Loop or Iterative Statements

Loop Control Statements

- We have seen in conditional statements, each instruction was executed only once when the condition true or false.
- But many programs required that a group of instructions to be executed repeatedly until some condition has been satisfied .This is known as **looping** or **Iteration.**
- A loop is a group of instructions the computer executes repeatedly a certain number of times.
- The repetition continue while a condition is true. When the condition becomes false, the loop terminates and the control passes to the statement(if any) following the loop.
- The loop consists of two segments , one is known as the **control statement** and other is the **body of the loop**.
- There are two kinds **of loop structures** or **Iterative statements** in python.

  while

  for

While Loop Control Statements

- The **while** Loop is often used if you don't know how many times the loop is to be executed before you start the loop.
- The **for** Loop is used in programming that you want to do something a fixed number of times.
- The **while** loop provides a mechanism to repeat one or more statements while a particular condition is **True**.
- Syntax of the **while** loop:-

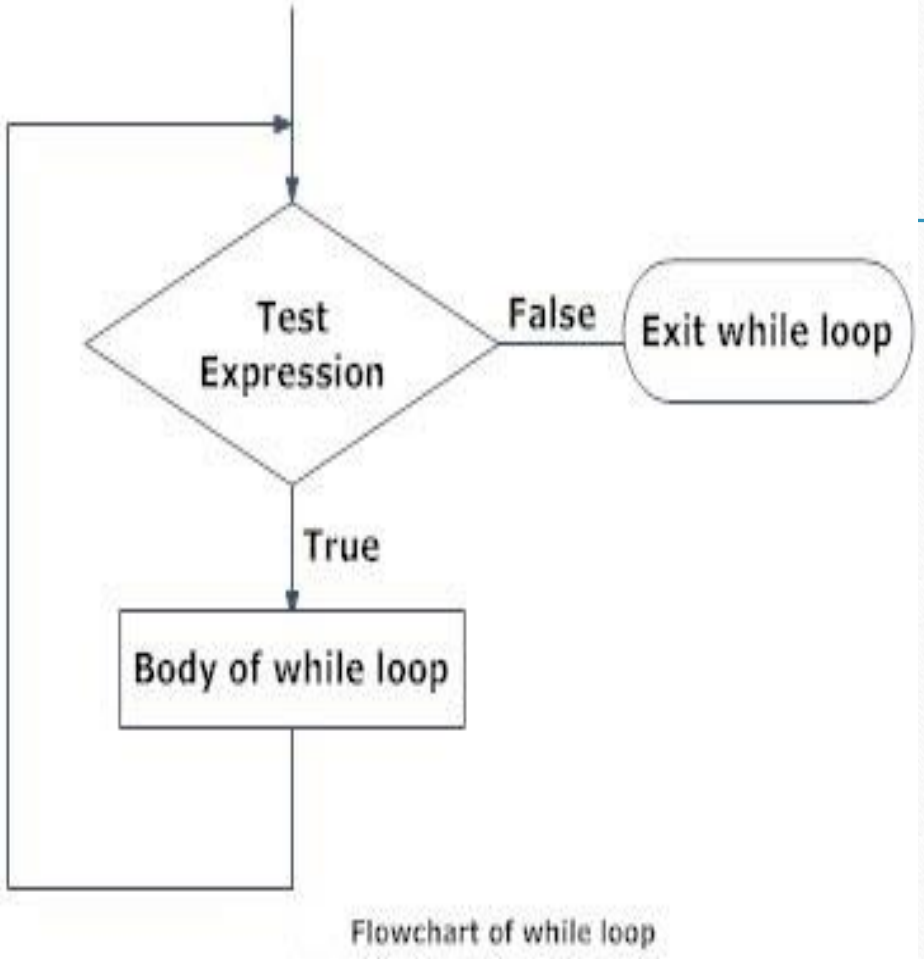
```
statement x
while Condition :
        statement block
statement y
```

- Note in the while loop , the condition is tested before any of the statements in the statement block is executed.

