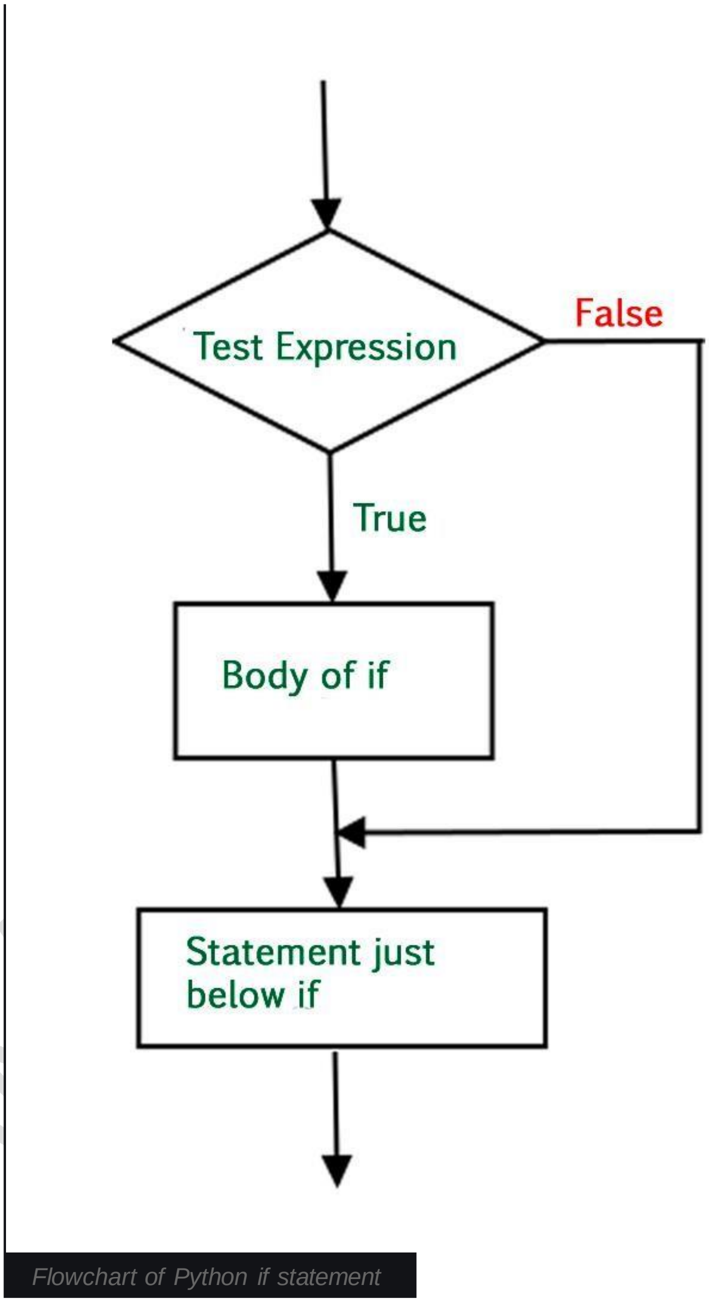
Python conditional Statement Explanation

Python If Statement

The 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.

Flowchart of If Statement

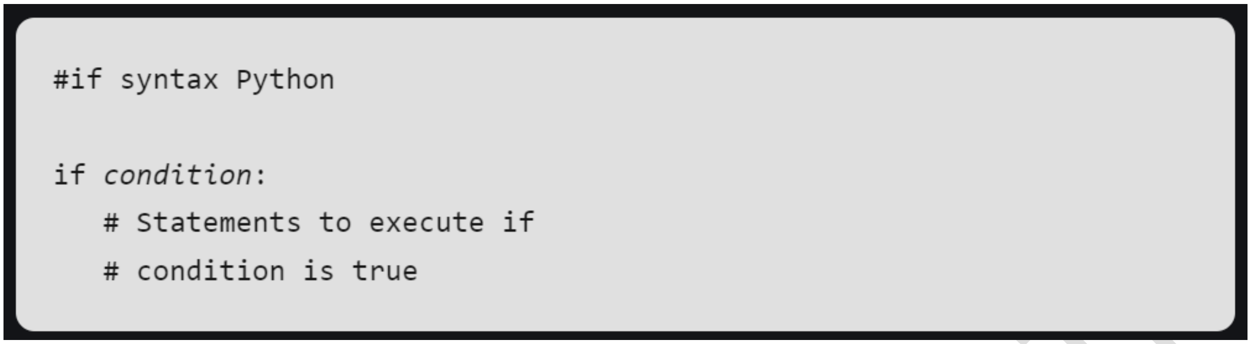
Let’s look at the flow of code in the Python If statements.



