

Introduction

Decision-making is as important in any programming language as it is in life. Decision-making in a programming language is automated using conditional statements, in which Python evaluates the code to see if it meets the specified conditions.

The conditions are evaluated and processed as true or false. If this is found to be true, the program is run as needed. If the condition is found to be false, the statement following the If condition is executed.

Python has six conditional statements that are used in decision-making:-

1. If the statement
2. If else statement
3. Nested if statement
4. If...Elif ladder
5. Short Hand if statement
6. Short Hand if-else statement

If Statement

The If statement is the most fundamental decision-making statement, in which the code is executed based on whether it meets the specified condition. It has a code body that only executes if the condition in the if statement is true. The statement can be a single line or a block of code.

The if statement in Python has the subsequent syntax:

if expression

Statement

#If the condition is true, the statement will be executed.

Examples for better understanding:

Example – 1

num = 5

```
if num > 0:
    print(num, "is a positive number.")
print("This statement is true.")
#When we run the program, the output will be:
5 is a positive number.
This statement is true.
```

Example – 2

```
a = 25
b = 170
if b > a:
    print("b is greater than a")
output : b is greater than a
```

If Else Statement

This statement is used when both the true and false parts of a given condition are specified to be executed. When the condition is true, the statement inside the if block is executed; if the condition is false, the statement outside the if block is executed.

The if...Else statement in Python has the following syntax:

```
if condition :
    #Will executes this block if the condition is true
else :
    #Will executes this block if the condition is false
```

Example for better understanding:

```
num = 5
if num >= 0:
    print("Positive or Zero")
else:
    print("Negative number")
output : Positive or Zero
```

If...Elif..else Statement

In this case, the If condition is evaluated first. If it is false, the Elif statement will be executed; if it also comes false, the Else statement will be executed.

The If...Elif..else statement in Python has the subsequent syntax:

if condition :

 Body of if

elif condition :

 Body of elif

else:

 Body of else

Example for better understanding:

We will check if the number is positive, negative, or zero.

```
num = 7
```

```
if num > 0:
```

```
    print("Positive number")
```

```
elif num == 0:
```

```
    print("Zero")
```

```
else:
```

```
    print("Negative number")
```

output: Positive number

Nested IF Statement

A Nested IF statement is one in which an If statement is nestled inside another If statement.

This is used when a variable must be processed more than once. If, If-else, and If...elif...else statements can be used in the program. In Nested If statements, the indentation (whitespace at the beginning) to determine the scope of each statement should take precedence.

The Nested if statement in Python has the following syntax:

```
if (condition1):
```

```
#Executes if condition 1 is true
```

```
if (condition 2):
```

```
    #Executes if condition 2 is true
```

```
    #Condition 2 ends here
```

```
#Condition 1 ends here
```

Examples for better understanding:

Example-1

```
num = 8
```

```
if num >= 0:
```

```
    if num == 0:
```

```
        print("zero")
```

```
    else:
```

```
        print("Positive number")
```

```
else:  
    print("Negative number")
```

output: Positive number

Example-2

```
price=100  
quantity=10  
amount = price*quantity  
if amount > 200:  
    if amount >1000:  
        print("The amount is greater than 1000")  
    else:  
        if amount > 800:  
            print("The amount is between 800 and 1000")  
        elif amount > 600:  
            print("The amount is between 600 and 1000")  
        else:  
            print("The amount is between 400 and 1000")  
elif amount == 200:  
    print("Amount is 200")  
else:  
    print("Amount is less than 200")
```

The output : "The amount is between 400 and 1000."

Short Hand if statement

Short Hand if statement is used when only one statement needs to be executed inside the if block. This statement can be mentioned in the same line which holds the If statement.

The Short Hand if statement in Python has the following syntax:

```
if condition: statement
```

Example for better understanding:

```
i=15  
# One line if statement  
if i>11 : print ("i is greater than 11")  
The output of the program : "i is greater than 11."  
Short Hand if-else statement
```

It is used to mention If-else statements in one line in which there is only one statement to execute in both if and else blocks. In simple words, If you have only one statement to execute, one for if, and one for else, you can put it all on the same line.

Examples for better understanding:

#single line if-else statement

```
a = 3
```

```
b = 5
```

```
print("A") if a > b else print("B")
```

output: B

#single line if-else statement, with 3 conditions

```
a = 3
```

```
b = 5
```

```
print("A is greater") if a > b else print("=") if a == b else print("B is greater")
```

output: B is greater

To summarise,

- The If the condition is used to print the result when only one of the conditions listed is true or false.
- When one of the conditions is false, the If-else condition is used to print the statement.
- When there is a third possible outcome, the Elif statement is used. In a program, any number of Elif conditions can be used.
- By declaring all of the conditions in a single statement, we can reduce the number of codes that must be executed.
- Nested if statements can be used to nest one If condition inside another.

Conclusion

If you're reading this, you're most likely learning Python or trying to become a Python developer. Learning Python or another programming language begins with understanding the fundamental concepts that form its foundation.