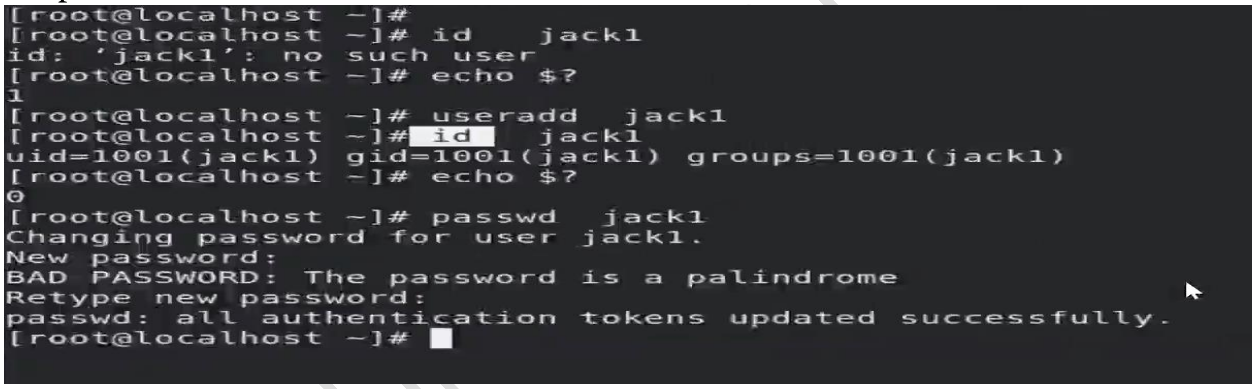
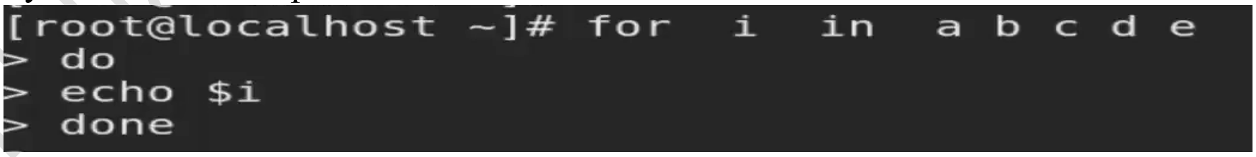
Shell and Shell Scripting Session No.1.3

- In the previous class we have learnt how we can pass multiple arguments for the script but now we want to perform some operation on each argument like useradd etc.
- So, for this operation we will use the iteration process. We are going to perform a user operation. We will put the below command inside the script, "id" command will first check whether the user exists or not, and with the help of the status code we create some conditions and create a script that will add the user if it does not exist.



```
[root@localhost ~]#
[root@localhost ~]# id   jack1
id: 'jack1': no such user
[root@localhost ~]# echo $?
1
[root@localhost ~]# useradd  jack1
[root@localhost ~]# id   jack1
uid=1001(jack1) gid=1001(jack1) groups=1001(jack1)
[root@localhost ~]# echo $?
0
[root@localhost ~]# passwd  jack1
Changing password for user jack1.
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]#
```

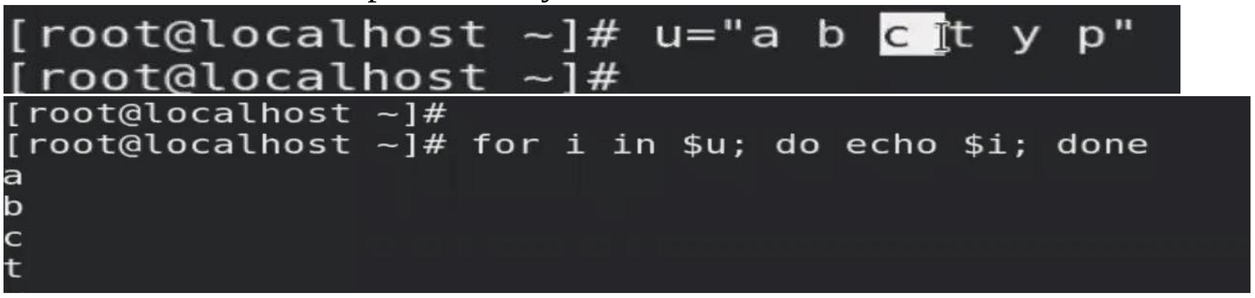
- What is Iteration? -> Iteration is the process that allows you to iterate over the collection of the data like Lists or tuples. So here if we pass multiple arguments in the command it will first go to the 1st element to operate and then go to the 2nd element and perform the operation and so on.
- And for iteration we will majorly use the "For" loop.
- Syntax of "for" loop in shell is:



```
[root@localhost ~]# for  i  in  a b c d e
> do
> echo $i
> done
```

