**Name: Wei Wang**
**Date: 8/9/2022**
**Course: IT FDN 110 B Su 22: Foundations Of Programming: Python**

# Assignment05: Lists and Dictionaries

## Introduction

In this module we will learn to work with Lists and Dictionaries by creating a script. We will also learn some basics about error handling, functions, script templates and how to post work to GitHub, as is common in the industry.

## Create a Python Script(in a Mac book)

Assignment requirements:

Modify a new script that manages a "ToDo list." The "ToDo" 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**

**Step1:** Create a new sub-folder called Assignment05 inside the _PythonClass folder

**Step2:** Create a new project in PyCharm that uses the _PythonClass\Assignment05 folder as its location

**Step3:** Add the starter file, "Assigment05_Starter.py," to the project.

**Step4:** Add code to your script that will perform that assignment's task. Update the change log in the script's header

**Part1:** # Data
# declare variables and constants

First declare the variables

**Part2:** # Processing
# Step 1 - When the program starts, load any data you have
# in a text file called ToDoList.txt into a python list of dictionaries rows

Open the strFile, use a "for" loop to go through each row and load the data in the ToDoList.txt into a python list of dictionaries rows and then close the file.

**Part3:** # Input/Output
# Step 2 - Display a menu of choices to the user. Use print( ) to let the user know that if the data has been saved or not saved to the file
# Step 3 - If the user choose to input "1" then print the current items in the table

**Part4:** # Step 4 - If the user choose to input "2" then use "input( )" function to let the user input their data and use "append( )" to add the data to the list/Table

**Part5:** # Step 5 - If the user choose to input "3" then use the "input( )" to ask the user what to remove, then use a "for" loop to go through each row to see if can find the item that the user wanted to remove. if found then remove it and print "row found and removed", otherwise print ""row not found" to inform the user.

**Part6:** # Step 6 - Save tasks to the ToDoList.txt file
# If the user choose to input "4" then use "open( )" to open the strFile and use "write( )" to save the data to the ToDoList.txt file. Close the file and print "Now in file!" to inform the user.

**Part7:** # Step 7 - Exit program
# If the user choose to input "5" then print "Exit? ('y/n'): " to ask the user's confirmation. If "Yes", then exit. see the code below(Figure 1)

```
# ------------------------------------------------------------------ #
# Title: Assignment 05
# Description: Working with Dictionaries and Files
```

```
#           When the program starts, load each "row" of data
#           in "ToDoList.txt" into a python Dictionary.
#           Add each dictionary "row" to a python list "table"
# ChangeLog (Who,When,What):
# WWang,08.08.2022,Created started script
# <Wei Wang>,<08/08/2022>,Added code to complete assignment 5
# ------------------------------------------------------------------------ #

# -- Data -- #
# declare variables and constants
objFile = ""  # 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
strFile = "ToDoList.txt"

# -- Processing -- #
# Step 1 - When the program starts, load any data you have
# in a text file called ToDoList.txt into a python list of dictionaries rows (like Lab
5-2)

objFile = open(strFile, "r")
for row in objFile:
    lstRow = row.split(",")
    dicRow = {"task": lstRow[0], "priority": lstRow[1].strip()}
    lstTable.append(dicRow)
objFile.close()

# -- 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] -
")).strip()
    print()  # adding a new line for looks
    # Step 3 - Show the current items in the table
    if strChoice == '1':
        for dicRow in lstTable:
            print(dicRow["task"] + ',' + dicRow["priority"])
            continue
    # Step 4 - Add a new item to the list/Table
    elif strChoice.strip() == '2':
        strItem = input("task: ")
        strValue = input("priority: ")
        lstTable.append({"task": strItem, "priority": strValue})
    # Step 5 - Remove a new item from the list/Table
    elif strChoice.strip() == '3':
        strItem = input("task to remove: ")
        found = False
        for row in lstTable:
            if row["task"].lower() == strItem.lower():
                found = True
                lstTable.remove(row)
                break
        if found:
            print("row found and removed")
        else:
            print("row not found")

    # Step 6 - Save tasks to the ToDoList.txt file
    elif strChoice.strip() == '4':
        objFile = open(strFile, "w")
        for row in lstTable:
            objFile.write(str(row["task"]) + ',' + str(row["priority"] + '\n'))
        objFile.close()
        print("Now in file!")

    # Step 7 - Exit program
    elif strChoice.strip() == '5':
        strChoice = input("Exit? ('y/n'): ")
        if strChoice.lower() == 'y':
            break
```

