Kate Beverly

August 10, 2022

Foundations of Programming: Python

Assignment 05

https://github.com/kbev12/IntotoProg_Python

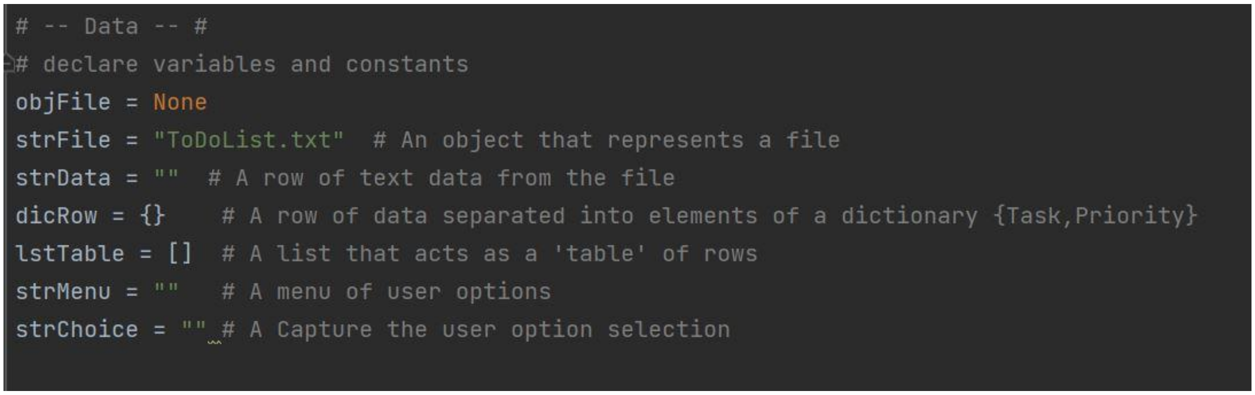
To Do Script

Introduction

For assignment 05 for the class Foundations of Programming: Python I worked on a python script that had been started previously. The script reads from a file to memory and then gives the user the option to display the information, add a task, remove a task, write to the file, and end the program.

Creating the Script

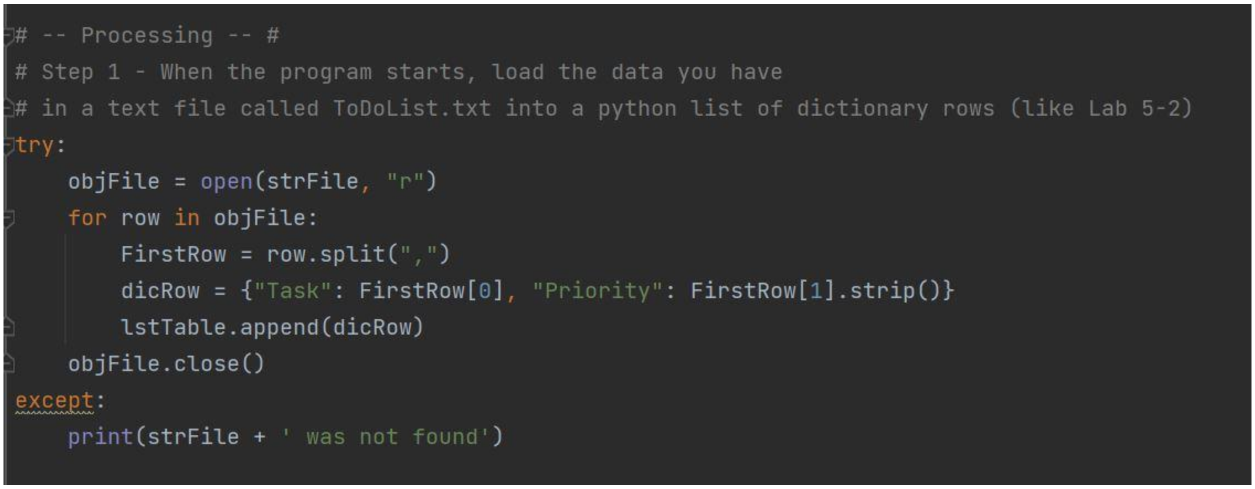
The script begins with declaring the variables (**Figure 1**). Most of the variables are empty except for strFile which is set to the name of the text file.



```
# -- Data -- #
# declare variables and constants
objFile = None
strFile = "ToDoList.txt"  # An object that represents a file
strData = ""  # A row of text data from the file
dicRow = {}    # A row of data separated into elements of a dictionary {Task,Priority}
lstTable = []  # A list that acts as a 'table' of rows
strMenu = ""   # A menu of user options
strChoice = ""# A Capture the user option selection
```

