Loops In Python

Looping means repeating something over and over until a particular condition is satisfied.

There are 2 loops or control statements in Python,
for loop
while loop

for loop:

A for loop in Python is a control flow statement that is used to repeatedly execute a group of statements as long as the condition is satisfied. Such a type of statement is also known as an iterative statement.

A for loop is also used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string). This is less like the for keyword in other programming languages, and works more like an iterator method as found in other object-orientated programming languages.

While working with sequence, for loop gets one by one element at a time from the sequence.

For loop with iterables:

#Iterating a list

```
colors = [‘Red’,’White’,’Green’,’Blue’]
for x in colors:
        print(x)
```

OUTPUT:

```
Red
White
Green
Blue
```

#Iterating a string

```
for i in ‘Python’:
        print(i)
```

OUTPUT:

```
P
y
```

t
h
o
n

For loop with range() function:

The range() function is used to generate the sequence of the integer numbers.

Syntax:
range(start,stop,step_size)

where,

- The start represents the beginning of the iteration. It is optional and by default it is 0.
- The stop represents that the loop will iterate till stop-1. The range(1,5) will generate numbers 1 to 4 iterations. It is mandatory.
- The step_size is used to skip the specific numbers from the iteration. It is optional to use. By default, the step_size is +1.

Examples:
#Print 1 to 10

```
for i in range(1,11,1):  #range(start,end,step)
	print(i)
```

or

```
for i in range(1,11):  #range(start,end) step by default is +1
	print(i)
```

#Print 0 to 10

```
for i in range(11):  #range(end) start = 0, step = +1 by default
	print(i)
```

while loop:

In python, while loop is used to execute a block of statements repeatedly until a given condition is satisfied. And when the condition becomes false, the line immediately after the loop in the program is executed.

Also while True loop is used when we do not know the end where we need to stop. Such loop is called an infinite loop.

Syntax of infinite while loop:

```
while True:
        ---
        ---    body of while loop
        ---
```

Syntax of while loop with condition:

```
initialization of counter
while condition:
        ---
        ---         body of while
        ---
        Increment/decrement of counter
```

Example:

```
#Print 1 to 10
i = 1
while i <= 10:
        print(i)
        i = i+1
```

Loop control statements in Python:

break, continue and **pass** are the loop control statements in python.

break keyword:

The break statement is used inside the loop to exit out of the loop. In Python, when a break statement is encountered inside a loop, the loop is immediately terminated, and the program control transfer to the next statement following the loop. Statements written after break keyword which are same indented are not executed.

```
#Print the numbers until 5 only between 1 to 10 using for loop
for i in range(1,11):
        if i > 5:
                break
        print(i)
```

#Print the numbers until 5 only between 1 to 10 using while loop

```
i = 1
while i <= 10:
	if i > 5:
		break
	print(i)
	i += 1
```

continue keyword:

The continue statement skip the current iteration and move to the next iteration. In Python, when the continue statement is encountered inside the loop, it skips all the statements below it and immediately jumps to the next iteration.

In simple words, the continue statement is used inside loops. Whenever the continue statement is encountered inside a loop, control directly jumps to the start of the loop for the next iteration, skipping the rest of the code present inside the loop’s body for the current iteration.

#Skip the printing of space in given string using for loop

```
for i in 'Hello World':
	if i == ' ':
		continue
	print(i,end='')
```

OUTPUT : HelloWorld

#Skip printing of 5 in a sequence from 1 to 10 using while loop

```
i = 0
while i < 10:
	i += 1
	if i == 5:
		continue
	print(i,end = ‘ ‘)
```

OUTPUT : 1 2 3 4 6 7 8 9 10

pass keyword:

The pass is the keyword In Python, which won’t do anything. Sometimes there is a situation in programming where we need to define a syntactically empty block. We can define that block with the pass keyword.

A pass statement is a Python null statement. When the interpreter finds a pass statement in the program, it returns no operation. Nothing happens when the pass statement is executed. Pass is also used for empty control statement, function and classes.

For e.g,

1.
```
for i in ‘Python’:
        pass
```

2.
```
def m1():
        pass
```

3.
```
class A:
        pass
```