5 Python - Control Flow Statements

A program is a sequential set of statements arranged in logical order which are executed from top to bottom to perform the desired task. But sequential flow of instructions does not help us in solving the problems, in many cases, especially when there are two or more conditions available to process the data values, we tend to skip the unwanted code and execute the required part with the help of control statements.

LESSON SKILLS

- if Statement
- if..else Statement
- if..elif..else Statement
- Nested if Statement

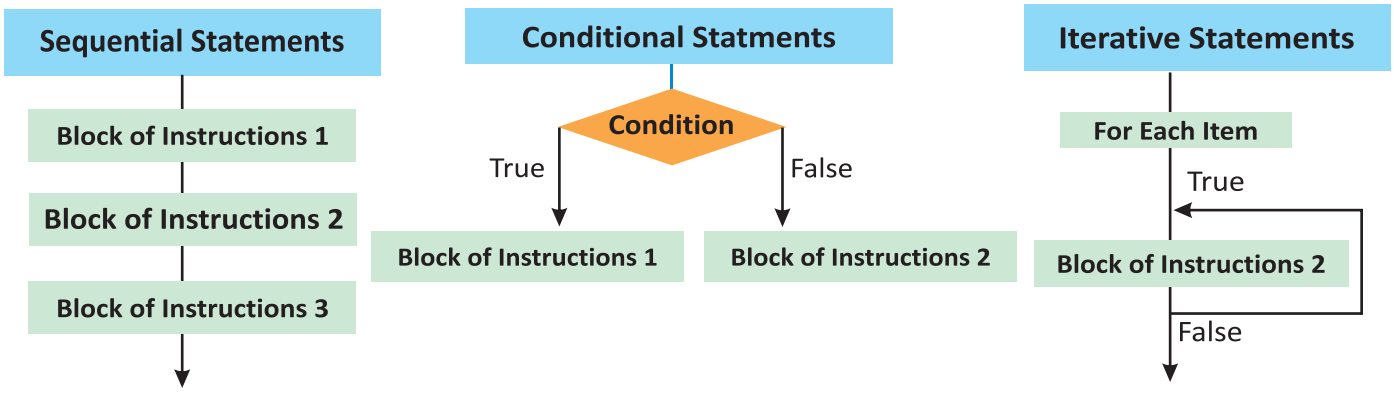
In this chapter, we will discuss the use of control statements which are used for creating decision based programs.

TYPES OF CONTROL FLOW STATEMENTS

The control flow statements in Python Programming Language are:

1. **Sequential Statements :** This refers to the line by line execution, in which the statements are executed sequentially, in the same order in which they appear in the program.
2. **Conditional Statements :** Depending on whether a condition is true or false, the decision structure may skip the execution of an entire block of statements or even execute one block of statements instead of other. **if, if...else** and **if...elif...else** are the decision control flow statements.
3. **Iterative Statements :** This is a control structure that allows the execution of a block of statements multiple times until a loop termination condition is met. **for loop** and **while loop** are the two loop control flow statements.

   Loop Control Flow Statements are also called **Repetition** statements or **Iterative** statements.

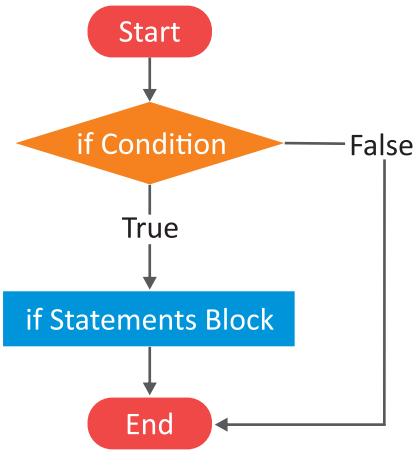


Syntax

```
if <conditional expression>:
    statement 1
    statement 2
    ...
    statement N
.....
```

Condition

if Statements Block



- The if conditional statement starts with **if** keyword and ends with a **colon (:).**
- The expression in an if statement should be a conditional expression. The if statement decides whether to run some particular statement or not depending upon the value (True or False) of the conditional expression.
- If the conditional expression evaluates to true then statements in the if block will be executed; otherwise none of the statements are executed.
- It is mandatory in Python to indent the statements in the if block, else it will display an error.

The following examples demonstrate the if condition:

Example 1 : Program to determine whether a person is eligible to vote according to the age entered.

Program

```
age = int(input("Enter the age: "))
if(age >= 18):
    print ("You are eligible to vote")
```

Output

```
= RESTART: C:/Users/Administrator.SCHOOLSOFT/De
sktop/Desktop/Python Programs/Chapter 4/Chapter
4_3.py
Enter the age: 20
You are eligible to vote
>>>
```

In the above example, the condition age>=18 evaluates to true, so it will execute the statements in if block. Above if block contains only one print statement.

