Looping Control Statements

Looping Control statements are used to repeat one or more than one statement number of times or until given condition.

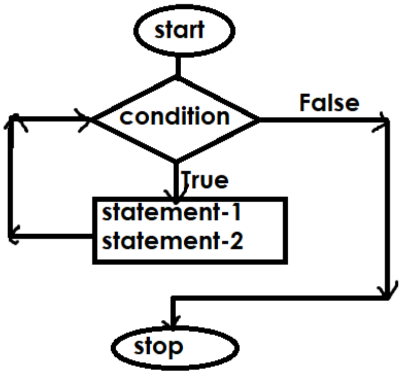
Python support 2 looping control statements

1. while loop
2. for loop

Note: python does not support do..while

While loop

"while" is a keyword which represents while loop in python. While loop is used in order to repeat one or more than one statement until given condition is True.

Syntax:
 	
while condition:
    statement-1
    statement-2
 	
while loop repeat statement-1, statement-2 until given condition is True, if condition is False it stop repeating.
	 start
	condition
	False
	True
	statement-1
	statement-2
	stop

While loop required 3 statements

1. initialization statement
2. condition
3. update statement

Initialization statement defines initial value of condition

Condition is a boolean expression which defines how many times while loop has to be repeated

Update statement which updates condition.

Example:

```
n=1  #initialization statement
while n<=10: # condition
 print("Naresh")
 n=n+1 # Update Statement
```

Output

```
Naresh
Naresh
Naresh
Naresh
Naresh
Naresh
Naresh
Naresh
Naresh
Naresh
```

Example:

```
# Write a program to print sum of 5 numbers
# input 5 numbers from keyboard

i=1
s=0
while i<=5:
    num=int(input("Enter any number "))
    s=s+num
    i=i+1

print(f'Sum of 5 numbers {s}')
```

Output

```
Enter any number 10
Enter any number 60
Enter any number 30
Enter any number 90
Enter any number 100
Sum of 5 numbers 290
```

Example:

```
# Write a program to input 5 numbers
# and find maximum number

i=1
max_num=0
while i<=5:
    num=int(input("Enter any number :"))
    if num>max_num:
        max_num=num
    i=i+1

print(f'Maximum number is {max_num}')
```

Output

```
Enter any number :40
Enter any number :10
Enter any number :90
Enter any number :30
Enter any number :20
Maximum number is 90
```

f-string (OR) format string

F-strings, also known as formatted string literals, are a way to embed expressions inside string literals for formatting in Python. Introduced in Python 3.6, f-strings provide a concise and readable way to create strings with embedded variables, calculations, or function calls.

To create an f-string, you prefix a string with the letter f or F. Inside the string, you can enclose expressions within curly braces {}. These expressions are evaluated at runtime, and their values are inserted into the string

f-string or format string is very much helpful in formatting output.

Syntax: f’string’

Example:

```
name="naresh"
age=56
print("My name is",name,"and age is",age)
print(f"My name is {name} and age is {age}")
num1=100
num2=200
print(f'Sum of {num1} and {num2} is {num1+num2}')
print(f'Diff of {num1} and {num2} is {num1-num2}')
```

Output

```
My name is naresh and age is 56
My name is naresh and age is 56
Sum of 100 and 200 is 300
Diff of 100 and 200 is -100
```

Example:

```
# Write a program to print numbers from
# 1 to 10 (while loop)

num=1
while num<=10:
    print(num,end=' ')
    num=num+1
```

Output

```
1 2 3 4 5 6 7 8 9 10
```

Example:

```
#Write a program to print numbers from
# 10 to 1
num=10
while num>=1:
    print(num)
    num-=1
```

Output

```
10
9
8
7
6
5
4
```

```
3
2
1
```

Example:

```
# Write a program to count digits of input number

num=int(input("Enter any number "))
count_digits=0
while num>0:
    count_digits+=1
    num//=10

print(f'Count of digits {count_digits}')
```

Output

```
Enter any number 583
Count of digits 3
```

Example:

```
# Write a program to find sum of digits
# of input number

num=int(input("Enter any number "))
s=0
while num>0:
    d=num%10
    s=s+d
    num=num//10

print(f'Number is {num}')
print(f'Sum of digits {s}')
```

Output

```
Enter any number 378
Number is 0
Sum of digits 18
```

Example:

```
# Write a program to print count of even and odd digits
# in input number

num=int(input("Enter any number "))
ec=0
oc=0

while num>0:
   d=num%10
   if d%2==0:
      ec+=1
   else:
      oc+=1
   num=num//10

print(f'Even Digits count {ec}')
print(f'Odd Digits count {oc}')
```

Output

```
Enter any number 19456
Even Digits count 2
Odd Digits count 3
```