

Chapter-6

LOOPS

In this chapter we will look **Loops** in more detail. You may recall,

Loop (Iteration) Structure – In this structure, condition is asked over and over and over again until a certain task is complete.

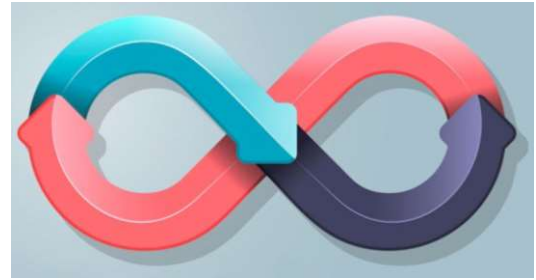


Repetition – **LOOPS** are things that **REPEAT**. Like in the above picture, the iCodeT trainer will teach you the concept of computers until you understand it completely. In this “**Your understanding**” is a condition. Once you understand a particular topic then only the trainer will stop teaching it.

REPETITION IS BORING !

Let's do a simple exercise –

1. Count from 1 to 10
2. Again, Count from 1 to 10
3. Again, Count from 1 to 10
4. One more time. Count from 1 to 10
5. One more time. Count from 1 to 10
6. Are you tired & bored. If no. Count again from 1 to 10
7. One more time count from 1 to 10



For humans, doing the same thing over and over again is very boring. But, computer is our good friend and a machine.

Computers are programmed to do repetitive tasks.

Computer can easily and simply count from 1 to 10 for 7 times.

This process of telling computer to do same task repetitive is done using Loops in programming language.

Why Loops

- Loops allow software developers to use one command for a particular action and repeat the same for multiple times.
- Loops allow software developers to greatly reduce the programming time.
- Loops allow software developers to keep the code clean

Types of Loops – Broadly speaking, there are two types of Loops in computer programming

1. Counting Loops
2. Conditional Loops

Counting Loops – As the word suggests, in counting loop developer writes a code to tell the computer to **repeat** an action a *certain number of times*.

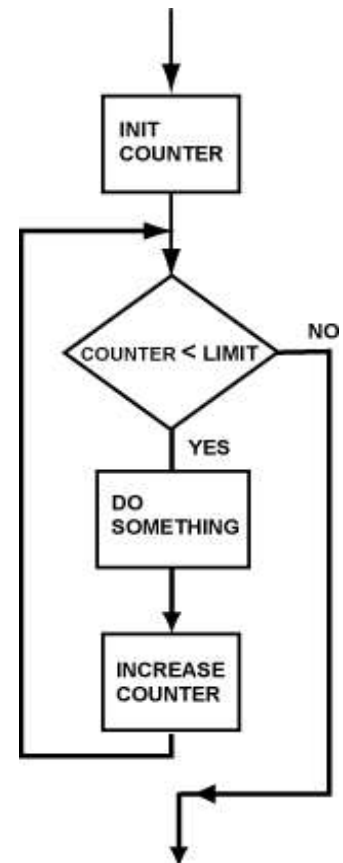
As in the previous example of “Count 1 to 10 for 7 times”. So this is counting loop for 7 counts.

Lets understand the same example using this flow chart.

1. Initial the counter $INIT = 0$. This means we have not started to count.
2. We check whether $INIT < 7$. Currently 0 so Yes
 - a. Do something. This is the action of counting from 1 to 10
 - b. Increase the counter. Now $INIT = 1$.
 - c. Repeat Step 2
3. Once $INIT$ is > 7 , stop

In code.

```
for i in range(7):  
    count from 1 to 10*
```



The code shown above is just for example. This is not a real code and is used to introduce “for loop”

Conditional Loops – As the word suggests, in conditional loop, developer writes a code to tell the computer to **repeat** an action until a **condition is satisfied**.

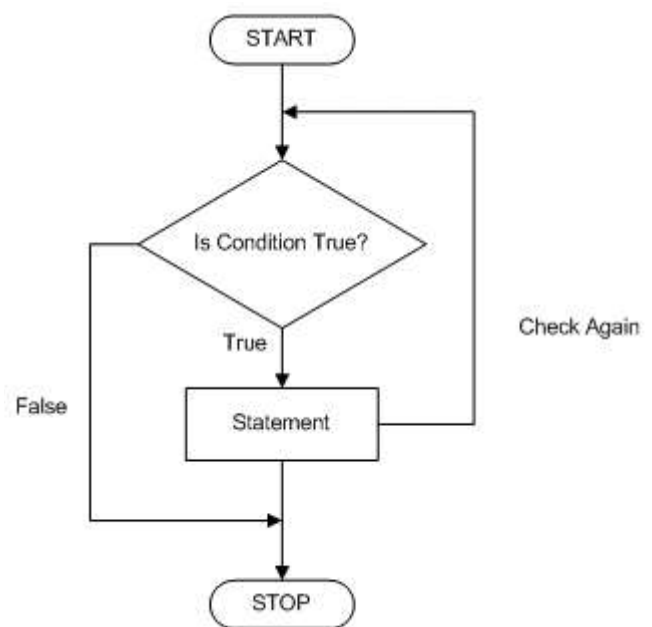
As in the previous example of “Count 1 to 10 for 7 times”. In this we are constantly checking whether the counting is done for 1 time or 2 time or 7 time. Once the counting is done for 7 times then only we stop.

