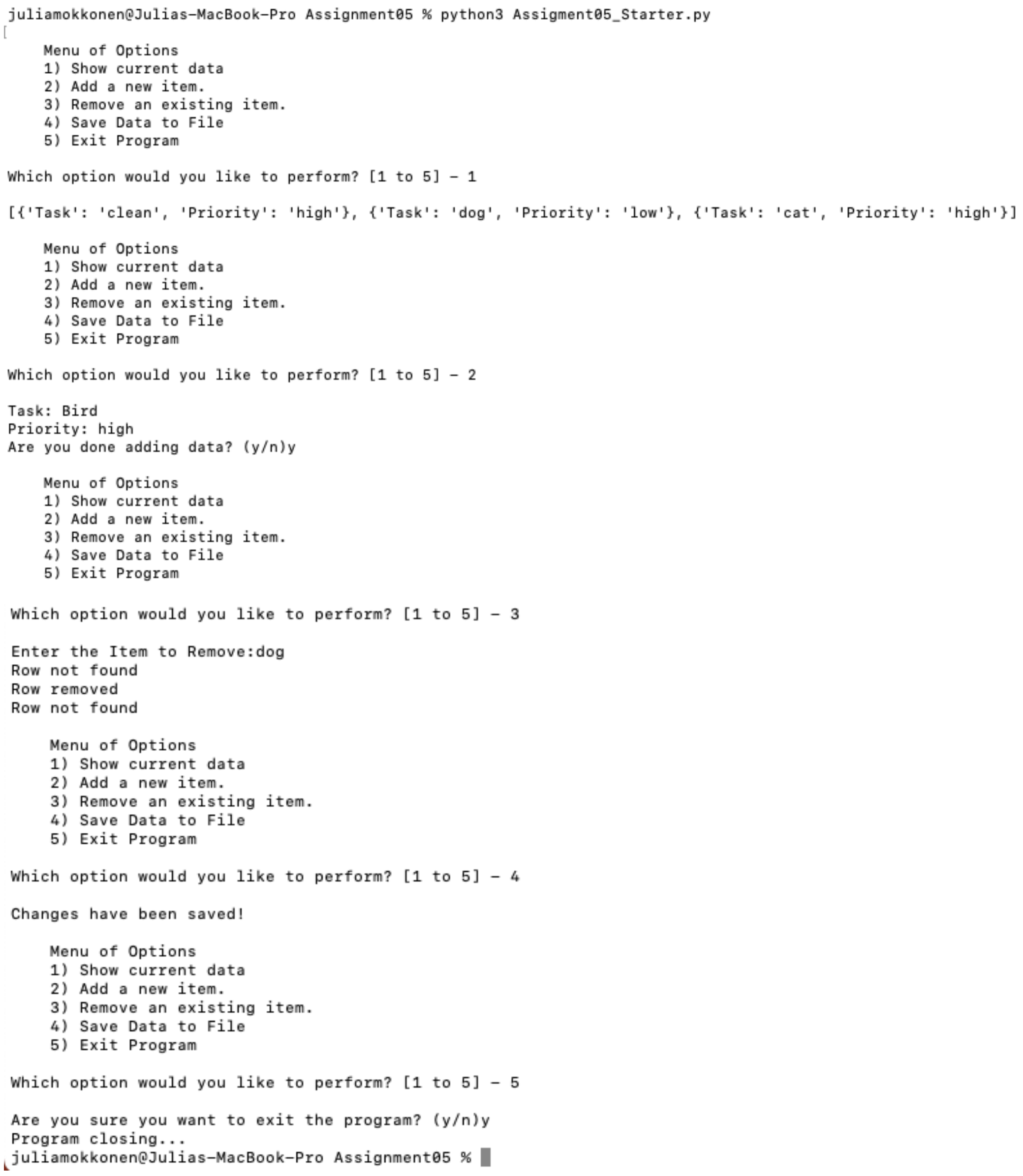
Julia Moekkoenen
August 8, 2023
IT FDN 110 A
Assignment 05
https://github.com/juliamoe/IntroToProg-Python

