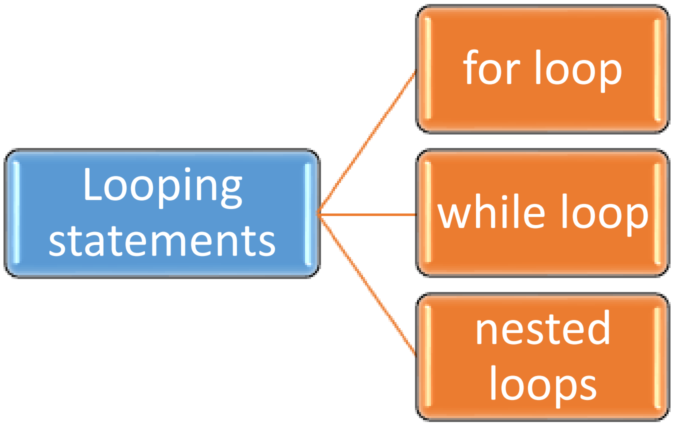
LOOPS IN PYTHON

There might be certain situations during programming when we want to repeat one or more statements for specific number of times. This task can be achieved using loops. Loops allows us to repeat a statement or a block of statements multiple times. Each repetition is called an iteration.

Following are the different types of loops in Python :



FOR LOOP

For loop is used for traversing or iterating over a sequence like string , lists and tuples.

'in' operator and 'range()' function can be used with for loops . These loops are fixed iteration loops i.e the number of times the loop executes is known beforehand.

Syntax :

```
for i in sequence :

        body of for loop
```

Here , 'i' is a variable that takes the values of items present inside the sequence in each iteration of the loop. Also, notice that correct indentation should be provided for the statements inside for loop.

Example :

```
#Program to calculate the product of all numbers stored in a list
#declaring list
list1= [1,2,3,4,5]
#variable to store the product initialized to 1
```

```
pro=1
#traversing the list
for i in list1:
  pro=pro*i
#printing the result
print ("The product is : ", pro)
```

Output:

```
The product is :  120
```

WHILE LOOP

While loop is used to repeat execution of a statement or a block of statements till the given condition is true.

The condition is checked at the entry of the loop. If the condition is true, control goes inside the loop and statements in body of loop are executed. If the testing condition becomes false, control moves out of the while loop and the next sequential statement in the program gets executed.

These loops are unfixed iteration loops i.e. the number of times while loop executes is not known beforehand, it depends how many times the condition evaluates to true.

Syntax :

```
while conditional_expression :

        body of while loop
```

Here, conditional_expression is any condition made up of logical and relational operators. Any non-zero value is treated as True.

Notice that correct indentation should be provided for the statements inside for loop.

Example :

```
#Program to calculate the product of all numbers stored in a list
#declaring list
list1=[1,2,3,4,5]
l= len(list1)
i=0
pro=1 #variable to store the product initialized to 1
while i<5:
  pro=pro*list1[i] #visiting each item in the list using list index
  i=i+1
#printing the result
print ("The product is : ", pro)
```

Output :

```
The product is :  120
```

NESTED LOOPS

In nesting of loops, one loop can be put inside another loop. For instance, While loop inside while loop or for loop inside while loop.

Example :

1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

```python
r=6
for i in range(1, r): #outer loop for rows
  for j in range (1, i+1): #inner loop for columns
    print(i, end=" ") #printing output in each line
  print() #skips to the next line after each row print
```