Flowchart of Python if statement

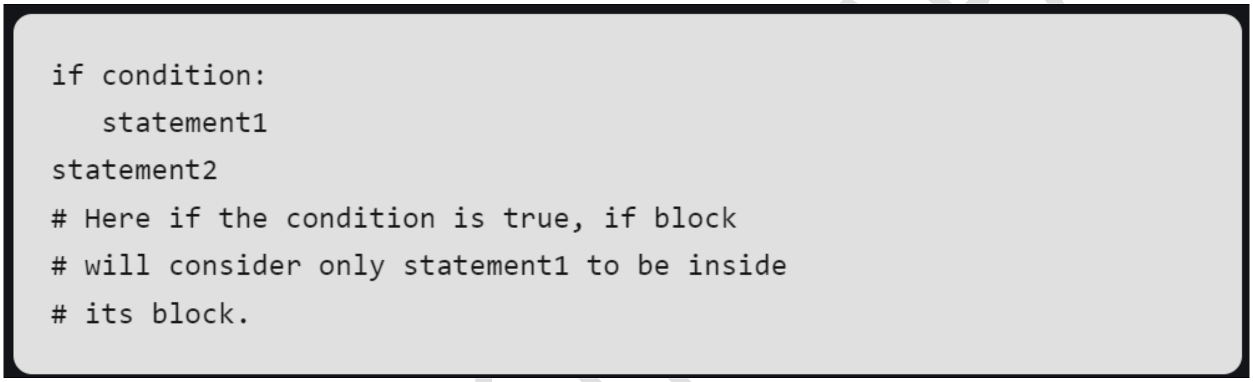
Syntax of If Statement in Python

Here, the condition after evaluation will be either true or false. if the statement accepts boolean values – if the value is true then it will execute the block of statements below it otherwise not.



```
#if syntax Python

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

As we know, Python uses indentation to identify a block. So the block under the Python if statements 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.
```

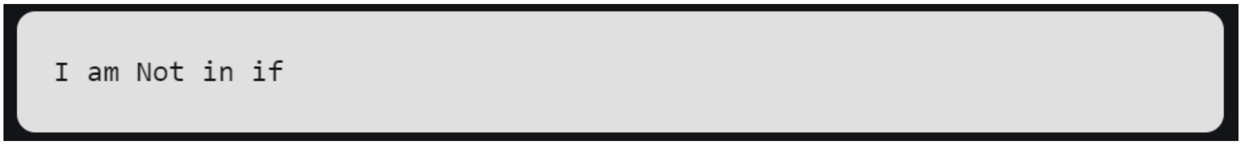
Example of Python if Statement

As the condition present in the if statements in Python is false. So, the block below the if statement is executed.

