Andrew Meng

IT FDN 100

11/21/22

Assignment 6

Utilizing Functions to Edit a Python Script

# Introduction:

In this assignment, we were provided a starter script resembling last week’s assignment, that I added functions to optimize and organize the script.

Methods:

A python function is a method that can group multiple statements, and only runs when the function is called. Functions can have parameters, data that can be run through a function, and can also be referred to as an argument. In this assignment, we were provided with a script like a previous assignment, which coded a to do list capable of recording tasks and their priorities. This script could also remove and save the recorded data to a specific text file. Functions were present in the starter file for this script, namely function document headers that denote notes in a docstring. An example of this is figure 1, where these notes can help organize and simplify the role of the overall function. These headers are easier to comprehend than long lines of code and can be read much quicker to get the general idea of the code.

Text

Description automatically generated

Figure 1: example of a function document header

Likewise, another method for utilizing functions we learned this week was classes. Classes can serve as a function for a function, a method of grouping up multiple functions, variables, or constants.

Text

Description automatically generated

Figure 2: an example of a class

Text

Description automatically generated

Figure 3: an example of a different class in the same assignment

In this assignment, as demonstrated by figures 2 and 3, these functions were grouped into different classes, IO, and processor, that performed different objectives. Class IO focused on input and output tasks, including printing the menu, and providing options to the user. Class processer focused on processing tasks, namely adding data to dictionary rows, reading, and removing data, and writing data to a file. The process of using classes and functions allows the coder to organize their code into distinct categories that can be read in detailed sectioned tasks. I was asked to add functions to this code to organize this code while ensuring that the code ran properly. I edited this code, adding code to define functions in both class IO and processor. For example, in the starter script, in class processor, I added code to open a file, used a for loop to dictate dictionary rows, and closed it. I followed the param requests in each function, added code to ensure that the code would work, and defined each function.

# Errors:

Like the last assignment, I struggled with the process of editing another person’s code as I had to think about their thought process and how to edit their code. Understanding how to edit the functions was interesting but proved difficult at first. I think being able to understand last week’s assignment helped greatly as I was able to effectively and efficiently determine what I had to do for each specific function, and then implement the code that would make it work. With constant troubleshooting to test whether my ideas were functioning as intended, eventually the code would run smoothly.

# Conclusion:

In this week’s assignment, we built off last week’s process and organized data into numerous class and function groups. Evidently, I could see how organizing the data allowed for an easier process in understanding the various roles and tasks performed in this assignment, and sectioning categories of classes allowed for an optimal method to search through the code.

# Results:

Figure 4: the code running in PyCharm:

Text

Description automatically generated

Figure 5: code cont:

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Figure 6: text file containing to do list data.