Bryan Mon

August 09, 2021

IT FDN 110 A Su 21: Foundations Of Programming: Python

Assignment 05

# Lists and Dictionaries

### Introduction

This week’s assignment was a continuation on our Lists lessons where we also expanded into working with Dictionaries. Our task was similar to week 4’s assignment, we needed to build an interactive menu program which allowed a user to review, add, delete, and save a list of tasks and their priorities, to a txt file.

### Creating the Program

The script this week had an added challenge. We were provided with a starter script which had some basic code already in place (figure 01). The challenge was to learn to work with code that was not our own. This meant that I had to continue building off of work that had different formatting and variable definitions than what I was accustomed to.

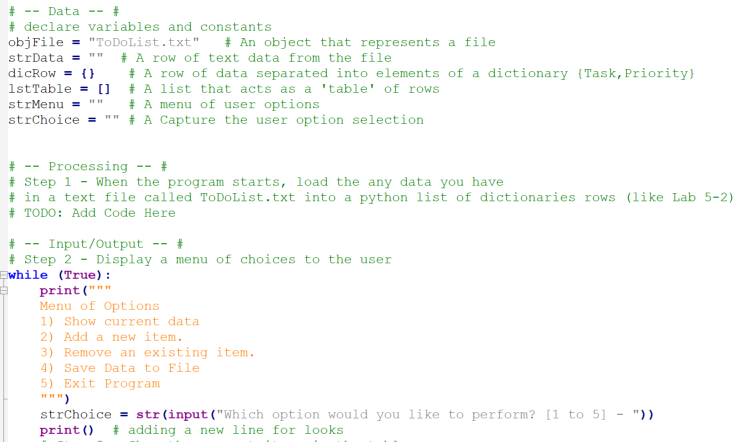


Figure 0 - the starter script. Variables were pre-defined and the pseudo-code was already in place

Nevertheless, I carried on with the exercise. We were to incorporate dictionary collections with this assignment. A dictionary in Python works similarly to a List, but where Lists can store a sequence of objects in a certain order, a dictionary relies on a key-value store. In addition, dictionaries are denoted by the curly bracket {}, as opposed to lists square brackets [].

Our first step was to read the data from a txt file and load the content into memory. Last week’s assignment had us perform a similar task using lists, so I used some of my old code to do the same only with a dictionary. I loaded the data from the txt file, split the rows by the commas, loaded the data into my dictionary where I had to specify the keys we were using (see figure 02). I enhanced this portion of the project by adding a try / except block to check whether a txt file even existed. This feature can be pretty handy for a user running the program, as it could give some detailed information should this step fail.

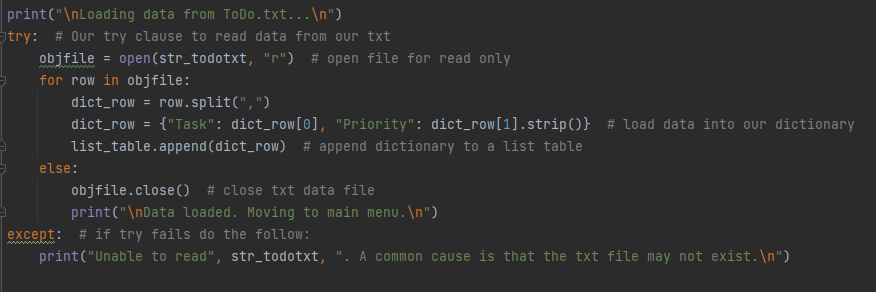


Figure - Opening and loading the data from the txt file to memory. The dictionary is denoted with the {}

Since we would be working with the data loaded on memory, we needed a way to display the data to the user through the menu. Menu option 1 was to read the table data and print it to the screen. I used the same code from last week to perform this task, but I added a feature to check if the list table was empty so that I could let the user know that there was no data to display.

Option 2 was to allow the user to add rows of data to our table. I spent some extra time on this step to try and improve beyond the basic requirements of the task. I added conditional if statements which would check whether the user had specified a ‘High’ or ‘Low’ priority to their task, and return an error if they did not (figure 03). This added constraint would keep the user from accidently adding the wrong type of data to the table.

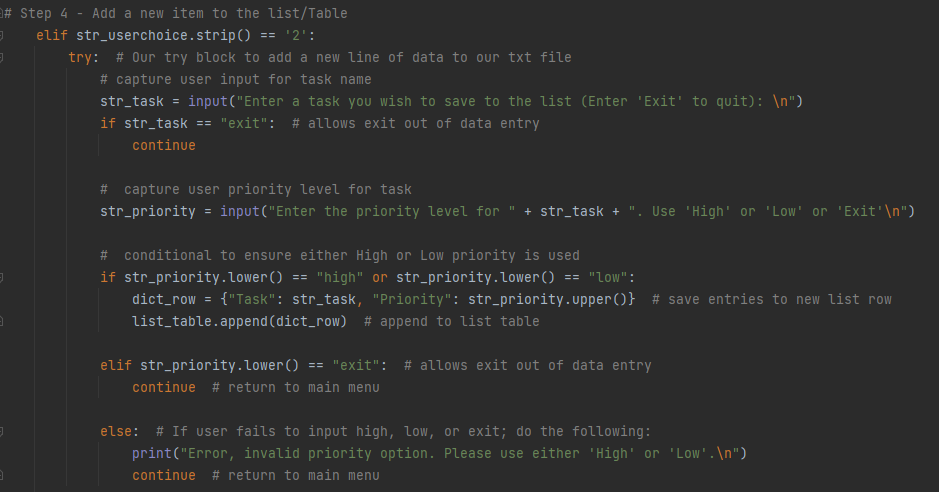


Figure - Adding data to the table uses a series of if / elif conditions to improve the user experience

Our next task was to allow the user to remove a row of data from the table. Since we didn’t have this option on our last assignment, I had to write this code from scratch. To perform this task, I first asked a user to enter the name of a Task they wished to remove, and then I had a For loop check each row in the table for a matching value. Once found, I simply removed the row using the list.remove() method and then printed the updated list to the user. If the entry could not be found, I provided an error message to the user, printed the contents of the table, and returned them to the main menu.

The last task for this project was to save the data to a file. Unlike our last assignment, I did not want to append the changes to the existing data file, since all of our changes were done to the data in memory and the program would constantly need to keep accessing the txt file. Instead, I simply overwrote the existing file with a new file of the same name. A simple solution to the problem, and it worked as intended.

### Conclusion

I ended up spending more time on this assignment compared to time spent on any of my previous assignments. I enjoyed the challenge, as this was a great opportunity for me to incorporate all of the skills I’ve learned up to this point into an actual working program. After completing this assignment, I feel confident in my understanding of these Python concepts, and look forward to working with Functions next week.