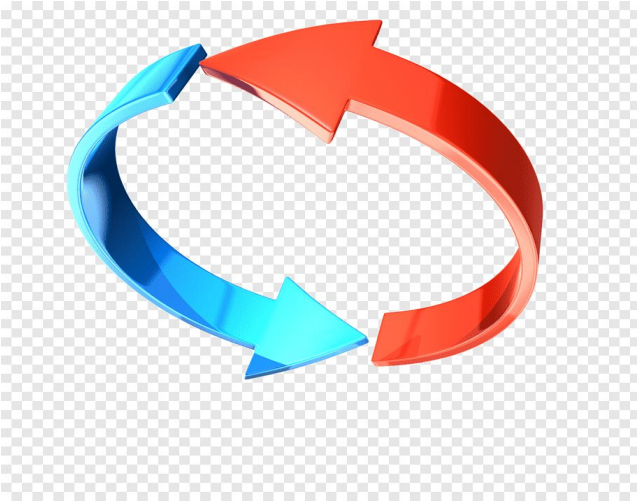
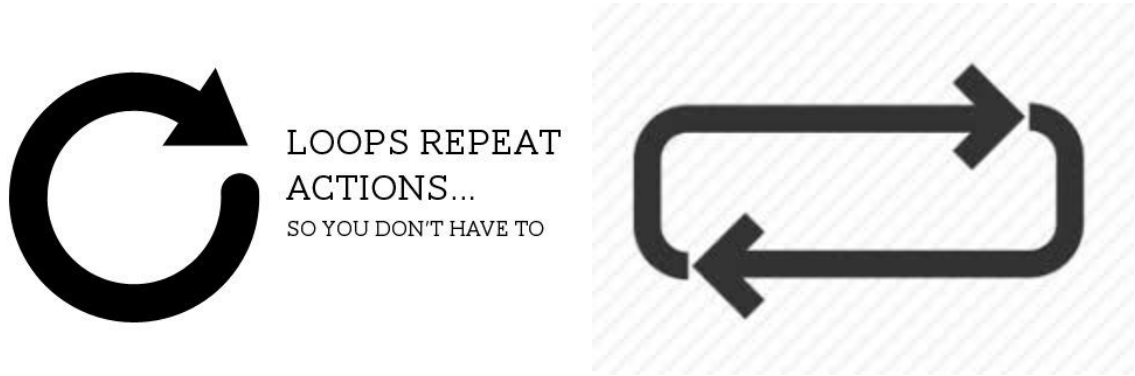
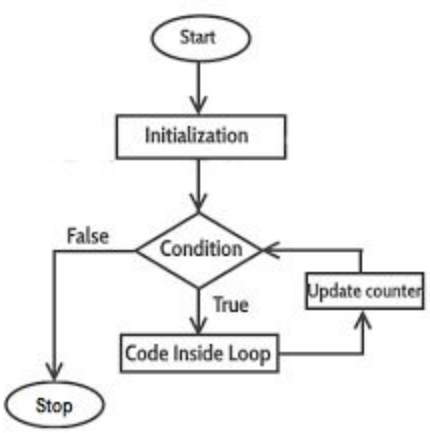
Week-3: Summary:

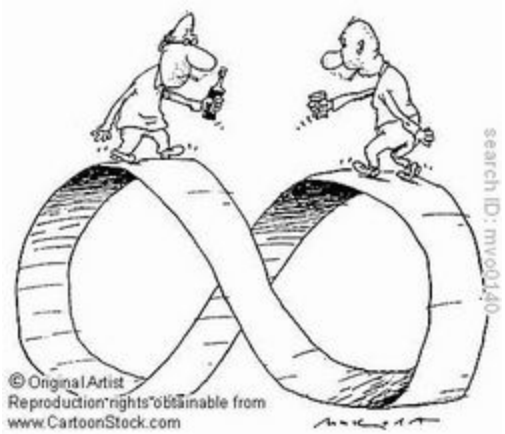
First, we will learn about the **iteration or loop**.