Python

```python
# python program to illustrate If statement
i = 10

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

Output:



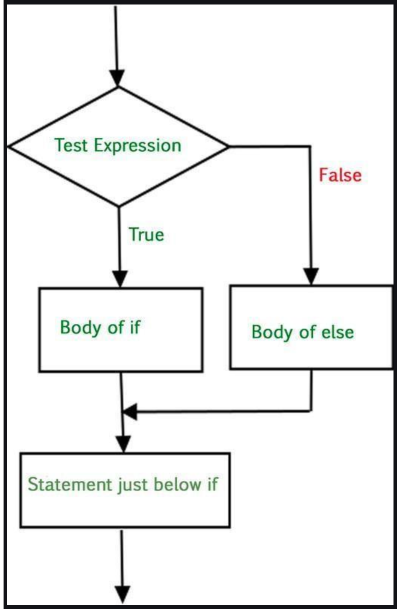
```
I am Not in if
```

Python If Else Statement

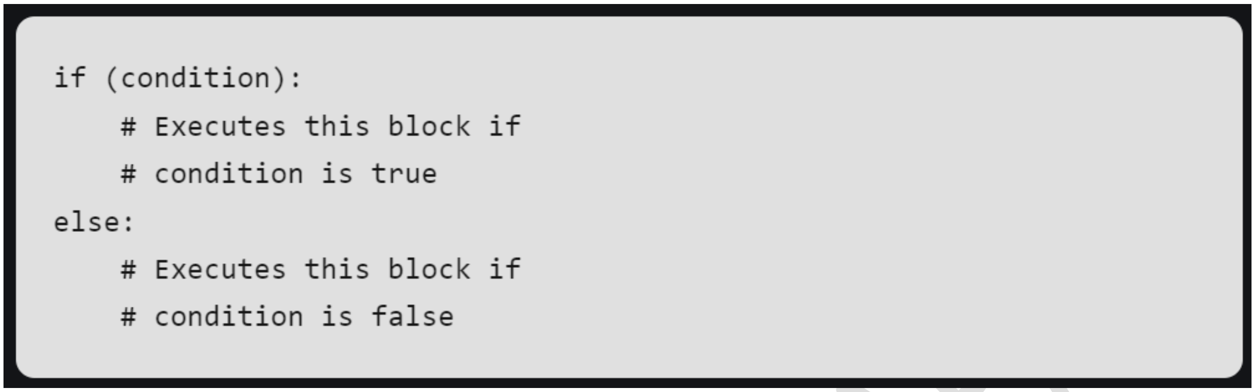
The if statement alone tells us that if a condition is true it will execute a block of statements and if the condition is false it won't. But if we want to do something else if the condition is false, we can use the else statement with the if statement Python to execute a block of code when the Python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

Flowchart of If Else Statement

Let's look at the flow of code in an if else Python statement.



Syntax of If Else in Python

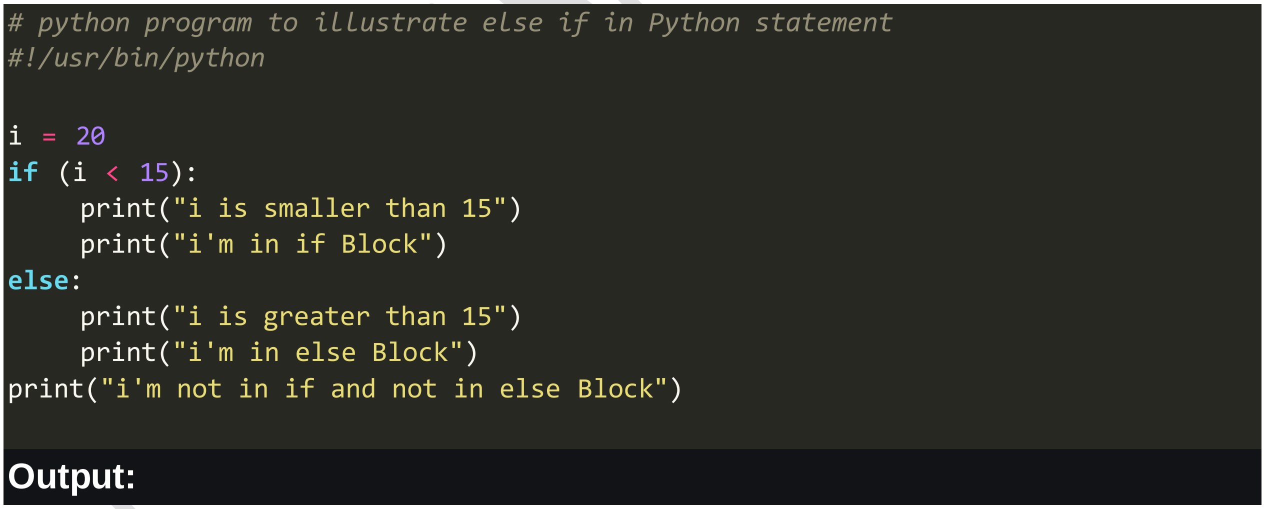