**Figure 1: task-priority script**

**Step5:** Run the code
Run the script both in PyCharm and an OS command/shell window and capture images of it working on my computer

This is the first screen shot of the script running in PyCharm(Figure 2)

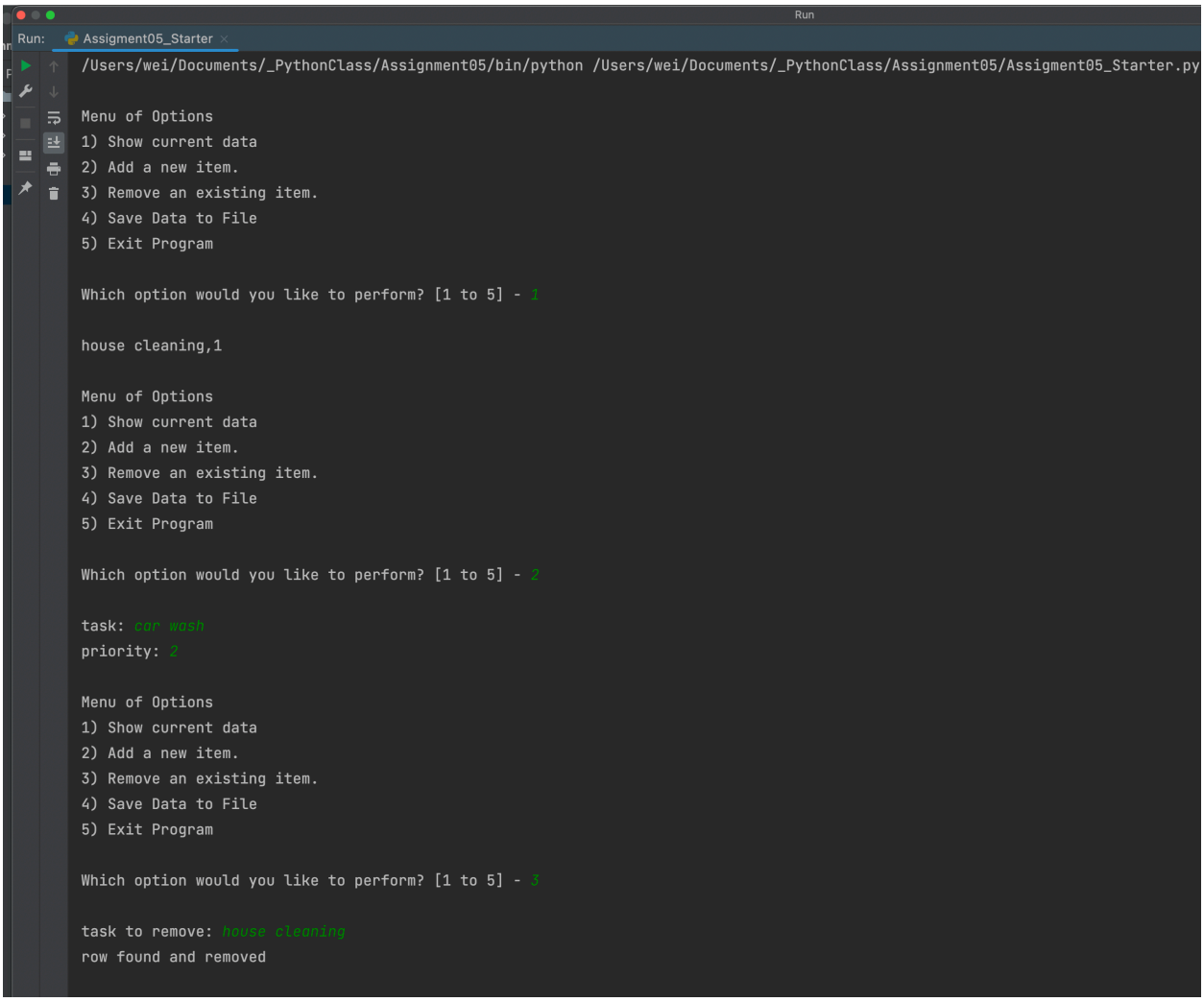


**Figure 2:** *the first screenshot of the script running in PyCharm*

This is the second screenshot of the script running in PyCharm(Figure 3)

