

Experiment 06

Aim: 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 Message

```
#!/bin/bash

SERVERS="s1 s2 s3"
for S in $SERVERS; do
    echo "Updating pkg on: $S"
done
```

```
#!/bin/bash

SERVERS="s1 s2 s3"
for S in $SERVERS; do
    echo "Updating pkg on: $S"
done

localhost:~/aatif# bash test1.sh
Updating pkg on: s1
Updating pkg on: s2
Updating pkg on: s3
```

2. Iterate Range of Numbers

```
#!/bin/bash

for value in {1..20}
do
    echo "Number: $value"
done
```

```
#!/bin/bash

for value in {1..20}
do
    echo "Number: $value"
done
```

Output:

```
localhost:~/aatif# bash test2.sh
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5
Number: 6
Number: 7
Number: 8
Number: 9
Number: 10
Number: 11
Number: 12
Number: 13
Number: 14
Number: 15
Number: 16
Number: 17
Number: 18
Number: 19
Number: 20
```

3. Iterate on Multiple Files

```
#!/bin/bash

for file in /root/*
do
    chmod 755 "$file"
    echo "Updated permission for: $file"
done
```

```
#!/bin/bash

for file in /root/*
do
    chmod 755 "$file"
    echo "Updated permission for: $file"
done
localhost:~/aatif# bash test3.sh
Updated permission for: /root/aatif
Updated permission for: /root/bench.py
Updated permission for: /root/hello.c
Updated permission for: /root/hello.js
Updated permission for: /root/readme.txt
```

4. Create Infinite Loop

```
#!/bin/bash

for (( ; ; ))
do
    echo "This is an infinite loop"
    echo "Use Ctrl+C to stop it"
done
```

```
#!/bin/bash

for (( ; ; ))
do
    echo "This is an infinite loop"
    echo "Use Ctrl+C to stop it"
done

localhost:~/aatif# bash test4.sh
This is an infinite loop
Use Ctrl+C to stop it
This is an infinite loop
Use Ctrl+C to stop it
```

5. Nested for Loop

```
#!/bin/bash

for server in Apache DB
do
    for app in apache php do
        echo "$server can run $app LAMP package"
    done
done
```

```
#!/bin/bash

for server in Apache DB
do
    for app in apache php do
        echo "$server can run $app LAMP package"
    done
done
localhost:~/aatif# bash test5.sh
Apache can run apache LAMP package
Apache can run php LAMP package
Apache can run do LAMP package
DB can run apache LAMP package
DB can run php LAMP package
DB can run do LAMP package
```

6. Use Array in for Loop

```
#!/bin/bash

apps=("apache" "mysql" "php")
for app in "${apps[@]}"
do
    echo "The application name is: $app"
done
```

```
#!/bin/bash

apps=("apache" "mysql" "php")
for app in "${apps[@]}"
do
    echo "The application name is: $app"
done
localhost:~/aatif# bash test6.sh
The application name is: apache
The application name is: mysql
The application name is: php
```

7. Use Break in for Loop

```
#!/bin/bash

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

```
#!/bin/bash

for file in ~/.* ; do
    if [[ "$file" == "./bash.sh" ]]
    then
        echo "$file is available"
        break
    fi
done
localhost:~/aatif# bash test7.sh
```

8. Use Command Substitution

```
#!/bin/bash

for log in $(cat ~/testfile)
do
    echo "Log entry: $log"
done
```

```
#!/bin/bash

for log in $(cat ~/testfile)
do
    echo "Log entry: $log"
done
localhost:~/aatif# bash test8.sh
```