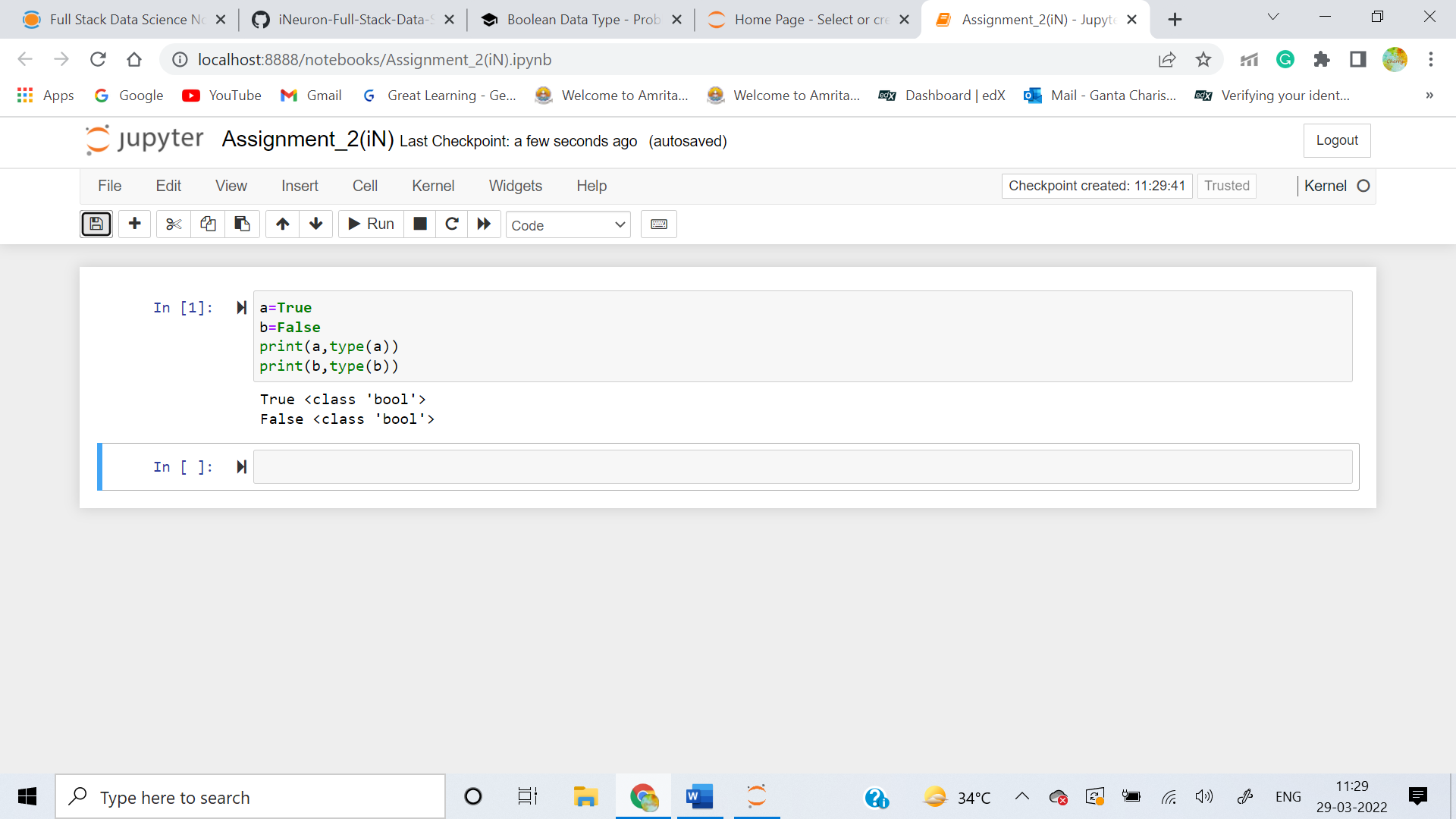
**1.What are the two values of the Boolean data type? How do you write them?**

Answer:

A Variable of the primitive data type Boolean can have two values: TRUE and FALSE ( Boolean literals).Boolean variables are used to indicate whether a condition is true or not, or to represent two states.



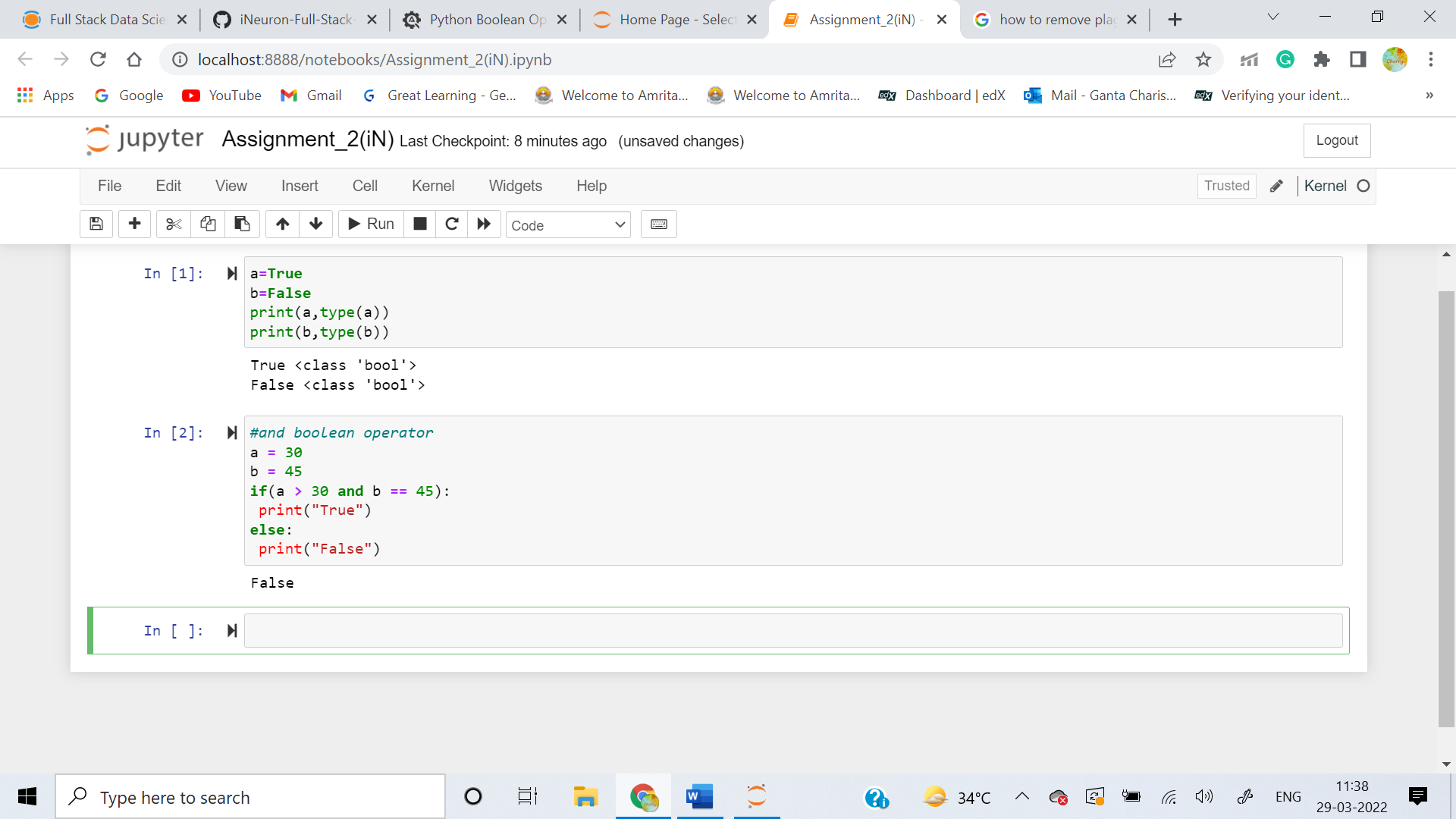
**2. What are the three different types of Boolean operators?**

