**LOOPING IN PYTHON**

**Loop :** A loop is a control flow structure that allows a set of statements that will be executed repeatedly until a certain condition is met.

Loops are used to automate repetitive tasks, to iterate over a data structure.

Python language support two types of loops, They are :

1. For loop
2. While loop

**While Loop :**

While loop is used to execute a block of statements repeatedly till a given condition is True.

while loop will check the condition before executing the loop.

It is used whenever the number of iterations is not known in advance.

**Syntax** of while loop is as follows

while ( condition ):

 # block of statements that are to be executed

 statements

**Example** : print numbers from 1 to 5

counter=1

###### while ( counter <=5):

######  print(counter, end=" ")

######  counter+=1

###### Output :1 2 3 4 5

**For Loop :** for loop in python is different from the for in other programming languages.

In Python a for loop is used to iterate over a set of items, list of items or any sequence of items.

**Syntax** of for loop is as follows

for variable in sequence:

 # Block of code to be executed for each iteration

Example1 : print all the given sequence using a for loop.

Input :

###### fruits=[“apple”, “mango”, “Guava”, “watermelon”]

###### for i in fruits:

###### print(i)

Output :

apple

mango

Guava

watermelon

**Example 2 :** Print numbers from 0 to 5

###### for i in range(6):

######  print(i, end=" ")

Output :

0 1 2 3 4 5

You might wonder what is the word range in the above example. Let me explain it in detail.

**range function :**

range function is used to generate a sequence of numbers within a specified range.

Mainly used with for loop to iterate over a sequence of numbers.

It auto increments the number until the specific number of iterations are reached.

**Syntax of range( ) function is** :

range ( start, stop, step)

Start : from where the sequence is to be started, it is optional if not mentioned it will be considered with default value to 0.

Stop : end value of the sequence (exclusive), but the stop value is not included in the sequence.

Step : it is the difference between two consecutive values in the sequence. If it is not mentioned it’s default value is 1.

**Examples:**

* range( 5 ) denotes the values from 0 to 4 excluding 5. If you mention a single value in range it means the value is a stopping value.
* range( 2, 5 ) denotes the values starting from 2 to 4 excluding 5.
* range( 3, 20, 2 ) denotes the values starting from 3 to 20 incrementing the sequence by 2 each time.

**Example 1 :** Give only stop value, it will take the start value by default to zero and step value to 1.

**Input:**

for i in range(5):

 print(i,end=" ")

Output :

0 1 2 3 4

**Example 2 :** Give starting with 1 and stopping with 6 , it will take the step value to 1 by default.

**Input :**

for i in range(1,6):

 print(i,end=" ")

Output :

1 2 3 4 5

**Example 3 :** Give all the three values starting with 1 , stopping value to 15, step value is 3 i.e. difference between consecutive numbers in sequence is 3.you can see that in the output.

Input :

for i in range(1,15,3):

 print(i,end=" ")

Output :

1 4 7 10 13

**Break Statement :** Break Statement is used to terminate a loop based on a certain condition when it encounters to terminate the loop it is in and execution of the program continues to the next statement after the loop.

Example 1 :

###### for i in range(5):

######  if i==3:

######  break

######  print(i,end=" ") # Output 0 1 2

**Continue Statement :**

The continue statement is used to skip the rest of the code inside a loop for the current iteration and move to the next iteration of the loop.

Example :

##### for i in range(5):

#####  if i==3:

#####  continue

#####  print(i, end=" ") #Output 0 1 2 4

##### When the given condition is met the statements under that loop block will be skipped so it skipped print resulting in no number 3 in output.

**pass statement :**

The pass statement is a no-operation statement.it acts as a temporary placeholder when code is needed for syntactical reasons but no actual action is intended.

It permits the program to proceed without executing any particular operation.

Example :

for i in range(4):

 if i==2:

 pass # This place is for future code.

 print(i) # it prints 0 1 2 3