004_Python_Nested_if_statement

March 9, 2023

1 Python Nested **if** statement

We can have a nested-[**if-else**] or nested-[**if-elif-else**] statement inside another **if-else** statement. This is called **nesting** in computer programming. The nested if statements is useful when we want to make a series of decisions.

Any number of these statements can be nested inside one another. Indentation is the only way to figure out the level of nesting. They can get confusing, so they must be avoided unless necessary.

We can use nested if statements for situations where we want to check for a **secondary condition** if the first condition executes as **True**.

1.0.1 Syntax:

Example 1:

```
if conditon_outer:
    if condition_inner:
        statement of nested if
    else:
        statement of nested if else:
    statement ot outer if
else:
    Outer else
statement outside if block
```

Example 2:

```
if expression1:
    statement(s)
    if expression2:
        statement(s)
    elif expression3:
        statement(s)
    elif expression4:
        statement(s)
    else:
        statement(s)
else:
    statement(s)
```

[1]:

```
# Example 1:

a=10
if a>=20:  # Condition FALSE
    print ("Condition is True")
else:  # Code will go to ELSE body
    if a>=15:  # Condition FALSE
        print ("Checking second value")
    else:  # Code will go to ELSE body
        print ("All Conditions are false")
```

```
All Conditions are false
```

[2]:

```
# Example 2:

x = 10
y = 12
if x > y:
    print( "x>y")
elif x < y:
    print( "x<y")
    if x==10:
        print ("x=10")
    else:
        print ("invalid")
else:
    print ("x=y")
```

```
x<y
x=10
```

[3]:

```
# Example 3:

num1 = 0
if (num1 != 0):  # For zero condition is FALSE
    if(num1 > 0):
        print("num1 is a positive number")
    else:
        print("num1 is a negative number")
else:  # For zero condition is TRUE
    print("num1 is neither positive nor negative")
```

```
num1 is neither positive nor negative
```

[4]:

```
# Example 4:

'''In this program, we input a number check if the number is
positive or negative or zero and display an appropriate message.
```

```
This time we use nested if statement'''

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

```
Enter a number: 9
Positive number
```

[5]:

```
# Example 5:

def number_arithmetic(num1, num2):
    if num1 >= num2:
        if num1 == num2:
            print(f'{num1} and {num2} are equal')
        else:
            print(f'{num1} is greater than {num2}')
    else:
        print(f'{num1} is smaller than {num2}')

number_arithmetic(96, 66)
# Output 96 is greater than 66
number_arithmetic(96, 96)
# Output 56 and 56 are equal
```

```
96 is greater than 66
96 and 96 are equal
```

[ ]:

```

```