Answer:  there are three types of Boolean operators:

* The AND operator (&& or "and")
* The OR operator (|| or "or")
* The NOT operator (not)

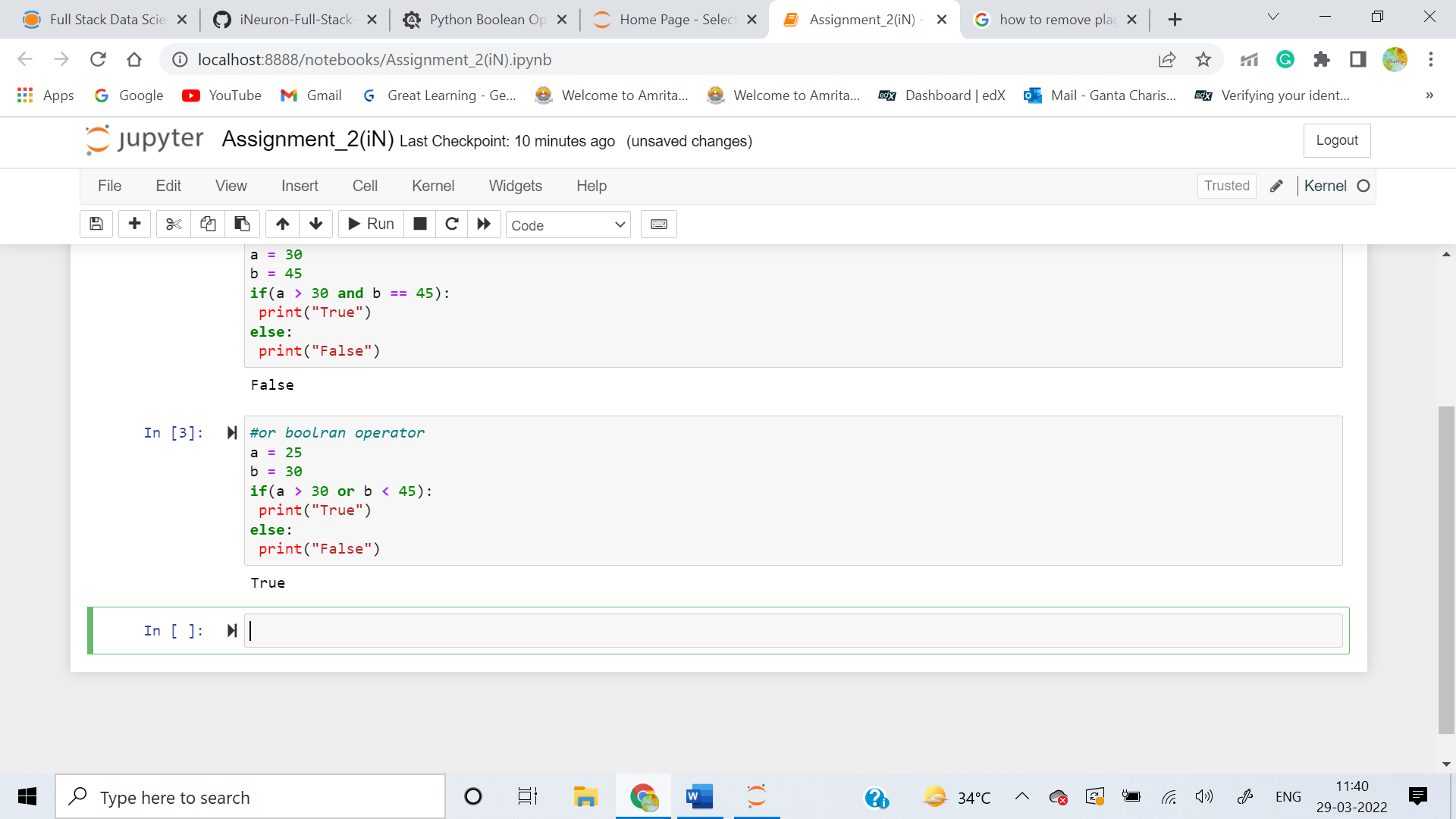
The **AND boolean operator** is similar to the bitwise **AND operator** where the operator analyzes the expressions written on both sides and returns the output.