Example 2 : Program to check if amount>1000 then apply 10% discount.

Program

```
price = int(input("Enter the price: "))
quantity = int(input("Enter the quantity: "))
amount = price * quantity
print("Amount: ", amount)
if(amount >1000):
    discount = amount*10/100
    print ("Discount = ", discount)
    amount = amount-discount
print("Amount payable: ", amount)
```

Output

```
>>>
====== RESTART: C:/Users/Administrator.SCHOOLSOFT
/Desktop/If Condition.py ======
Enter the price: 100
Enter the quantity: 20
Amount:  2000
Discount =  200.0
Amount payable:  1800.0
>>>
```

IF...ELIF...ELSE STATEMENT

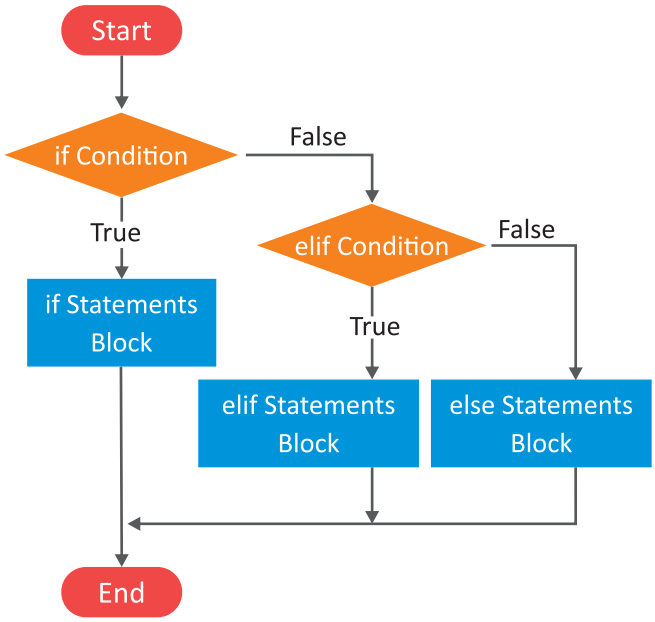
Sometimes, we need to work with multiple conditions. In this case, Python supports if...elif...else statement to test additional conditions. The **elif** condition is used to include multiple conditional expressions between if and else.

It is not necessary that every if statement should have an else block. After the first conditional expression or the first if branch, a Python program can have as many elif's branches depending upon the expressions that have to be tested.

A series of if..elif statement can have a final else block, which is executed if none of the if or elif expression is true.

Syntax

```
if <conditional expression1>:        → Condition 1
    statement(s)                     → if Statements Block
elif <conditional expression2>:      → Condition 2
    statement(s)                     → elif Statements Block
elif <conditional expression3>:
    statement(s)
else:
    statement(s)                     → else Statements Block
```



If the conditional expression 1 is false, it checks the conditional expression 2 and so on. If all the conditions are false, statement block of else is executed.

To understand the if...elif...else statement, let us understand with the following examples.

Example 1 : Program to check whether a number entered by the user is negative, positive or equal to zero.

Program

```
num = int(input("Enter a number: "))
if (num == 0):
    print("The number is equal to zero")
elif (num > 0):
    print("The number is positive")
else:
    print ("The number is negative")
```

Output

```
>>>
= RESTART: C:/Users/Administrator.SCHOOLSOFT/Desktop/Desktop/Python Programs/Chapter 4/Chapter 4_8.py
Enter a number: 15
The number is positive
>>>
```

Program

```
ch = input("Enter any character : ")
if(ch == "A" or ch == "E" or ch == "I" or ch == "O" or ch == "U"):
    print(ch, "is a vowel")
elif(ch == "a" or ch == "e" or ch == "i" or ch == "o" or ch == "u"):
    print(ch, "is a vowel")
else:
    print(ch, "is not a vowel")
```

Output

```
>>>
= RESTART: C:/Users/Administrator.SCHOOLSOFT/Desk
top/Desktop/Python Programs/Chapter 4/Chapter4_13
.py
Enter any character : O
O is a vowel
>>>
```

In the given example, the character entered is O, so the if condition results true and the statements in the if condition will get executed. Hence the result displayed will be 'O is a vowel'.

