

Conditional Statements in Python

By Bhadra Reddy

Overview of decision

When we consider our real-time scenario every day, we make some decisions and based on the decisions made we will take further actions. Hence all our daily life activities depend on the decisions we make.

A similar situation arises in the programming language as well where we have to make some decisions and based on that the program will execute.

Conditional Statements in Python

- In programming languages, most of the time in large projects we have to control the flow of execution of our program and we want to execute some set of statements only if the given condition is satisfied, and a different set of statements when it's not satisfied.
- Conditional statements are also known as decision-making statements. We need to use these conditional statements to execute the specific block of code if the given condition is true or false.

How many conditional statements available in python?

Python provides four conditional statements

- 1) If statement
- 2) If-else statements
- 3) elif statements
- 4) Nested if and if-else statements
- 5) elif ladder

1) If statement

- Python if statement is one of the most commonly used conditional statements in programming languages. It decides whether certain statements need to be executed or not.
- If the condition is true, then the set of code present inside the "if" block will be executed otherwise not.
- The if condition evaluates a Boolean expression and executes the block of code only when the Boolean expression becomes TRUE.

Syntax

If (EXPRESSION == TRUE):

Block of code

else:

Block of code

Here, the condition will be evaluated to a Boolean expression (true or false). If the condition is true, then the statement or program present inside the "if" block will be executed and if the condition is false, then the statements or program present inside the "else" block will be executed.

Example:

```
age = 20
```

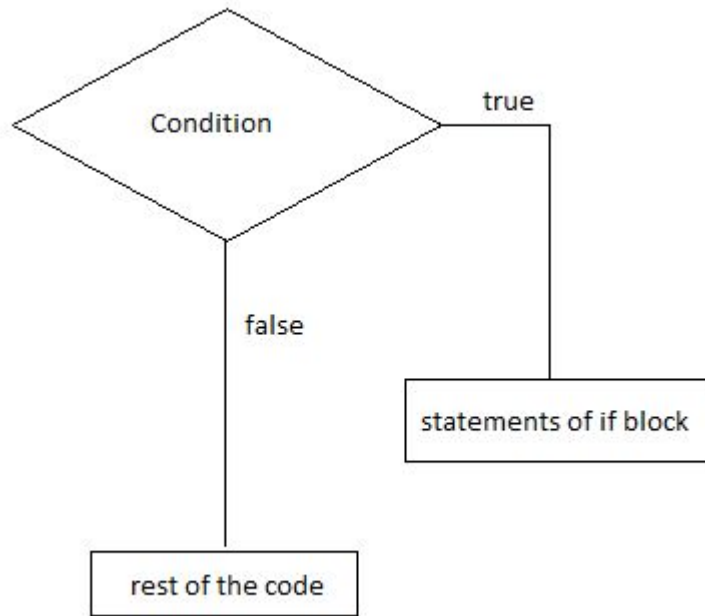
```
If age > 30:
```

```
    print("Age is greater than 30")
```

```
else :
```

```
    print("Age is less than 30")
```