Figure 1. Declaring the program variables

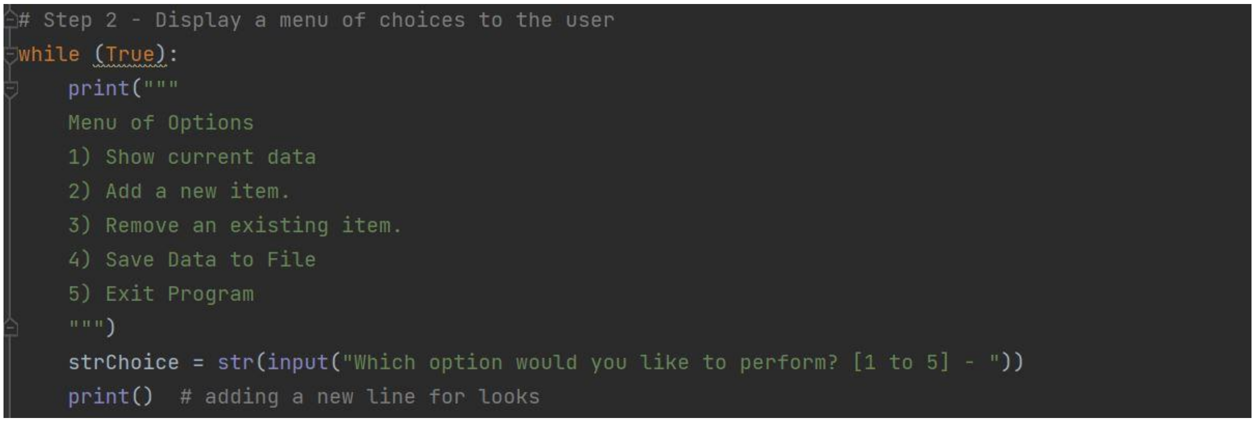
In step one (**Figure 2**) the script opens the text file in read mode. It splits each row by the comma and creates a dictionary identifying the first object as the task and the second as the priority. Each row is appended to a table and stored it in the system's memory.



```
# -- Processing -- #
# Step 1 - When the program starts, load the data you have
# in a text file called ToDoList.txt into a python list of dictionary rows (like Lab 5-2)
try:
    objFile = open(strFile, "r")
    for row in objFile:
        FirstRow = row.split(",")
        dicRow = {"Task": FirstRow[0], "Priority": FirstRow[1].strip()}
        lstTable.append(dicRow)
    objFile.close()
except:
    print(strFile + ' was not found')
```

Figure 2. Step 1 – Opening the file and saving to memory

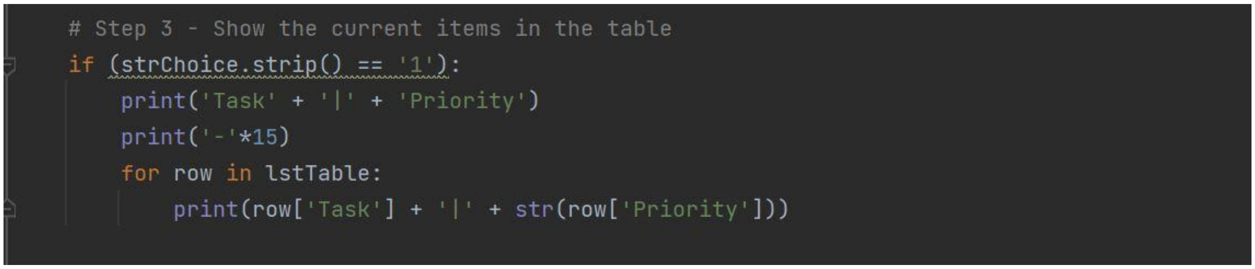
Step two (**Figure 3**) displays the menu of options to the user. It requests an input for which option and assigns their response to the variable strChoice.



```
# 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
```

