Dominic Ferenczy  
May 3, 2021  
Foundations of Programming: Python  
Assignment05  
[uwferenczy/IntroToProg-Python: This Rep. will be used for reviewing homework files (github.com)](https://github.com/uwferenczy/IntroToProg-Python)

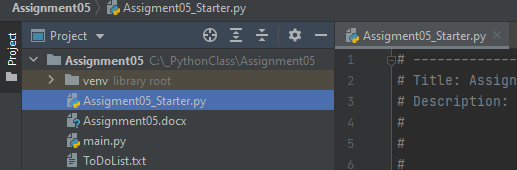
How to Develop a ToDo List Application

# Introduction

This document details how I applied what I have learned in module 5 of the Introduction to Programming class to modify an existing script that manages a “ToDo list.” Topics covered in this document include how to add an existing script to a new project, how to load columns into a Python Dictionary object, how to display data from a list, how to add a new row of data to a list, how to remove a dictionary object from a list, how to add a list of dictionaries to a text file, how to perform error handling, how to write a list containing dictionary items to a txt file, and how to test the code.

# Creating a PyCharm Project Using an Existing Script

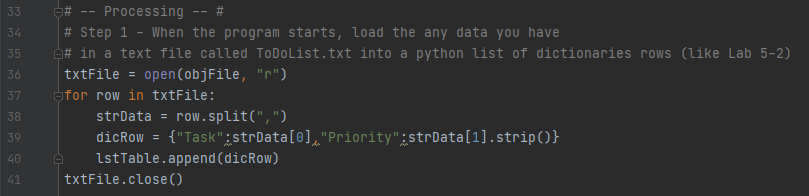
I created a new Python project in PyCharm that uses the \_PythonClass\Assignment05 folder as its location and added the starter file “Assignment05\_Starter.py” to the project as shown in Figure 1.

   
Figure 1: PyCharm showing Assignment05\_Starter.py file in correct folder location

I updated the header to include my name, the date I added the starter file and what I did.

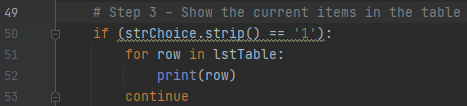
# Loading Columns from a File into a List

I started by creating a new variable for the text file and setting it to open and read the objFile. After opening it, I iterated through each row, split it and created a new dictionary by passing the data elements from the row into the dicRow variable and setting them equal to “Task” and “Priority”. I then appended each dictionary into the lstTable variable and closed the text file. The resulting code can be seen in figure 2.

  
Figure 2: PyCharm showing the Processing portion of the code

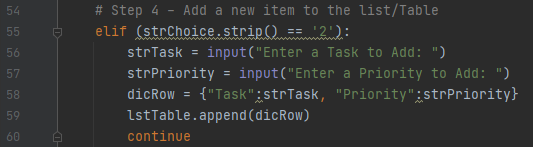
# Displaying the Contents of a List

Because the contents of the list was loaded with data from the text file right after the application started, I was able to print each iteration (row) of data in the lstTable variable. The resulting code can be seen in figure 3.

  
Figure 3: PyCharm showing code to print rows of a list

# Add a New Row of Data to a List

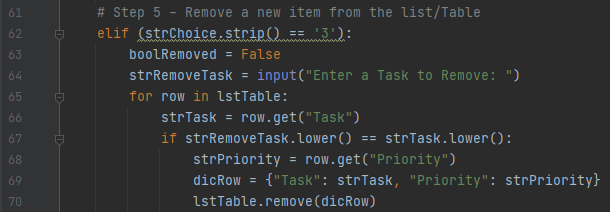
I started by setting the strTask and strPriority variable equal to an input from the user. I then set the dicRow variable “Task” and “Priority” equal to the inputs gathered from the user and appended the resulting dictionary into the lstTable using the .append function. The lstTable is a list of dictionary objects, and therefore I had to add the inputs from the user into a structured dictionary format that matched the other dictionary objects that were read into the program in the processing portion of the script. The resulting code can be seen in figure 4.

  
Figure 4: PyCharm showing code to add new data to a dictionary object and append it to a list

Note: I added a print statement after appending to the table. This is not reflected in figure 4.

