

1. Write a shell script that prompts the user to enter a number. Check if the number is either between 20 and 30 (inclusive) or less than 10, and is not equal to 2. Display "Number meets the criteria" if both conditions are true. Otherwise, display "Number does not meet the criteria."

Solution:

```
#!/bin/bash
echo "Enter a number:"
read number
if { [ $number -ge 20 ] && [ $number -le 30 ] || [ $number -lt 10 ]; } && [ $number -ne 2 ]; then
    echo "Number meets the criteria."
else
    echo "Number does not meet the criteria."
fi
```

There are 4 type of loops in shell scripting.

The While Loop

The for Loop

The select loop

The until loop

While Loop

❑ Example

```
n=1
while [ $n -le 10 ]; do
    echo $n
    n=$(( n+1 ))          or    n=$(( n+1 )) or (( n++ )) or (( ++n ))
done
```

**Write a shell script that uses a while loop to print the numbers from 1 to 5.**

```
#!/bin/bash
counter=1
while [ $counter -le 5 ]; do
    echo $counter
    counter=$((counter + 1))
done
```

**Exercise:**

Write a shell script that uses a while loop to calculate the sum of numbers from 1 to 10 and display the result.

```
#!/bin/bash
sum=0
counter=1
# Calculate the sum
while [ $counter -le 10 ]; do
    sum=$((sum + counter))
    counter=$((counter + 1))
done
echo "The sum of numbers from 1 to 10 is: $sum"
```

### **For Loop**

```
for (( initiate; condition; increment/decrement ))
do
    statement
done
```

**Write a shell script that uses a for loop to print all 1 to 20 numbers.**

```
#!/bin/bash

for number in {1..20}; do
    echo $number
done
```

Or

```
#!/bin/bash
```

```
for ((number =1; number <=20; number++)); do
```

```
    echo $number
```

```
done
```

### **Exercise:**

Write a shell script that uses nested for loops to generate and print a multiplication table for numbers 1 through 5.

### **Output:**

1 2 3 4 5

2 4 6 8 10

3 6 9 12 15

4 8 12 16 20

5 10 15 20 25

```
#!/bin/bash
for i in {1..5}; do
    for j in {1..5}; do
        echo -n "$((i * j)) "
    done
    echo
done
```

**Exercise:** Write a shell script to display left-handed triangle shape using “\*”

```
#!/bin/bash
for ((i=1; i<=5; i++)); do
    for ((j=1; j<= $i; j++)); do
        echo -n $i          [echo -n: Ensures that the * symbols are printed on the same line.]
    done
    echo
done
```

```
for command in ls pwd date; do
    echo "Command Name: $command"
    echo "Command Output:"
    $command
done
```

Command Name: ls

Command Output:

file1.txt

file2.txt

Command Name: pwd

Command Output:

/home/user

Command Name: date

Command Output:

Fri Jan 4 12:34:56 UTC 2025

### Select Loop With Cases

select variable in alal dulal rahim

do

case \$variable in

alal ) # Case alal

echo "Alal is selected"

;;

dulal ) # Case dulal

echo "Dulal is selected"

;;

rahim ) # Case rahim

echo "Rahim is selected"

;;

\* ) # Default case

echo "Default"

;;

esac

done

**Output:**

- 1) alal
- 2) dulal
- 3) rahim
- #?

**User input: 1**

**Output:** Alal is selected



## **Until Loop:**

```
until [ condition ] ; do  
    statement  
done
```

## **Example**

```
#!/bin/sh  
a=0  
until [ ! $a -lt 10 ]; do  
    echo $a  
    a=`expr $a + 1`  
done
```

**Output:**

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

Write a shell script that uses an until loop to print the numbers from 1 to 10.

```
#!/bin/bash
counter=1
until [ $counter -gt 10 ]; do
    echo $counter
    counter=$((counter + 1))
done
```

Write a shell script that uses a for loop to print numbers from 1 to 10, but **skip** printing **even numbers** using the continue statement.

```
#!/bin/bash
```

```
for number in {1..10}; do
```

```
    if [ $((number % 2)) -eq 0 ]; then
```

```
        continue
```

```
    fi
```

```
    echo $number
```

```
done
```