Python-Conditional Statements and Loop

What are Conditional Statements in Python?

Conditional Statement in Python perform different computations or actions depending on whether a specific Boolean constraint evaluates to true or false. Conditional statements are handled by IF statements in Python.

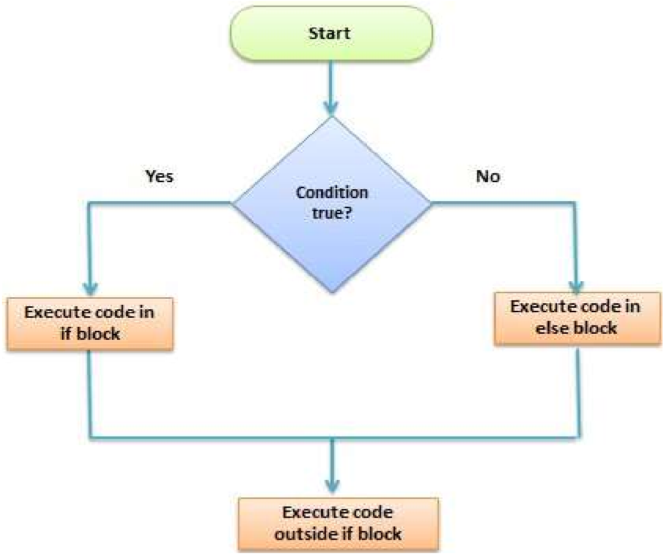
IF CONDITION:

- Python if Statement is used for decision-making operations. It contains a body of code which runs only when the condition given in the if statement is true. If the condition is false, then the optional else statement runs which contains some code for the else condition.

- When you want to justify one condition while the other condition is not true, then you use Python if else statement.

```
SYNTAX:
        If <Condition>:
                Statement
        Else:
                Statement
```

Python if…else Flowchart:



How to use “else condition”?

The “else condition” is usually used when you have to judge one statement on the basis of other. If one condition goes wrong, then there should be another condition that should justify the statement or logic.

When “else condition” does not work?

- There might be many instances when your “else condition” won’t give you the desired result. It will print out the wrong result as there is a mistake in program logic. In most cases, this happens when you have to justify more than two statement or condition in a program.

ELIF:

```
SYNTAX:
        If <Condition>:
                Statement
```

```
Elif<Condtion>:
        Statement

Else:
        Statement
```

Nested If:

- One if condition has another if condition inside that.

SYNTAX:

```
If<condition>:
    Statement
            If<condition>:
                Statement
            Elif<condition>:
                Statement
           Else<condition>:
                Statement
Elif<condition>:
        Statement
Else:
  Statement
```

INLINE IF:

- Single line if condition is called Inline IF
- Fast executation
- It can be embedded into another python object
- It return only one value

  SYNTAX:

```
<Statement>if<Statement>
Eg:

Res= <"True Output">if age>=18<"False Output">
Print(Res)
```

Switch Case Statement in Python:

- A switch statement is a multiway branch statement that compares the value of a variable to the values specified in case statements.
- Python language doesn't have a switch statement.
- Python uses dictionary mapping to implement Switch Case in Python.

Iterative Statements/Loops:

It is a process of repeated execution block of code until a condition is satisfied.

Repeated execution of a block of code is called Iteration.

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

Types of Loops/ Iterative Statement:

Indefinite Loop:

- A loop which is executed infinite iterations is called Indefinite loop.
- In this no. of iteration cannot be predefined.
- While loop

Definite Loop:

- A loop which is executed finite number of times is called Definite loop.
- In this no. of iterations are predefined.
- For loop

IMPORTANT:

Number(While Loop)

Loop…..endloop

While….- minimum 0 time execute

Do….while- minimum 1 time execute

For loop-

Sequential (For Loop)

For each- Sequential data types/extract the data from internal memory.

Perfect Loop:

1. Initialization
   Assigning starting value to a variable.
2. Condition
   Giving a condition
3. Incrementation
   Increasing the value

What is 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 exactly same thing 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:

```
<Initialization>
While <Condition>:
      Statement
      <Increment>
```

What is 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" number of time.

- It is specially designed to extract data from sequential data type.
- It is exact replica of "for each loop" in other programming language.