As loops repeat actions so you don't have to do this multiple times…

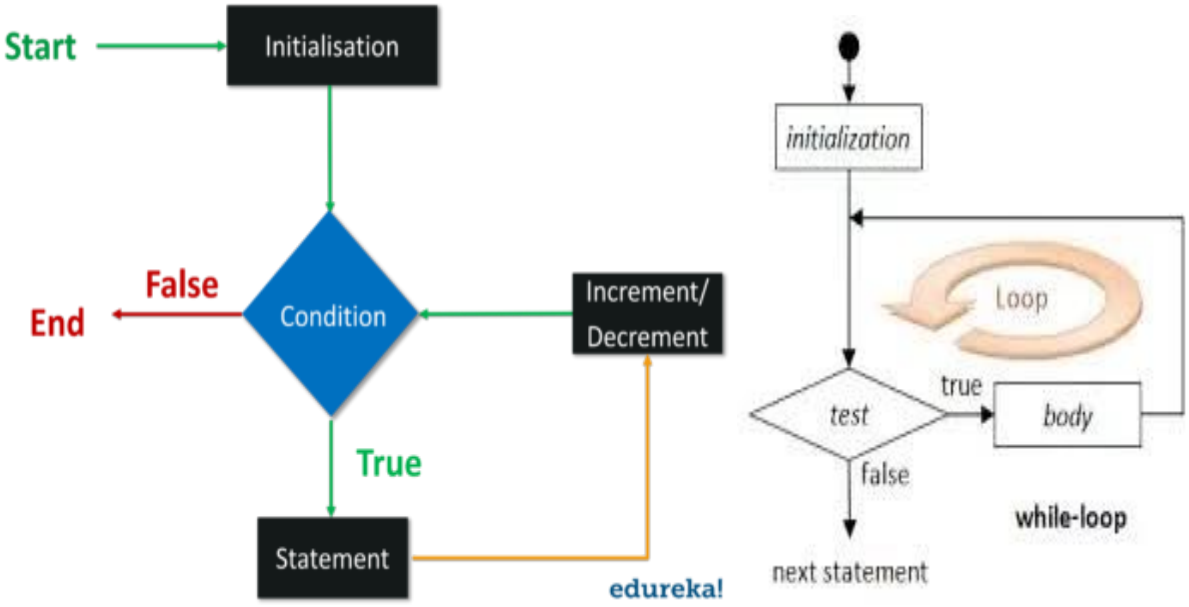






Confused? Don’t worry we will describe one by one?

We can repeat the same task using **a loop.**



After the **initialization**, it will **check the condition** until false and it will run.
Caution: Don’t use the infinite conditions.

Python Loops

Python has two primitive loop commands:

- **while loops**
- **for loops**

What is Loop?

Loops can execute a block of code number of times until a certain condition is met. Their usage is fairly common in programming. Unlike other programming language that have For Loop, while loop, do while, etc.

What is a For Loop?

For loop is used to iterate over elements of a sequence. It is often used when you have a piece of code which you want to repeat "n" a number of times.

What is a While Loop?

While Loop is used to repeat a block of code. Instead of running the code block once, It executes the code block multiple times until a certain condition is met.

How to use "While Loop"

While loop does the exact same thing as what "if statement" does, but instead of running the code block once, they jump back to the point where it began the code and repeats the whole process again.

Syntax:

while **expression**: # eg: **a<5**

    statement # eg: **a = a + 1**

Print i as long as i is less than 6:

```
i = 1
while i < 6:
  print(i)
  i += 1
```

The while loop requires relevant variables to be ready, in this example we need to define an indexing variable, i, which we set to 1.

The break Statement

With the break statement we can stop the loop even if the while condition is true:

```
i = 1
while i < 6:
  print(i)
  if i == 3:
    break
  i += 1
```

How to use "For Loop"

In Python, "for loops" are called **iterators.**

Just like while loop, "For Loop" is also used to repeat the program.

But unlike while loop which depends on condition true or false. "For Loop" depends on the elements it has to iterate.

Example:

```
for i in range(0, 5):
    print(i)
```

Output:

0

1

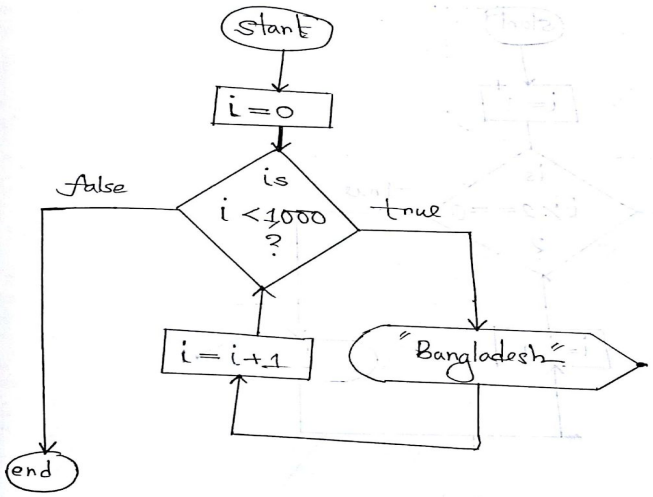
2

3

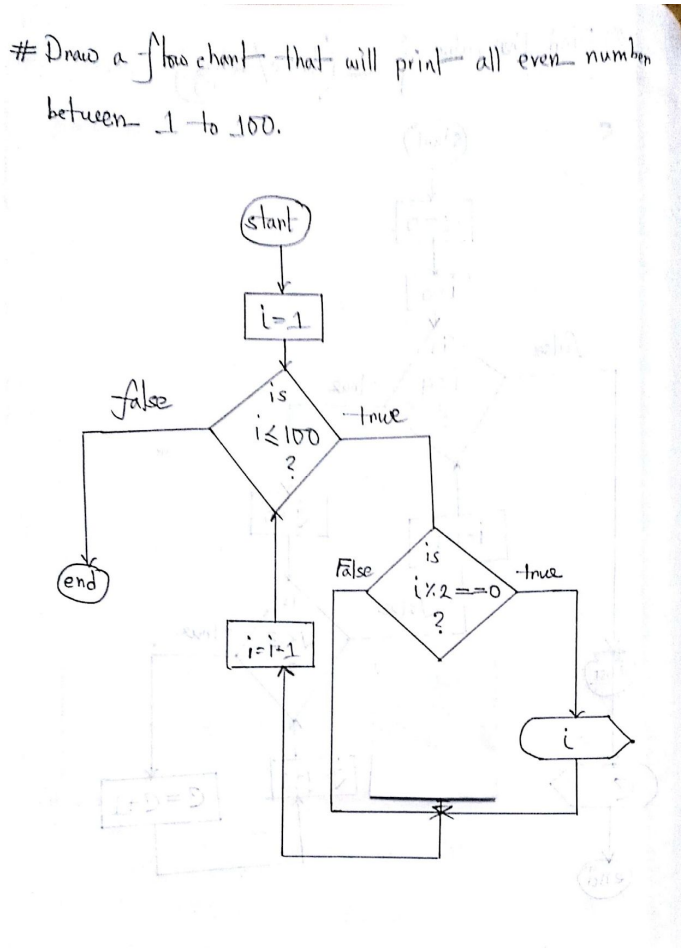
4

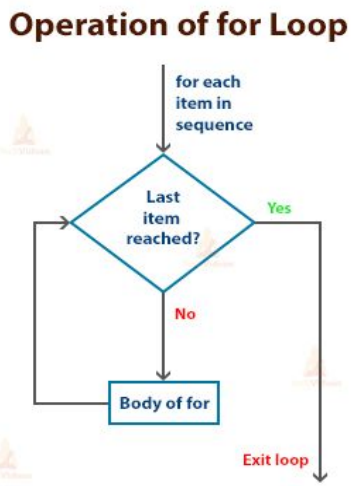
In this problem, we'll print **'Bangladesh'** 1000 times using a simple loop. To do this task, we will use a for a loop.

#Draw a flowchart that will print "Bangladesh" 1000 times.



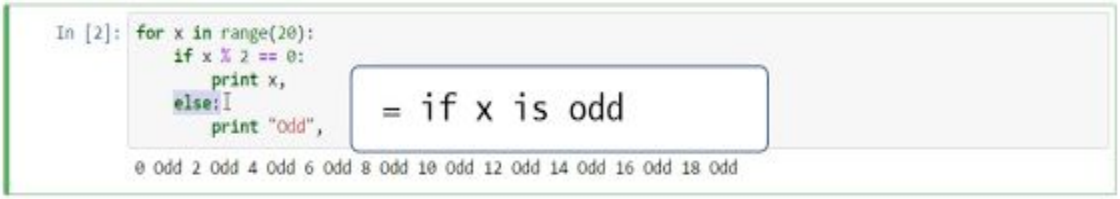
On the next problem, we will print all the even numbers between 1 to 100.





Now, we'll trace/debug the code. That means we'll follow the value of c on every step. This time we are working with a nested loop. A nested loop means a loop inside another loop.

Print the odd-even until 20.



Let's draw a flow chart then, we will implement it.
Condition until 20, and check.

Practical Example:

Let see another example for For Loop to repeat the same statement over and again.

Python loop	Working Code for all exercises

Code for while loop	`x=0`
	`while (x<4):`
	`  print (x)`
	`  x= x+1`

For Loop Simple Example	`x=0`
	`for x in range (2,7):`
	`  print (x)`

Use of for loop in string	`Months = ["Jan","Feb","Mar","April","Ma`
	`y","June"]`
	`for m in (Months):`
	`  print (m)`

Practice Problems

---

1. Input a number, then reverse it. (1234 to 4321; 596254 to 452695).
2. Check a number whether it is palindrome or not.
3. Count the digit of a number.
4. Count the unique-digit of a number.
5. Take two inputs, base(b) and power(p) determine the value.
6. Fibonacci Series (0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89 ... upto N)
7. Print series: 3, 6, 9, 12, 15, 18, 21, 24, 27 upto N.
8. Print 3n+1 series (if number is even then n=n/2, otherwise n=3n+1).
9. Factorial (3! = 6, 4! = 24, 5! = 120, 6! = 720, 7! = 5040)