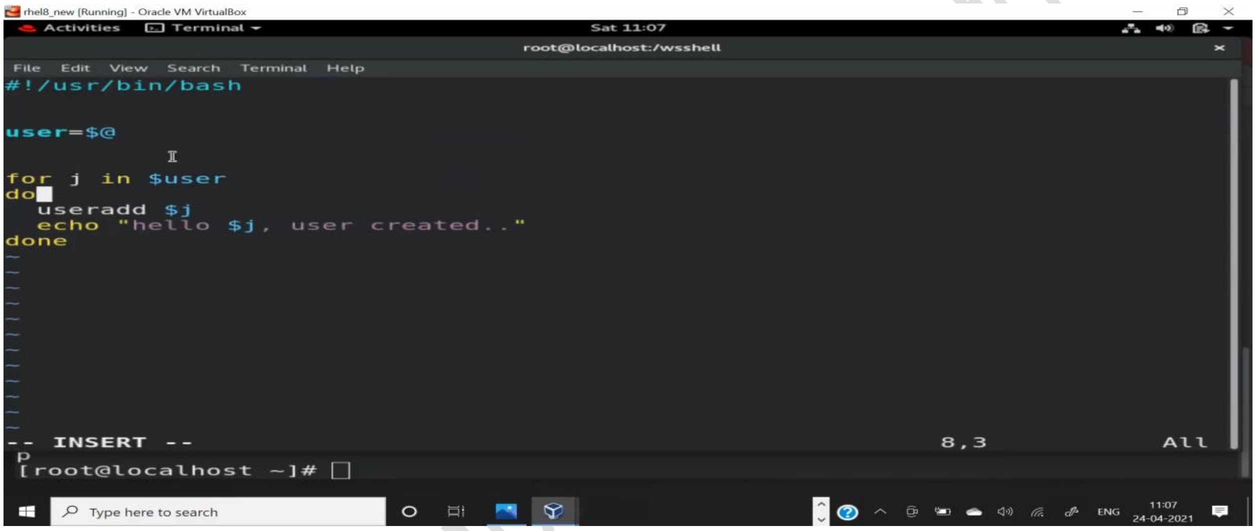
- For will go in the list a, b, c, d and store each value in "i" variable and perform on it, do and do is the keyword that indicates the block of code just like "{ }" in C++ programming language.

- We can write the for loop in this way also:



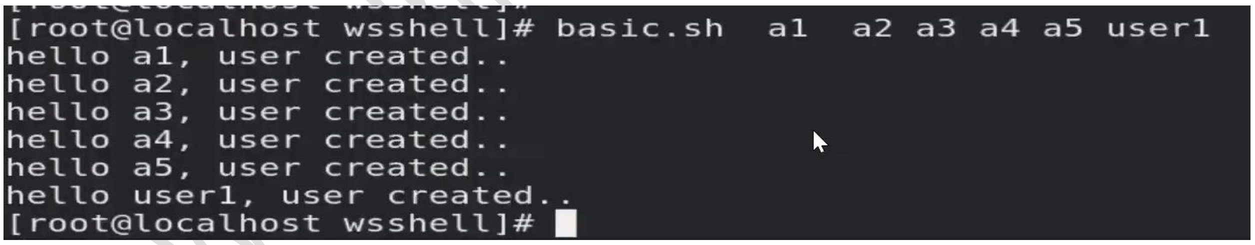
```
[root@localhost ~]# u="a b c t y p"
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# for i in $u; do echo $i; done
a
b
c
t
```

- Script for performing user-add operation on each argument we provide in the command



