Date : 27th - 08 - 2020
Morning Session : 9am – 11.00 PM
By ~ Rohan Kumar

Topics: Python Syntax, comments, Variables, understand 1st python program

Concatenating: String Concatenation is the technique of combining two strings. String Concatenation can be done using many ways.

```
In [6]: str1 = "hello"
        str2 = " world"
        str1 + str2

Out[6]: 'hello world'
```

```
In [7]: str1 = "2"
        str2 = "3"
        str1 + str2

Out[7]: '23'
```

Comment : **we can use # for comment.**

Star Operator:

```
:  #using star operator
   str1 * 3

:  'PythonPythonPython'
```

```
:  str2 = "Program"
   str2 * 5
```

In Operator:

```
In [13]: test = "it's my life"

In [14]: test
Out[14]: "it's my life"

In [15]: #in operator
         "my" in test
Out[15]: True

In [16]: "hi" in test
Out[16]: False
```

If Statement:

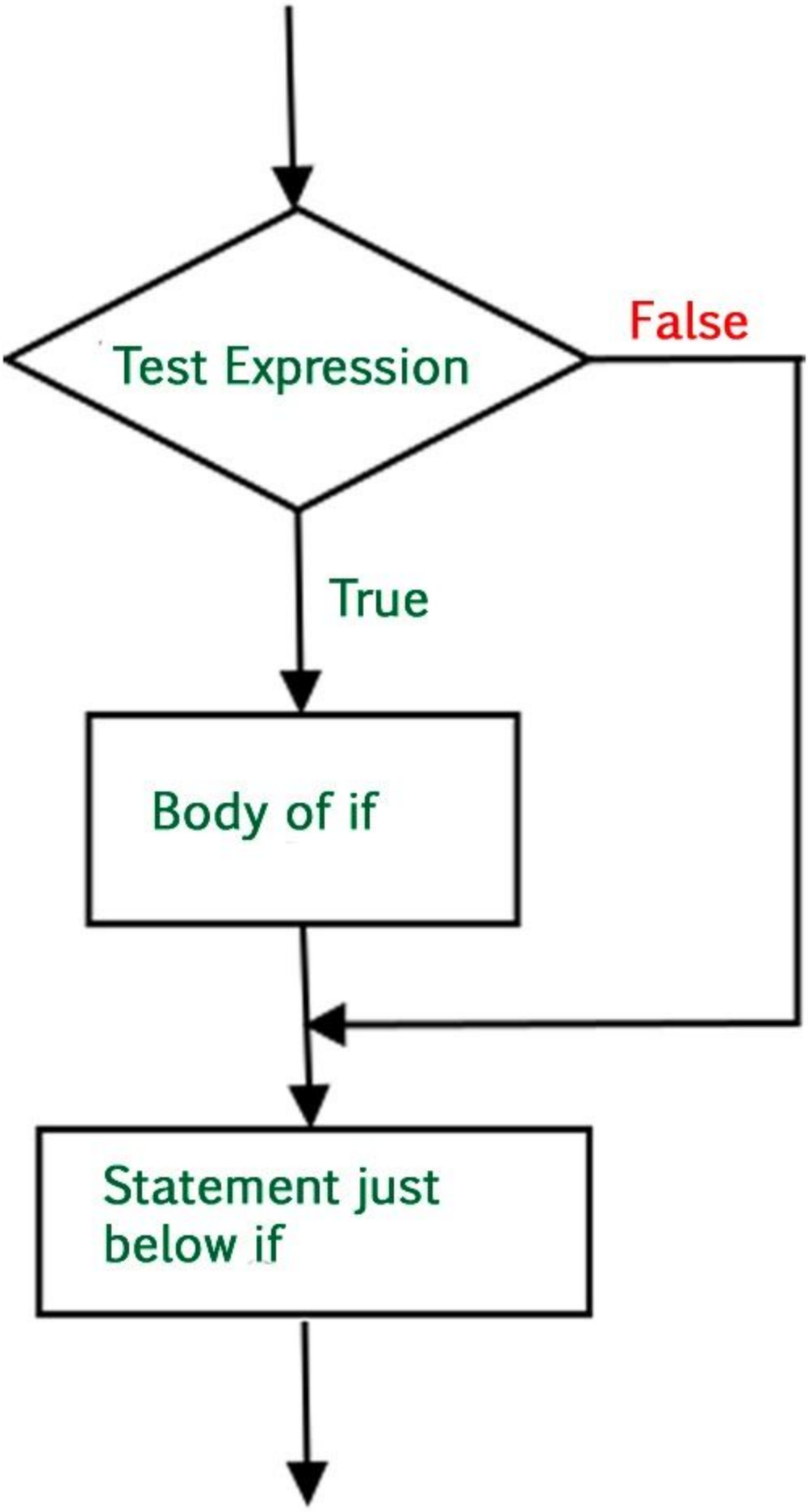
if statement is the most simple decision making statement. It is used to decide whether a certain statement or block of statements will be executed or not i.e if a certain condition is true then a block of statement is executed otherwise not.