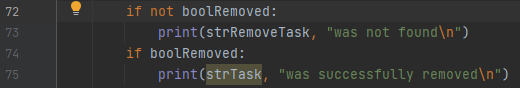
# Remove an Object from a List

This was the most difficult portion of the program in my opinion. I started by creating a new variable that collects the input for which task to remove from the user. I then iterate through each row in the lstTable and set the strTask variable equal to the “Task” in the dictionary for that row using the .get function. From there I check to see if the strTask and the strRemoveTask are equal (after casting both to lowercase using the .lower() function). If they are equal, I set the strPriority variable equal to the “Priority” of that of that row’s dictionary object and set the dicRow variable equal to the dictionary object. I then remove the dictionary object from the lstTable variable. Initially, I tried to use the .remove function and pass row into it (from the for loop), but it did not work. Re-creating the dictionary for the row if there was a match from the input of the user fixed this issue. The resulting code can be seen in figure 5.

  
Figure 5: PyCharm showing code to remove a dictionary object from a list

# Perform Error Handling

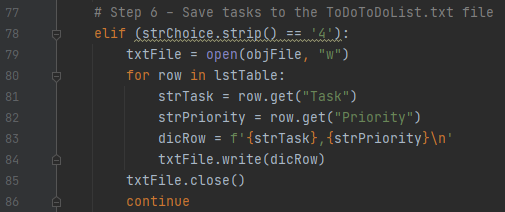
I wanted to provide a feedback loop to the user that would display if the requested task was successfully removed from the list or not. To accomplish this, I started off by adding a boolRemoved variable before the for loop and setting it equal to False. I set this equal to True inside of the nested if statement of the for loop after I removed the dictionary object from the lstTable list. I then added two if statements based on the boolRemoved variable, one that displayed the task was not found, the other displaying that the task was successfully removed. I decided to print the strTask variable if it was successfully removed and the strRemovedTask if it was not to accurately display the case of each. The resulting code can be seen in figure 6.

  
Figure 6: PyCharm showing the status of the removal of the dictionary object from the list

Note: I removed the breaklines from the end of the print statement because it added an extra line between the Menu. This is not reflected in Figure 6.

# Writing a List of Dictionaries to a Text File

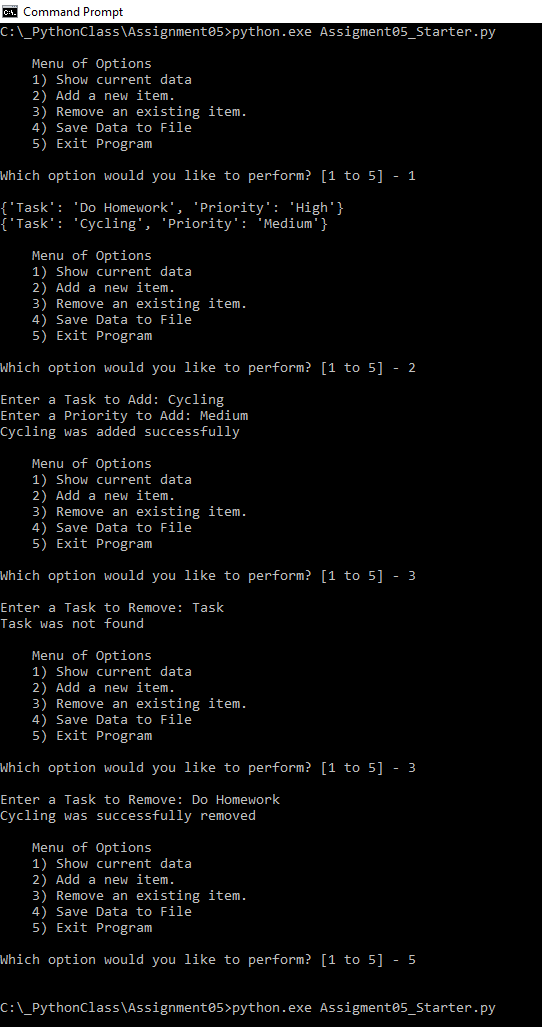
I started by setting the txtFile variable to open the obFile in write mode. I used write mode to ensure that the data from the list (combining the existing data with any changes by the application) overwrites the original data each time it is saved. I then used a for loop to iterate through each dictionary in the lstTable list, set the dicRow variable equal to an fstring that contained the strTask and strPriority from the dictionary, separated by a comma with a new line at the end, and then wrote the resulting string to the txtFile. I made sure to close the txtFile after all iterations were completed. The resulting code can be seen in figure 7.

