Vidyavardhini’s College of Engineering & Technology

Department of Computer Engineering

Experiment No. 2
To implement Conditional Statements and Loop in python
Name : Vaidehi D. Gadag
Branch/Div.: Comps-1 (C47)
Date of Performance: 10/01/2024
Date of Submission: 31/01/2024

Experiment No. 2

Title: To implement Conditional Statements and Loop in python

Aim: To study, and implement Conditional Statements and Loop in python

Objective: To introduce Conditional Statements and Loop in python

Theory:

- Conditional Statements

There comes situations in real life when we need to do some specific task and based on some specific conditions and, we decide what should we do next. Similarly there comes a situation in programming where a specific task is to be performed if a specific condition is True. In such cases, conditional statements can be used. The following are the conditional statements provided by Python.

if

if..else

Nested if

if-elif statements.

Let us go through all of them.

if Statement

If the simple code of block is to be performed if the condition holds true than if statement is used. Here the condition mentioned holds true then the code of block runs otherwise not.

if..else Statment

In conditional if Statement the additional block of code is merged as else statement which is performed when if condition is false.

Nested if Statement

if statement can also be checked inside other if statement. This conditional statement is called nested if statement. This means that inner if condition will be checked only if outer if condition is true and by this, we can see multiple conditions to be satisfied.

if-elif Statment

The if-elif statement is shoutcut of if..else chain.While using if-elif statement at the end else block is added which is performed if none of the above if-elif statement is true.

- Looping in python

Python programming language provides following types of loops to handle looping requirements. Python provides three ways for executing the loops. While all the ways provide similar basic functionality, they differ in their syntax and condition checking time.

While Loop:

In python, while loop is used to execute a block of statements repeatedly until a given a condition is satisfied. And when the condition becomes false, the line immediately after the loop in program is executed.

for in Loop:

For loops are used for sequential traversal. For example: traversing a list or string or array etc. In Python, there is no C style for loop, i.e., for (i=0; i<n; i++). There is "for in" loop which is similar to for each loop in other languages. Let us learn how to use for in loop for sequential traversals.

Code:

```
a = input('Enter the first number : ')
b = input('Enter the second number : ')
c = input('Enter the third number : ')
# if statement
if a > b:
    print('{0} is greater than {1}'.format(a,b))
# if...else statement
if a > b:
    print('{0} is greater than {1}'.format(a,b))
else:
    print('{1} is greater than {0}'.format(a,b))
# nested if statement
if a > c:
    print('{0} is greater than {1}'.format(a,c))
    if a > b:
        print('{0} is greater than {1}'.format(a,b))
    else:
        print('{0} is less than {1} and {2}'.format(a,b,c))
# if..elif statement
age = 18
if age < 13:
    print("You are a child.")
elif age < 20:
    print("You are a teenager.")
elif age < 65:
    print("You are an adult.")
else:
    print("You are a senior citizen.")
```

```
#for loop
characters = ['RJ','Koya','Schooky','Mang','Chimmy','Tata','Kookie']
for x in characters:
    print(x)
#while loop
i = 1
while i < 4:
    print('Hello')
    i+=1
```

Output:

```
Enter the first number : 12
Enter the second number : 11
Enter the third number : 10
12 is greater than 11
12 is greater than 11
12 is greater than 10
12 is greater than 11
You are a teenager.
RJ
Koya
Schooky
Mang
Chimmy
Tata
Kookie
Hello
Hello
Hello
```

Conclusion:

In conclusion, conditional and looping statements are fundamental concepts in Python that allow for the creation of dynamic and responsive code. By using 'if', 'elif', and 'else' statements, you can control the flow of your program based on whether certain conditions are met. This enables you to write code that can handle a variety of inputs and situations, making your programs more robust and versatile.

Looping statements, on the other hand, allow you to repeat a block of code as many times as necessary. The 'for' loop is useful for iterating over sequences and iterables, while the 'while' loop can repeatedly execute a block of code as long as a certain condition is true. By mastering these looping constructs, you can write code that can efficiently and effectively process large amounts of data or perform repetitive tasks.

Overall, the implementation of conditional and looping statements in Python is essential for creating complex and dynamic programs. By combining these concepts with other Python features, you can build powerful and flexible applications that can handle a wide range of inputs and situations.