```
root@9a4a8a5799315e0: ~
GNU nano 7.2                                forloop4.sh *
#!/bin/bash
#Array Declaration
arr=( "Welcome" "to" "Javatpoint" )
for i in "${arr[@]}"
do
echo $i
done
```

Step 4: Output

```
root@9a4a8a5799315e0:~# ./forloop4.sh
Welcome
to
Javatpoint
root@9a4a8a5799315e0:~# _
```

8.A Shell Program to demonstrate the use of ‘For Loop’ to read white spaces in string as word separators.

Step 1: Create an “forloop4.1.sh” script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0:~# touch forloop4.1.sh
root@9a4a8a5799315e0:~# nano forloop4.1.sh
root@9a4a8a5799315e0:~# chmod +x forloop4.1.sh
```

Step 3: Write the Code in nano forloop4.sh script file

```
root@9a4a8a5799315e0: ~
GNU nano 7.2                                forloop4.1.sh *
#!/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
```

Step 4: Output

```
root@9a4a8a5799315e0:~# ./forloop4.1.sh
Let's
start
learning
from
Javatpoint.
root@9a4a8a5799315e0:~# _
```

9. A Shell Program to define 'For Loop' to read each line in string as a word.

Step 1: Create an "forloop5.sh" script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0:~# touch forloop5.sh
root@9a4a8a5799315e0:~# nano forloop5.sh
root@9a4a8a5799315e0:~# chmod +x forloop5.sh
```

Step 3: Write the Code in nano forloop5.sh script file

```
root@9a4a8a5799315e0: ~
GNU nano 7.2                                forloop5.sh *
#!/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
```

Step 4: Output

```
root@9a4a8a5799315e0:~# ./forloop5.sh
Let's start
learning from
javatpoint.
```

10. A Shell Program to define 'For Loop' to read three-expression.

Step 1: Create an "forloop6.sh" script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0:~# touch forloop6.sh
root@9a4a8a5799315e0:~# nano forloop6.sh
root@9a4a8a5799315e0:~# chmod +x forloop6.sh
```

Step 3: Write the Code in nano forloop6.sh script file

```
root@9a4a8a5799315e0: ~
GNU nano 7.2                                forloop6.sh *
#!/bin/bash
#For Loop to Read Three-expression
for ((i=1;i<=10;i++ ))
do
echo "$i"
done
```

Step 4: Output

```
root@9a4a8a5799315e0:~# ./forloop6.sh
1
2
3
4
5
6
7
8
9
10
```

11.A Shell Program to define a 'For Loop with the Break Statement'.

Step 1: Create an "forloop7.sh" script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0:~# touch forloop7.sh
root@9a4a8a5799315e0:~# nano forloop7.sh
root@9a4a8a5799315e0:~# chmod +x forloop7.sh
```

Step 3: Write the Code in nano forloop7.sh script file

```
root@9a4a8a5799315e0: ~
GNU nano 7.2 forloop7.sh *
#!/bin/bash
#Table of 2
for table in {2..100..2}
do
echo $table
if [$table==20]; then
break
fi
done_
```

Step 4: Output

```
root@9a4a8a5799315e0:~# ./forloop7.sh
2
4
6
8
10
12
14
16
18
20
```

12.A Shell Program to define 'For Loop with a Continue Statement'.

Step 1: Create an "forloop8.sh" script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0:~# touch forloop8.sh
root@9a4a8a5799315e0:~# nano forloop8.sh
root@9a4a8a5799315e0:~# chmod +x forloop8.sh
```

Step 3: Write the Code in nano forloop8.sh script file

```
root@9a4a8a5799315e0: ~
GNU nano 7.2 forloop8.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
```

Step 4: Output

```
root@9a4a8a5799315e0:~# ./forloop8.sh
1
2
3
4
5
16
17
18
19
20
```

13.A Shell Program to define a ‘Infinite bash For Loop’.

Step 1: Create an “forloop9.sh” script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0:~# touch forloop9.sh
root@9a4a8a5799315e0:~# nano forloop9.sh
root@9a4a8a5799315e0:~# chmod +x forloop9.sh
```

Step 3: Write the Code in nano forloop9.sh script file

```
root@9a4a8a5799315e0: ~
GNU nano 7.2                                forloop9.sh *
#!/bin/bash
i=1;
for (( ; ; ))
do
sleep 1s
echo "Current Number: ${i++}"
done
```

Step 4: Output

```
root@9a4a8a5799315e0:~# ./forloop9.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
^C
```

WHILE LOOP

14.A Shell Program to define ‘While Loop’ to print series of numbers as per user input.