* True and True = True
* True and False = False
* False and True = False
* False and False = False



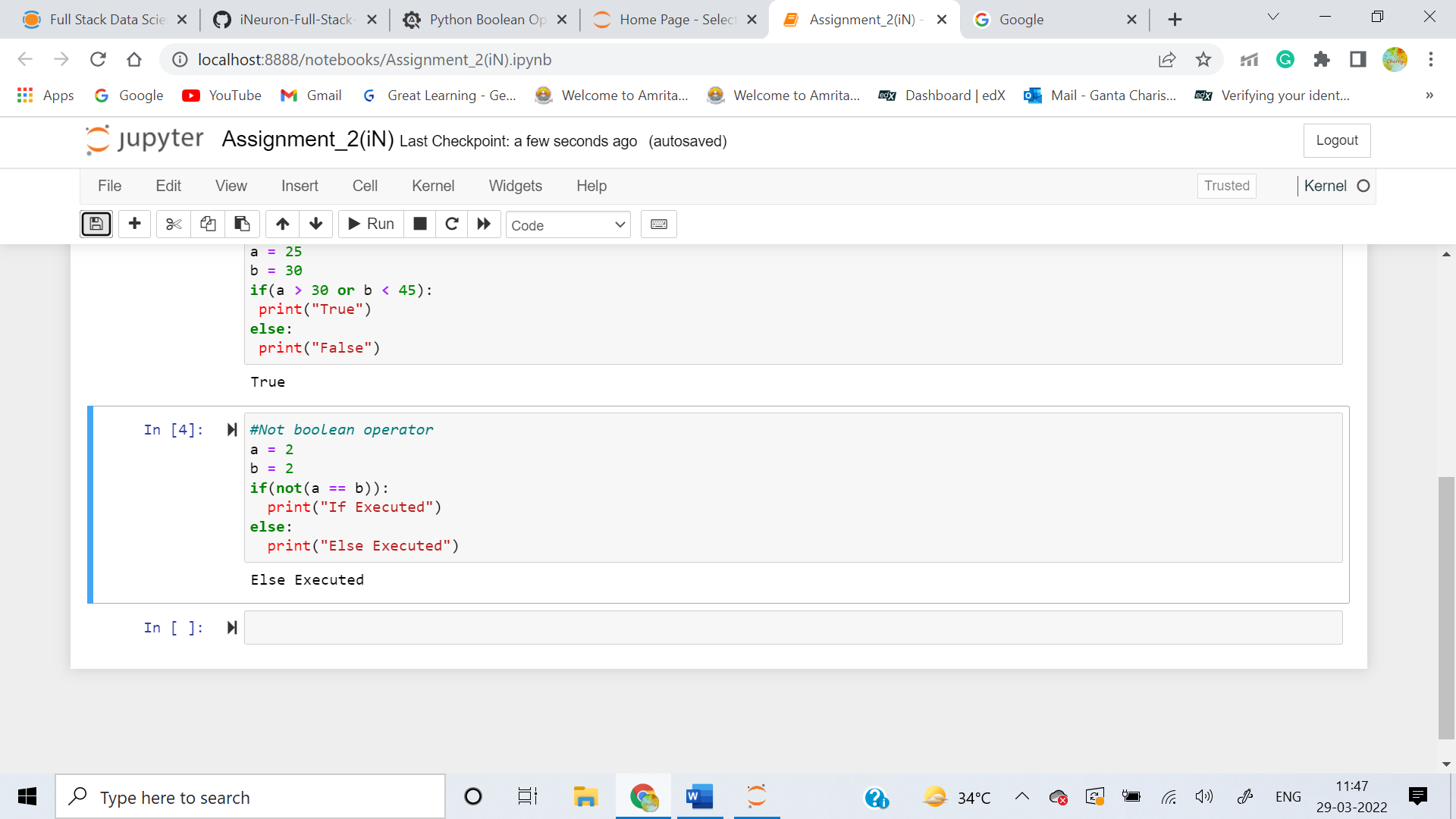
The **OR operator** is similar to the **OR bitwise operator**. In the bitwise OR, we were focussing on either of the bit being 1. Here, we take into account if either of the expression is true or not. If at least one expression is true, consequently, the result is true.

* True or True = True
* True or False = True
* False or True = True
* False or False = False



The **NOT operator** reverses the result of the boolean expression that follows the operator. It is important to note that the NOT operator will only reverse the final result of the expression that **immediately follows.** Moreover, the NOT operator is denoted by the keyword **"not".**

* not(True) = False
* not(False) = True



**3. Make a list of each Boolean operator's truth tables (i.e. every possible combination of Boolean values for the operator and what it evaluate ).**

**Answer:** The Truth tables for the boolean Operators are as follows:

**Truth Table for (AND) Operator:**

* True and True = True
* True and False = False
* False and True = False
* False and False = False