```
if (condition):
    # Executes this block if
    # condition is true
else:
    # Executes this block if
    # condition is false
```

Example of Python If Else Statement

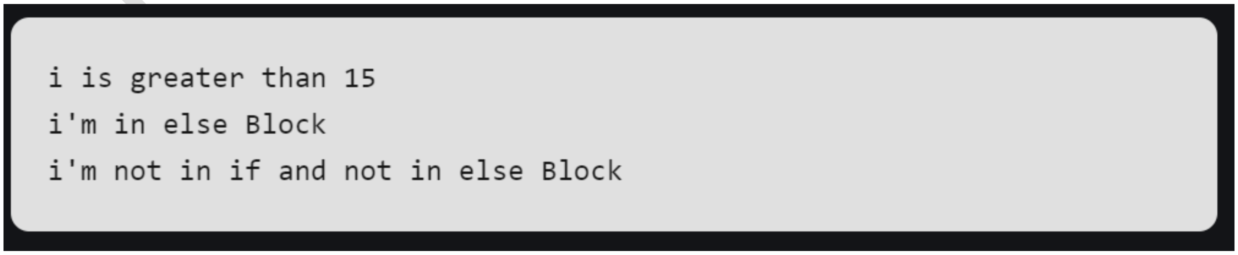
The block of code following the else if in Python, the statement is executed as the condition present in the if statement is false after calling the statement which is not in the block(without spaces).

Python



```
# python program to illustrate else if in Python statement
#!/usr/bin/python

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")
```

Output:



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

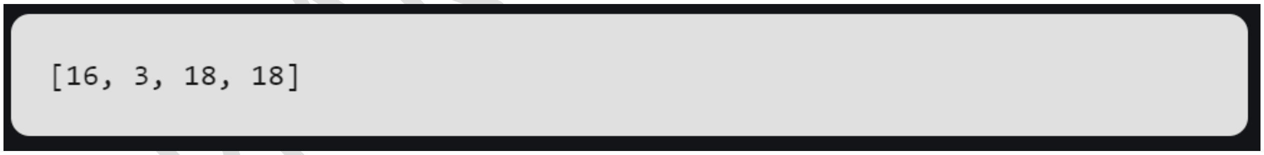
If Else in Python using List Comprehension

In this example, we are using an Python else if statement in a list comprehension with the condition that if the element of the list is odd then its digit sum will be stored else not.

Python

```
# Explicit function
def digitSum(n):
     dsum = 0
     for ele in str(n):
          dsum += int(ele)
     return dsum