SYNTAX:

```
For <ITERATOR> in <ITERABLE>:
   Print (<ITERATOR>)
```

- **ITERATION:**

  It is process of repeatedly executing a block of code.

- **ITERABLE:**

  An object on which a process of iteration is perform is called Iterable.

  It are sequential data type(string, list, tuple)

  Everything we can loop over is called Iterable

  This Iterable gives Iterators

  It is an object which is gives elements to Iterators.

  - This Iterables will calls and built-in function i.e. iter().
  - Iter(l)-<list_Iterator object at 001D098….>
  - Next(): It is used to call the next element in list.

- **ITERATOR:**

  It is an object which acts like an agent in extracting data from internal memory location of sequential data types.

Eg:

```
L1= ["BLUE","RED","GREEN","ORANGE"]
For x in L1:
    Print(x)
```

Loop:

- Initialization—iter()
- Iter()- It initialize the iterator with itrable memory location
- Incrementation—next()
- Next()- It extracts next value

- To extend functionalities of for loop towards numbers,we have to use an external function called as range().
- **RANGE():**

► It generates the values considering starting value, ending value and step.

► Range is a independent function.

► **SYNTAX:** range(<starting value>, <ending value>, <step>)

► Starting Value: From where values has to be generated.

► Default value will be 0.

► Ending Value: Till where values has to be generated.

- **STEP:** It defines how to generate value using starting and ending value.

► Default step is 1.

► Range(, ,)-error

► In range function out of these three one is mandatory, i.e 'Ending value'

► Accept only integer value.

► Dimension is nothing but direction.

INTERRUPTIVE STATEMENT:

It interrupts the flow of statement.

- Break: It is used to stop the execution of the loop.
  After break statement…execution continues with rest of the program.
- Continue: It is used to skip current iteration based on the condition.
- Exit(): It stops the execution of the program and terminate complete program.
- Pass: It helps us to execution the program without any new block implementation.

PASS:

```
For I range(1,10):
    Pass
Print("THIS IS A CAT")
```

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.

What is enumerate() in Python?

enumerate() IN PYTHON is a built-in function used for assigning an index to each item of the iterable object. It adds a loop on the iterable objects while keeping track of the current item and returns the object in an enumerable form. This object can be used in a for loop to convert it into a list by using list() method.

Example:
Enumerate function is used for the numbering or indexing the members in the list.
Suppose, we want to do numbering for our month (Jan, Feb, Marc, ….June), so we declare the variable i that enumerate the numbers while m will print the number of month in list

Switch Case Statement in Python:

What is Switch statement?

- A switch statement is a multiway branch statement that compares the value of a variable to the values specified in case statements.
- Python language doesn't have a switch statement.
- Python uses dictionary mapping to implement Switch Case in Python

Summary:
A conditional statement in Python is handled by if statements and we saw various other ways we can use conditional statements like Python if else over here.

- "if condition" – It is used when you need to print out the result when one of the conditions is true or false.
- "else condition"- it is used when you want to print out the statement when your one condition fails to meet the requirement
- "elif condition" – It is used when you have third possibility as the outcome. You can use multiple elif conditions to check for 4th,5th,6th possibilities in your code
- We can use minimal code to execute conditional statements by declaring all condition in single statement to run the code
- Python If Statement can be nested

Like other programming languages, Python also uses a loop but instead of using a range of different loops it is restricted to only two loops "While loop" and "for loop".