  If the condition is **True** , only then the loop statements will be executed

  Otherwise - If the condition is **False**, then the control will jump to outside of the loop statements

Flowchart and Example



Flowchart of while loop

```
#Write a program to print 1 to 10 numbers?
i=1
while i<=10:
    print(i,end=" ")
    i+=1
>>>
1 2 3 4 5 6 7 8 9 10
>>>
```

Initially **i**=1 and is less than 10, that is, the condition is True, so in the while loop the value of **i** is printed and the condition is updated so that every execution of the loop , the condition becomes more approachable.

Nesting of while loops

- Nesting of while loops, that is , one while loop within another while loop

```
# Write a program that calculate and prints
# the factorial of first  n natural numbers ?
# using nested while
i=1
n=int(input("Enter a Positive Integer :"))
while i<=n:
    fact=j=1
    while j<=i:
        fact=fact*j
        j+=1
    print("%d! = %d"%(i,fact))
    i+=1
```

```
Enter a Positive Integer :10
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
```

Practice Problems

- Write a program to calculate sum and Average of the first 100 natural numbers using **while**?
- Write a program to print 30 horizontal asterisks(*) ?
- Write a program to calculate sum and Average of numbers m to n using **while**?
- Write a program to read the number until -1 is encountered .also count the no of Negative, Positive and Zeros and find the Sum,Avg of Positive and Negative numbers?
- Write a program to find sum of even and odd numbers from m to n using **while** ?
- Write a program to print the factorial of a given number using while loop?
- Write a program to find the given number is an Armstrong Number or not?
- Write a program to convert decimal to binary and binary to decimal ?
- Write a program to find the GCD of two positive integer numbers using while loop?

for loop control structures

- The **for** loop statement is used to repeat a statement or group of statements a fixed number of times.
- It's usually used when you know in advance how many times you want to execute a group of consecutive instructions.
- The **for …..in** statement is a looping statement used to iterate over a sequence (list, tuple, string) or other iterable objects. Iterating over a sequence is called traversal.
- There is a difference in for loop syntax. Python syntax uses the **range function** ,which makes the loop simpler ,more expensive and less prone error(s).
- Syntax of **for** loop:- **for loop_control_var in sequence :**

  statement block
- When a for loop is used, a range of sequence is specified (only once).The items of the sequence are assigned to the loop control variable one after the other.
- Loop continues until we reach the last item in the sequence. The body of for loop is separated from the rest of the code using indentation.