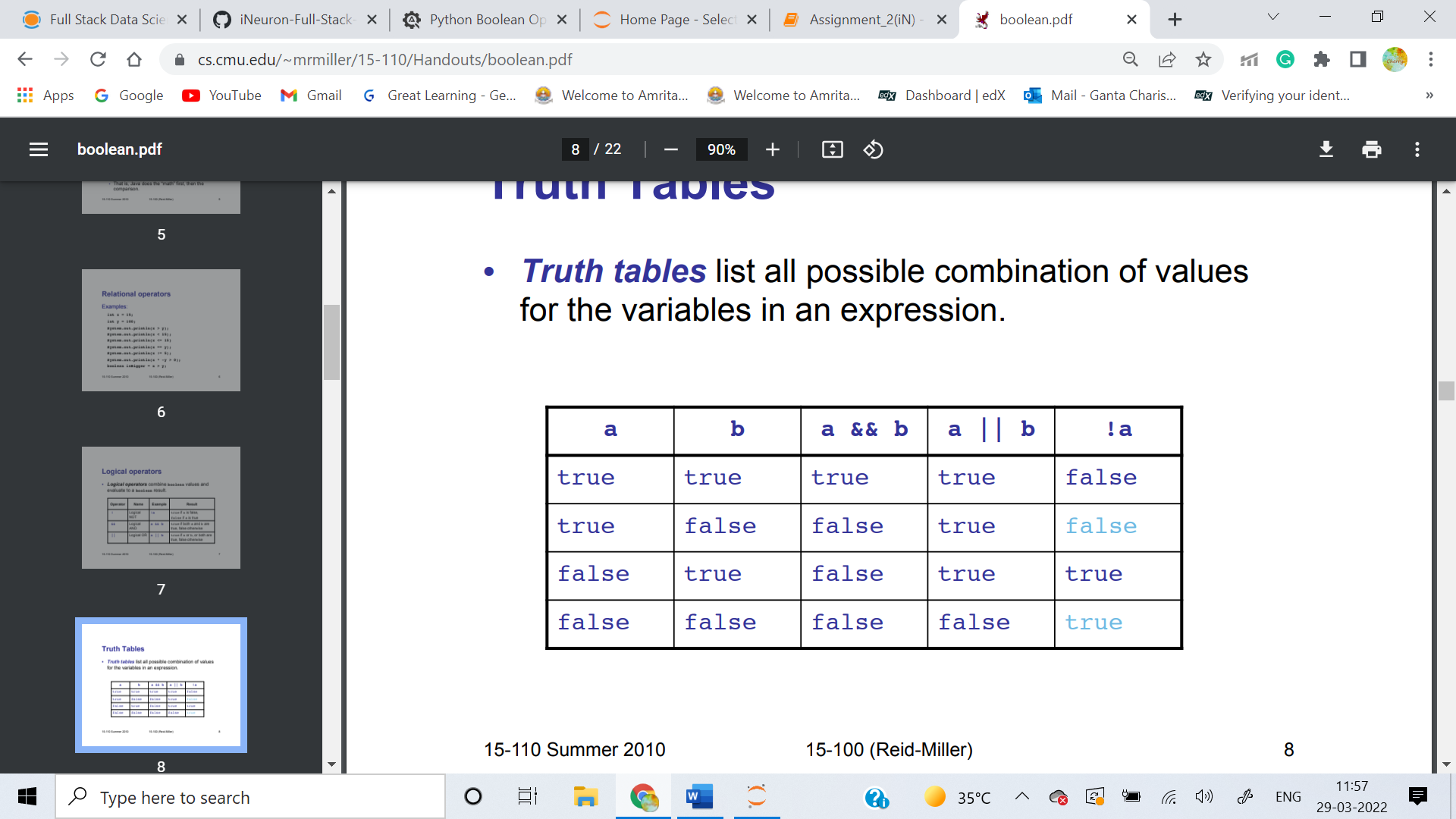
**Truth Table for (OR) Operator:**

* True or True = True
* True or False = True
* False or True = True
* False or False = False

**Truth Table for (NOT) Operator:**

* not(True) = False
* not(False) = True

Truth tables list all possible combination of values for the variables in an expression.



