classmate
Date
Page

[LSS DAY-02]

Q Explain conditional statement with examples?

⇒ Conditional statements are used to check a particular condition and if true then the statement of that condition will be executed.

Conditional Operator

-eq : equal to
-lt - less than
-gt : greater than
== two strings are equal
!= : Not equal to

2) If statements

- A condition is checked if true then the statement with then condition will be executed

Syntax:

```
if [condition];
    commands/statements

fi
```

2) Ex:

```
echo "Enter a number"
read n
if [$n -lt 100];
then
    echo "$n is less than zero"
```

⇒ if-else:

If condition is written if true then if block will be executed or else block will be performed.

Syntax:

```
'if [condition];
 then
 command (s)
 else
    command (s)
 fi
```

Ex:

```
a=10
b=20
if [a-gt b]
 echo "a is greater"
else
   echo "b is greater"
```

⇒ if ~~else~~ elif-else statement → Multiple of statements are written and their condition are checked and if statement if is false then else block will be performed

Syntax:

```
if [condition]
then
    commands
elif [condition]
then
    commands
fi
```

Ex:
```
a=10;
b=20;
c=20;
if [a -gt if b + a -if c]
then
    who "a is greater"
elif [b -gt a + b -gt c]
then
    echo "b is greater"
else
    who "c is greater"
fi
```

⇒ **Nested if:**

if condition are written inside another if condition

Syntax:
```
if [condition] then
    commands
    if [condition] then
    command
    fi
fi
```

⇒ **Case statement:**

Case statement can be used as an alternative to if statement

Syntax:
```
case in
    pattern 1) command;
            2) command;
    else
```

Ex:
```
N1 = $1
N2 = $2
N3 = $3
(case . $ op in f)
(( Result = $N1 + $N2));
(( Result = $N1 - $N2));
(( Result = $N1 * N2));
(( Result = $N11 /$N2));
```

*)
```
echo "wrong number of arguments"
exit 0;
case
echo "$ N11 $op $N2 = $ Result"
```

(2) Explain Looping statement with example!

(1) Until statements:

Here command is ~~executed~~ evaluated and based on the result loop will be executed if condition is ~~false~~ false then loop will be terminated.

Ex:
```
n=10
until [ $a -17 10]
do
    who $a
    a="expr $ a+1"
```

Classmate
Date
Page

③ While!

While loop enables you to execute set of commands repeared by untill some conditional ...

Ex: !

```
   i=1
while[$1 -le 10]
do
   echo $i
   i='expr $i+1'
   done
```

③ for loop

A for loop is a statement which allows Code to be repeatedly executed

Ex!
```
for no in {1-10}
do
echo $no
done.
```

④ Switch statement!

Case statements works as a switch statement if ~~required~~ specified value match with the pattern then it will execute a block of that particular pattern

Syntax!
```
Case in
        pattern 1) Statement:
                2) Statement n
```