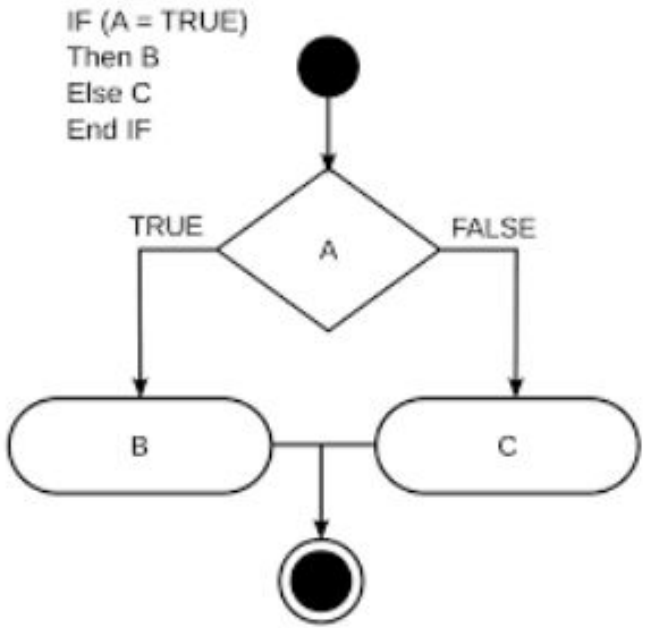
Python Programming Basics

if..else Conditional Statements

Contents

Introduction

- The most simplest form is `if` statement. `else` and `else if` statements can also be used depending on the number of conditions to be checked.

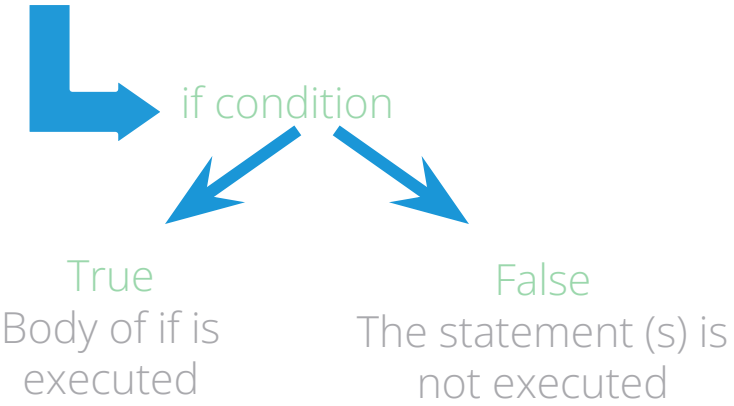


if Statement

Syntax:

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

The program evaluates



if Statement

If the condition is TRUE, the statement gets executed. But if it's FALSE, nothing happens.

```
# Check if x is a positive number or not
x = 5
if x > 0:
    print("Positive number")
```

```
Positive number
```

- Comparison operator '**>**' is used in condition.
- Here, **x >0** returns TRUE; hence the print statement is executed

```
# Print the value of the object if it is of a character type
x = "Hello there"
if isinstance(x,str):
    print(x)
```

```
Hello there
```

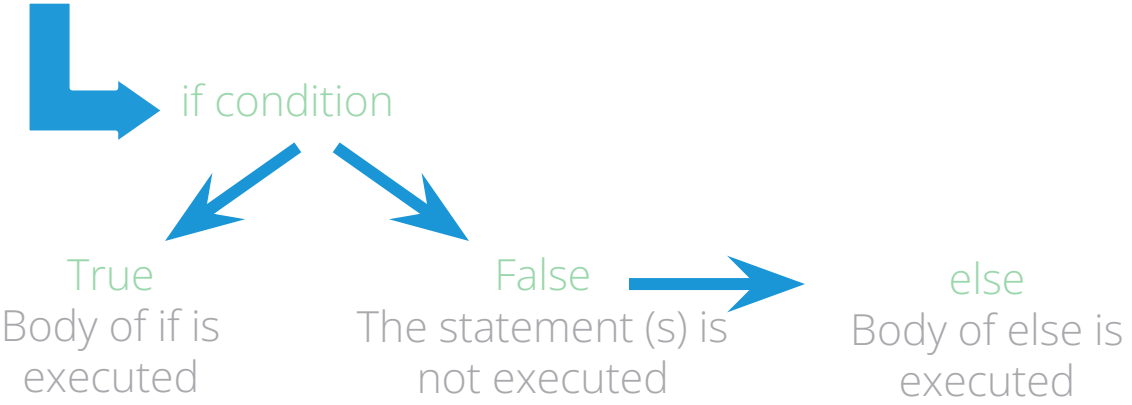
- **is.instance()** returns TRUE or FALSE depending on whether the argument is of string (str) type or not.
- Here **x** is a string, hence the condition returns **TRUE** and "Hello there" is printed

if..else Statement

Syntax:

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

The program evaluates



* The **else** statement is optional and there could be at most only one **else** statement following **if.**

if..else Statement

`if..else` Statements are used when you want to execute one block of code when a condition is true, and another block of code when it is false.

```
# Check if x is a positive number or not and print the status

x = -5
if x > 0:
    print('Positive number')
else:
    print('Negative number')
```

```
Negative number
```

This condition evaluates to false; hence it will skip the body of **if** and the body of **else** will be executed.