  
Figure 7: PyCharm showing the code for saving the updated tasks to the text file

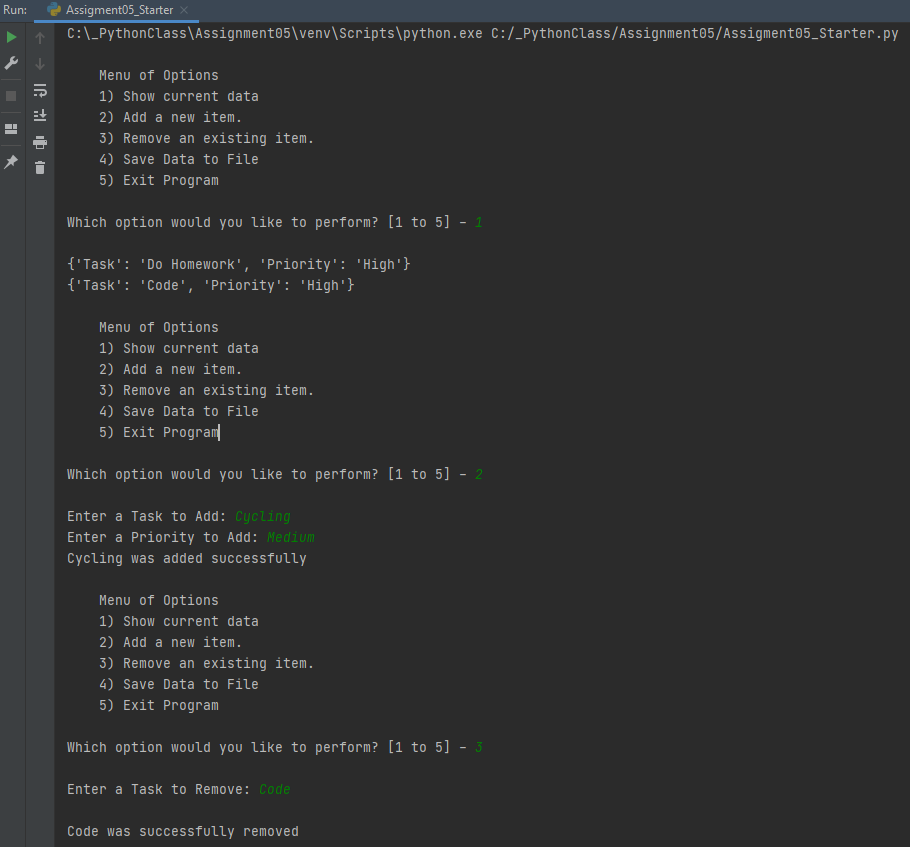
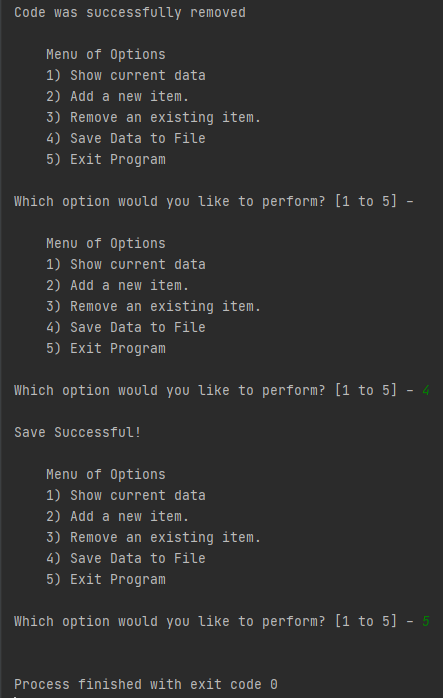
Note: I added a print statement after closing the file to display that the save was successful. This is not reflected in figure 7.

# Testing Application

I tested the application in command prompt as shown in figure 7. It ran correctly in command prompt and matched the assignment requirements.

  
Figure 7: Command Prompt showing program running correctly

I then tested the application in PyCharm as shown in figure 8. It ran correctly in PyCharm and matched the assignment requirements.

  
  
Figure 8: PyCharm showing program running correctly

I then opened the ToDoList.txt file to verify that the new items were added as shown in figure 9. The items were added correctly.

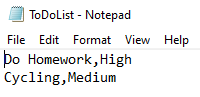


Figure 9: Notepad showing items added correctly

# Summary

This document captured what I learned in module 4 of the class. I documented the steps I took to add an existing script to a new project, load columns into a Python Dictionary object, display data from a list, add a new row of data to a list, remove a dictionary object from a list, add a list of dictionaries to a text file, perform error handling, write a list containing dictionary items to a txt file, and test the code.