Jonathan Den Haan

11/18/2022

Foundations of Programming: Python

Assignment 06

**Module 6**

**Introduction:**

In this module we learn about using functions to simplify code writing. Functions can also be incorporated into a function class, similar to a list of lists where each function is part of a larger group of functions. We also learn about the use of doc strings to document what is happening in our code. At the end of the module we begin learning about variable scope, which is a consideration when writing functions where the definition of a variable can be local or global based on the code writers intention.

**Labs:**

Lab06-A

Text

Description automatically generated

This script works first by assigning empty variables intNumA and intNumB and then asking for input to create values for those two variables. It then defines 4 functions using the parameters value1 and value2 as arguments (?). the results of each function are defined by result, which is equal to the type of mathematical operation we want to happen. The results are then assigned to variables Sum1, Dif1, etc. finally, these functions are called, and the input variables of intNumA and intNumB are passed in as arguments. The results are assigned to the aforementioned variables, and the results are printed with supporting text.

Lab06-B

Text

Description automatically generated

This script keeps the variables intNumA and intNumB and adds a list variable that is empty for now. This will contain the result of the function once it performs work on the variables. The function works by assigning the results of various math operations to variables, and then returning those variables in a tuple. The contents of that tuple are saved in the result variable, and then that variable is unpacked using two methods, one by using the .format function to print the indices of the tuple one at a time, and the other by assigning each item in the tuple to a variable, and then printing each variable with supporting text and formatting.

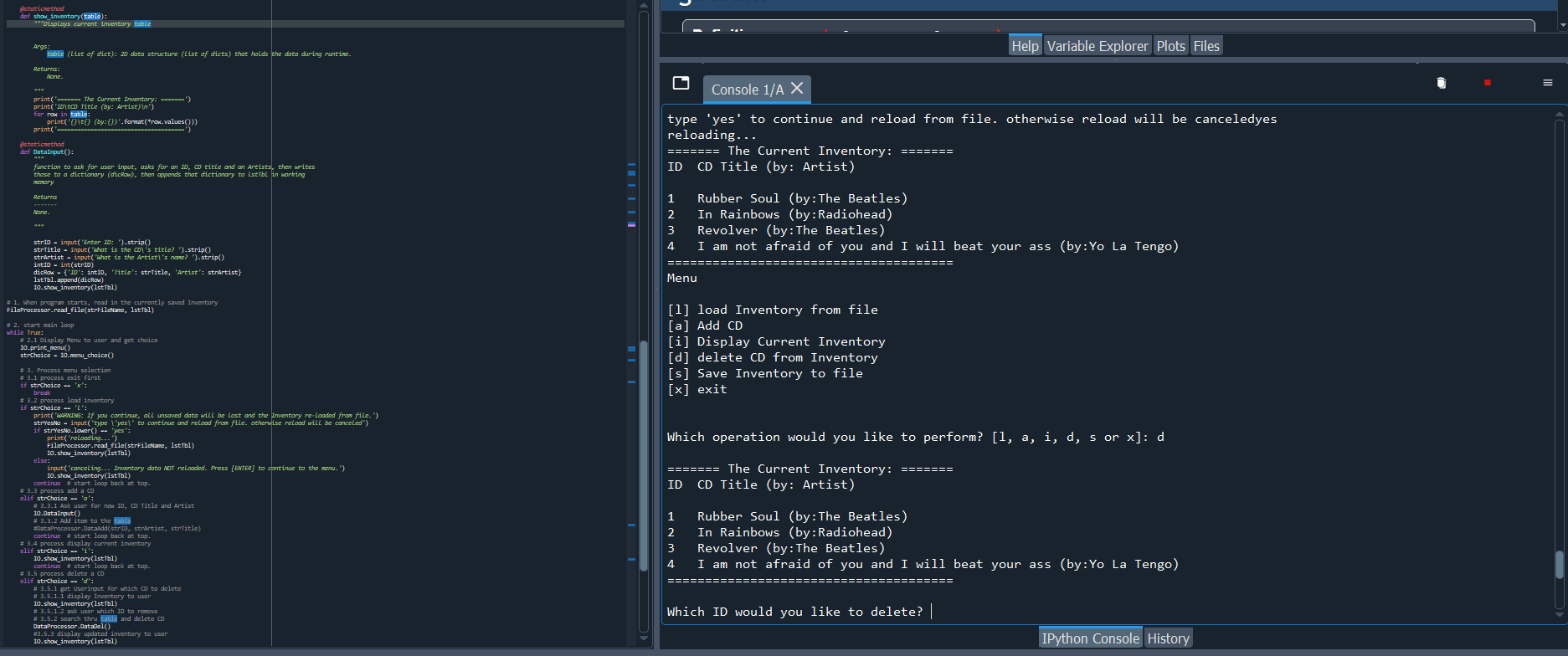
Lab06-C

Text

Description automatically generated

This script works much the same as the others, except it creates a class of functions so that when you call the class and specify the function you want, it uses your variables and performs the specified function. A major difference is the addition of doc strings which are strings that tell the user what the code does without them having to decipher the code itself. I use the technique of assigning the output of each function to a variable here, but I could just as easily index them or assign the result of the function to a variable in the function, and then ask the script to print each item in the index for the function.

Assignment 06



Text

Description automatically generated

This assignment was challenging in a new way. Firstly, it was difficult to remember where I was in the script, for two reasons. The first being that the doc strings create huge lines of text between function names and the rest of the code so that I was often scrolling up and down just trying to remember what I was looking at. Another reason was that since the functions are usually defined first, the usual top to bottom order of the script was not quite what I was used to. I also found it hard to get oriented to someone else’s code, so it took a while to understand what to do. To complete this assignment, I first mapped out the functions that I needed to create by checking all the TODO’s in the script. I then moved each block of code into a function one by one, testing it each time to make sure that it functioned in a scratch document before I added it back into the main one. I then would add the function call to the main script and test it individually to make sure it worked. The most common error I found was that python would see variables as undefined. This threw me but then I saw why the module talks about global vs local variables and that when you have variables defined in the parent and in the child, they are actually not the same thing. I usually fixed this by making sure that variables used in the function were also consumed in the function, unless they were truly global variables, like the lstTbl or File Name variables.

**Summary:**

This module taught us a lot about writing code more efficiently. It is far simpler to write generic code that can accept any variable for task you want to accomplish than to write blocks of code every time you want that task accomplished. This does not come without complications however, so it is important to understand variable scope, as well as how to organize the blocks so that information can be passed in between functions. Another important piece of this module was learning how to format and use doc strings. These are usually unique to an organization but serve the essential purpose of being embedded parts of the code that describe what is happening and how a function functions. In this way a new user can quickly decipher the purpose and function of blocks of code and how the script as a whole works together.