Syntax:

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

```
In [23]: if True:
             print("i am true")
             print("i am alos true")
         print("i am out of if condition")

i am true
i am alos true
i am out of if condition
```

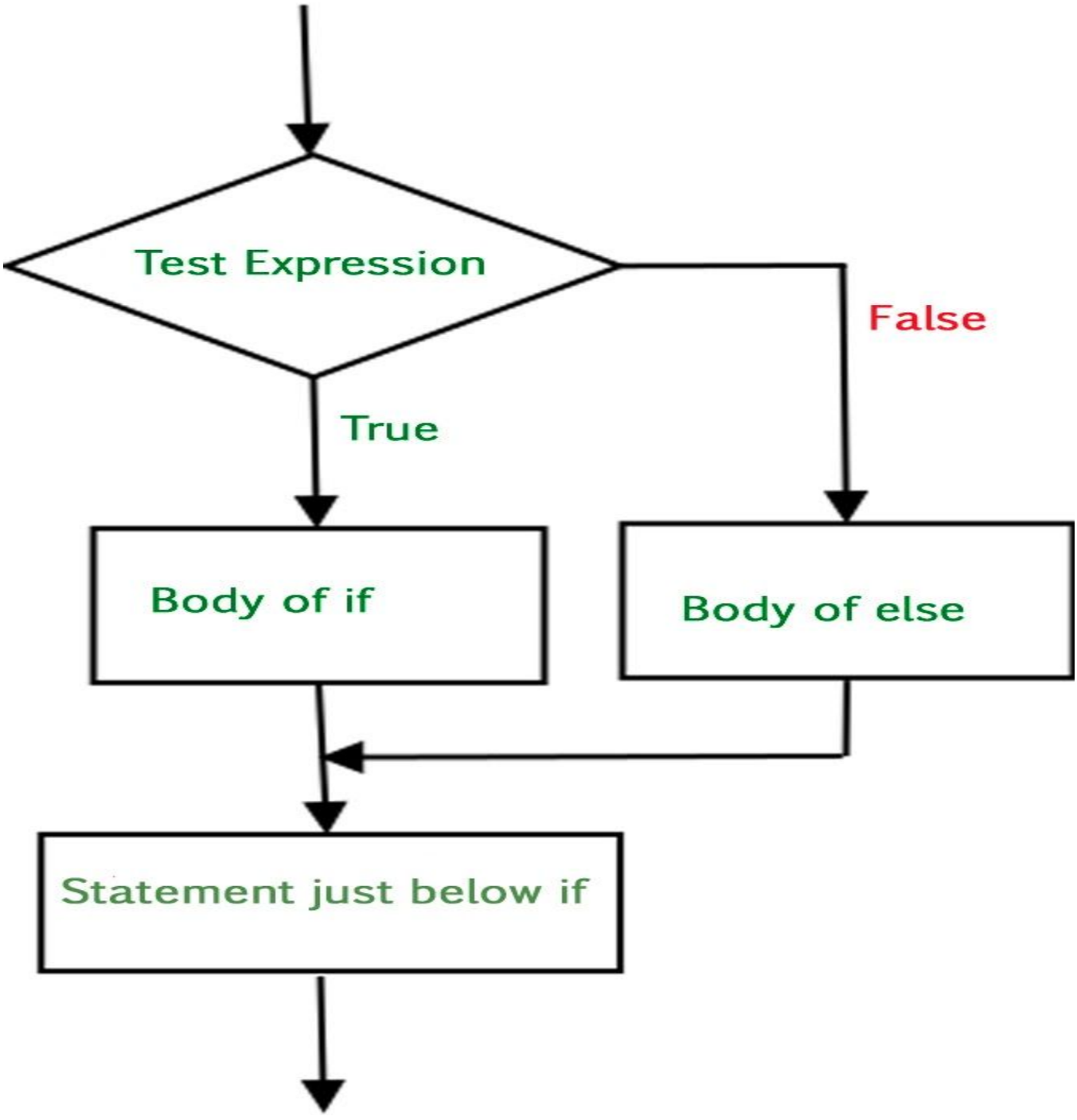
```
In [24]: if False:
             print("i am false")
         print("i am true")

i am true
```