NESTED IF STATEMENT

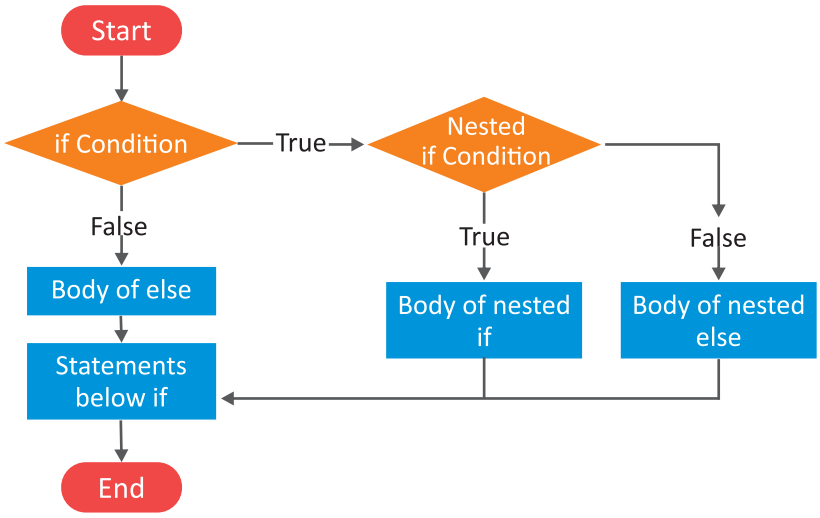
There may be a situation when you want to check for another condition after a condition resolves to true. In such a situation, you can use the **nested if** construct.

In a nested if construct, you can have an **if...elif...else** construct inside another **if...elif...else** construct.

The inner condition must be with increased indentation than the outer condition, and all the statements under one block should be with the same indentation.

Syntax

```
if <conditional expression1>:
    if <conditional expression2>:
        statement(s)
    else:
        statement(s)
else:
    statement(s)
```



Example 1: Program to check whether a number entered by the user is negative, positive or equal to zero, using nested if construct.

Program

```
num = int(input("Enter a number: "))
if num >= 0:
    if num == 0:
        print("Zero")
    else:
        print("Positive number")
else:
    print("Negative number")
```

Output

```
Type "help", "copyright", "credits" or "license()
" for more information.
>>>
= RESTART: C:\Users\Administrator.SCHOOLSOFT\Desk
top\Desktop\Python Programs\Chapter 4\Chapter4_16
.py
Enter a number: 5
Positive number
>>>
```

C Fill in the blanks

Hints Else | Conditional | Iterative | if...elif...else | Indentation

1. Loop control statements are also called ..... statements.
2. All the statements under the if condition start with an increased ..... .
3. A series of if...elif statements have a final ..... block, which is executed if none of the if or elif expressions is true.
4. In Python, ... statement is used to check multiple conditions.
5. Statements that allows you to give conditions are called as ..... statement.

D Answer the following questions

1. Differentiate between sequential and conditional statements.

   .....

   .....

2. It is necessary for every 'if' block to be accompanied with an 'else' block. Comment on this statement with the help of an example.

   .....

   .....

3. Write the syntax of the if statement.

   .....

   .....

4. When is if...elif...else statement used?

   .....

   .....

5. Explain nested if statement.

   .....

   .....

B **Find the error and rewrite the code.**

Program

```
num = 100
if num>100:
    print(num)
```

Solution

Program

```
x = 3
  if x>0
     print("POSITIVE")
print("This is already printed")
```

Solution

C **Write conditional expressions for the following :**

Program

a. If a<10 b = 20, else b = 30

b. Print 'Morning' if time < 12, otherwise print 'Afternoon'

c. If marks >=70, set remarks to True, otherwise False.

Solution

LAB TIME

A **Write Python programs for the following:**

1. Commission on sales by a salesman is calculated as per the following policy:

Amount of Sales (in Rs.)	Commission Rate
0 - 10000	Nil
10001 - 40000	5%
40001 - 60000	7.5%
>60000	10%

Write a Python program that accepts sales made by a salesman and displays the commission due.

TERM I - SKILL SHEET
Chapter (1, 2, 3, 4, 5)

A. Tick (✓) the correct answer

1. The data entered into the computer is converted into _____ code for processing.
   a. Decimal ☐ b. Binary ☐ c. Octal ☐
2. Which of the following is not one of the types of Sparkline?
   a. Line ☐ b. Area ☐ c. Win/Loss ☐
3. To make the layer nearly transparent, set the opacity to _____ .
   a. 50% ☐ b. 100% ☐ c. 1% ☐
4. The values stored in the tuple are enclosed within the _____.
   a. {} ☐ b. () ☐ c. [] ☐
5. _____ is a short form of 'else...if' statement.
   a. Elif ☐ b. Else If ☐ c. If ☐

B. State T for True and F for False

1. 1 + 1 equals to 0 in binary addition. ☐
2. You can create a chart instantly on a new worksheet, by pressing F11 key. ☐
3. You cannot delete a layer once it is added in Layers Panel. ☐
4. Comparison operators return either True or False according to the condition. ☐
5. You cannot use logical operators in 'if' statement condition. ☐

C. Fill in the blanks

Hints: Column | Dictionary | Nibble | Eye

1. _____ consists of 4 bits.
2. _____ chart is used for plotting more than one data series.
3. _____ is a collection of unordered key-value pairs.
4. _____ icon is used to hide/unhide the layer in Krita.