# Initializing list
List = [367, 111, 562, 945, 6726, 873]

# Using the function on odd elements of the list
newList = [digitSum(i) for i in List if i & 1]

# Displaying new list
print(newList)
```

Output :



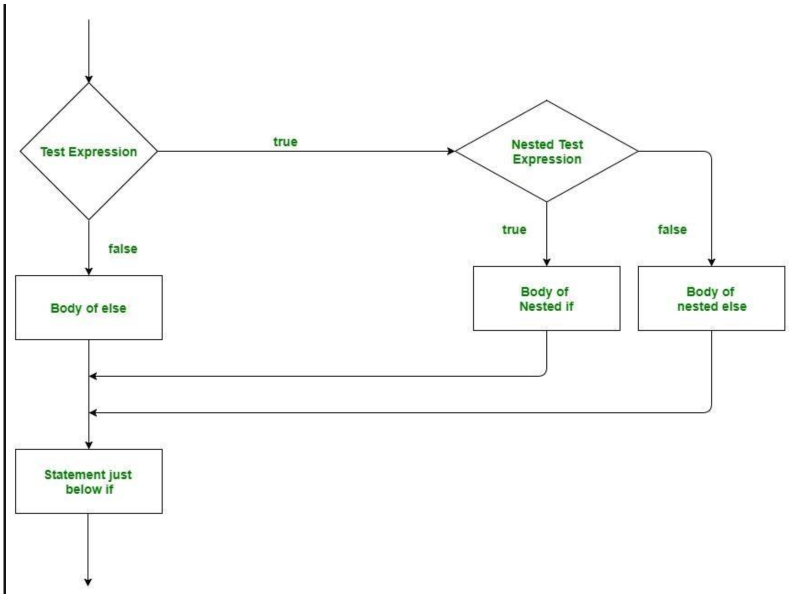
```
[16, 3, 18, 18]
```

Python Nested If Statement

A nested if is an if statement that is the target of another if statement. Nested if statements mean an if statement inside another if statement.

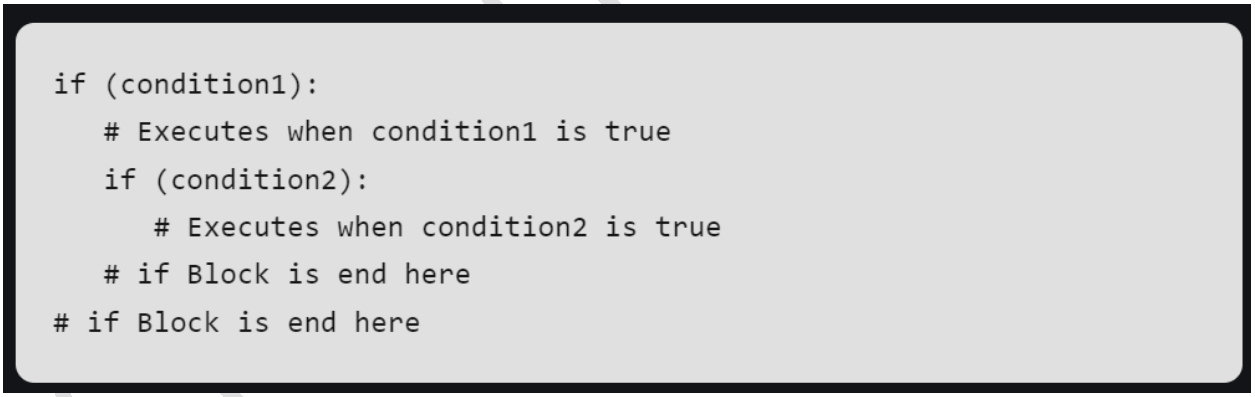
Yes, Python allows us to nest if statements within if statements. i.e., we can place an if statement inside another if statement.

Flowchart of Python Nested if Statement



Flowchart of Python Nested if statement

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
```

Example of Python Nested If Statement

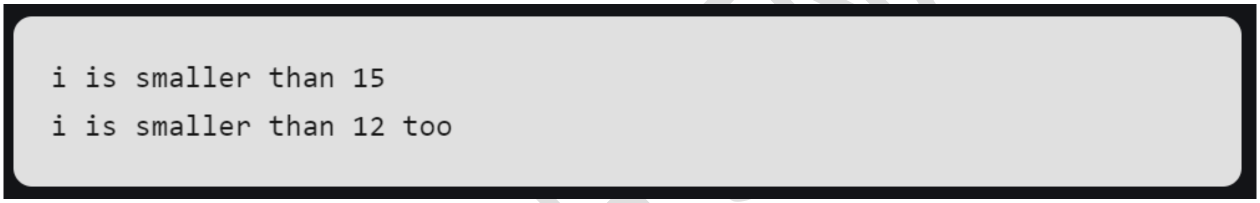
In this example, we are showing nested if conditions in the code, All the If condition in Python will be executed one by one.

Python

```
# 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")
```

Output:



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

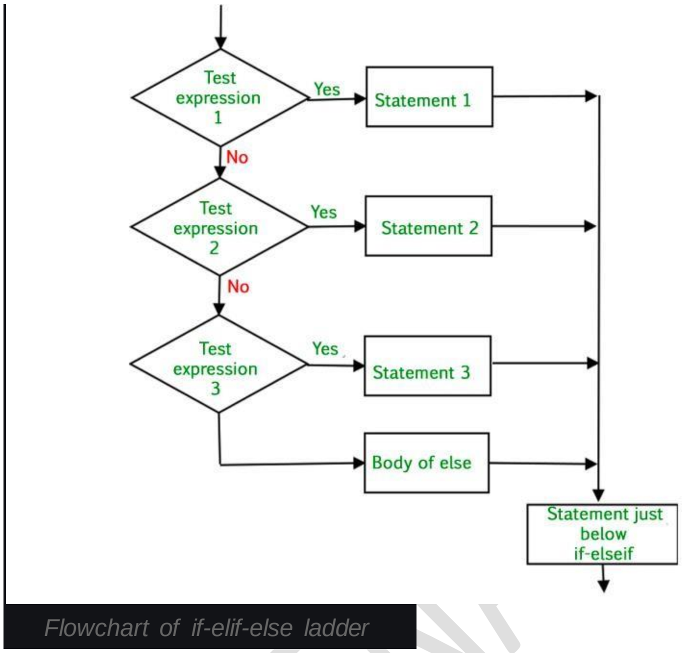
Python Elif

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.

Flowchart of Elif Statement in Python

Let's look at the flow of control in if-elif-else ladder:



Flowchart of if-elif-else ladder

Syntax:

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

Example of Python if-elif-else ladder

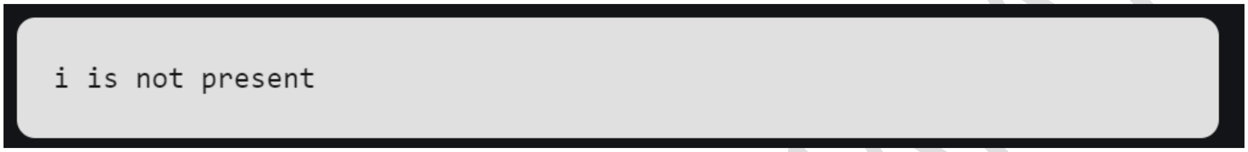
In the example, we are showing single if in Python, multiple elif conditions, and single else condition.

Python

```
# Python program to illustrate if-elif-else ladder
#!/usr/bin/python

i = 25
```

```python
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")
```

Output:



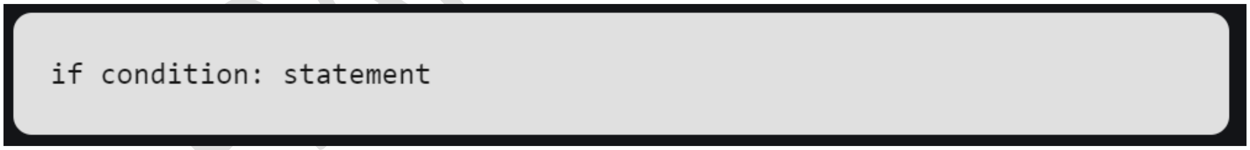
```
i is not present
```

Ternary Statement | Short Hand If Else Statement

Whenever there is only a single statement to be executed inside the if block then shorthand if can be used. The statement can be put on the same line as the if statement.

Example of Python If shorthand

In the given example, we have a condition that if the number is less than 15, then further code will be executed.

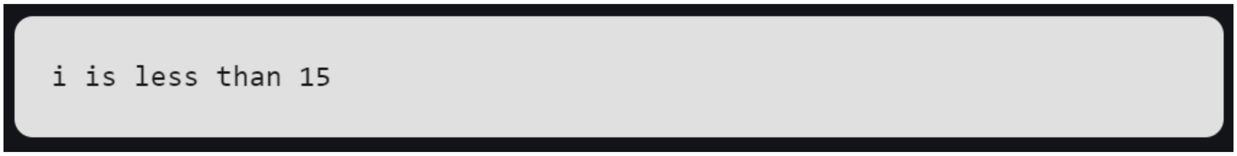


```
if condition: statement
```

Python

```python
# Python program to illustrate short hand if
i = 10
if i < 15: print("i is less than 15")
```

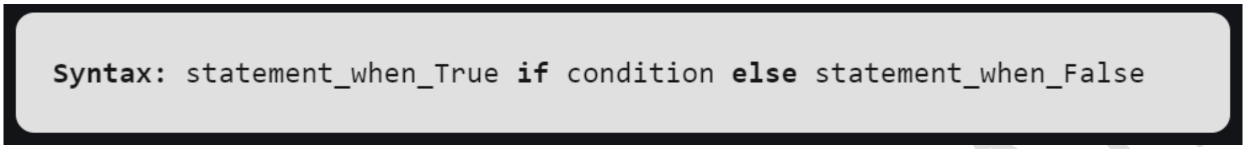
Output



```
i is less than 15
```

Example of Short Hand If Else Statements

This can be used to write the if-else statements in a single line where only one statement is needed in both the if and else blocks.



