

Experiment – 6

Aim: To study the usage of For loop in shell.

- Using a for loop in shell scripting can be handy for iterating through lists of items or performing operations on files.
- In shell scripting, for loops typically follow this syntax:

```
For item in list  
do  
    #commands to be executed for each item  
done
```

1. Echo Basic Manage

```
cat > testfor1.sh
```

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

```
SERVERS="s1 s2 s3"
```

```
for S in $SERVERS; do
```

```
    echo "Updating pkg on: $S"
```

```
done
```

Save as : testfor1.sh

Execute using : bash testfor1.sh

```
localhost:~/Ayushi# cat > testfor1.sh  
#!/bin/bash  
  
SERVERS="s1 s2 s3"  
for S in $SERVERS; do  
    echo "Updating pkg on: $S"  
done  
localhost:~/Ayushi# bash testfor1.sh  
Updating pkg on: s1  
Updating pkg on: s2  
Updating pkg on: s3  
localhost:~/Ayushi#
```

2. Iterating through range of numbers

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

```
For value in {1:5}
```

```
do
```

```
    echo "Number : $value"
```

```
done
```

Save as: testfor2.sh

Execute using: bash testfor2.sh

Output:

```
#!/bin/bash  
for value in {1..5}  
do  
    echo "number : $value"  
done  
localhost:~/Ayushi# bash testfor2.sh  
number : 1  
number : 2  
number : 3  
number : 4  
number : 5
```

3. Iterate on Multiple Files

```
cat > testfor3.sh
```

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

```
for file in /root/*
```

```
do
```

```
    chmod 755 "$file"
```

```
    echo "Updated permission for: $file"
```

```
done
```

Save as: testfor3.sh

Execute using: bash testfor3.sh

```
localhost:~/Ayushi# cat > testfor3.sh  
#!/bin/bash  
for file in /root/*  
do  
    chmod 755 "$file"  
    echo "update permission for : $file"  
done  
localhost:~/Ayushi# bash testfor3.sh  
update permission for : /root/Ayushi  
update permission for : /root/bench.py  
update permission for : /root/hello.c  
update permission for : /root/hello.js  
update permission for : /root/readme.txt  
localhost:~/Ayushi#
```

4. Creating an Infinite Loop

```
cat > testfor4.sh
#!/bin/bash
for(( ; ; ))
do
    echo "This is an infinite loop"
    echo "Use Ctrl+C to stop it"
done
```

Save as: testfor4.sh**Execute using:** bash testfor4.sh

```
localhost:~/Ayushi# rm testfor4.sh
localhost:~/Ayushi# cat > testfor4.sh
#!/bin/bash
for(( ; ; ))
do
    echo "This is infinite loop"
    echo "Use Ctrl+C to stop it"
done
localhost:~/Ayushi# bash testfor4.sh
This is infinite loop
Use Ctrl+C to stop it
This is infinite loop
Use Ctrl+C to stop it
This is infinite loop
```

5. Implementing a Nested for Loop

```
cat > testfor5.sh
#!/bin/bash
for serverd in A B C; do
    for app in apache dp; do
        echo "$serverd can run $app LAMP package"
    done
done
```

Save as: testfor5.sh**Execute using:** bash testfor5.sh

```
localhost:~/Ayushi# cat > testfor5.sh
#!/bin/bash
for serverd in A B C; do
    for app in apache dp; do
        echo "$serverd can run $app LAMP package"
    done
done
localhost:~/Ayushi# bash testfor5.sh
A can run apache LAMP package
A can run dp LAMP package
B can run apache LAMP package
B can run dp LAMP package
C can run apache LAMP package
C can run dp LAMP package
```

6. Use Array in for loop

```
cat > testfor6.sh
#!/bin/bash
apps=("apache" "mysql" "php")
for app in "${apps[@]}"
do
    echo "The application name is $app"
done
```

Save as: testfor6.sh**Execute using:** bash testfor6.sh

```
localhost:~/Ayushi# cat > testfor6.sh
#!/bin/bash

for file in ~/.* ; do
    if [[ "$file" == "./bash.sh" ]]
    then
        echo "$file is available"
        break
    fi
done
localhost:~/Ayushi# bash testfor6.sh
```

7. Use Break in for Loop

```
cat > testfor7.sh
#!/bin/bash
for file in ~/.* ; do
    if [[ "$file" == "./bash.sh" ]]
    then
        echo "$file is available"
        break
    fi
done
```

```
localhost:~/Ayushi# cat > testfor7.sh
#!/bin/bash
for file in ~/.* ; do
    if [[ "$file" == "./bash.sh" ]]
    then
        echo "$file is available"
        break
    fi
done
localhost:~/Ayushi# bash testfor7.sh
localhost:~/Ayushi#
```

8. Use Command Substitution`cat > testfor8.sh``#!/bin/bash``for log in $(cat ~/testfile)``do` `echo "Log entry: $log"``done`**Save as:** testfor8.sh**Execute using:** bash testfor8.sh

```
localhost:~/Ayushi# cat > testfor8.sh
#!/bin/bash

for log in $(cat ~/testfile)
do
    echo "Log entry: $log"
done
localhost:~/Ayushi# bash testfor8.sh
cat: can't open '/root/testfile': No such file or directory
localhost:~/Ayushi#
```