Test Expression
False
True
Body of if
Statement just
below if

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.

Syntax:

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



```
#even number program and odd number
num = 5
rem = num % 2
if rem == 0:
    print("even")
else:
    print("odd")
print("end")
```

```
odd
end
```

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.

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
```

```
#nested If Else statements
num  =8
rem = num % 2 #output - 0
if rem == 0:
    print("even")  #this will get executed
    if num > 5:
        print("greater") #this line get executed
    else:
        print("smaller")
else:
    print("odd")
print("end") #this line
```

```
even
greater
end
```

Elif - Else If : 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.

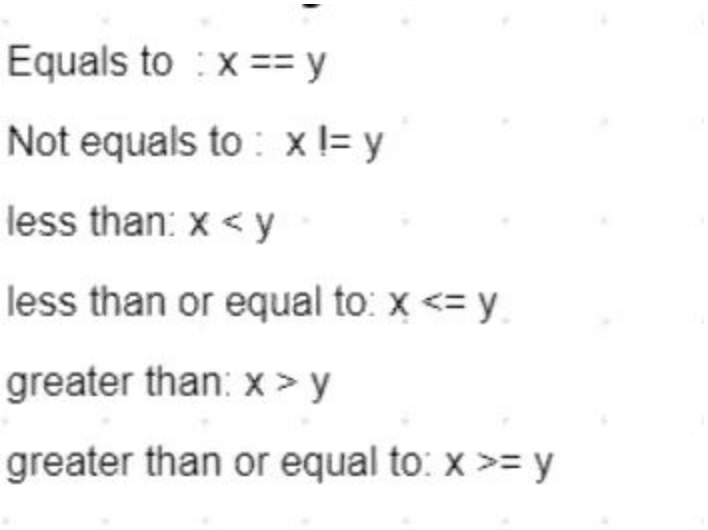
Syntax:-

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

```
#using elif - else-if
num = 1
if num == 1:
    print("one")
elif num == 2:
    print("two")
elif num == 3:
    print("three")
elif num == 4:
    print("four")
```

```
one
```

Python Conditions:



Based on these conditions we take decisions.

Operators:

1) **Arithmetic Operator. (+, - , * , / , % , //)**
2) **Assignments Operator. (= , +=, -=, *=, //=)**
3) **Relational Operator. (<, >, <=, => , == , !=)**
4) **Logical Operator.**
5) **Unary Operator.**

Logical operators: Logical operators perform Logical AND, Logical OR and Logical NOT operations.

and	Logical AND: True if both the operands are true
or	Logical OR: True if either of the operands is true
not	Logical NOT: True if operand is false

```
a = 10
b = 6
c = 8
if a > b:
    print("a is greater than b")
if a > c:
    print("a is grater than c")
```

```
a is greater than b
a is grater than c
```

```
if a >b and a>c :
    print("a is greater than evry no.")
```

```
a is greater than evry no.
```

```
if a >b or a >c:
    print("a is gretre")
```

```
# and , or, not --- and > or > not
name = "rohan"
age = 25
if ((name == "rohan") or (name == "aniket" and age >= 20)):
    print("this one")
else:
    print("second one")
```

```
this one
```