

Task 4 - Decision Making in Python

There comes situations in real life when we need to make some decisions and based on these decisions, we decide what should we do next. Similar situations arises in programming also where we need to make some decisions and based on these decision we will execute the next block of code.

Decision making statements in programming languages decides the direction of flow of program execution. Decision making statements available in python are:

- 1) if statement
- 2) if..else statements
- 3) nested if statements
- 4) if-elif ladder

1. if statement

if statement is the most simple decision making statement. It is used to decide whether a certain statement or block of statements will be executed or not i.e if a certain condition is true then a block of statement is executed otherwise not.

1.1 Syntax:

```
if condition:
    # Statements to execute if
    # condition is true
```

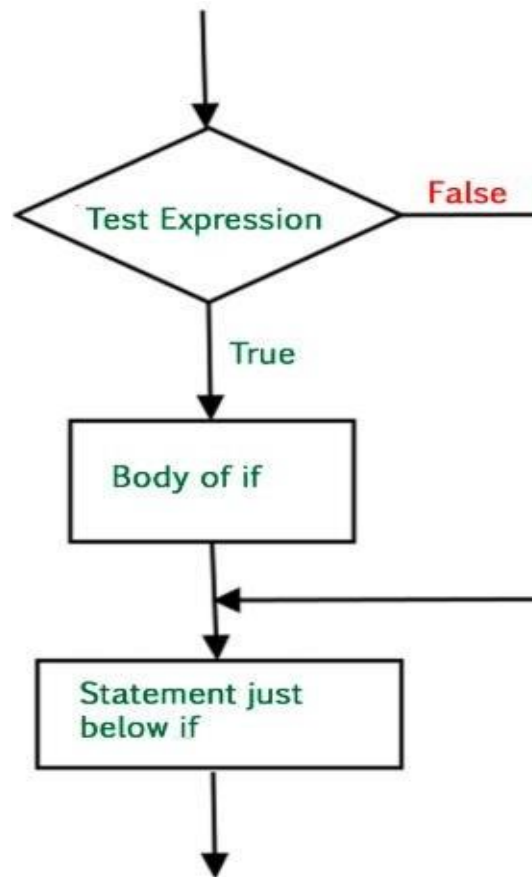
Here, condition after evaluation will be either true or false. if statement accepts boolean values

– if the value is true then it will execute the block of statements below it otherwise not. We can use *condition* with bracket ‘(‘)’ also. As we know, python uses indentation to identify a block. So the block under an if statement will be identified as shown in the below example:

```
if condition:
    statement1
    statement2

# Here if the condition is true, if block
# will consider only statement1 to be inside
# its block.
```

1.2 Flowchart:



1.3 Example:

python program to illustrate If statement

```
i = 10
if (i > 15):
    print ("10 is less than 15")
print ("I am Not in if")
```

1.4 Output:

```
I am Not in if
```

As the condition present in the if statement is false. So, the block below the if statement is not executed.