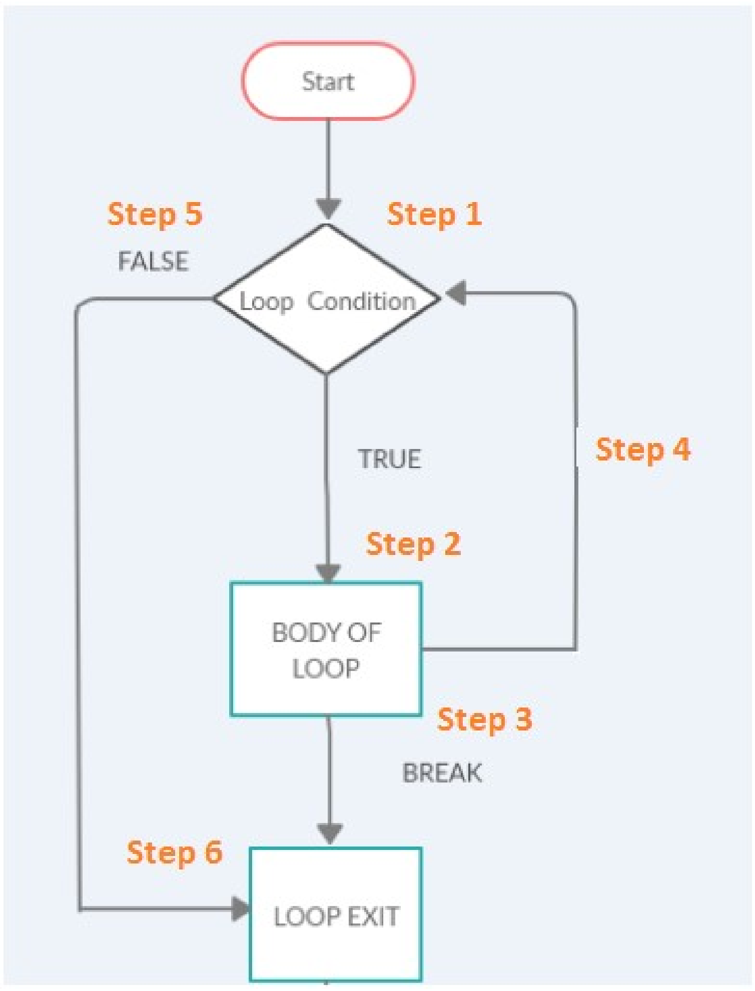
- While loops are executed based on whether the conditional statement is true or false.
- For loops are called iterators, it iterates the element based on the condition set
- Python For loops can also be used for a set of various other things (specifying the collection of elements we want to loop over)

- Breakpoint is used in For Loop to break or terminate the program at any particular point
- Continue statement will continue to print out the statement, and prints out the result as per the condition set
- Enumerate function in “for loop” returns the member of the collection that we are looking at with the index number

Python break statement:

The break statement takes care of terminating the loop in which it is used. If the break statement is used inside nested loops, the current loop is terminated, and the flow will continue with the code followed that comes after the loop.

The flow chart for the break statement is as follows:



The following are the steps involved in the flowchart.

Step 1)
The loop execution starts.
Step 2)
If the loop condition is true, it will execute step 2, wherein the body of the loop will get executed.
Step 3)
If the loop's body has a break statement, the loop will exit and go to Step 6.
Step 4)
After the loop condition is executed and done, it will proceed to the next iteration in Step 4.
Step 5)
If the loop condition is false, it will exit the loop and go to Step 6.
Step 6)
End of the loop.

Break statement execution flow:

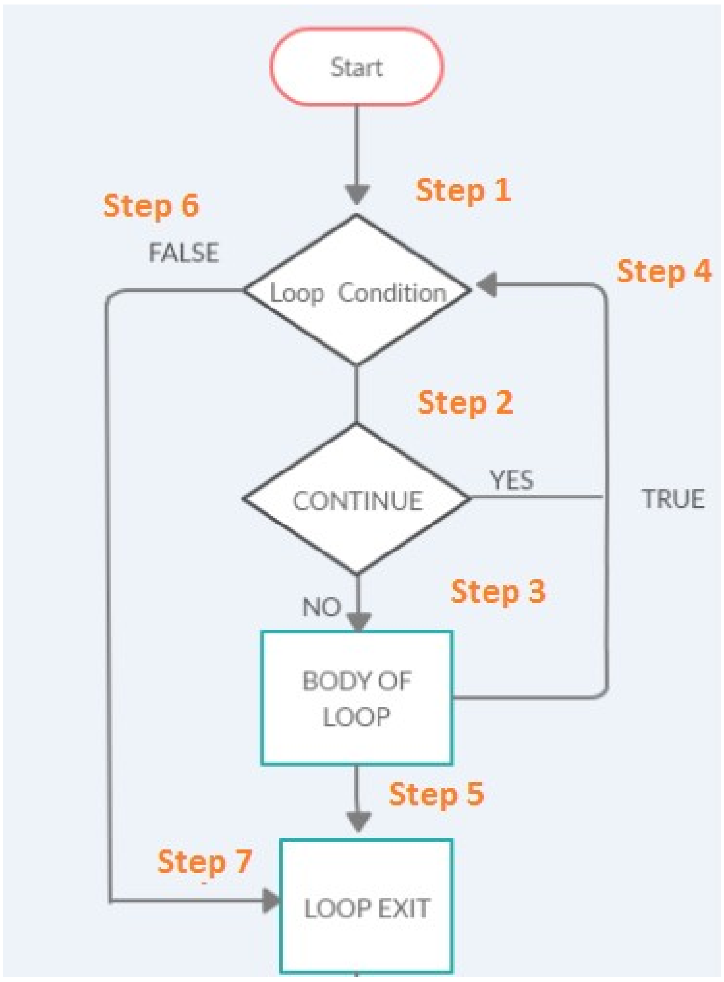
- When the for-loop starts executing, it will check the if-condition. If **true**, the break statement is executed, and the for–loop will get terminated. If the condition is false, the code inside for-loop will be executed.