Creating a To Do List

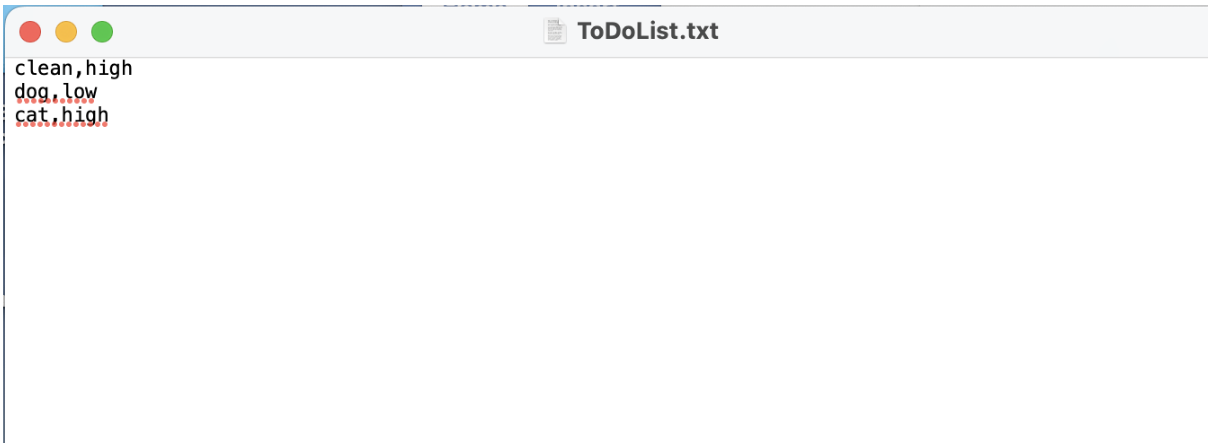
Introduction

The purpose of the code was to create a To do list utilizing an existing code that had been already started but not finished. The script had to be able to read, write, remove, and save data into a .txt file located in the same folder as where the program was saved. This was accomplished by using dictionaries, lists, loops, and if statements. The user provided a “Task” and a “Priority” for each to do item that were then saved into a two dimensional list that created a table. The working code is presented in the figure 1.1 and the before and after editing .txt file is presented in the figures 1.2 and 1.3.



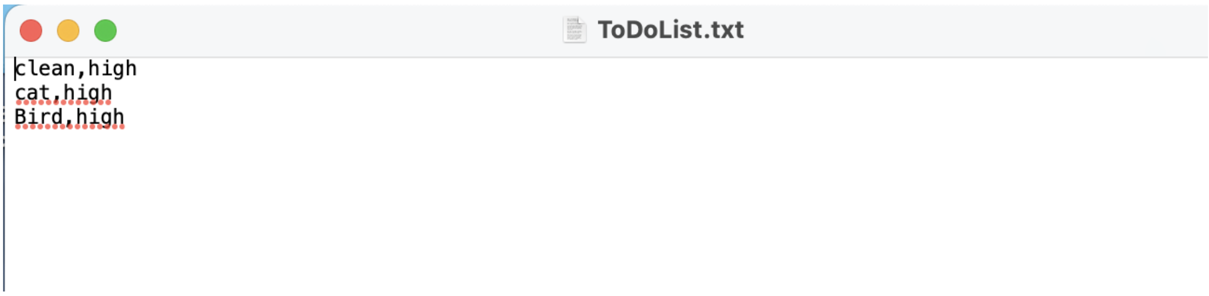
```
juliamokkonen@Julias-MacBook-Pro Assignment05 % python3 Assigment05_Starter.py

    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': 'clean', 'Priority': 'high'}, {'Task': 'dog', 'Priority': 'low'}, {'Task': 'cat', 'Priority': 'high'}]

    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

Task: Bird
Priority: high
Are you done adding data? (y/n)y

    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] - 3

Enter the Item to Remove:dog
Row not found
Row removed
Row not found

    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

Changes have been 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] - 5

Are you sure you want to exit the program? (y/n)y
Program closing...
juliamokkonen@Julias-MacBook-Pro Assignment05 %
```

Figure 1.1: Working code in the Mac Terminal



```
clean,high
dog,low
cat,high
```