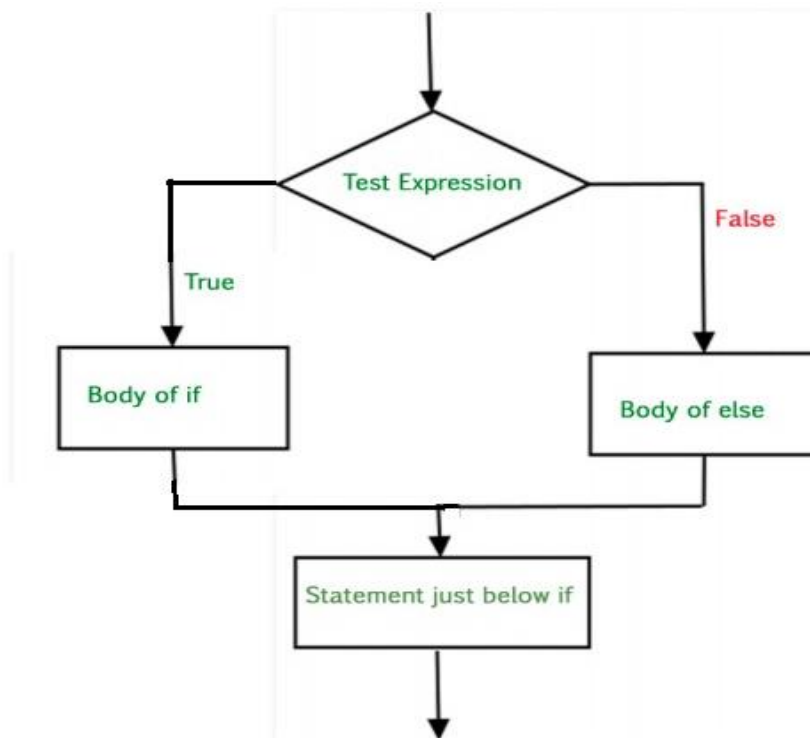
2. if- else

The if statement alone tells us that if a condition is true it will execute a block of statements and if the condition is false it won't. But what if we want to do something else if the condition is false. Here comes the *else* statement. We can use the *else* statement with *if* statement to execute a block of code when the condition is false.

2.1 Syntax:

```
if (condition):  
    # Executes this block if  
    # condition is true  
else:  
    # Executes this block if  
    # condition is false
```

2.2 Flow Chart:



2.3 Example:

python program to illustrate If else statement

```
i = 20;  
if (i < 15):  
    print ("i is smaller than 15")  
    print ("i'm in if Block")  
else:  
    print ("i is greater than 15")  
    print ("i'm in else Block")  
print ("i'm not in if and not in else Block")
```

2.4 Output:

```
i is greater than 15  
i'm in else Block  
i'm not in if and not in else Block
```

The block of code following the else statement is executed as the condition present in the if statement is false after call the statement which is not in block(without spaces)