**4. What are the values of the following expressions?**

**(5 > 4) and (3 == 5)**

**not (5 > 4)**

**(5 > 4) or (3 == 5)**

**not ((5 > 4) or (3 == 5))**

**(True and True) and (True == False)**

**(not False) or (not True)**

**Answer:**

((5**>**4)**and**(3**==**5)) : False

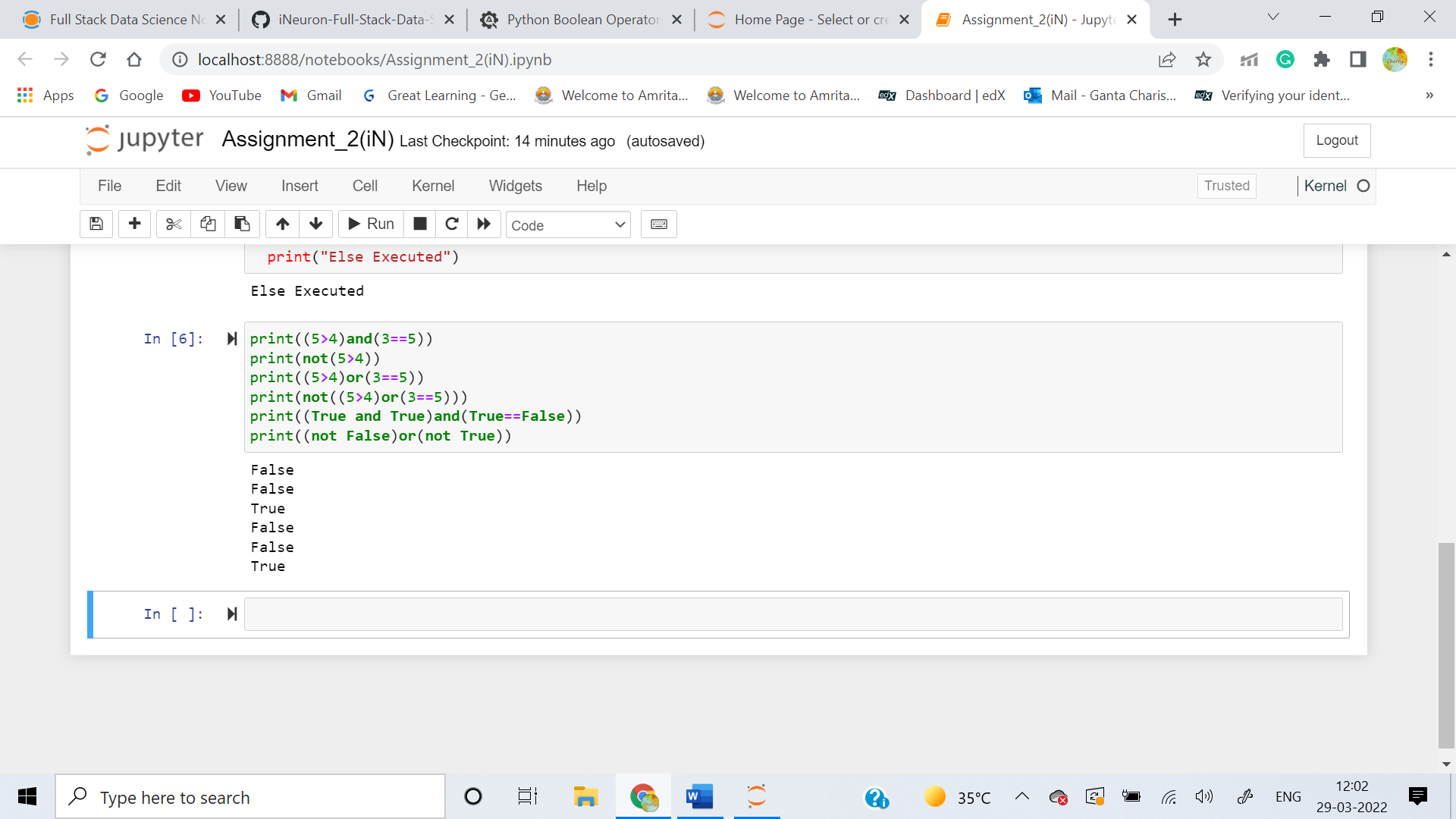
(**not**(5**>**4)) :False

((5**>**4)**or**(3**==**5)) : True