Figure 1.2: Output printed into a text file before editing

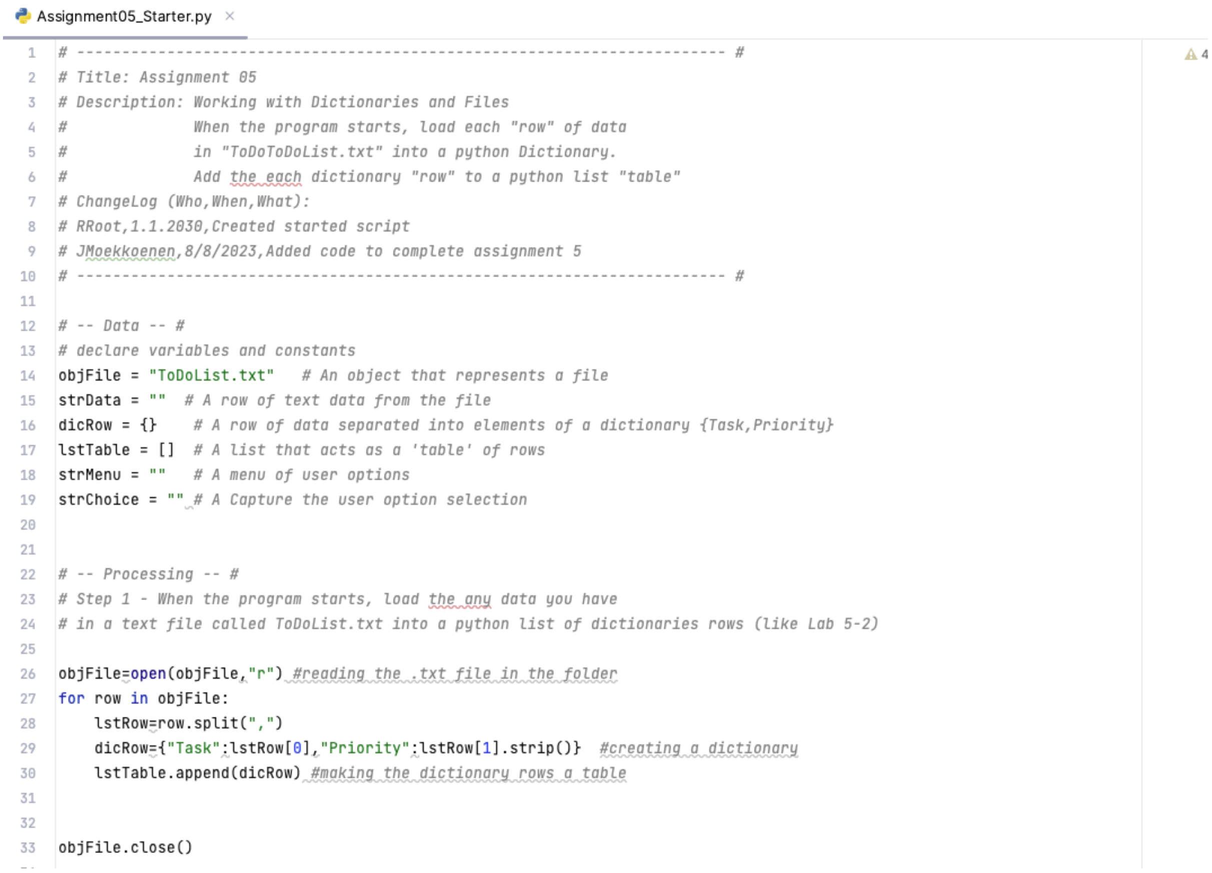


```
clean,high
cat,high
Bird,high
```

Figure 1.3: Output printed into a text file after removing the task row "dog" and adding the task row "bird"

