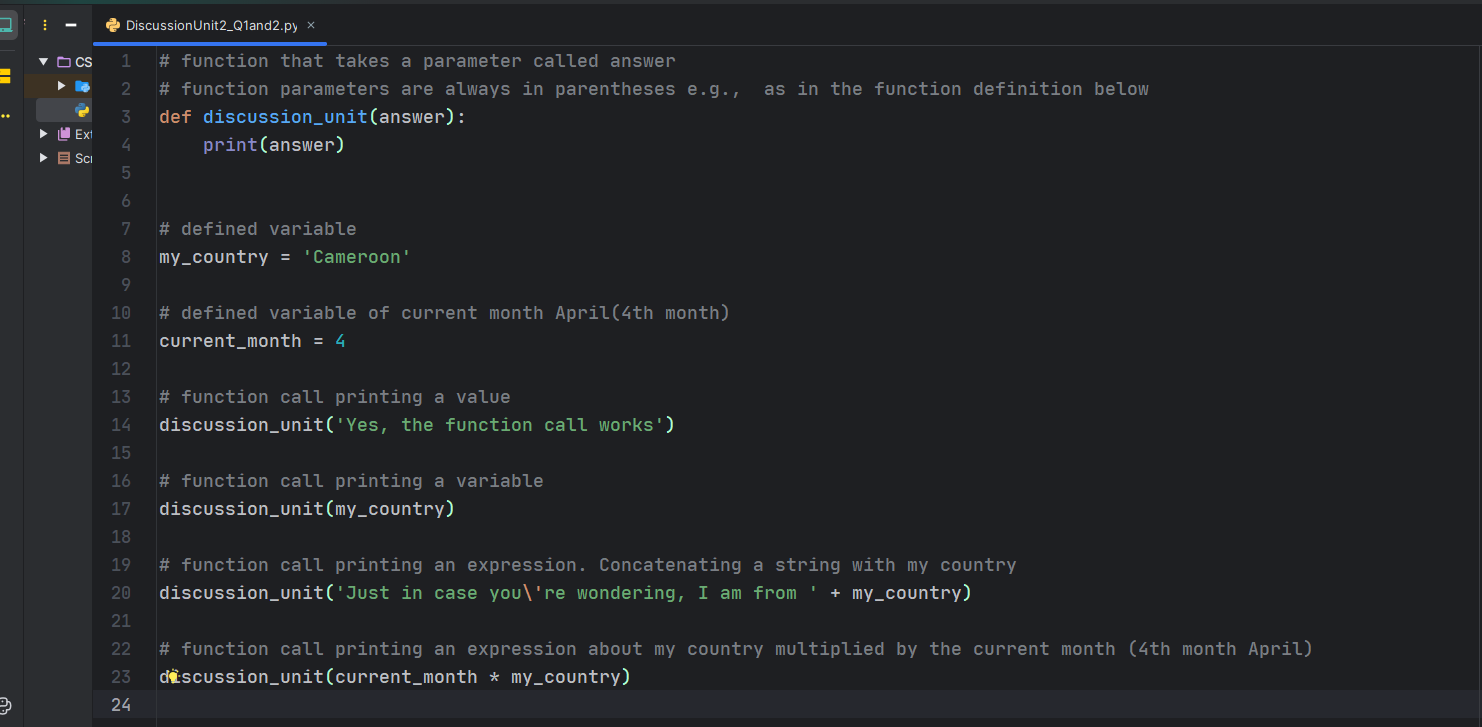
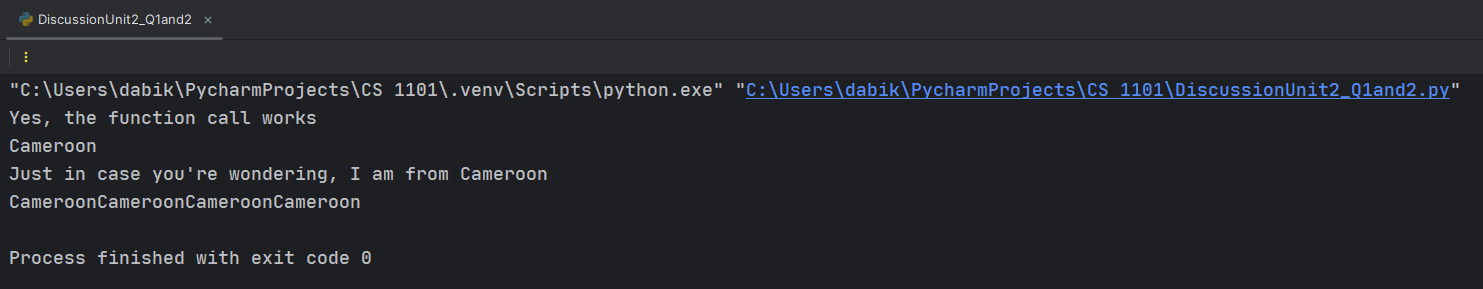
**Discussion Unit 2**

