

LINUX

BASH LOOPING

Loops are handy when you want to run a series of commands over and over again until a certain condition is reached. In scripting languages such as Bash, loops are useful for automating repetitive tasks.

Bash Looping

There are total 3 looping statements which can be used in bash programming.

- For statement
- While statement
- Until statement

➤ For statement:

The for loop operate on lists of items. It repeats a set of command for every item in a list.

- Make a directory.
`gedit for.sh` (here for.sh is your file name with bash file extinction)

First method

Input:

```
#!/bin/bash
For i in {1..10}
do
echo "Hi Welcome to $i Time"
done
```

LINUX

Output:

```
ubuntu@mr-pglu:~$ gedit for.sh
ubuntu@mr-pglu:~$ chmod +x for.sh
ubuntu@mr-pglu:~$ ./for.sh
Hi Welcome to 1 Time
Hi Welcome to 2 Time
Hi Welcome to 3 Time
Hi Welcome to 4 Time
Hi Welcome to 5 Time
Hi Welcome to 6 Time
Hi Welcome to 7 Time
Hi Welcome to 8 Time
Hi Welcome to 9 Time
Hi Welcome to 10 Time
ubuntu@mr-pglu:~$
```

(Here chmod +x for.sh used for executable file)

Second method

Input:

```
#!/bin/bash
For i in 1 2 3 4 5
do
echo " Welcome to ADIT for $i time "
done
```

Output:

```
ubuntu@mr-pglu:~$ gedit for.sh
ubuntu@mr-pglu:~$ chmod +x for.sh
ubuntu@mr-pglu:~$ ./for.sh
Welcome to ADIT for 1 time
Welcome to ADIT for 2 time
Welcome to ADIT for 3 time
Welcome to ADIT for 4 time
Welcome to ADIT for 5 time
ubuntu@mr-pglu:~$
```

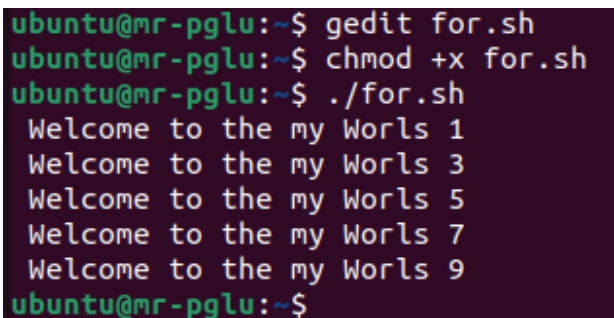
LINUX

Third method

Input:

```
#!/bin/bash
For i in {1..10..2}
do
echo “ Welcome to the my Worls $i ”
done
```

Output:



```
ubuntu@mr-pglu:~$ gedit for.sh
ubuntu@mr-pglu:~$ chmod +x for.sh
ubuntu@mr-pglu:~$ ./for.sh
Welcome to the my Worls 1
Welcome to the my Worls 3
Welcome to the my Worls 5
Welcome to the my Worls 7
Welcome to the my Worls 9
ubuntu@mr-pglu:~$
```

➤ While statement:

while command in Linux is used to repeatedly execute a set of command as long as the *COMMAND* returns true. The test command is given and all other commands are executed till the given command's result satisfies, when the command's result become false, the control will be out from the while command.

- Make a directory.
gedit while.sh (here while.sh is your file name with bash file extinction)

Input:

```
#!/bin/bash
i=0
while [ $i -le 10 ]
do
echo “ True ”
```

LINUX

```
((i++))  
done  
echo " False "
```

Output:

```
ubuntu@mr-pglu:~$ gedit while.sh  
ubuntu@mr-pglu:~$ chmod +x while.sh  
ubuntu@mr-pglu:~$ ./while.sh  
True  
True  
True  
True  
True  
True  
True  
True  
True  
True  
True  
True  
True  
False  
ubuntu@mr-pglu:~$
```

➤ Until statement:

until command in Linux used to execute a set of commands as long as the final command in the 'until' Commands has an exit status which is not zero. It is mostly used where the user needs to execute a set of commands until a condition is true.

- Make a directory.
gedit until.sh (here until.sh is your file name with bash file extinction)

Input:

```
#!/bin/bash  
Count=10  
i=20  
until [ $i -lt $Count ]  
do  
echo " False "  
((i--))
```

LINUX

done

echo " True "

Output:

```
ubuntu@mr-pglu:~$ gedit until.sh
ubuntu@mr-pglu:~$ chmod +x until.sh
ubuntu@mr-pglu:~$ ./until.sh
False
False
False
False
False
False
False
False
False
False
False
False
True
```