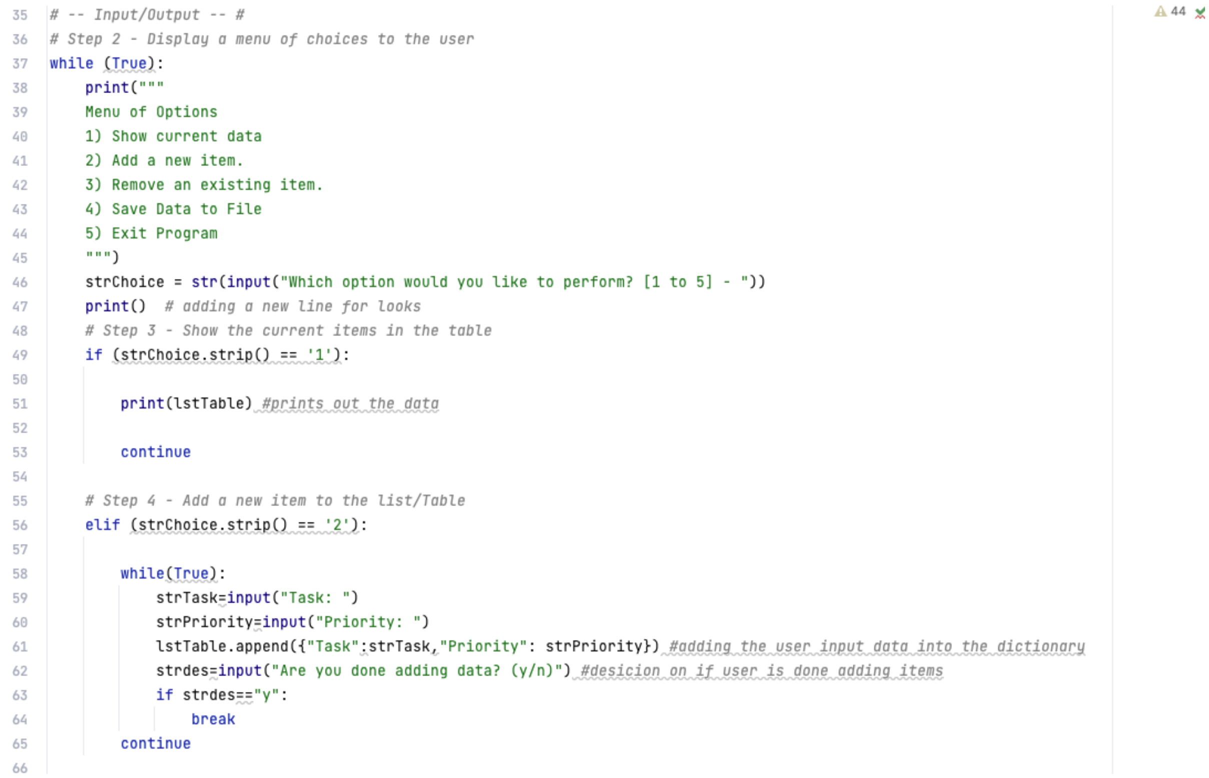
Creating the program

The task was to complete an existing code file that had been left unfinished. The unfinished code created a challenge of understanding what each variable definition was assigned to do and the editor needed to understand the flow of code also.