Figure 3. Displays the menu and requests the user's choice

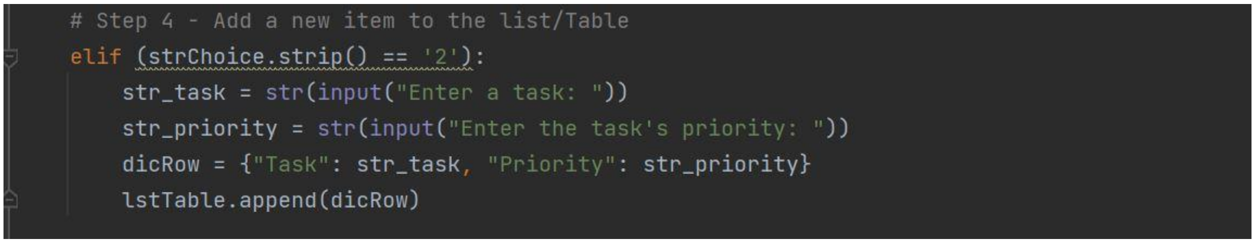
Step three (**Figure 4**) is triggered if the user selects option 1. Here the header Task|Priority is displayed with a line below. A for loop begins printing each row of tasks and their priority from the table in memory.



```
    # Step 3 - Show the current items in the table
    if (strChoice.strip() == '1'):
        print('Task' + '|' + 'Priority')
        print('-'*15)
        for row in lstTable:
            print(row['Task'] + '|' + str(row['Priority']))
```

Figure 4. Displaying the current items in the table

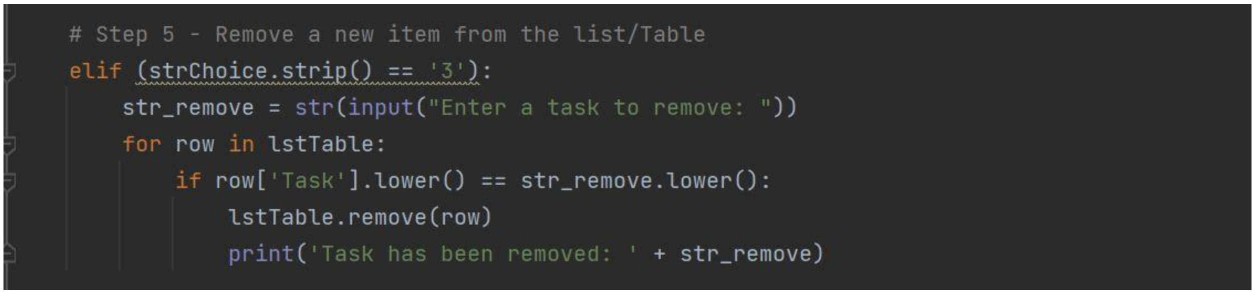
Step four (**Figure 5**) is triggered if the user selects 2. Here the script prompts the user to enter a task and it’s priority and assigning them to variables. They are added to the dictionary and appended to the table.



```
# Step 4 - Add a new item to the list/Table
elif (strChoice.strip() == '2'):
    str_task = str(input("Enter a task: "))
    str_priority = str(input("Enter the task's priority: "))
    dicRow = {"Task": str_task, "Priority": str_priority}
    lstTable.append(dicRow)
```