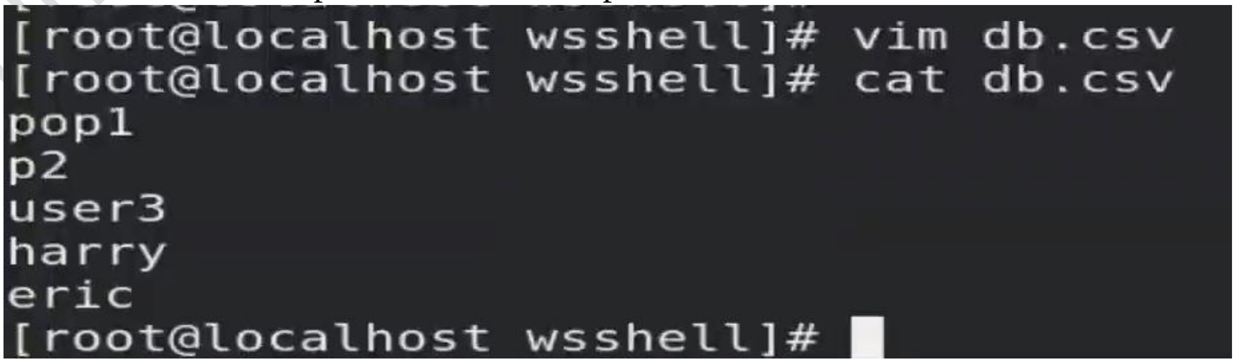
```
#!/usr/bin/bash

user=$@

for j in $user
do
   useradd $j
   echo "hello $j, user created.."
done
```

- This will create users for all



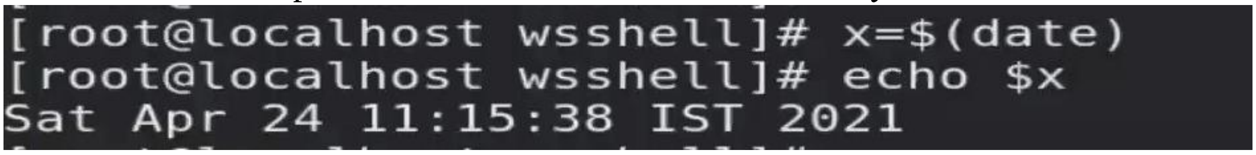
```
[root@localhost wsshell]# basic.sh  a1  a2 a3 a4 a5 user1
hello a1, user created..
hello a2, user created..
hello a3, user created..
hello a4, user created..
hello a5, user created..
hello user1, user created..
[root@localhost wsshell]#
```

- But this is not how we work in the real world, we have the entire file of user names and we have to iterate over that file, not any list or tuple.
- So for this we create a file name db.csv where we write some usernames and now we want to perform the same operation but this time with the file



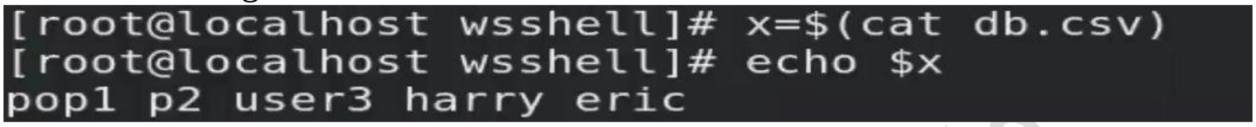
```
[root@localhost wsshell]# vim db.csv
[root@localhost wsshell]# cat db.csv
pop1
p2
user3
harry
eric
[root@localhost wsshell]#
```

- So for this we need to understand one more thing if we want to store the command or its output in a variable, then we use the “$” symbol



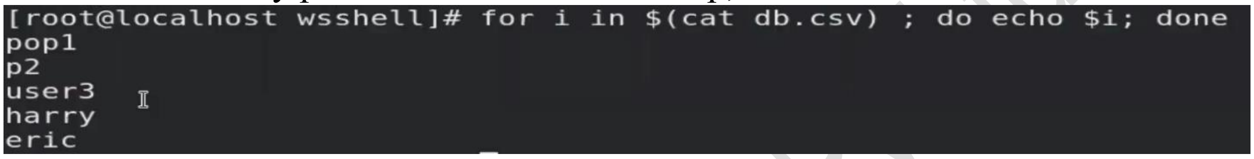
```
[root@localhost wsshell]# x=$(date)
[root@localhost wsshell]# echo $x
Sat Apr 24 11:15:38 IST 2021
```