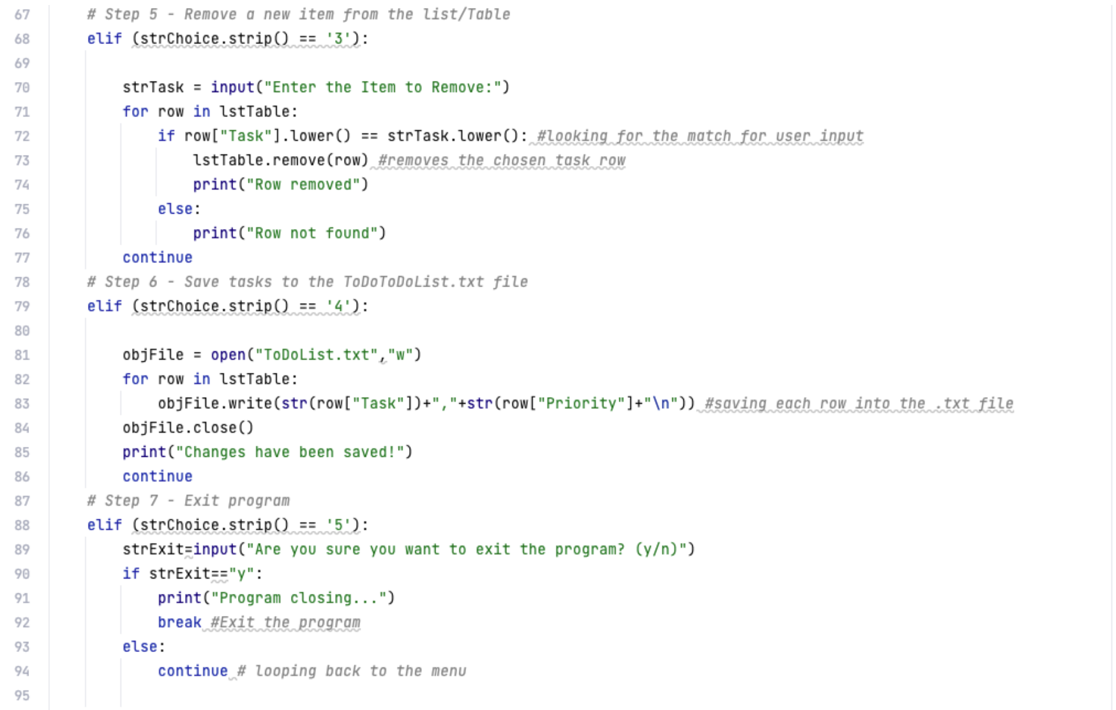
```
# ------------------------------------------------------------------------ #
# Title: Assignment 05
# Description: Working with Dictionaries and Files
#              When the program starts, load each "row" of data
#              in "ToDoToDoList.txt" into a python Dictionary.
#              Add the each dictionary "row" to a python list "table"
# ChangeLog (Who,When,What):
# RRoot,1.1.2030,Created started script
# JMoekkoenen,8/8/2023,Added code to complete assignment 5
# ------------------------------------------------------------------------ #

# -- Data -- #
# declare variables and constants
objFile = "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


# -- 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)

objFile=open(objFile,"r") #reading the .txt file in the folder
for row in objFile:
    lstRow=row.split(",")
    dicRow={"Task":lstRow[0],"Priority":lstRow[1].strip()}  #creating a dictionary
    lstTable.append(dicRow) #making the dictionary rows a table


objFile.close()
```

Figure 1.4a: Code for To Do list



```
# -- 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
    # Step 3 - Show the current items in the table
    if (strChoice.strip() == '1'):

        print(lstTable) #prints out the data

        continue

    # Step 4 - Add a new item to the list/Table
    elif (strChoice.strip() == '2'):

        while(True):
            strTask=input("Task: ")
            strPriority=input("Priority: ")
            lstTable.append({"Task":strTask,"Priority": strPriority}) #adding the user input data into the dictionary
            strdes=input("Are you done adding data? (y/n)") #desicion on if user is done adding items
            if strdes=="y":
                break
        continue

```

Figure 1.4b: Code for To Do list



```python
    # Step 5 - Remove a new item from the list/Table
    elif (strChoice.strip() == '3'):

        strTask = input("Enter the Item to Remove:")
        for row in lstTable:
            if row["Task"].lower() == strTask.lower(): #looking for the match for user input
                lstTable.remove(row) #removes the chosen task row
                print("Row removed")
            else:
                print("Row not found")
        continue
    # Step 6 - Save tasks to the ToDoToDoList.txt file
    elif (strChoice.strip() == '4'):

        objFile = open("ToDoList.txt","w")
        for row in lstTable:
            objFile.write(str(row["Task"])+","+str(row["Priority"]+"\n")) #saving each row into the .txt file
        objFile.close()
        print("Changes have been saved!")
        continue
    # Step 7 - Exit program
    elif (strChoice.strip() == '5'):
        strExit=input("Are you sure you want to exit the program? (y/n)")
        if strExit=="y":
            print("Program closing...")
            break #Exit the program
        else:
            continue # looping back to the menu

```

Figure 1.4c: Code for To Do list

Challenges

The biggest challenges were faced on the debugging side of the code. The approach was to code one step at a time and then debug it to make sure the portion worked before moving onto the next step. Some difficulty was faced with keeping up with what each variable meant and what values they were assigned to. Understanding the dictionaries and lists took additional time, and trial and error, to get the wanted outcome.

Summary

By using an existing unfinished code, a To Do list program was created. The purpose of the program was to let the user view, add, remove, and save data into a .txt by giving a task and priority inputs. The code was tested on the PyCharm application and Mac terminal to ensure functionality on both platforms. The technical knowledge gained during the coding were the use of dictionaries, removing data from a .txt file, and appending more data into the two dimensional list that creates a table.