Figure 5. Adding tasks and their priority to the table in memory

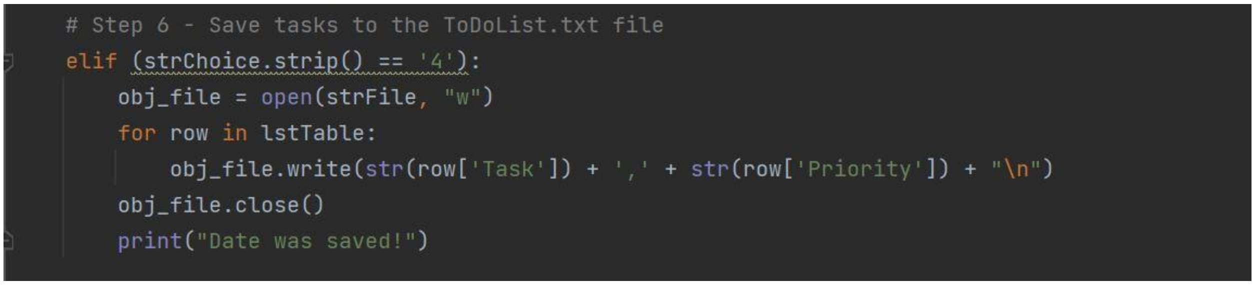
Step five (**Figure 6**) is triggered when the user selects 3. It requests what task the user would like to remove and assigns that to a variable. It loops through each row and if the task in lowercase matches the variable in lowercase the task is removed.



```
# Step 5 - Remove a new item from the list/Table
elif (strChoice.strip() == '3'):
    str_remove = str(input("Enter a task to remove: "))
    for row in lstTable:
        if row['Task'].lower() == str_remove.lower():
            lstTable.remove(row)
            print('Task has been removed: ' + str_remove)
```

Figure 6. Removing an item from the table

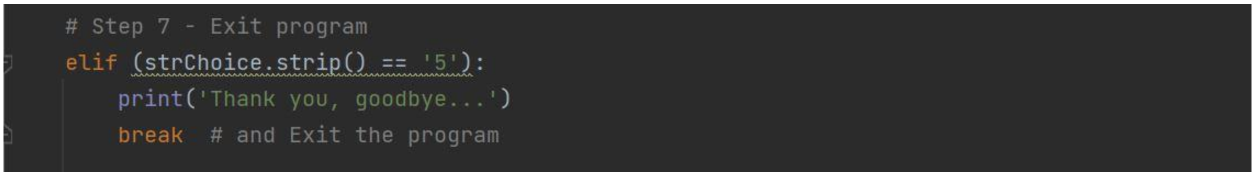
Step six (**Figure 7**) is triggered when the user selects 4. The text file is opened in write mode. The script loops though the table that is located in memory and writes each row with the task and the priority separated by a comma. The file is closed and the script displays “Data was saved”.



```
# Step 6 - Save tasks to the ToDoList.txt file
elif (strChoice.strip() == '4'):
    obj_file = open(strFile, "w")
    for row in lstTable:
        obj_file.write(str(row['Task']) + ',' + str(row['Priority']) + "\n")
    obj_file.close()
    print("Date was saved!")
```

Figure 7. Saves tasks to text file

Step seven (Figure 8) is triggered when the user selects option 5. This displays a goodbye message and breaks to exit the program.

