**Samir Abad**

**May 15, 2022**

**IT FDN 110A Sp. 22**

**Assignment 05**

**Github Link-** [mailsamir/Python-code: Create and promote python code (github.com)](https://github.com/mailsamir/Python-code)

**Task and Priority Script**

**Introduction:** In this paper I am going to talk about how I created a new python script that asks the user to add or remove a task and the corresponding priority from a list of tasks and priorities. The "ToDoList" file will contain two columns of data, "Task" and "Priority." Load the columns into a Python Dictionary object. Each dictionary object represents one row of data, and these rows must be added to a Python *List* object to create a table of data. A starting template was provided and the missing code was added inorder to execute the option selected by the user.

The following information shows the detailed steps demonstrating how I went about writing the script.

**Step0:** Header was updated to reflect the name, date and the update details were provided.

**Step1:** Load the data in a text file ‘ToDoList.txt’

Error handling was utilized to check for if the file was found. If the file is present then the file was opened with a read option. For every row in the file the split functionality was used to split each row of data in order to separate the tasks and priorities. The separated data was then stored into a dictionary using a Key/Value pair. Subsequently each dictionary row was added to a list table.

If the file was not found a print function showed a message saying “File not found”.

The code for Step 1 is shown below-

# -- Processing -- #  
# Step 1 - When the program starts, load the any data you have  
# in a text file called ToDoList.txt into a python list of dictionaries rows (like Lab 5-2)  
# *TODO: Add Code Here*try: # Error handling introduced to check for the file  
 strFile = open(objFile, "r")  
 for row in strFile:  
 t, p = row.split(",")  
 dicRow = {"Task": t, "Priority": p.strip()}  
 lstTable.append(dicRow)  
 print(dicRow["Task"] + ',' + dicRow["Priority"])  
 strFile.close()  
except: # if file not found  
 print("File not found. New file will be created when you save.")

**Step2**: Display a menu of choices to the user

No code changes were made to Step2. The Step2 code is shown below. A menu of selections are shown to the user using the print built in function. A while loop was used to keep asking the user for input until the program is Exited.

# -- Input/Output -- #  
# Step 2 - Display a menu of choices to the user  
while (True):  
 print("""  
 Menu of Options  
 1) Show current data  
 2) Add a new item.  
 3) Remove an existing item.  
 4) Save Data to File  
 5) Exit Program  
 """)  
 strChoice = str(input("Which option would you like to perform? [1 to 5] - "))  
 print() # adding a new line for looks

**Step3**: Show the current items in a table

If the user selects option 1- Display the current data. A for loop was used to print each row in the list table and show each task and its priority.

# Step 3 - Show the current items in the table  
if (strChoice.strip() == '1'):  
 # *TODO: Add Code Here* print("Task" + '|' + "Priority")  
 for row in lstTable:  
 print(row["Task"] + '|' + row["Priority"])  
 continue

**Step4:** Add a new task to the list table

If the user selects option 2- Use the input function to ask the user for a Task and Priority. Once the user provides those inputs store the corresponding data in a dictionary and then append that row of dictionary data into a list table. Print function was used to check if the new data was captured.

# Step 4 - Add a new item to the list/Table  
elif (strChoice.strip() == '2'):  
 # *TODO: Add Code Here* strTask = input("Enter a new task:")  
 strPriority = input("Enter the priority:")  
 dicRow={"Task":strTask, "Priority":strPriority}  
 lstTable.append(dicRow)  
 print("Task" + '|' + "Priority")  
 for row in lstTable:  
 print(row["Task"] + '|' + row["Priority"])

**Step5:** Remove an item from the list table

If the user selects option 3, then the user provides the input on the task to be removed. An if-else condition is used to check if the task entered by the user is present in any of the rows of the table list. If the task is present, then that row is removed. If the task is not found, then a message is printed saying “Task not found”. The code is shown below.

# Step 5 - Remove an item from the list/Table  
elif (strChoice.strip() == '3'):  
 # *TODO: Add Code Here* strtaskR = input("Enter the task to be removed:")  
 for row in lstTable:  
 if row["Task"].lower() == strtaskR.lower():  
 lstTable.remove(row)  
 print("Task removed")  
 print(lstTable)  
 else:  
 print("Task not found")  
 continue

**Step6:** Save task to a ToDoList.txt file

If the user selects option 4, then the text file ToDoList.txt is opened and the write functionality is used to write each row of data in the table list to the file and subsequently the file is closed.

# Step 6 - Save tasks to the ToDoToDoList.txt file  
elif (strChoice.strip() == '4'):  
 # *TODO: Add Code Here* strFile = open(objFile, "w")  
 for row in lstTable:  
 strFile.write(row["Task"] + ',' + row["Priority"] + '\n')  
 strFile.close()  
 print("Tasks successfully saved!")  
 continue

**Step7:** Exit the program

If the user selects option 5, then the break functionality is used to Exit the program.

# Step 7 - Exit program  
elif (strChoice.strip() == '5'):  
 # *TODO: Add Code Here* print("You have exited the program!")  
 break # and Exit the program

**Text

Description automatically generated**

**Fig1: Program output for Option 1 shown within pycharm**

**Text

Description automatically generated**

**Fig2: Program output for Option 2 shown within pycharm**

**Text

Description automatically generated**

**Fig3: Program output for Option 3 shown within pycharm**

**Text

Description automatically generated**

**Fig4: Program output for Option 4 shown within pycharm**

**Text

Description automatically generated**

**Fig5: Program output for Option 5 shown within pycharm**

**Summary:**

Assignment5 expanded upon the previous assignment by introducing concepts of creating a dictionary and a list of dictionaries. The usage of while loop to continue processing, for loop to print relevant data from a List Table in addition to writing data in a text file were some of the concepts that I learnt in this assignment. Also, I learnt how to add and remove items from an existing dictionaries. This program also allowed me to explore PyCharm by writing and testing the program in the PyCharm development environment. Overall, this assignment was very helpful in introducing basic python scripting to a novice programmer.