Flow diagram



Example with flow diagram

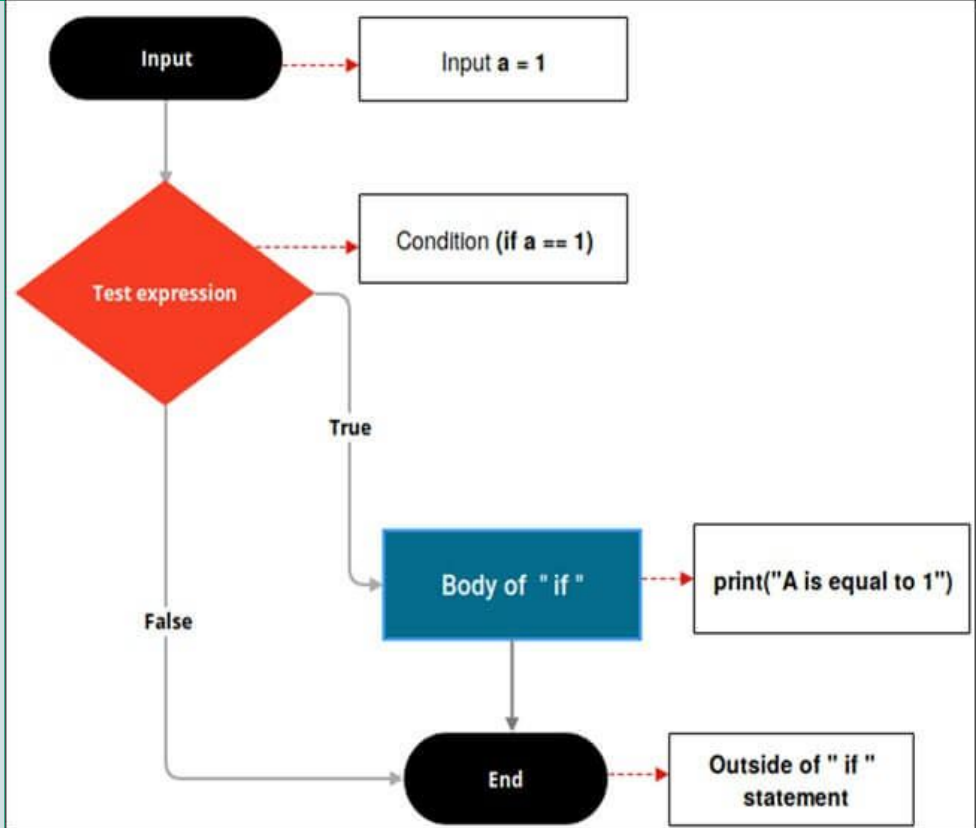
```
num = 1
```

```
if (num == 1):
```

```
    print("Num is 1.")
```

```
print("This statement will always be  
executed")
```

Output: Num is smaller than 1.



Examples

```
"""
Example #1
"""
a = 7
b = 0
if (a > b):
    print("a is greater than b")
```

```
"""
Example #2
"""
a = 7
b = 0
if (a):
    print("true")
```

```
"""
Example #3
"""
if ('Python' in ["Python", "C#", "java"]):
    print("true")

"""
Example #4:
-----
You have written an exam for a total score of 100
and if your score is above or equal to 60 then
you will be considered as PASS in the exam.
"""
passing_Score = 60
my_Score = 67
if(my_Score >= passing_Score):
    print("Congratulations! You have passed your exam")
```

2) If-else statement in python

2) if - else

- The statement itself says if a given condition is true then execute the statements present inside the “if block” and if the condition is false then execute the “else” block.
- The “else” block will execute only when the condition becomes false. It is the block where you will perform some actions when the condition is not true.
- if-else statement evaluates the Boolean expression. If the condition is TRUE then, the code present in the “ if “ block will be executed otherwise the code of the “else“ block will be executed

If - else syntax

Syntax:

If (EXPRESSION == TRUE):