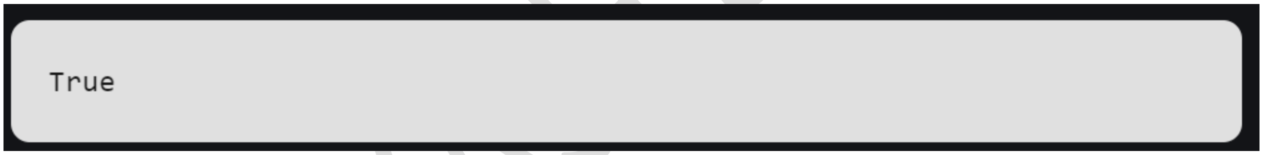
```
Syntax: statement_when_True if condition else statement_when_False
```

In the given example, we are printing True if the number is 15, or else it will print False.

Python

```python
# Python program to illustrate short hand if-else
i = 10
print(True) if i < 15 else print(False)
```

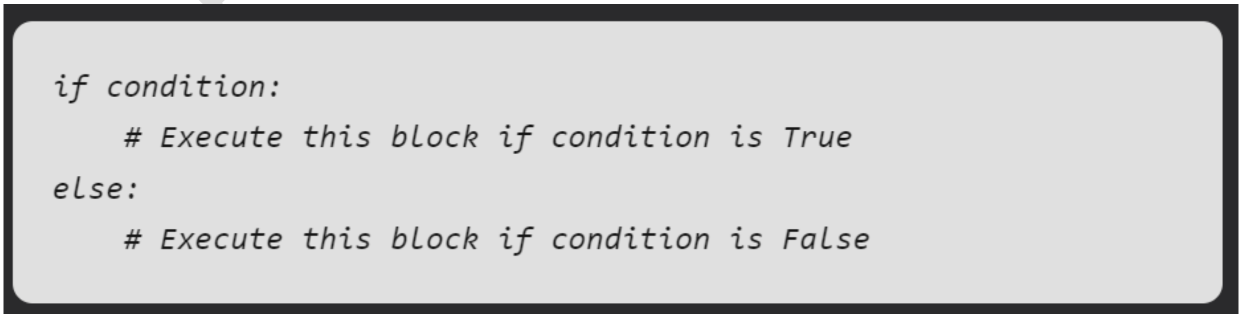
Output:



```
True
```

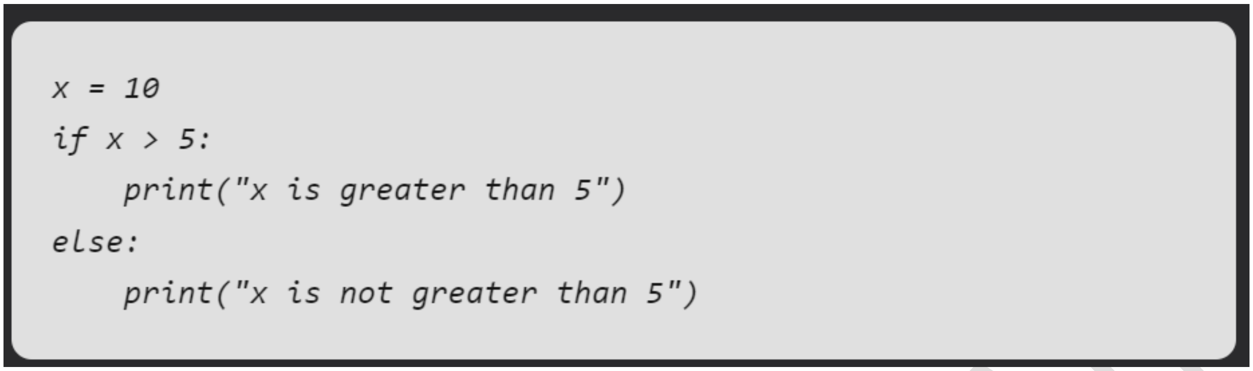
What is the conditional statement of if-else?