(**not**((5**>**4)**or**(3**==**5))) :False

((**True** **and** **True**)**and**(**True==False**)) : False

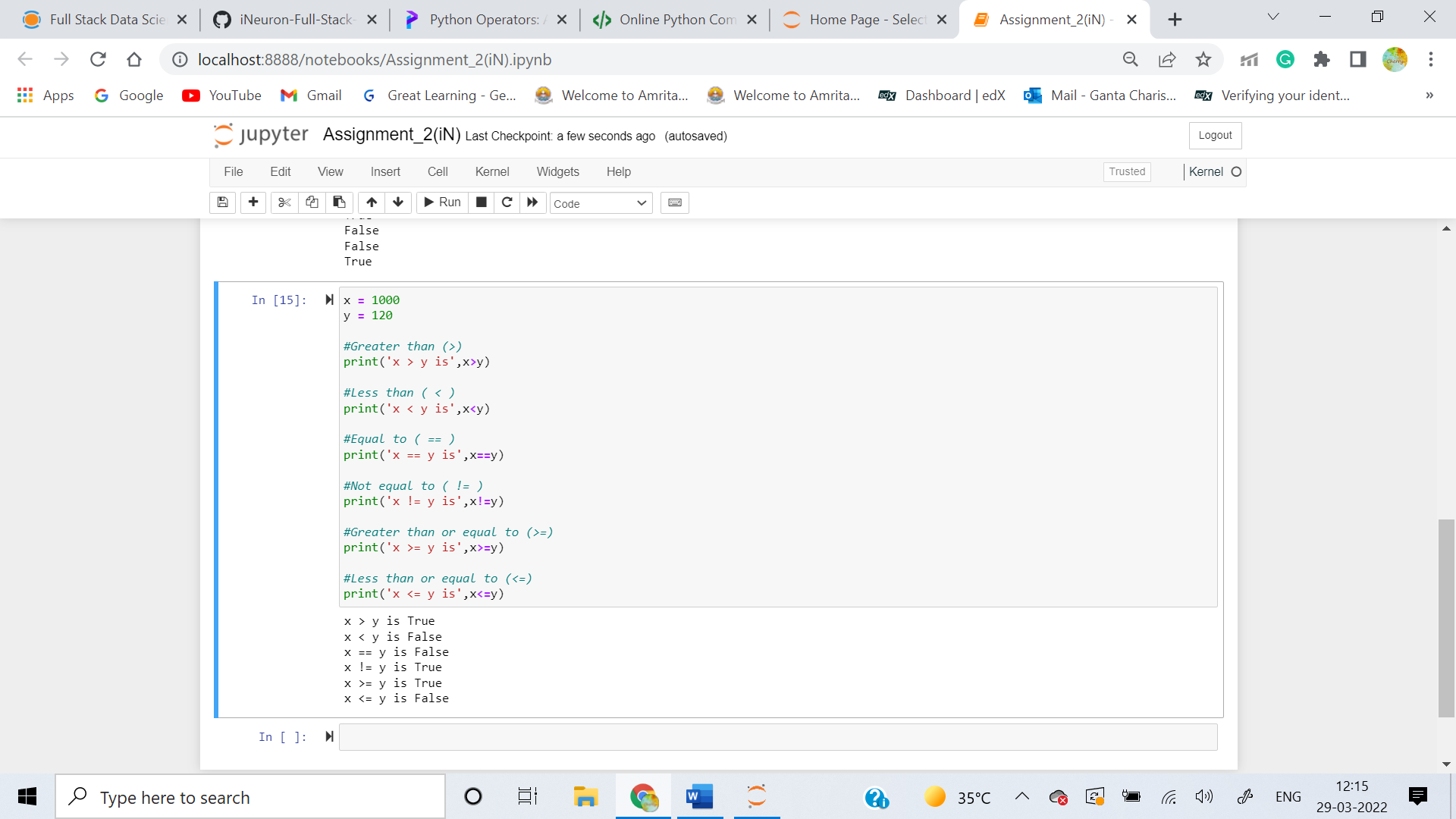
((**not** **False**)**or**(**not** **True**)) : True



**5. What are the six comparison operators?**

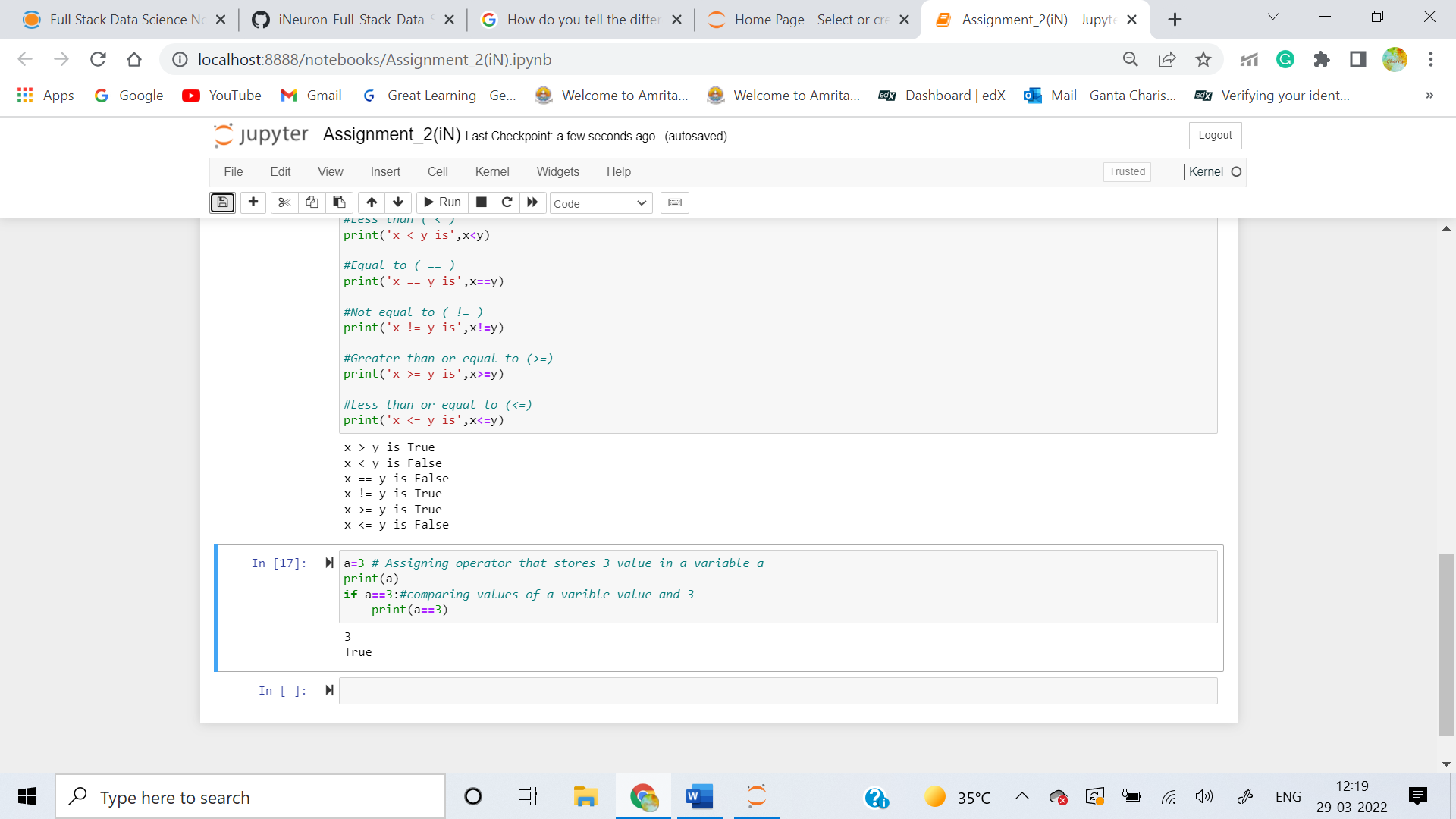
Python has six comparison operators, which are as follows:

* Less than ( < )
* Less than or equal to (<=)
* Greater than (>)
* Greater than or equal to (>=)
* Equal to ( == )
* Not equal to ( != )



**6. How do you tell the difference between the equal to and assignment operators?Describe a condition and when you would use one.**

Answer:The “=” is an assignment operator is used to assign the value on the right to the variable on the left. **The '==' operator checks whether the two given operands are equal or not**. If so, it returns true. Otherwise it returns false.



**7. Identify the three blocks in this code:**

**spam = 0**

**if spam == 10:**

**print('eggs')**

**if spam > 5:**

**print('bacon')**

**else:**

**print('ham')**

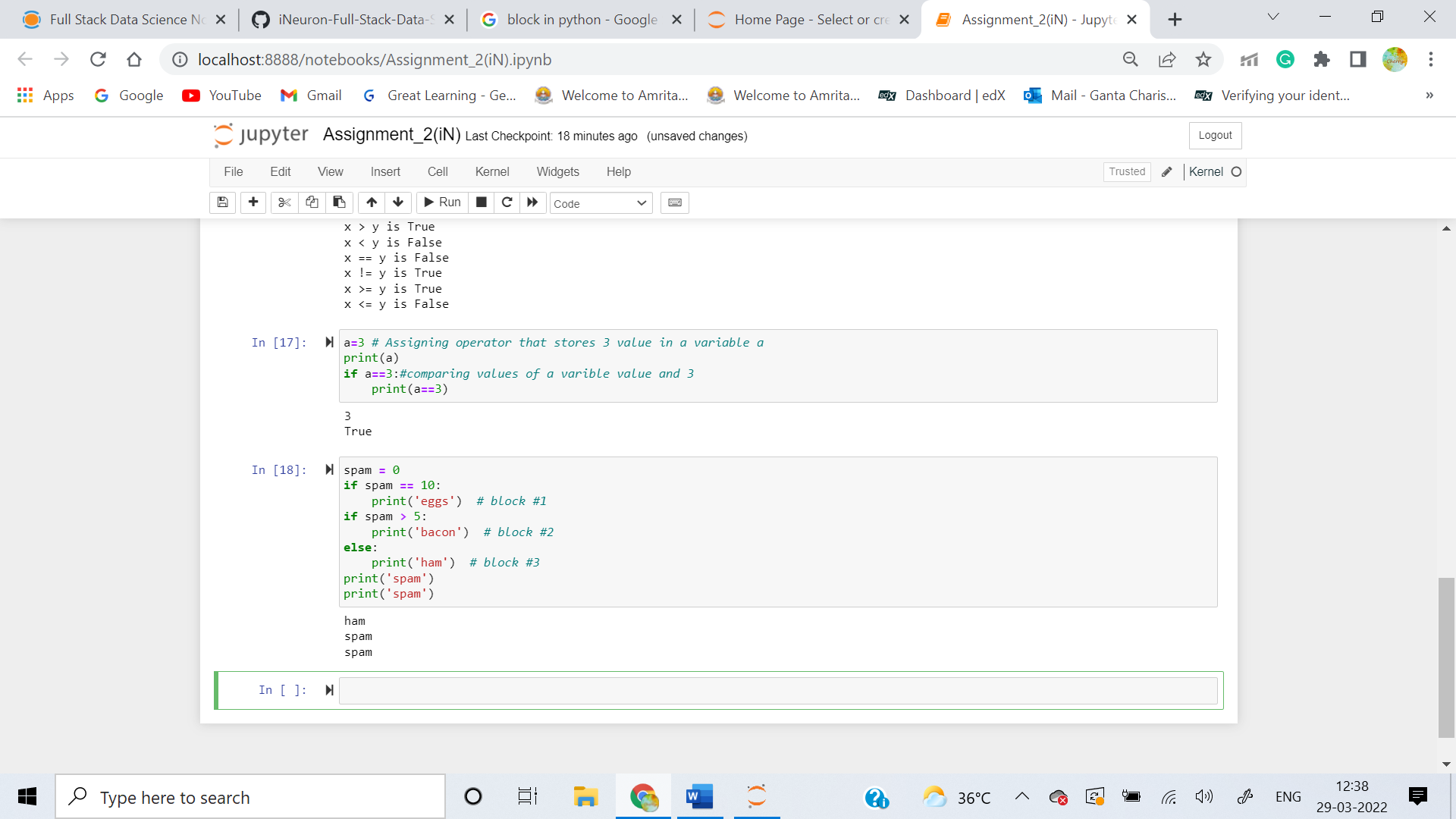
**print('spam')**

**print('spam')**

**Answer:**

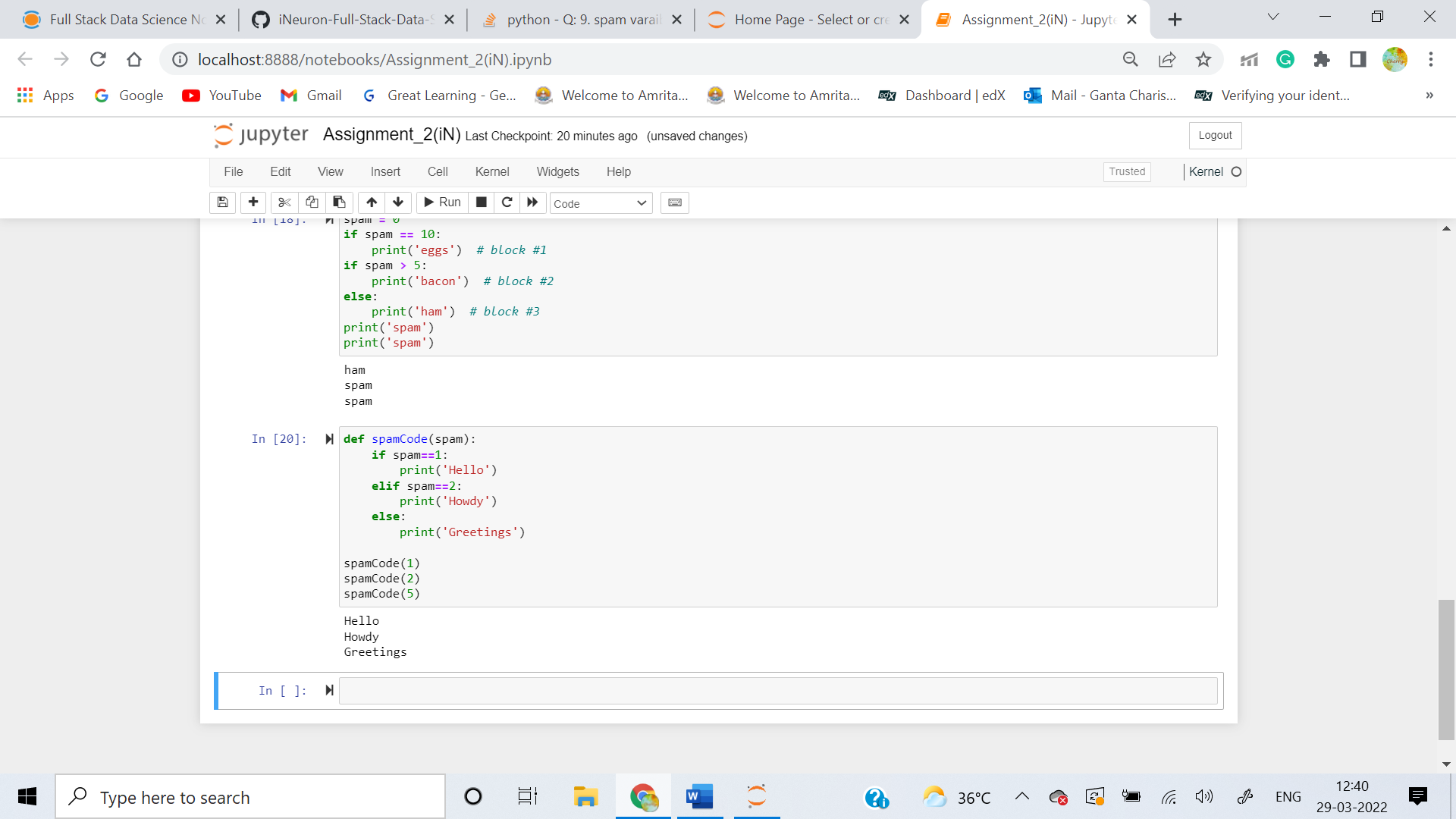
In Python, code block refers to **a collection of code that is in the same block or indent**.

The three blocks in this code are:



**8. Write code that prints Hello if 1 is stored in spam, prints Howdy if 2 is stored in spam, and prints Greetings! if anything else is stored in spam.**