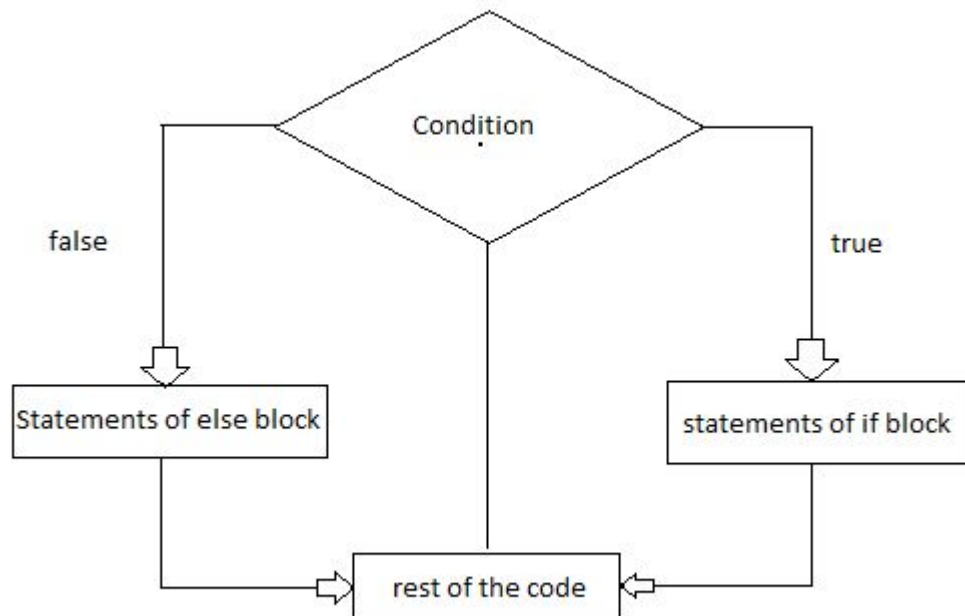
 Statement (Body of the block)

else:

 Statement (Body of the block)

- The statement itself says if a given condition is true then execute the statements present inside the “if block” and if the condition is false then execute the “else” block.
- The “else” block will execute only when the condition becomes false. It is the block where you will perform some actions when the condition is not true.
- if-else statement evaluates the Boolean expression. If the condition is TRUE then, the code present in the “ if “ block will be executed otherwise the code of the “else“ block will be executed

Flow Diagram (if - else)



Example with flow diagram

a = 1

```
if(a == 1):
```

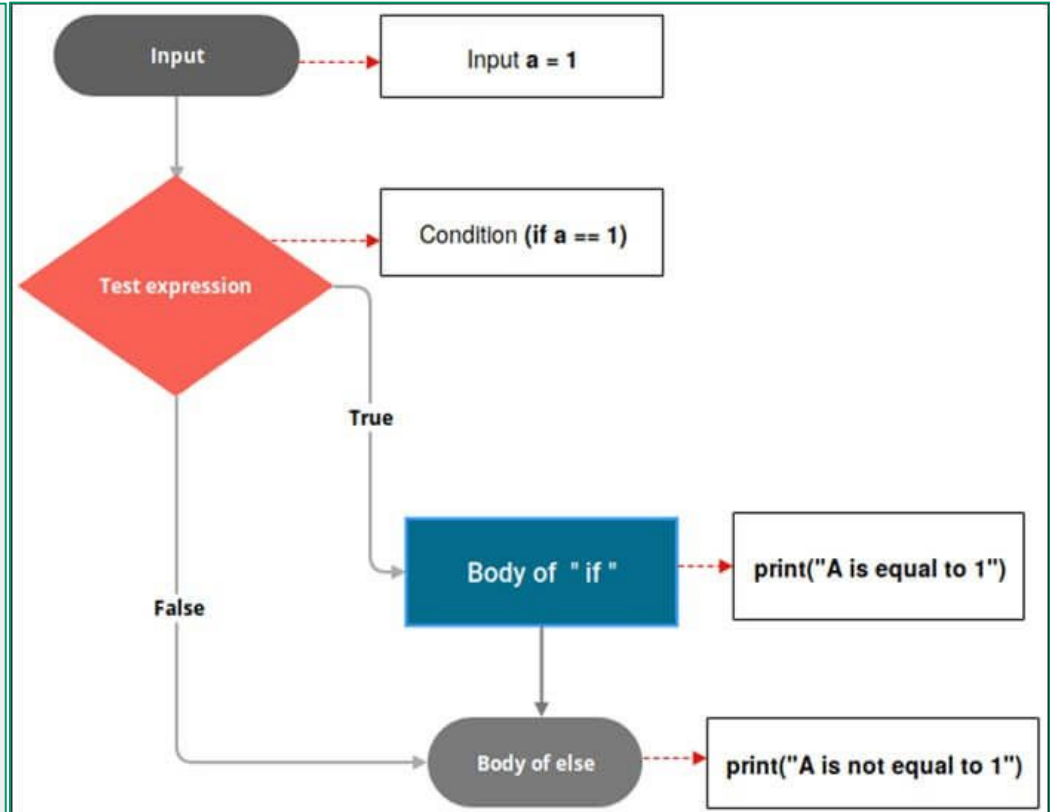
```
    print("A is equal to 1")
```

```
else:
```

```
    print("A is not equal to 1")
```

```
print ("This statement will always be  
executed" )
```

