TOTAL Python

unittest

> Unit Testing is a method used in programming to determine if a module or set of code modules is working properly. This evaluation is done in a separate file. In Python, it is implemented from the built-in unittest module.

```
import unittest
import mymodule
```

```
class TestName(unittest.TestCase):
   def test_marketingcode(self):
      fist_value = {smth}
      second_value = {output from  mymodule.function}
      self.assertEqual(fist_value, second_value, message)


if __name__ == '__main__':
   unittest.main()
```

It is key to respect the portions of code with a yellow background.

The first two arguments to **assertEqual** are two values that are compared to establish whether they are equal. For this reason, one must be obtained from a function of the evaluated module, and another must be the expected output for the same information input as in the first case. The third parameter (message) will contain a string with information that will be shown to the user in case the test fails.

Before even executing the code, Python reads the file to define some global variables. One of them is __name__, which takes the name "__main__" in case Python is running in that module individually. If, on the other hand, the module was imported, the __name__ variable takes the name of the module. This block of code evaluates that **the test is running directly**.