Samuel Nunn

May 22, 2020

Foundations of Programming: Python

Assignment 06

<https://github.com/stnunn/IntroToProg-Python-Mod06>

Working with Functions in Python

# Introduction and Assumptions

In this assignment, we will use functions to read/load “Task” and “Priority” values from a text file, and store them as Dictionary “rows” within a List “table” object. We will present the data to the user, then we will give the user five options: (1) Add a new Task, (2) Remove an existing Task, (3) Save data to File, (4) Reload Data from File, and (5) Exit program. This is similar to what we accomplished in Assignment 5, but we are now using functions for better organization and reusability.

# Instructions

1. Start a new PyCharm project called “Assignment06” within the C:\\_PythonClass\ directory. Copy the Assignment06\_Starter.py file into this directory. Modify the Header section with your information, as seen in Figure 1.

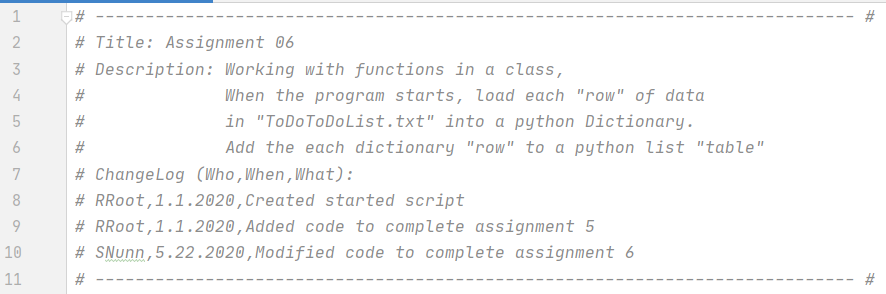


Figure . Script Title and Header Section

1. In the “Data” section, declare your variables and constants, as seen in Figure 2.

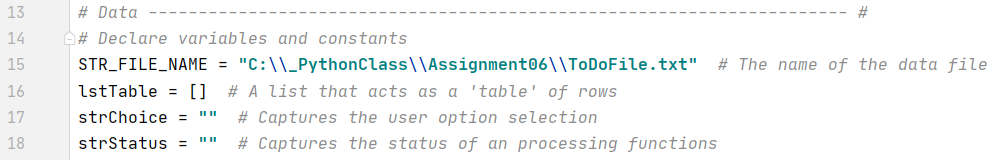


Figure . – Data. Declare variables and constants.

1. In the “Processing” section, create a “class” called “Processor”. Then, indent and create all the “Processor” class functions., as seen in Figure 3. The “class” allows us to better organize our functions. Adapt the code from Assignment 5 to work within the new functions, as seen in Figure 3.



Figure . Process the Data

1. In the Presentation (Input/Output) section, create a new class called IO. Adapt the code from Assignment 5 to work within the new functions, as seen in Figure 4 and Figure 5.



Figure - IO Class

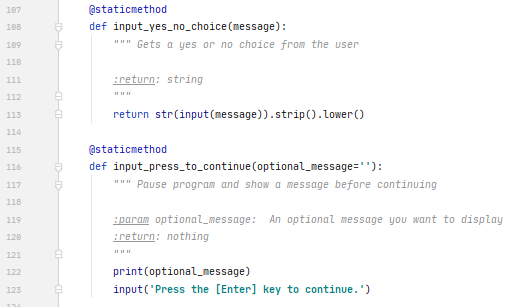


Figure . IO Class

1. In the Main Body of the Script, first use the Processor.read\_data\_from\_file() function to call the function which will load all the data from the file. Note the use of the “class” prefix “Processor”, then the “.”, then the function name. We continue to use conditional logic and calling our IO functions to determine the user choice. Based on user choice, we prompt for inputs and pass the parameters/arguments back to the functions via function calls. This makes our Main Body much more readable, and allows us to reuse our functions, as seen in Figure 6 and Figure 7.

