MIT World Peace University

Python Programming
Second Year B. Tech, Semester 4

Learning about If Else statements in Python

Assignment No. 2

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 20

February 28, 2023

Contents

1 Aim

To Write a python program to find largest of three numbers.

2 Objectives

1. To learn and Implement different types of If-else statements.

3 Problem Statement

Learn different kinds of If statements in python, and understand some Data Types in Python.

4 Theory

4.1 Decision Making

Following is the general form of a typical decision making structure found in most of the programming languages.

```
if condition:
    statement(s)
elif condition:
    statement(s)
else:
    statement(s)
```

Python programming language assumes any non-zero and non-null values as *true*, and if it is either zero or null, then it is assumed as *false* value.

Python programming language provides the following types of decision making statements.

4.2 If Statements

An *if statement* consists of a boolean expression followed by one or more statements.

```
if expression:
    statement(s)
```

[1]:
```
var = 100
if var == 100:
    print("Value of expression is 100")
print("Good bye!")
```

```
Value of expression is 100
Good bye!
```

4.3 If . . . Else Statements

An *if statement* can be followed by an optional *else statement*, which executes when the boolean expression is *false*.

```
if expression:
    statement(s)
else:
    statement(s)
```

[3]:
```
if var == 200:
    print("Value of expression is 200")
else:
    print("Value of expression is not 200")
```

```
Value of expression is not 200
```

4.4 IF...ELIF...ELSE Statements

An *if statement* can be followed by an optional *elif statement*, which is short for *else if*. It is very useful to test various conditions for decision making.

```
if expression:
    statement(s)
elif expression:
    statement(s)
else:
    statement(s)
```

[9]:
```
import random
num = random.randint(-10, 10)
print(num)
if num > 0:
    print("Positive number")
elif num < 0:
    print("Negative number")
else:
    print("The value is Zero")
```

```
3
Positive number
```

4.5 Nested IF Statements

You can use one *if* or *else* statement inside another *if* or *else* statement(s).

```
if expression:
    statement(s)
    if expression:
        statement(s)
    else:
        statement(s)
else:
    statement(s)
```

[11]:
```
Number_of_passengers = 6
Number_of_baggage = 2
Security_check = True

if Number_of_passengers > 5:
    print("The number of passengers is more than 5")
    if Number_of_passengers and Number_of_baggage > 5:
        print("The number of passengers and baggage is more than 5")
    else:
        print("The number of passengers and baggage is less than 5")
```

```
The number of passengers is more than 5
The number of passengers and baggage is less than 5
```

5 Platform

Operating System: Arch Linux x86-64
IDEs or Text Editors Used: Visual Studio Code with Jupyter
Interpreter: python 3.10.8

6 Libraries Used with pip

No additional Libraries are used with pip. The only libraries used are the default libraries that come with python.

7 Input

1. Take 3 numbers from the User.

8 Output

1. The Largest of those numbers.

9 Code

[1]:
```
# splitting the input string into a list of strings and then converting each␣
 ↪string into an integer
(a, b, c) = [int(i) for i in input("Enter 3 Numbers: ").split()]
print("The Numbers that you have entered are: ", a, b, c)


# checking the largest number
if (a >= b and a >= c):
    print("The largest number is:", a)
```

```
elif b >= a and b >= c:
    print("The largest number is: ", b)

else:
    print("The largest number is: ", c)
```

```
The Numbers that you have entered are:  1 2 3
The largest number is:  3
```

10 Conclusion

Studied implementation of different forms of if...else statements using python programming.

11 FAQ

1. Describe following logical and relational operators with suitable examples.
   Logical Operators: and, or, not
   Relational Operators: ==, !=, >, <, >=, <=

   Answer:

   Logical Operators:

   and: This operator is used to check if both the conditions are true.

   or: This operator is used to check if any one of the conditions is true.

   not: This operator is used to check if the condition is false.

   Relational Operators:

   ==*:* This operator is used to check if the two operands are equal.

   !=: This operator is used to check if the two operands are not equal.

   >*:* This operator is used to check if the left operand is greater than the right operand.

   <*:* This operator is used to check if the left operand is less than the right operand.

   >=*:* This operator is used to check if the left operand is greater than or equal to the right operand.

   <=*:* This operator is used to check if the left operand is less than or equal to the right operand.

   Example for All the above mentioned Operators are:

```
    # Logical Operators
    a = 10
    b = 20
    c = 30
    if a > b and a > c:
      print("a is the largest number")
    elif b > a or b > c:
      print("b is greater than a or b is greater than c")

    # Relational Operators
```

```
    a = 10
    b = 20
    if a == b:
      print("a and b are equal")
    else:
      print("a and b are not equal")

    if a != b:
      print("a and b are not equal")
    else:
      print("a and b are equal")

    if a > b:
      print("a is greater than b")
    else:
      print("a is less than b")

    if a < b:
      print("a is less than b")
    else:
      print("a is greater than b")

    if a >= b:
      print("a is greater than or equal to b")
    else:
      print("a is less than b")

    if a <= b:
      print("a is less than or equal to b")
    else:
      print("a is greater than b")

```