1. Example 1 & 2

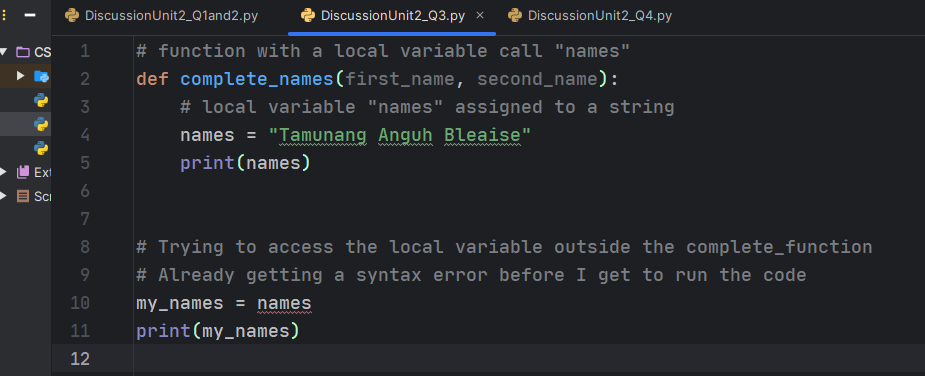


* For example 1:
  + The code argument for the define function is **answer**
  + The parameter for the function is in parenthesis **(answer)**
* For example 2:
  + The function is called to print an argument (string value) provided to it.
  + The function is also called to print a defined function my\_country (print variable)
  + This same function prints an expression( a string concatenated with my\_country)

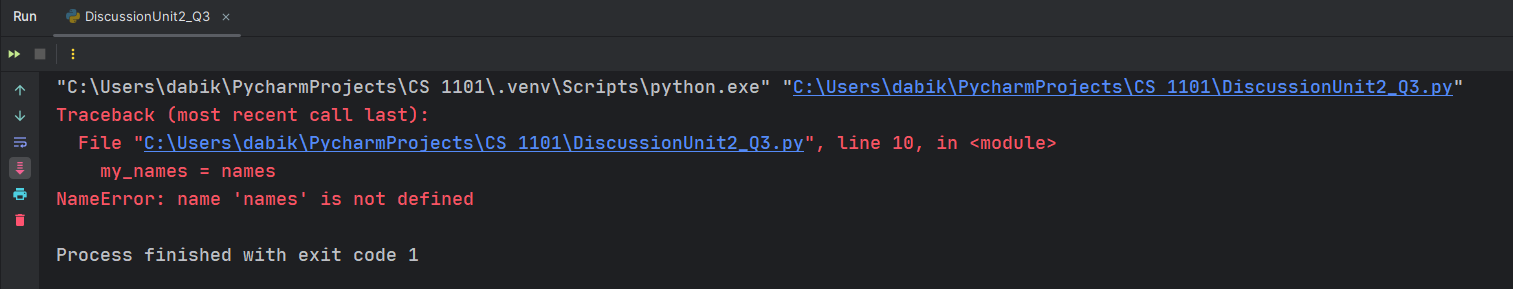
See the results below when I run the code:



1. Example 3



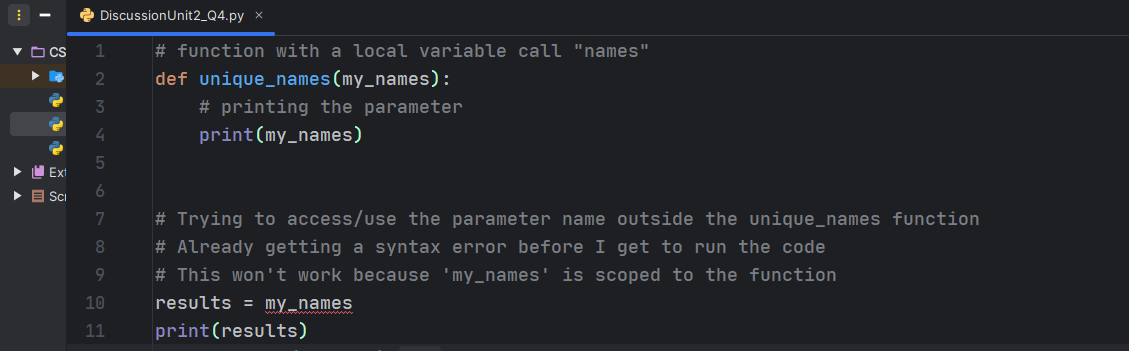
After running the code, I get the following results



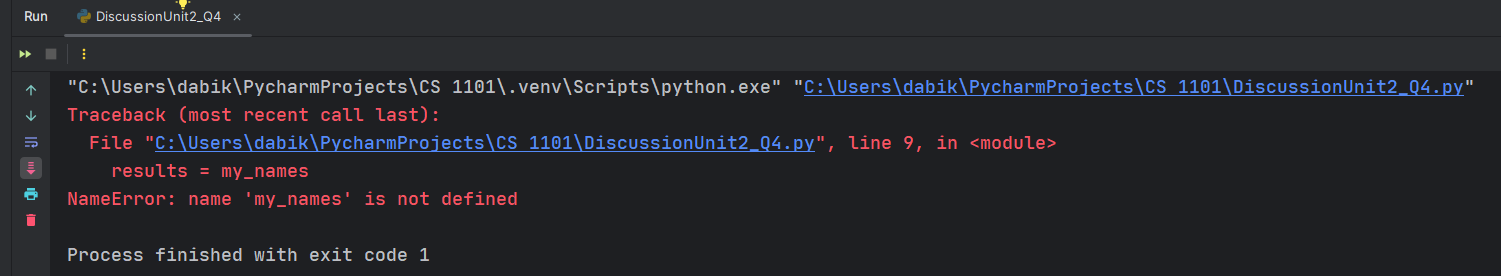
Explanation:

* We get the above error because all variables inside a function are local. Meaning, I can only access function variables inside the function and not outside the function as the interpreter does not recognize the variable ***“names”.***  This gives a ***NameError*** in indicating that the ***“names*”** variable is not defined since it’s local and not Global

1. Example 4



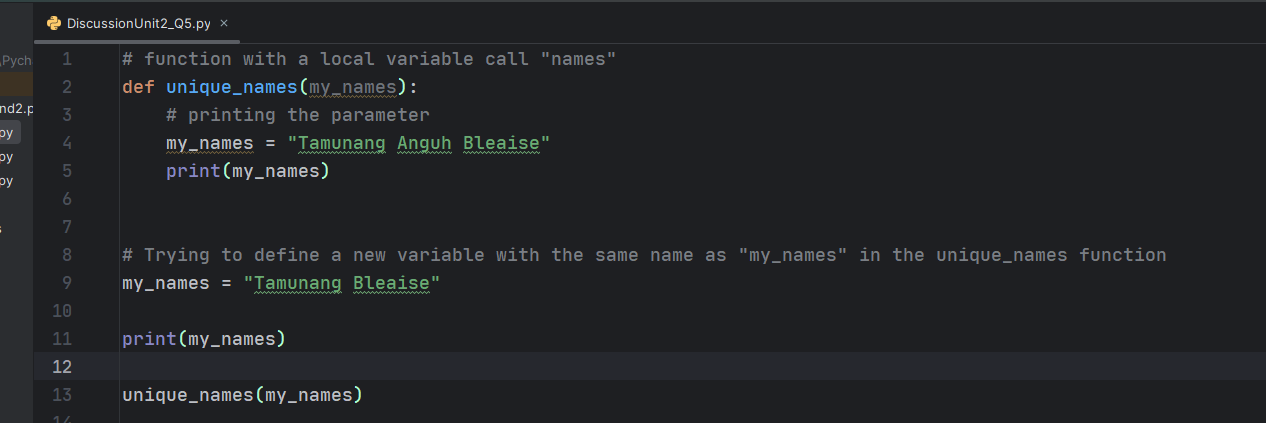
Results:



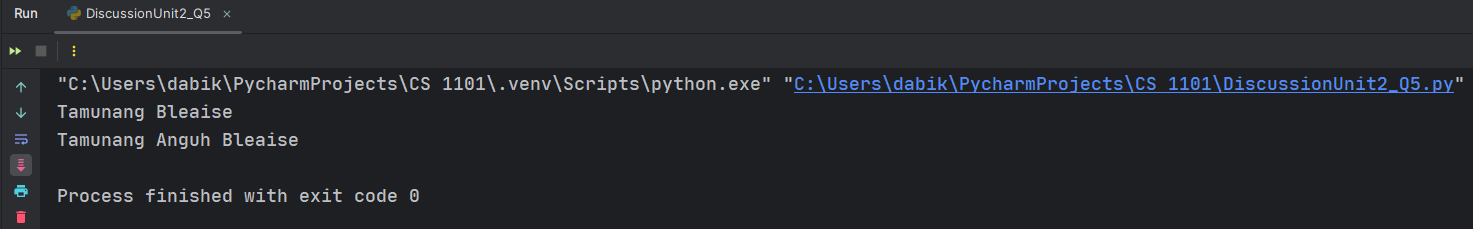
Explanation:

* Just like in example 3, We get the above error because all variables inside a function are local. Meaning, I can only access function variables inside the function and not outside the function as the interpreter does not recognize the variable ***“my\_names”.***  This gives a ***NameError*** in indicating that the ***“my\_names*”** variable is not defined since it’s local and not Global
* The ***Traceback*** in the code error tells us what program file the error occurred in, what line, and what functions were executing at the time. It also shows the line of code that caused the error (Allen Downey, 2015).

1. Example 5



Results:

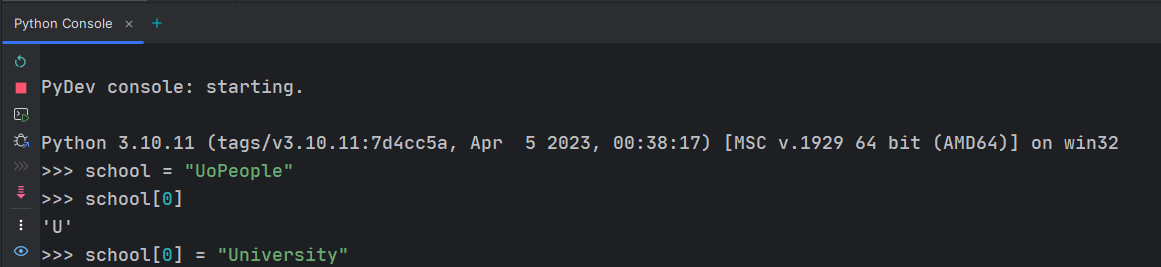


Explanation:

* From the screenshot, I am already an error from my IDE (Sematic Error)
* The code runs successfully without any errors but generates 2 different unexpected results(what semantic errors do best)
* When we run the code, the unique\_names function is been called. The function calls other statements inside the function following the flow of execution in Python.
* The function jumps into the function body and executes all the statements inside. This explains why even though the my\_name variables can have changed, the function is still able to print the original value of my\_name.
* The print function prints the value of the variable outside of the unique\_function because as it stands, there exists no function with the name “my\_names”

Question:

* Can we change the value of a string once it has been created (Are strings immutable)?



# References

Allen Downey, G. T. (2015). *Think Python: How to Think Like a Computer Scientist.* Needham, Massachusetts: Green Tea Press.