Ryan Shipman

Nov 24, 2020

IT FDN 110 A

Assignment06

TheFunctionFiasco.py

**Introduction.**

I will start out by saying, what a challenge this assignment has been! This has been the toughest assignment thus far for me to complete. I do believe I am about 85% confident in the concepts for this module and this assignment. I will quickly go over some of the steps I took to complete certain portions of this assignment. Bear with me as I am not 100% solid with this module's knowledge, but I am close! The beautiful thing about programming is it doesn't matter where you are in your journey there is always more to learn. Lets jump in.

**Global Variables**

The first thing you will notice on the starter file is that the global variables are pre defined at the top. See figure 1 for reference.

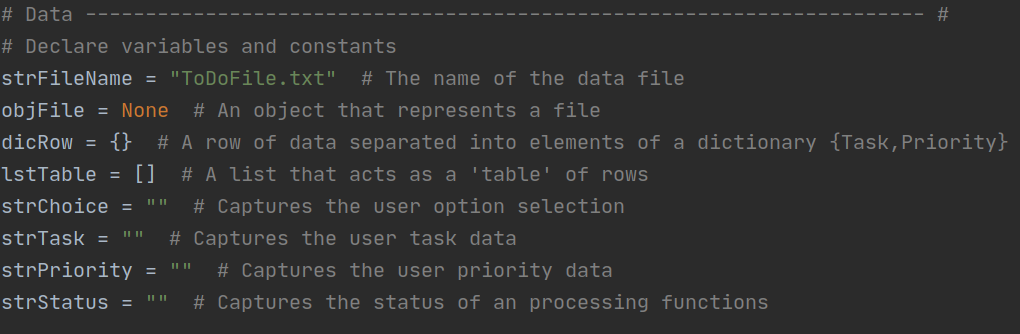


Figure 1.

**Class and Functions**

Next you will notice the code has a class of all the processing functions in one tidy group and the IO code in another. Taking the processing code, IO/User code and splitting it from the main body of the script can be beneficial for organizational reasons. The code will be much easier to work with and decipher when it is like this. In order to have the different parts of the code split from the main script in python you have to use and call a function and they can be a bit tricky to understand.

A function in python is a block of code which only runs when it is called. In this case we will be calling all the functions from within the while loop in the main body of the script. You can pass data (known as parameters) into a function. also functions utilizes its own set of variables which don't work outside of the functions itself. Next we will take a look at a function that was used in completing assignment 06. See figure 2. For reference of a function used in assignment 6.

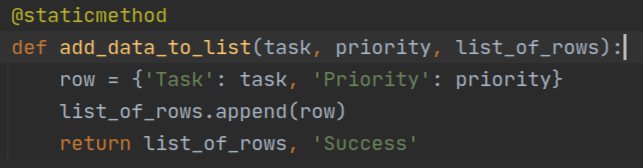


Figure 2

So if we were to call this function it would look something like this. Processor.add\_data\_to\_list(strTask, strPriority, lstTable)

Notice that we call the class.function(and pass in the values as parameters here).

This is a very basic concept for functions. Notice in figure 2. It is essentially a simple dictionary row being appended into a list of data and it will only do this if you call the function and pass in the values you want to add to the dictionary row. Also take note that the variables inside of the function are not the same as the global variables of the main script.

Now that you have called the function it has returned data back to you and in order to use this data in the main you would do so by using tuple unpacking. In this case now that our function has run its code and returned us some data we then unpack that data like so.



**Summary**

This was a quick and dirty explanation on my thought processes behind adding some of the code to the script for this assignment. I hope that what I wrote makes sense and helps you on your journey in Python. Thanks for reading.