Step 1: Create an “whileloop.sh” script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0:~# touch whileloop.sh
root@9a4a8a5799315e0:~# nano whileloop.sh
root@9a4a8a5799315e0:~# chmod +x whileloop.sh
```

Step 3: Write the Code in nano whileloop.sh script file

```
root@9a4a8a5799315e0: ~
GNU nano 7.2                                whileloop.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."
```

Step 4: Output

```
root@9a4a8a5799315e0:~# ./whileloop.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.
```

15.A Shell Program to define a ‘While Loop’ with multiple conditions

Step 1: Create an “whileloop1.sh” script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0:~# touch whileloop1.sh
root@9a4a8a5799315e0:~# nano whileloop1.sh
root@9a4a8a5799315e0:~# chmod +x whileloop1.sh
```

Step 3: Write the Code in nano whileloop1.sh script file

```
root@9a4a8a5799315e0: ~
GNU nano 7.2                               whileloop1.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."
```

Step 4: Output

```
root@9a4a8a5799315e0:~# ./whileloop1.sh
Enter starting number: 11
Enter ending number: 20
11
12
13
14
15
16
17
18
19
20
This is the sequence that you wanted.
```

16.A Shell Program to define a ‘Infinite bash While Loop’.

Step 1: Create an “whileloop2.sh” script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0:~# touch whileloop2.sh
root@9a4a8a5799315e0:~# nano whileloop2.sh
root@9a4a8a5799315e0:~# chmod +x whileloop2.sh
```

Step 3: Write the Code in nano whileloop2.sh script file

```
root@9a4a8a5799315e0: ~
GNU nano 7.2                               whileloop2.sh *
#!/bin/bash
#An infinite while loop
while :
do
echo "Welcome to Javatpoint."
done
```


Step 4: Output

```
root@9a4a8a5799315e0: ~  
Welcome to Javatpoint.  
Welcome to Javatpoint.  
Welcome to Javatpoint.  
Welcome to Javatpoint.  
Welcome to Javatpoint.  
Welcome to Javatpoint.  
Welcome to Javatpoint.  
Welcome to Javatpoint.  
Welcome to Javatpoint.  
Welcome to Javatpoint.  
Welcome to Javatpoint.  
Welcome to Javatpoint.  
Welcome to Javatpoint.  
Welcome to Javatpoint.  
Welcome to Javatpoint.  
Welcome to Javatpoint.  
Welcome to Javatpoint.  
Welcome to Javatpoint.
```

17.A Shell Program to define a ‘Infinite bash While Loop’.

Step 1: Create an “whileloop3.sh” script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0:~# touch whileloop3.sh  
root@9a4a8a5799315e0:~# nano whileloop3.sh  
root@9a4a8a5799315e0:~# chmod +x whileloop3.sh
```

Step 3: Write the Code in nano whileloop3.sh script file

```
root@9a4a8a5799315e0: ~  
GNU nano 7.2                               whileloop3.sh *  
#!/bin/bash  
#An infinite while loop  
while true  
do  
echo "Welcome to Javatpoint"  
done
```

Step 4: Output

```
root@9a4a8a5799315e0: ~  
Welcome to Javatpoint  
Welcome to Javatpoint  
Welcome to Javatpoint  
Welcome to Javatpoint  
Welcome to Javatpoint  
Welcome to Javatpoint  
Welcome to Javatpoint  
Welcome to Javatpoint  
Welcome to Javatpoint  
Welcome to Javatpoint  
Welcome to Javatpoint  
Welcome to Javatpoint  
Welcome to Javatpoint  
Welcome to Javatpoint  
Welcome to Javatpoint  
Welcome to Javatpoint
```

18.A Shell Program to define a ‘While Loop with a Break Statement’.

Step 1: Create an “whileloop4.sh” script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0:~# touch whileloop4.sh  
root@9a4a8a5799315e0:~# nano whileloop4.sh  
root@9a4a8a5799315e0:~# chmod +x whileloop4.sh
```

Step 3: Write the Code in nano whileloop4.sh script file

```
root@9a4a8a5799315e0: ~  
GNU nano 7.2                               whileloop4.sh *  
#!/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
```

Step 4: Output

```
root@9a4a8a5799315e0:~# ./whileloop4.sh
Countdown for Website Launching...
10
9
8
7
6
5
4
3
Mission Aborted, Some Technical Error Found.
```

19.A Shell Program to define a ‘While Loop with a Continue Statement’.

Step 1: Create an “whileloop5.sh” script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0:~# touch whileloop5.sh
root@9a4a8a5799315e0:~# nano whileloop5.sh
root@9a4a8a5799315e0:~# chmod +x whileloop5.sh
```

Step 3: Write the Code in nano whileloop5.sh script file

```
root@9a4a8a5799315e0: ~
GNU nano 7.2                               whileloop5.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."
```

Step 4: Output

```
root@9a4a8a5799315e0:~# ./whileloop5.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.
```

20.A Shell Program to define a while loop in bash script as similar as a ‘While Loop in C programming language’.

Step 1: Create an “whileloop6.sh” script file using touch command

Step 2: Create a nano file to write the code

```
root@9a4a8a5799315e0:~# touch whileloop6.sh
root@9a4a8a5799315e0:~# nano whileloop6.sh
root@9a4a8a5799315e0:~# chmod +x whileloop6.sh
```

Step 3: Write the Code in nano whileloop6.sh script file

```
root@9a4a8a5799315e0: ~
GNU nano 7.2                               whileloop6.sh *
#!/bin/bash
#While loop example in C style
i=1
while((i <= 10))
do
  echo $i
  let i++
done
```

Step 4: Output

```
root@9a4a8a5799315e0:~# ./whileloop6.sh
1
2
3
4
5
6
7
8
9
10
```