The `if-else` statement in Python is used to control the flow of the program based on a condition. It has the following syntax:



```
if condition:
    # Execute this block if condition is True
else:
    # Execute this block if condition is False
```

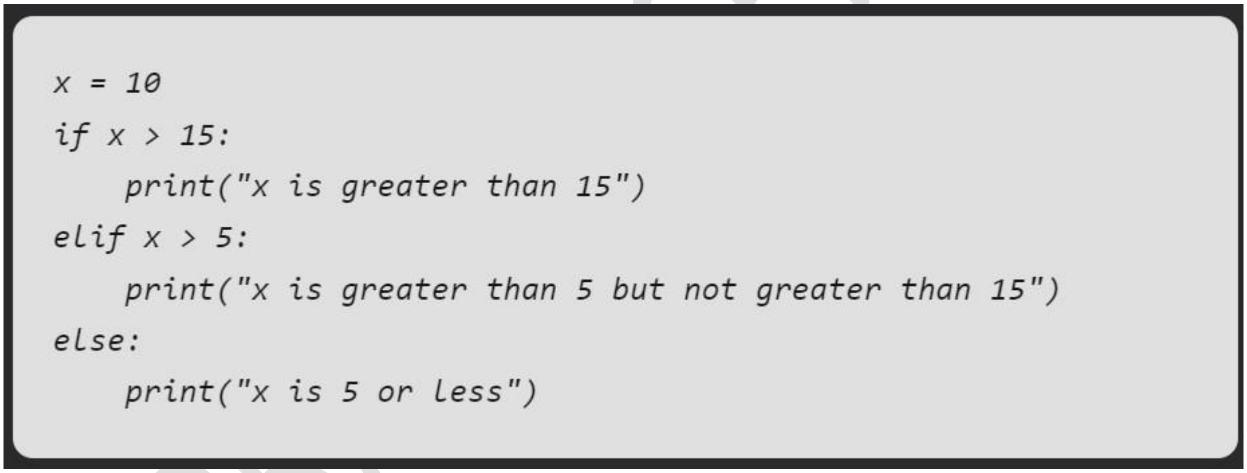
For example:



```
x = 10
if x > 5:
    print("x is greater than 5")
else:
    print("x is not greater than 5")
```

How many else statements can a single if condition have in Python?

A single `if` condition can have at most one `else` statement. However, you can have multiple `elif` (else if) statements to check additional conditions if needed:



```
x = 10
if x > 15:
    print("x is greater than 15")
elif x > 5:
    print("x is greater than 5 but not greater than 15")
else:
    print("x is 5 or less")
```

What are the different types of control statements in Python?

In Python, control statements are used to alter the flow of execution based on specific conditions or looping requirements. The main types of control statements are:

- ***Conditional statements**: `if`, `else`, `elif`*
- ***Looping statements**: `for`, `while`*
- ***Control flow statements**: `break`, `continue`, `pass`, `return`*

What are the two types of control statements?

The two primary types of control statements in Python are:

- ***Conditional statements**: Used to execute code based on certain conditions (`if`, `else`, `elif`).*
- ***Looping statements**: Used to execute code repeatedly until a condition is met (`for`, `while`).*

Are control statements and conditional statements the same?

No, control statements and conditional statements are not exactly the same.

- ***Conditional statements** (`if`, `else`, `elif`) specifically deal with checking conditions and executing code based on whether those conditions are `True` or `False`.*
- ***Control statements** encompass a broader category that includes both conditional statements (`if`, `else`, `elif`) and looping statements (`for`, `while`), as well as other statements (`break`, `continue`, `pass`, `return`) that control the flow of execution in a program.*