**Answer:**



**9.If your programme is stuck in an endless loop, what keys you’ll press?**

**Answer:** If a programme is stuck in an endless loop, Press **Ctrl-c**to stop a program stuck in an infinite loop.

**10. How can you tell the difference between break and continue?**

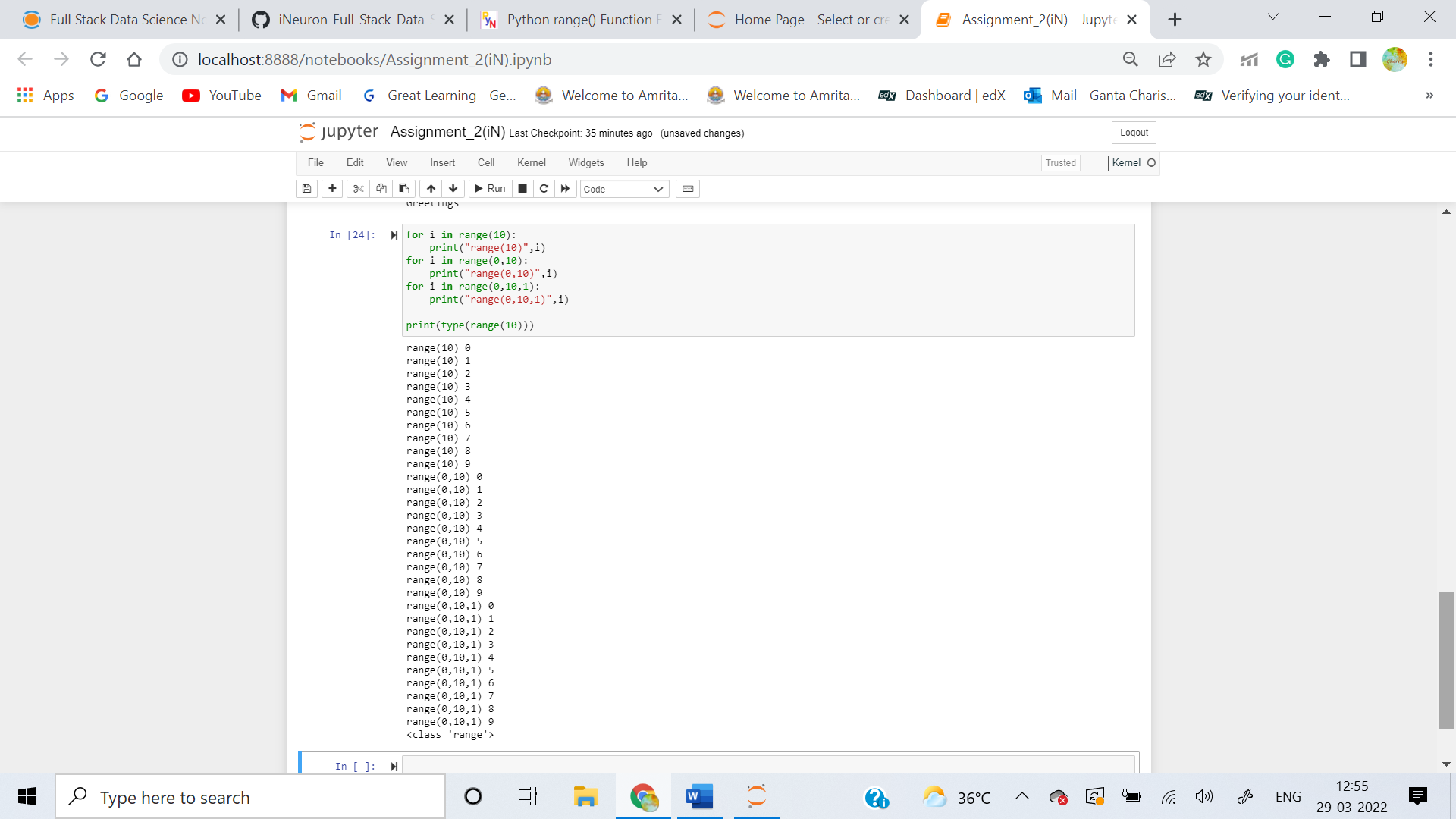
**Answer:**

The Python break statement stops the loop in which the statement is placed. A Python continue statement skips a single iteration in a loop. The break statement will move the execution outside the loop if break condtion is satisfied. Whereas the continue statement will move the execution to the start of the loop. Both break and continue statements can be used in a for or a while loop.

**11. In a for loop, what is the difference between range(10), range(0, 10), and range(0, 10, 1)?**

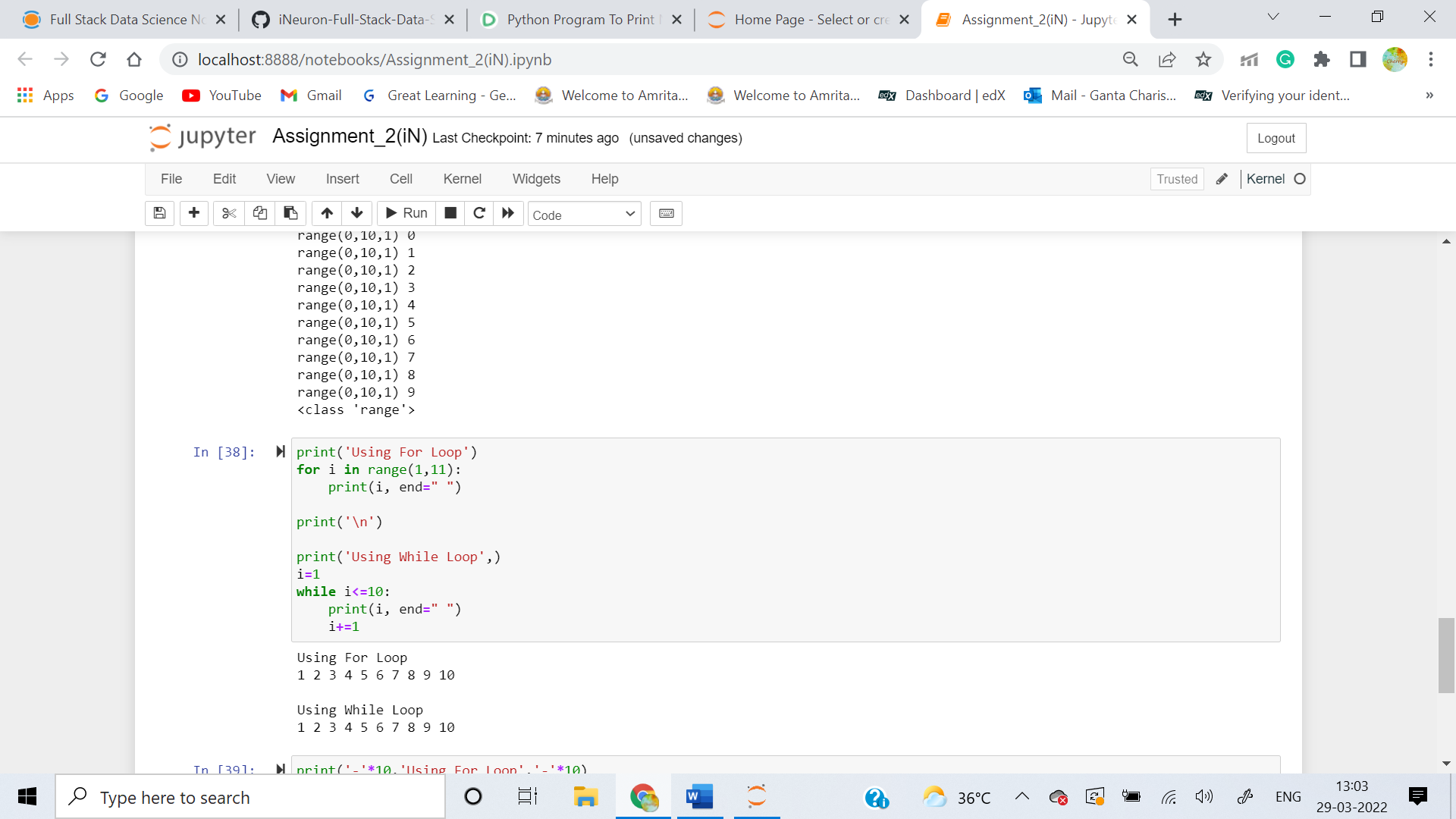
**Answer:** The Differences are as follows:

1. The ***range(10)*** call range from 0 to 9 (but not include 10)
2. The ***range (0,10)*** explicitly tells the loop to start at 0
3. The ***range(0,10,1)*** explicitly tells the loop to increase the variable by 1 on each iteration



**12. Write a short program that prints the numbers 1 to 10 using a for loop. Then write an equivalent program that prints the numbers 1 to 10 using a while loop.**

**Answer:**



**13. If you had a function named bacon() inside a module named spam, how would you call it after importing spam?**

**Answer:** This function can be called with **spam.** **bacon()**.