

* CHAPTER - 7: LOOPS IN PYTHON

Sometimes we want to repeat a set of statements in our program - for instance: Print 1 to 100.

Loops make it easy for a programmer to tell the Computer, which set of instructions to repeat.

→ Types of loops in Python

Primarily, there are two types of loops in Python:

1) while loop

2) for loop

→ while loop

The syntax of a while loop looks like this:

While Condition:

Body of the loop

⇒ The block keeps executing until the condition is true.

In while loop, the condition is checked first. If it evaluates to true, the body of the loop is executed otherwise not!

If the loop is entered, the process of [Condition check & Execution] is continued until the condition becomes false.

Quick Quiz: Write a program to print 1 to 50 using a while loop.

• An Example:

```
i = 0
```

```
while i < 5:
```

⇒ Prints "Haavey" - 5 times!

print("Hareey")
i = i + 1

Note: If the condition never becomes false, the loop keeps getting executed.

Quick Quiz: Write a program to print the content of a list using while loops.

→ ~~while loop~~ For loop

A for loop is used to iterate through a sequence like list, tuple or string [iterables]

The syntax of a for loop looks like this:

l = [1, 7, 8]

for item in l:

print(item)

→ print 1, 7 and 8

→ Range function in Python

The range function in python is used to generate a sequence of numbers.

We can also specify the start, stop and step-size as follows:

range(start, stop, step-size)

↳ step-size is usually not used with range().

→ An Example demonstrating range () function

```
for i in range(0,7):  
    print(i)
```

→ range(7) can also be used
→ prints 0 to 6

→ For loop with else

An Optional else can be used with a for loop if the code is to be Executed when the loop exhausts
Example:

```
l = [1, 7, 8]  
for item in l:  
    print(item)  
else:  
    print("Done")
```

→ This is printed when the loop exhausts!

Output:

1

7

8

Done

→ The break Statement

'break' is Used to Come Out of the loop when encountered
It ~~includes~~ instructs the program to - Exit the loop now.

Example:

```
for i in range(0,80):  
    print(i)  
    if i == 3:  
        break
```

→ This will print 0, 1, 2, and 3

→ The Continue Statement

'Continue' is used to stop the current iteration of the loop and continue with the next one.
It instructs the program to "Skip" this iteration.

Example :

```
for i in range(4):
```

```
    print("Printing")
```

```
    if i == 2:    ⇒ if i is 2, the iteration
```

```
        continue    is skipped.
```

```
    print(i)
```

→ Pass Statement

Pass is a null statement in python.
It instructs to "Do nothing".

Example :

```
l = [1, 7, 8]
```

```
for item in l:
```

```
    pass
```

→ without pass, the program will throw an error.