- So, for retrieving the data of this file we have to use the same method



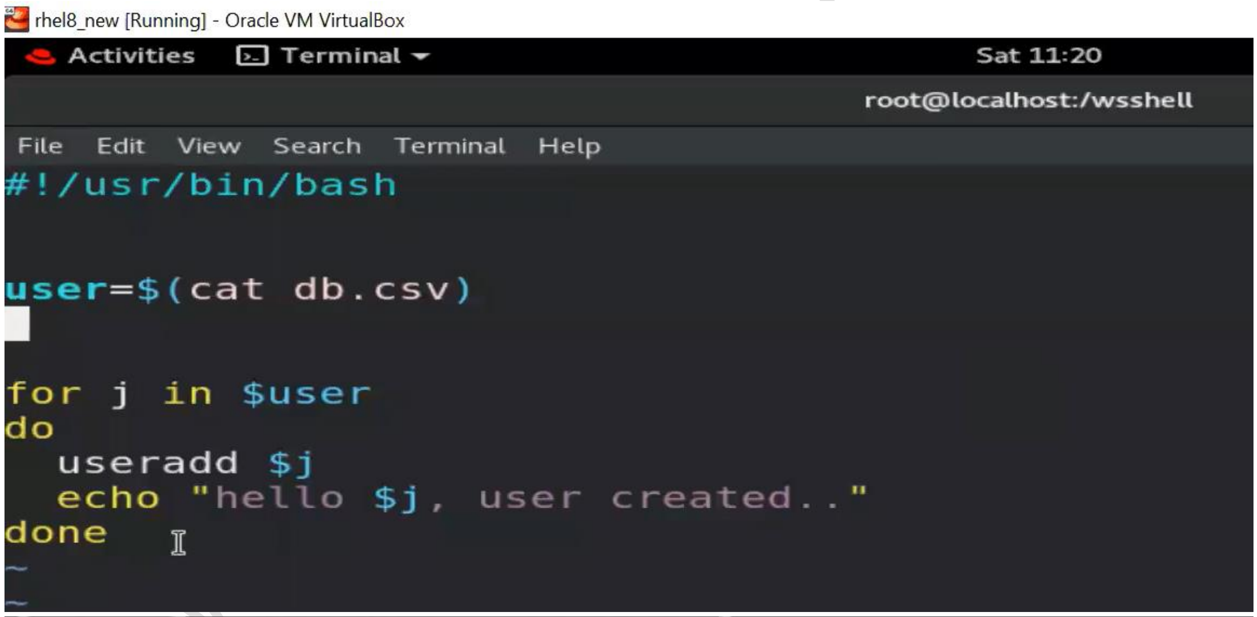
```
[root@localhost wsshell]# x=$(cat db.csv)
[root@localhost wsshell]# echo $x
pop1 p2 user3 harry eric
```

- And now we easily put this inside the “for” loop,

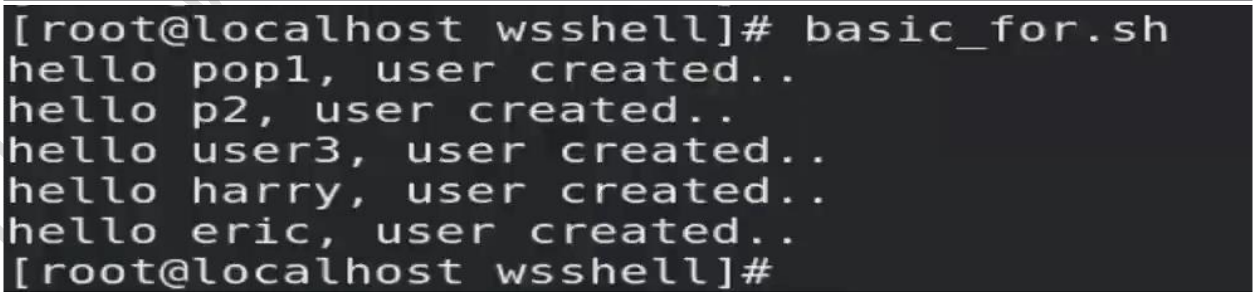


```
[root@localhost wsshell]# for i in $(cat db.csv) ; do echo $i; done
pop1
p2
user3
harry
eric
```

- We create the file and put the script there


rhel8_new [Running] - Oracle VM VirtualBox
Activities
Terminal
Sat 11:20
root@localhost:/wsshell
File Edit View Search Terminal Help

```
#!/usr/bin/bash


user=$(cat db.csv)


for j in $user
do
  useradd $j
  echo "hello $j, user created.."
done
~
~
```



```
[root@localhost wsshell]# basic_for.sh
hello pop1, user created..
hello p2, user created..
hello user3, user created..
hello harry, user created..
hello eric, user created..
[root@localhost wsshell]#
```

- one way to verify is by going to this file: “/etc/passwd”, this file is the one where the details of all the users are stored.