- When the while loop executes, it will check the if-condition; if it is **true,** the break statement is executed, and the while –loop will exit. If if the condition is false, the code inside while-loop will get executed

Python continue statement:

The **continue** statement skips the code that comes after it, and the control is passed back to the start for the next iteration.

Continue flow Chart



The following are the steps involved in the flowchart.

Step 1)
The loop execution starts.
Step 2)
The execution of code inside the loop will be done. If there is a continued statement inside the loop, the control will go back to Step 4, i.e., the start of the loop for the next iteration.
Step 3)
The execution of code inside the loop will be done.
Step 4)
If there is a continue statement or the loop execution inside the body is done, it will call the next iteration.
Step 5)
Once the loop execution is complete, the loop will exit and go to step 7.
Step 6)
If the loop condition in step 1 fails, it will exit the loop and go to step 7.
Step 7)
End of the loop.

Continue statement execution flow:

- The for –loop, loops through my_list array given. Inside the for-loop, the if-condition gets executed. If the condition is **true**, the continue statement is executed, and the control will pass to the start of the loop for the next iteration.

- When the while loop executes, it will check the if-condition, if it is **true,** the continue statement is executed. The control will go back to the start of while –loop for the next iteration. If if the condition is false, the code inside while-loop will get executed.

What is pass statement in Python?

Python pass is a null statement. When the Python interpreter comes across the across pass statement, it does nothing and is ignored.

When to use the pass statement?

- Consider you have a function or a class with the body left empty. You plan to write the code in the future. The Python interpreter will throw an error if it comes across an empty body.
- A comment can also be added inside the body of the function or class, but the interpreter ignores the comment and will throw an error.
- The pass statement can be used inside the body of a function or class body. During execution, the interpreter, when it comes across the pass statement, ignores and continues without giving any error.

When to use a break and continue statement?

- A **break** statement, when used inside the loop, will terminate the loop and exit. If used inside nested loops, it will break out from the current loop.
- A **continue** statement will stop the current execution when used inside a loop, and the control will go back to the start of the loop.

❖ The main difference between break and continue statement is that when break keyword is encountered, it will exit the loop.
❖ In case of continue keyword, the current iteration that is running will be stopped, and it will proceed with the next iteration.

Summary:

- Python break and continue are used inside the loop to change the flow of the loop from its normal procedure.
- A for-loop or while-loop is meant to iterate until the condition given fails. When you use a break or continue statement, the flow of the loop is changed from its normal way.
- A **break** statement, when used inside the loop, will terminate the loop and exit. If used inside nested loops, it will break out from the current loop.
- A **continue** statement, when used inside a loop, will stop the current execution, and the control will go back to the start of the loop.
- The main difference between **break** and **continue** statement is that when **break** keyword is encountered, it will exit the loop.
- Python Pass Statement is used as a placeholder inside loops, functions, class, if-statement that is meant to be implemented later.
- Python pass is a null statement. When the execution starts and the interpreter comes across the pass statement, it does nothing and is ignored