Flowchart and Example

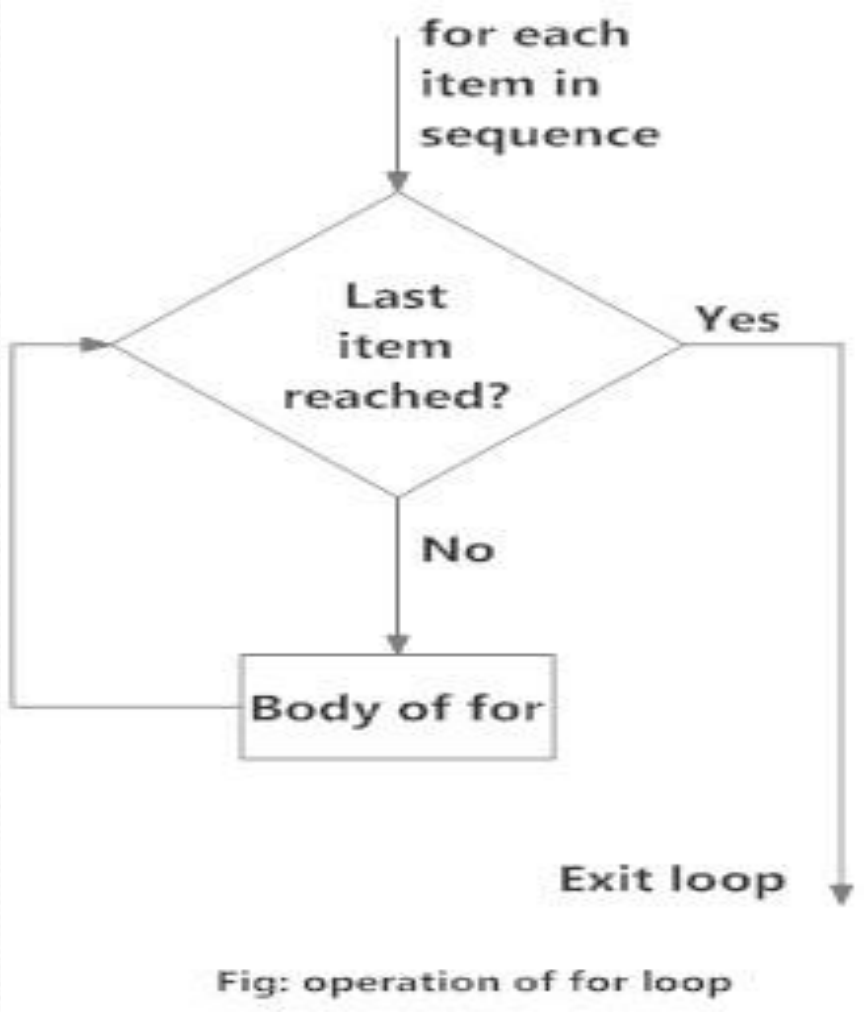


Fig: operation of for loop

```
# Program to find the sum of all numbers stored in a list

# List of numbers
numbers = [6, 5, 3, 8, 4, 2, 5, 4, 11]

# variable to store the sum
sum = 0

# iterate over the list
for val in numbers:
    sum = sum+val

# Output: The sum is 48
print("The sum is", sum)
```

The **range()** Function

- The **range()** is built in function in python that is used to iterate over a sequence of numbers
- Syntax is : - range(**beg , end,[step]**)
- The range() produces a sequence of numbers starting with **beg** and ending with one less than the number **end**.
- The **step** argument is optional, by default, every number in the range is incremented by 1 but we can specify a different increment using **step**.
- step can be both, negative and positive ,but not zero.

```
for i in range(10):
    print(i,end=" ")
```

```
0 1 2 3 4 5 6 7 8 9
```

```
for i in range(0,10):
    print(i,end=" ")
```

```
0 1 2 3 4 5 6 7 8 9
```

```
for i in range(0,10,3):
    print(i,end=" ")
```

```
0 3 6 9
```

- If range() is given single argument ,it produce an object values from **0** to **argument-1**
- If range() is given two arguments , it produce an object values from **beg** to **end .**
- If range() is given 3 arguments ,then third arguments specifies the interval of the sequence.

Nesting of for loops

- When the body of one loop contains another, the second is said to be nested inside the first.
- The inner and outer loop need not be generated by the same type of control structure.
- One loop can be completely embedded within the other-there can be no overlap.
- The way if statements can be nested, similarly for loops can also be nested.
- Syntax:
- **for loop_control_var in sequence :**

for loop_control_var in sequence :

statement block

Example:-

```
for i in  range(5) :
    print(i,"-",end=" ")
    for j in  range(5) :
        print(j,"-",end=" ")
    print()
```

Practice Problems

- Write a program to calculate sum and Average of the first 100 natural numbers using **for**?
- Write a program to print even numbers between 1 and 20 ?
- Write a program to print multiplication table for n number ?
- Write a program to display squares, square roots, cubes and cube roots of the 1 to 10 ?
- Write a program to print the factorial of a given number using **for** loop?
- Write a program to find the maximum of a given set of numbers ?
- Write a program to generate the first n Fibonacci numbers where n is the value entered by user ?
- Write a program to generate Prime numbers up to n where n is the value entered by user ?
- Write a program to find the pow(x,n) ?