Output: A is equal to 1



```
"""
```

Example #1

```
"""
```

```
a = 7
```

```
b = 0
```

```
if (a > b):
```

```
    print("a is greater than b")
```

```
else:
```

```
    print("b is greater than a")
```

```
"""
```

Example #2

```
"""
```

```
passing_Score = 60
```

```
my_Score = 67
```

```
if(my_Score >= passing_Score):
```

```
    print("Congratulations! You passed the exam")
```

```
else:
```

```
    print("Sorry! You failed the exam, better luck next time")
```

3) elif statement

3) elif statement

- In Python, we have one more conditional statement called “elif” statements. “elif” statement is used to check multiple conditions only if the given condition is false. It’s similar to an “if-else” statement and the only difference is that in “else” we will not check the condition but in “elif” we will check the condition.
- “elif” statements are similar to “if-else” statements but “elif” statements evaluate multiple conditions.

Syntax (elif)

if (condition):

#Set of statement to execute if condition is true

elif (condition):

#Set of statements to be executed when if condition is false and elif condition is true

else:

#Set of statement to be executed when both if and elif conditions are false

Examples (elif)

Example #1

```
num = 10
```

```
if (num == 0):
```

```
    print("Number is Zero")
```

```
elif (num > 5):
```

```
    print("Number is greater than 5")
```

```
else:
```

```
    print("Number is smaller than 5")
```

Example #2

```
num = -7
```

```
if (num > 0):
```

```
    print("Number is positive")
```

```
elif (num < 0):
```

```
    print("Number is negative")
```

```
else:
```

```
    print("Number is Zero")
```

4) Nested if-else statements

Nested if else statements

- Nested “if-else” statements mean that an “if” statement or “if-else” statement is present inside another if or if-else block.
- An “if” statement is present inside another “if” statement which is present inside another “if” statements and so on.

Nested if-else syntax

Syntax:

if(condition):

#Statements to execute if condition is true

if(condition):