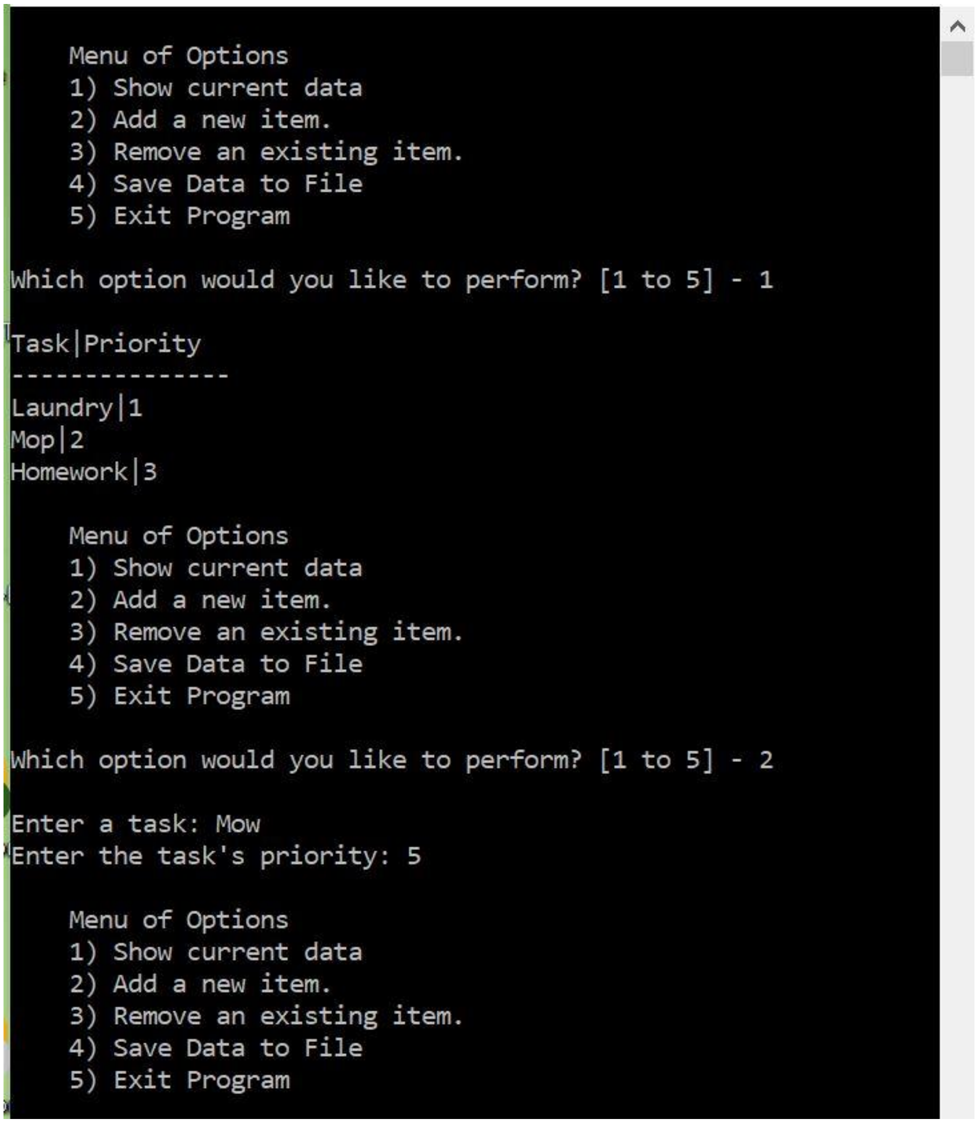


```
# Step 7 - Exit program
elif (strChoice.strip() == '5'):
    print('Thank you, goodbye...')
    break  # and Exit the program
```

