

UNIT-2

LOOPING, CONDITIONAL & CONTROL STATEMENTS

- The statements in the program are executed one by one by Python interpreter. This type of execution is called '**sequential execution**'.
- **Decision Making** is required when we want to execute a code only if certain conditions are satisfied.

CONTROL STATEMENTS

- Control or change the flow of execution. (**Consider the PPT as well**)
 1. If Statements
 2. If...else Statements
 3. If...elif...else statement
 4. While loop
 5. For loop
 6. Else suite
 7. Break statement
 8. Continue statement
 9. Pass statement
 10. Assert statement
 11. Return statement

NOTE: Switch statement is not available in Python.

1. IF STATEMENT

- The group of statements in Python is called a **suite**.

Example:

#Python program to display a digit in words

```
num=1
```

```
if num==1:
```

```
    print("ONE")
```

To display a group of messages when condition is true

```
str='Yes'
```

```
if str=='Yes':
```

```
    print("Yes")
```

```
print("This is Python Session")
print("Control Structures")
```

A Word on Indentation

- Refers to spaces that are used in the beginning of a statement.

Example:

```
if x==1:
    print('a')
    print('b')
    if y==2:
        print('c')
        print('d')
print('end')
```

2. IF ELSE STATEMENT

Example:

To check number is even or odd

```
x=16
if x%2==0:
    print(x,"is even number")
else:
    print(x,"is odd number")
```

3. IF...ELIF...ELSE STATEMENT

Example:

To check number is zero, positive or negative

```
num=-17
if num==0:
    print(num,"is zero")
elif num>0:
    print(num,"is positive")
else:
    print(num,"is negative")
```

4. THE WHILE LOOP

- Useful to execute a group of statements several times repeatedly depending on the whether a condition is True or False.

Example:

To display number from 1 to 10

```
x=1
while x<=10 :
    print(x)
    x+=1
print("Finish")
```

5. THE FOR LOOP

- Called **iterators**.
- Iterating over a sequence is called **traversal**.

Example:

To display characters of string

```
str="Hello"
n=len(str)
print(n)
for ch in str:
    print(ch)
```

#Difference between while and for loop

```
#with while loop
x=0
while(x<5):
    print(x)
    x=x+1;
```

```
#with For loop
for x in range(0,5):
    print(x)
```

The range function ()

- Range(10) –generate 0 to 9
- **range (start, stop, step_size)** – step_size defaults to 1 if not provided.

Example:

```
print(range(10))
print(list(range(10)))
print(list(range(2,8)))
print(list(range(20,40,2)))
```

Program to iterate through a list using indexing

```
genre=['pop','rock','jazz']
for i in range(len(genre)):    #iterate over the list using index
    print("I like",genre[i])
```

```
list=[10,20, 30, 'A', 'Amit']
for element in list:
    print(element)
```

#For Loop for string

```
Months=["Jan", "Feb", "March", "April", "May"]
for m in Months:
    print(m)
```