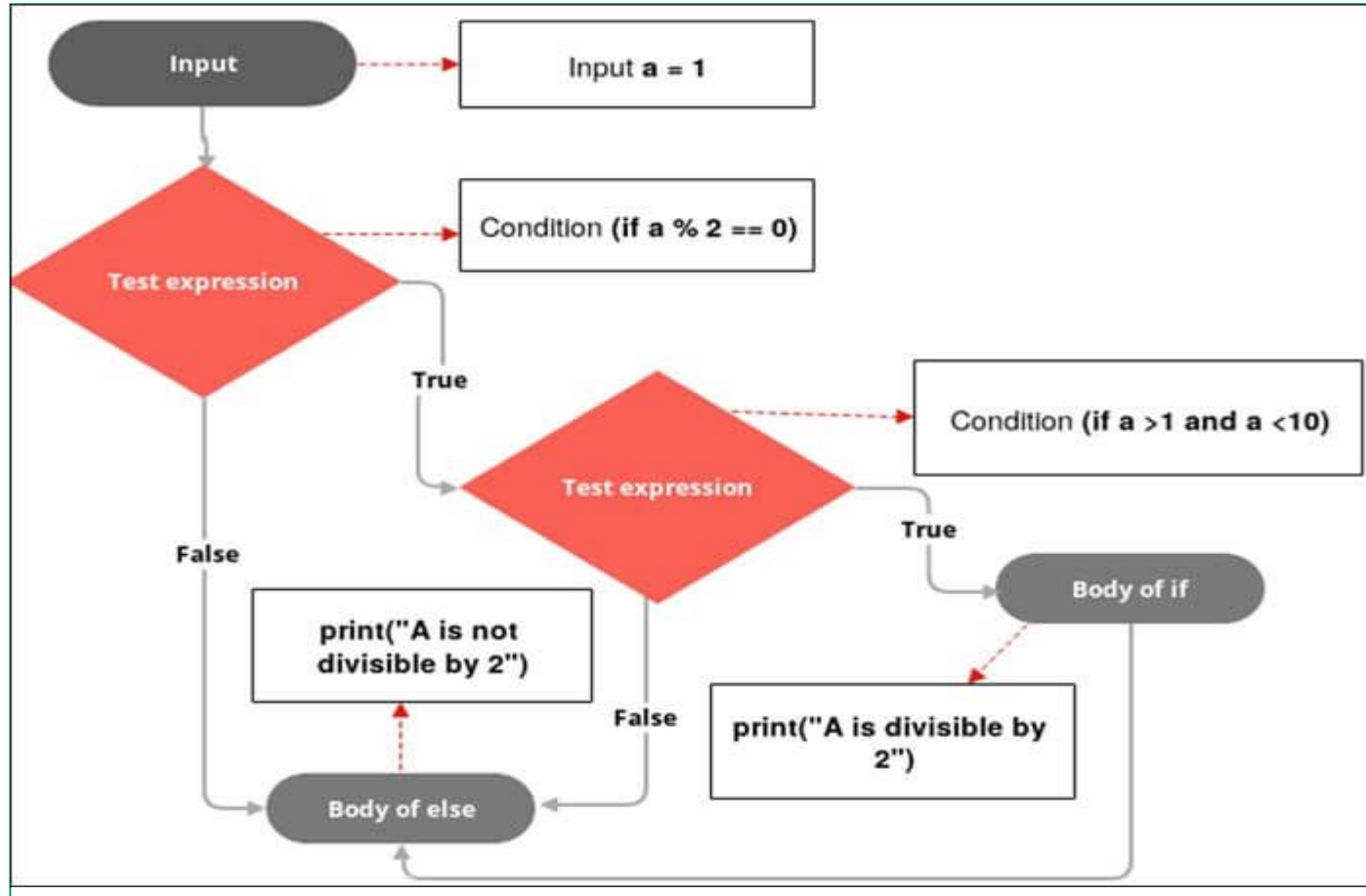
#Statements to execute if condition is true

#end of nested if

#end of if

The above syntax clearly says that the if block will contain another if block in it and so on. If block can contain 'n' number of if block inside it.

Example with flow diagram



Example on nested if else

Example #1

num = 7

if (num != 0):

if (num > 0):

print("Number is greater than Zero")

Example #2:

i = 10

if (i == 10):

if (i < 20):

print (i, "is smaller than 20")

if (i < 21):

print (i, "is smaller than 21")

Example #3

num = -7

if (num != 0):

if (num > 0):

print("Number is positive")

else:

print("Number is negative")

else:

print("Number is Zero")

5) elif ladder statement

elif ladder syntax:

elif ladder syntax

this elif ladder is used to test multiple expressions

if (condition):

#Set of statement to execute if condition is true

elif (condition):

#Set of statements to be executed when if condition is false and elif condition is true

elif (condition):

#Set of statements to be executed when both if and first elif condition is false and second elif condition is true

elif (condition):

#Set of statements to be executed when if, first elif and second elif conditions are false and third elif statement is true

else:

#Set of statement to be executed when all if and elif conditions are false

Example on elif ladder

#Example

```
my_marks = 90
```

```
if (my_marks < 35):
```

```
    print("Sorry!, You failed the exam")
```

```
elif(my_marks > 60 and my_marks > 100):
```

```
    print("Passed in First class")
```

```
else:
```

```
    print("Passed in First class with distinction")
```