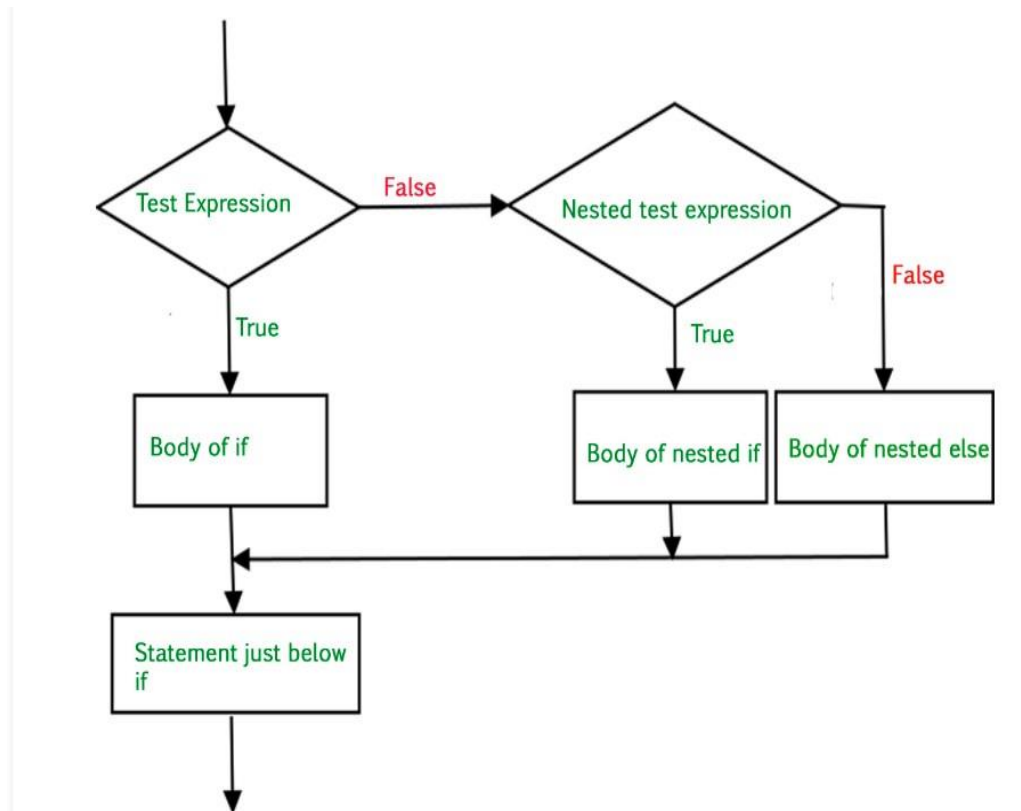
3. nested-if

A nested if is an if statement that is the target of another if statement. Nested if statements means an if statement inside another if statement. Yes, Python allows us to nest if statements within if statements. i.e, we can place an if statement inside another if statement.

3.1 Syntax:

```
if (condition1):  
    # Executes when condition1 is true  
    if (condition2):  
        # Executes when condition2 is true  
        # if Block is end here  
    # if Block is end here
```

3.2 Flow chart:



3.3 Example:

```
# python program to illustrate nested If statement

i = 10
if (i == 10):
    # First if statement
    if (i < 15):
        print ("i is smaller than 15")
    # Nested - if statement
    # Will only be executed if statement above
    # it is true
    if (i < 12):
        print ("i is smaller than 12 too")
    else:
        print ("i is greater than 15")
```

3.4 Output:

```
i is smaller than 15
i is smaller than 12 too
```

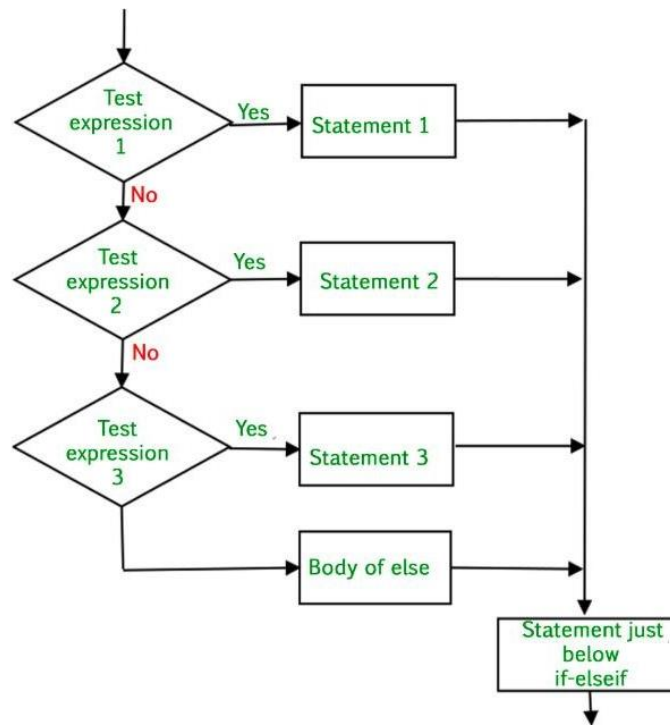
4. if-elif-else ladder

Here, a user can decide among multiple options. The if statements are executed from the top down. As soon as one of the conditions controlling the if is true, the statement associated with that if is executed, and the rest of the ladder is bypassed. If none of the conditions is true, then the final else statement will be executed.

4.1 Syntax:

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

4.2 Flow Chart:



4.3 Example:

Python program to illustrate if-elif-else ladder

```
i = 20
if (i == 10):
    print ("i is 10")
elif (i == 15):
    print ("i is 15")
elif (i == 20):
    print ("i is 20")
else:
    print ("i is not present")
```

4.4 Output:

```
i is 20
```

Video Link: <https://youtu.be/PqFKRqpHrjw>

References:

- 1) <https://www.geeksforgeeks.org/decision-making-python-else-nested-elif/>
- 2) <https://www.tutorialsteacher.com/python/python-if-elif>
- 3) <https://www.w3schools.in/python-tutorial/decision-making/>