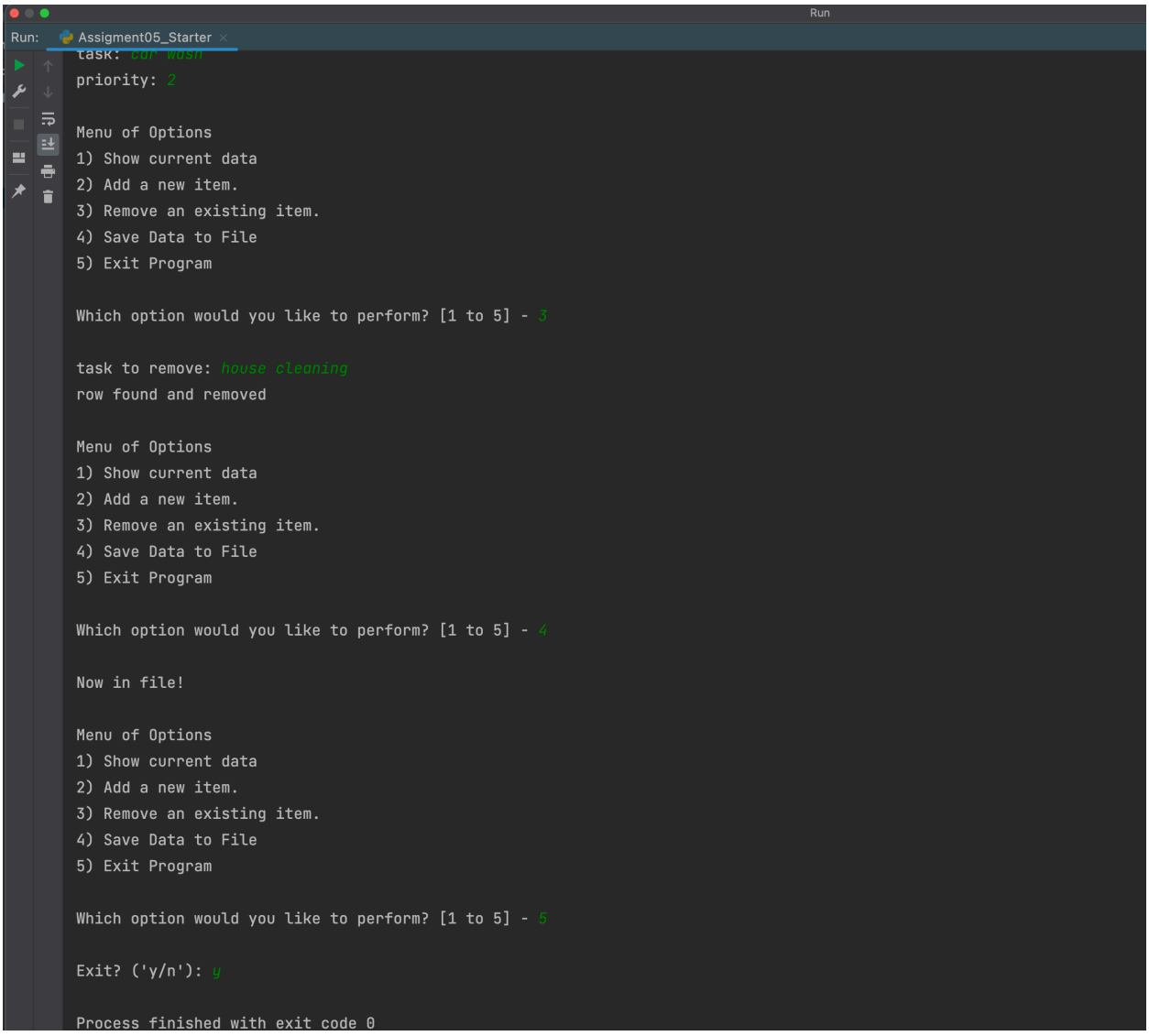


**Figure 3:** *the second screenshot of the script running in PyCharm*

This is the picture of the screen shot of the script running in IDEL (Figure 4)

```
Python 3.10.5 (v3.10.5:f377153967, Jun  6 2022, 12:36:10) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>
= RESTART: /Users/wei/Documents/_PythonClass/Assignment05/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

car wash,2

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: roof cleaning
priority: 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] - 3

task to remove: roof cleaning
row found and removed

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

Now in file!

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

Exit? ('y/n'): y
>>
```

