DAY 8

ITERATIVE CONSTRUCTS

In common programming, iterative constructs are used to repeat the execution of a part of code.
In bash shell scripting, there are three types of loop constructs available.

→ for loop
→ while loop
→ until loop

where for-loop is count controlled and while-loop & until-loop are condition controlled.

for-loop:

for loop is used in bash shell scripting to repeat the execution of a section of code for definite number of iterations.

Syntax:

```
for var in argument_list
do
        <block of commands>
done
```

Iterating through a collection of values

```
#!/bin/bash
for server in server1 server2 server3 server4 server5
do
echo “Processing the server: ${server}”
done
```

Iterating through the value of a variable

```
#!/bin/bash
read -p "Enter a statement:" input
for var in ${input}
do
echo “The current word is: $var”
done
```

while-loop:

while-loop is a condition-controlled loop. It is used to repeatedly execute a section of the script as long as the given condition evaluates to be true. To build these conditions, the bash comparison operators are used.

Syntax:

```
while <condition>
do
<block of commands>
done
```

```
#!/bin/bash
num=10
while [ $num -le 15 ]
do
echo $num
num=`expr $num + 1`
done
```

```
#!/bin/bash
server=“server1"
while [ $server != server10 ]
do
read -p "Enter a word" server
echo "The word you entered is: $server"
done
```

until loop:

until-loop is a like the while-loop except the fact that the until loop runs as long as the given condition evaluates to false. In other words, until loop runs until the given condition becomes true, when it will be terminated.

Syntax:

```
until <condition>
do
<block of commands>
done
```

```
#!/bin/bash
val=5
until [ $val -gt 10 ]
do
echo $val
val=`expr $val + 1`
done
echo “until loop is terminated”
```

Break and continue statements:

We have so far discussed three types of loops: for loop, while loop and until loop. All these three types of loops are used when a set of actions need to executed repeatedly. Sometimes in such loop, we want to skip a few iterations or need to stop the execution of loop even before completing the iterations.
In such scenarios, we need to use break/continue statements within the loop.

* Use break statement, when we need to terminate the execution of the entire loop.
* Use continue statement, when we need to skip the current iteration but continue with the execution of loop from next iteration.

```
#!/bin/bash
for command in date cal exit who
do
if [ $command == "exit" ]
then
break
fi
echo "The command is: $command"
done
echo "Out of the loop"
```

```
#!/bin/bash
num=1
while [ $num -le 5 ]
do
  if [ $num -eq 3 ]
  then
    num=`expr $num + 1`
    continue
  fi
  square=`expr $num \* $num`
  echo $square
  num=`expr $num + 1`
done
```

DAY 8 - END