There is an easier way to use **`if..else`** statement specifically for vectors. You can use **`if..else()`** function instead.

if..else Statement

```
# Take input as a number from user
# Print the sum of natural numbers up to that number.

num = int(input("Enter a number: "))
if num < 0:
    print("Enter a positive number")
else:
    sum=(num * (num + 1)) / 2
    print("The sum is", sum)
```

```
Enter a number: 5
The sum is 15.0
```

- **input()** interactively takes the user input values

Here the input is 5, hence the sum of 5 natural number is 15

if..else inside print Function

Using **`if..else`** statement for vector, we can use for loop to execute **`if..else`** condition in a single statement

- If the condition is TRUE it returns x else y
- **`if..else`** cannot take a vector as an input in python but results into a vector. Hence we use for loop along with `if..else()`

```
#Create a vector & put a condition to check & print even or odd depending
#on result.
```

```
a = [6,1,5,14]
print([("even" if x%2 == 0 else "odd") for x in a])
```

```
['even', 'odd', 'odd', 'even']
```

Nested if..else Statement

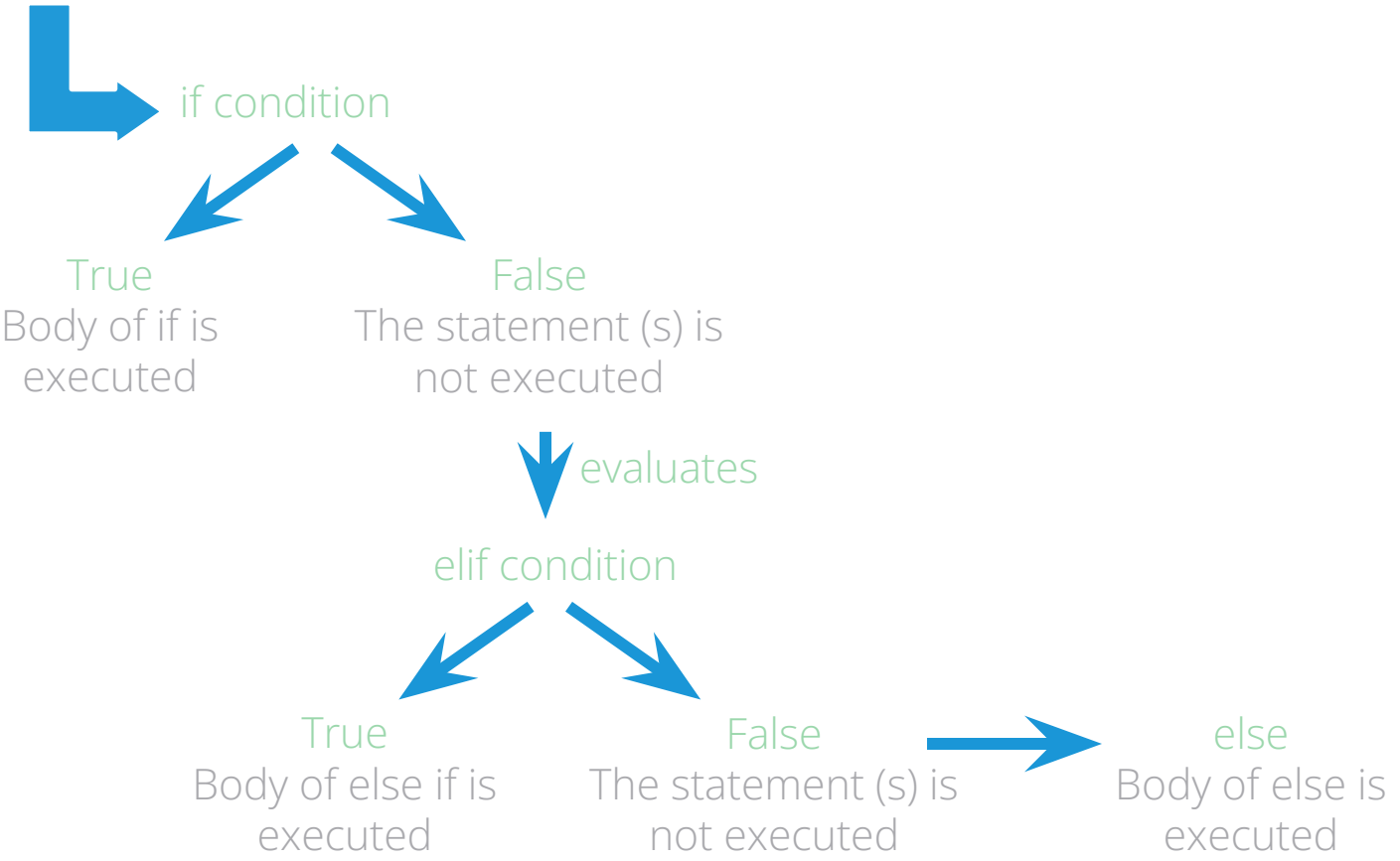
elif is short for "**else if**". It allows us to check multiple expressions. Similar to **else, elif** statement is optional.

Syntax:

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

Nested if..else Statement

The program evaluates



Only one statement gets executed depending on the conditions.

Nested if..else Statement

`if..elif..else` statements are used when there are several conditions. Any number of conditions can be added to `elif` statement after an if statement and before an else statement.

```
x=-11
if x>0:
    print('x is a positive number')
elif x==0:
    print('x is zero')
else:
    print('x is a negative number')
```

```
x is a negative number
```

The **if** condition and **elif** condition evaluates to False so it executes the statements in the block of **else**.

Quick Recap

In this session, we learnt all the **`if..else`** conditional statements with the help of examples. Here is a quick recap:

if statement	• If the condition is TRUE, the statement gets executed. But if it's FALSE, nothing happens.
if..else statement	• Used when you want to execute a statement when the condition returns FALSE. **`if..else()`** is the vector equivalent of the if ..else statement
Nested if..else statement	• Used when there are several conditions • Multiple conditions can be given using **`elif`** statement