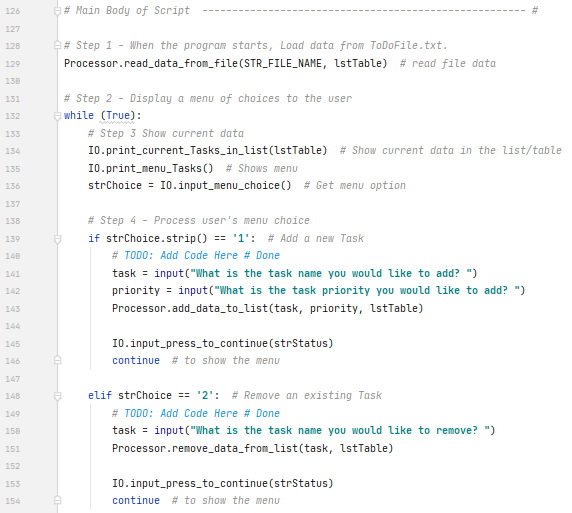


Figure . Main Body

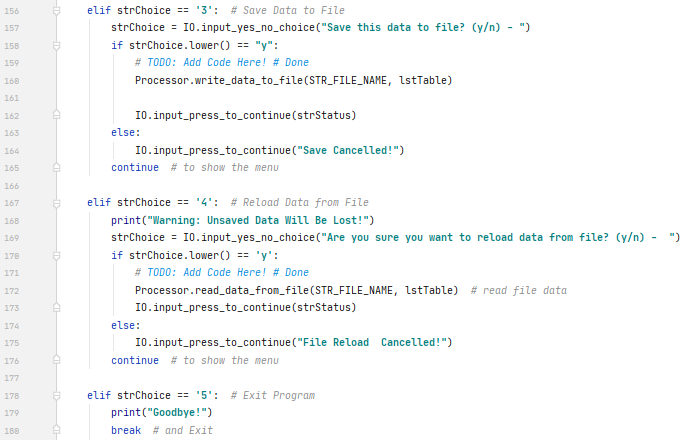


Figure - Main Body (cont')

1. Run the program in the command prompt, as seen in Figures 8 – 11, below.

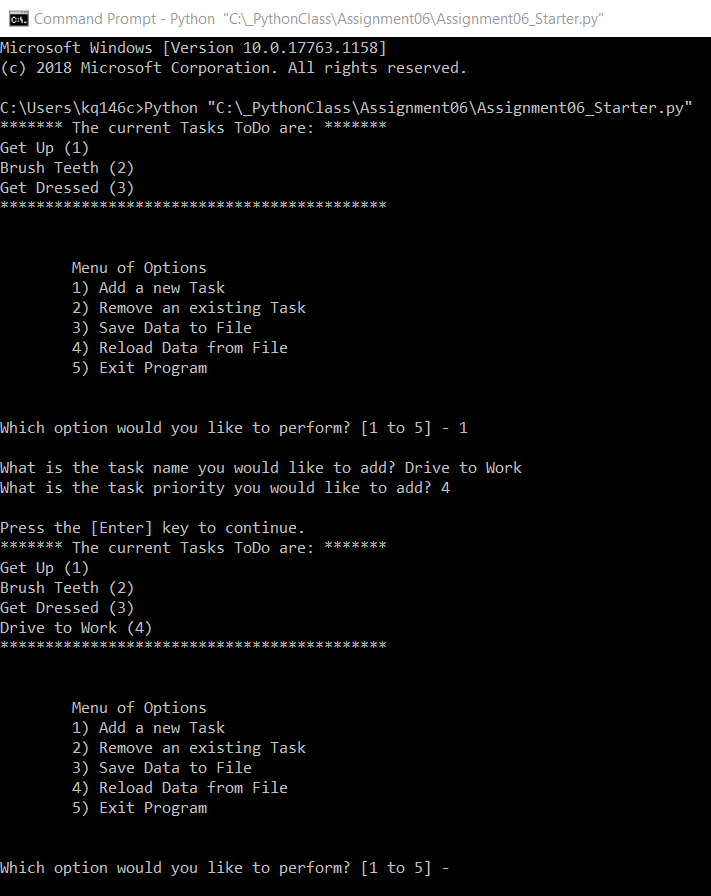


Figure – List is Loaded and Displayed on Program Start. Option 1 Selected.