Lets understand the same example using this flow chart.

1. Check if we have not counted yet for 7 times.
 - a. If not. Count from 1 to 10.
2. If the condition at step 1 is false.
That means we have counted for 7 times. We will break and stop.

In code.

*While counting < 8
count from 1 to 10*
increase counting*



- ♣ The code shown above is just for example.
- ♣ This is not a real code and is used to introduce “While loop”

Important Note – As you saw, the same example of counting from 1 to 10 for 7 times is achieved using **Counting loop (for)** as well as **Conditional loop (While)**.

Both **for** and **while** are popular and have different advantages. We will learn them soon.

One more example

1. Reading a 100 page book

○ Counting Loops

- You will read 100 pages of the book from page 1.
- You will read 10 pages of the book from page 1.

○ Conditional Loops

- You will start reading the book from page 1 and stop the book when its completed.
- You will start reading the book from page 1 and stop when you will feel tired.
- You will start reading the book from page 1 and stop when its midnight.
- You will start reading the book from page 1 and stop when lights are turned off.
- You will start reading the book from page 1 and stop when 3 chapters are completed.

FOR vs WHILE Loop

We saw in our last example that **all** “for” loops can be **written** as “while” loops and vice-versa. The difference between “for” and “while” is that in **for loop** the **number of repetition is already known** to obtain a certain result.

Whereas **in while loop, the number of repetition is decided** by the **outcome of condition**. The loop breaks when a certain condition is false.

In other words, you should use

- **for** loops when you know how many times the loop should run i.e, **counting** loop.
- **while** loops when you require to break the loop on certain condition i.e, **condition** loop.



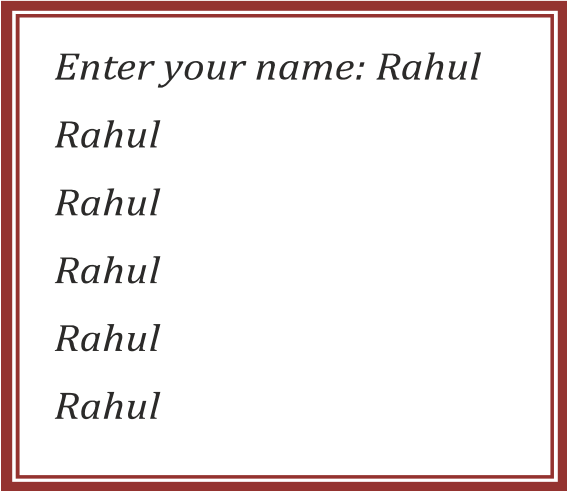
Fun Fact – Use whichever loop seems more appropriate for a given problem

Comparison		
Parameters	For Loop	While Loop
Type	Counting Loop	Conditional Loop
Use	For loop is used when we already know the number of iterations	While loop is used when we do not know the number of iterations
Format	Initialization Counter Condition Statement Increment Repeat	Condition Statement Repeat
Syntax	<pre>for (initialization; condition; iteration) { Statements; //body of for loop }</pre>	<pre>while (condition) { statements; //body of while loop }</pre>
Example	<pre>sum = 0 for i in range(10): sum = sum + i print(sum)</pre>	<pre>sum = 0 i = 0 while (sum<=44): sum = sum + i i = i+1 print(sum)</pre>
Result	<pre>C:\Users\Work\Desktop>python sum.py 0 1 3 6 10 15 21 28 36 45 C:\Users\Work\Desktop></pre>	<pre>C:\Users\Work\Desktop>python sum.py 0 1 3 6 10 15 21 28 36 45 C:\Users\Work\Desktop></pre>

Examples

- ✓ Computer should ask your name and print it for 5 times
Python Code -

```
name = input("Enter your name: ")  
for i in range(5):  
    print(name)
```



```
Enter your name: Rahul  
Rahul  
Rahul  
Rahul  
Rahul  
Rahul
```

- ✓ Add a number 2 by 3 and replace it with the result. Do this till the final result is not 14
Python Code-

```
num = 2  
to_add = 3  
while(num != 14):  
    num = num + to_add  
    print(num)
```



```
5  
8  
11  
14
```

Practice Scenarios

1. Brush your tongue 3 times

2. Do the below dance sequence till you are tired

- Right leg forward
- Left leg forward
- Clap

3. Climbing 5 Stairs. Sequence

- Left leg up
- Left leg forward
- Left leg down
- Right leg up
- Right leg forward
- Right leg down

4. Subtract scenario

- Subtract a given number by 2 until the number becomes 0
- Let's say a given number is 10. Then check a condition whether the number has become 0. If not then subtract by 2. Continue this cycle.