**Figure 4:** *A screenshot of the script running in IDLE*

## Step 6 Verify that it Worked

Locate the text file and open it in a text editor (Figure 5).

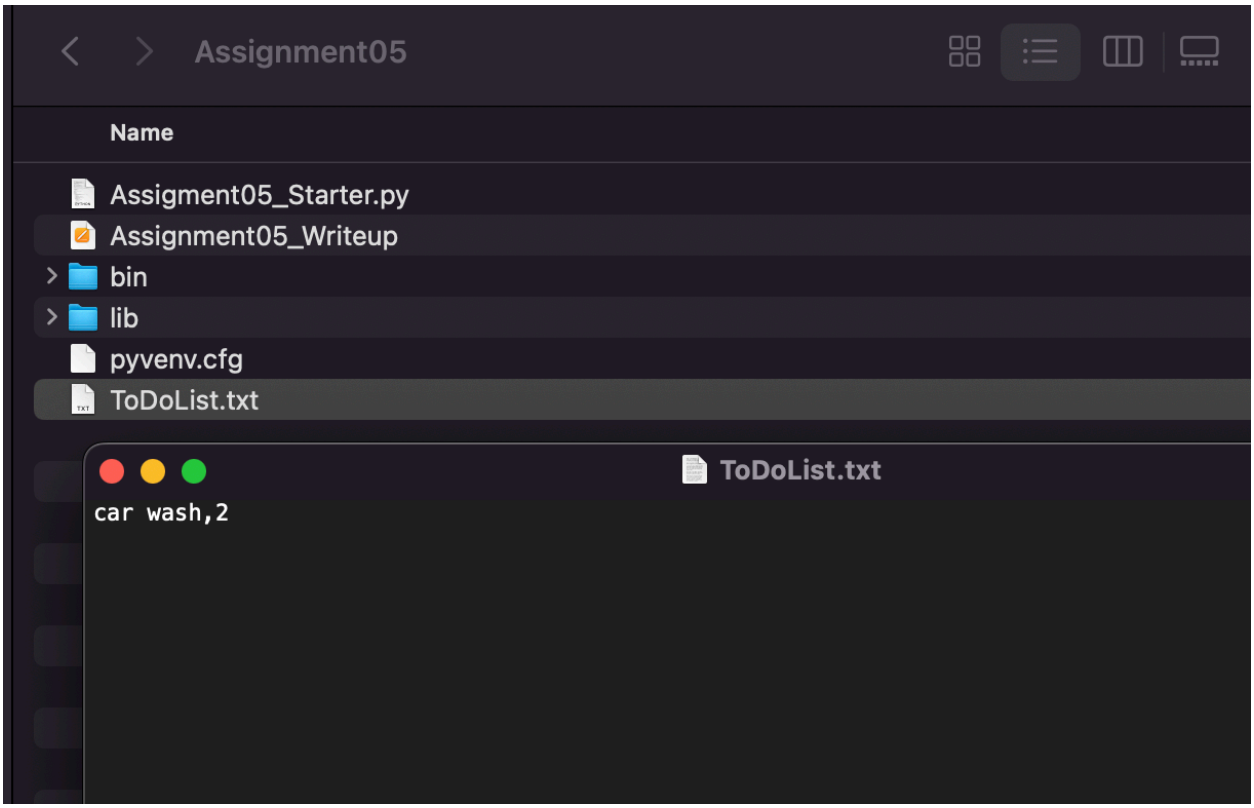


**Figure 5:** *Verifying that the file has data*

## summary

In this module, I learned the difference between a List and a Dictionary, the difference between an Index and a Key, How to read data from a file into a List, How to read data from a file into a Dictionary, why is error handling using Try-Except recommend and what is GitHub how to use it.