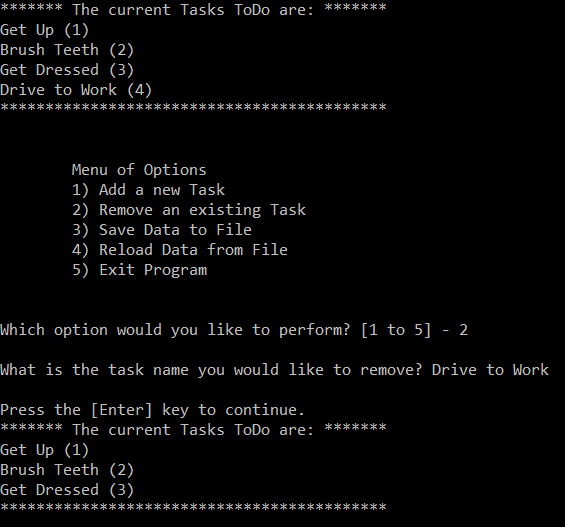


Figure - Option 2

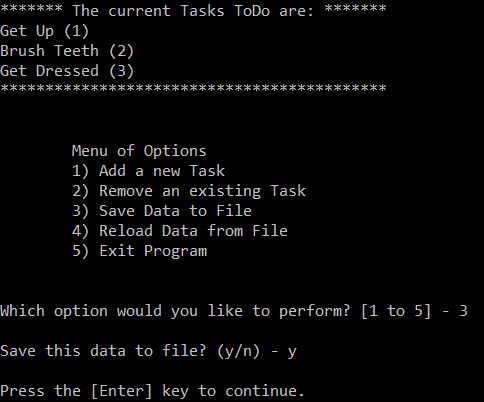


Figure - Option 3

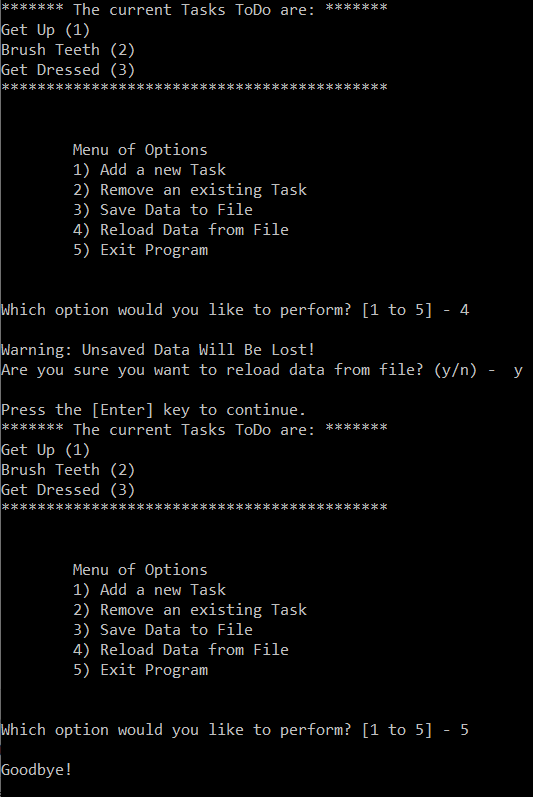


Figure – Option 4 and Option 5

1. Open the .txt file to see the Tasks and Priorities which have been saved, as seen in Figure 12.

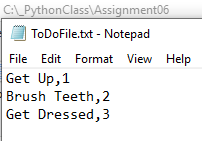


Figure . Open ToDoFile.txt to see the saved output

# Summary

In this assignment, we learned how to use functions to read/load “Task” and “Priority” values from a text file, and store them as Dictionary “rows” within a List “table” object. We presented the data to the user, then we gave the user five options: (1) Add a new Task, (2) Remove an existing Task, (3) Save data to File, (4) Reload Data from File, and (5) Exit program. This is similar to what we accomplished in Assignment 5, but we used functions for better organization and reusability.

Lastly, writing the results to a .txt file allowed the user to save the results to a file, so it could be reviewed at a later time.