Figure 8. Exiting the program

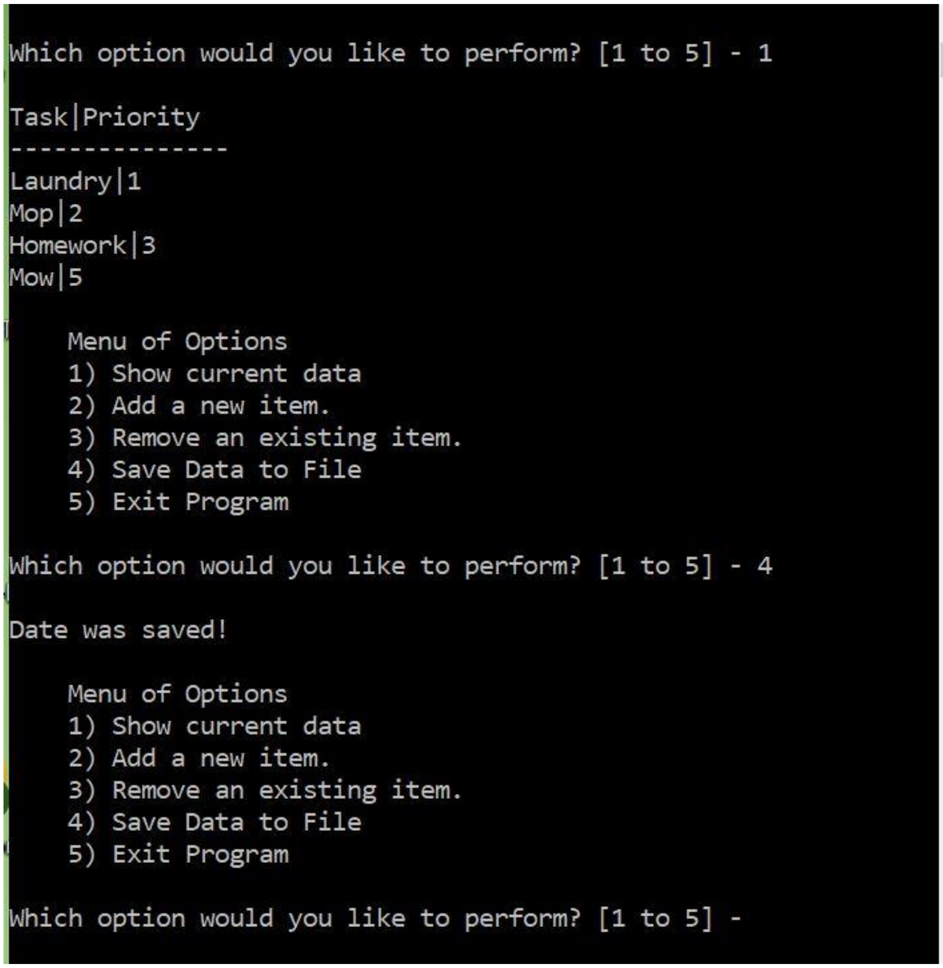
Testing

I tested the program in both PyCharm and in the command prompt and it successfully completed using both (Figure 9).



```

    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

Which option would you like to perform? [1 to 5] - 1

Task|Priority
---------------
Laundry|1
Mop|2
Homework|3

    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

Which option would you like to perform? [1 to 5] - 2

Enter a task: Mow
Enter the task's priority: 5

    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
```

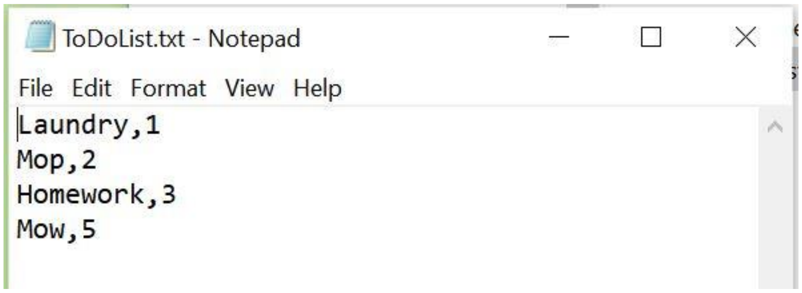
Figure 9. Testing the program



```
Which option would you like to perform? [1 to 5] - 1

Task|Priority
---------------
Laundry|1
Mop|2
Homework|3
Mow|5

    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

Which option would you like to perform? [1 to 5] - 4

Date was saved!

    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

Which option would you like to perform? [1 to 5] -
```

Figure 9 continued Testing the program

The script successfully ran and updated the text file ToDoList.txt (**figure 10**).



```
Laundry,1
Mop,2
Homework,3
Mow,5
```

Figure 10. Updated ToDoList.txt

Summary

For Foundations of Programming: Python fifth assignment I worked on a python script that was started previously to add in some functionality. The script reads from a file to memory and then gives the user the option to display the information, add a task, remove a task, write to the file, and end the program.