Jonathan Kung

11/30/2022

IT FDN 110A

Assignment 7

https://github.com/jonkung/IntroToProg-Python-Mod06

Pickling

Introduction

This week, the main topics were pickling and error handling. Pickling is a way to save data in a binary format. Error handling is anticipating an error by user input and can keep the code operating without giving out an error.

Assignment

In this assignment, I created a script that asks user for name and id number. Then the script can save the list into binary format using pickling.

Pickling

Pickling is a way to save data in a binary format and can obscure the file's content and may reduce the file size. However, pickling does not encrypt the data. To save data by pickling, you add "b" to the file command when opening. For example in figure 1, adding the "b" in "ab" or "rb" will write/read the file in binary format.

```
# Now we store the data with the pickle.dump method
objFile = open("AppData.dat", "ab")
pickle.dump(lstCustomer, objFile)
objFile.close()

# And, we read the data back with the pickle.load method
objFile = open("AppData.dat", "rb")
objFileData = pickle.load(objFile) #load() only loads one row of data.
objFile.close()
```

Figure 1: pickling commands

In the assignment, I created a script to ask and store the person's name and id number. The structure of the script is similar to previous assignments. In this assignment when opening the txt file, I included "ab" to append to the end of the list. Then I used pickle.dump to load the data into the file.

```
while(True):
    choice = input("Enter your choice (1/2/3): ")

    if choice == "1":
        name = str(input("enter your name: "))
        id = int(input("enter your numerical id: "))
        data.append([name, id])
        print("\n Current List: ")
        print(data, "\n")
        continue

    elif choice == "2":
        print("\n Current List: ")
        print(data)
        file = open(file_name, "ab")
        file_data = pickle.dump(data, file)
        file.close()
        print("data has been saved. ")
        continue

    elif choice == "3":
        break
```

Figure 2: saving list into file using pickling

Error Handling

To handle error by user's input, I used a try/except function. The try/except function will try some command and if an error occurs, then the code will jump to the command under except. In this assignment, I included a try/except command at the start of the code to read an existing txt file. If the txt file exists in the folder, then it will open it and print the data in the txt file. However, if no txt file exists, then it will skip the command and jump to the next line which is asking the user for inputs.

```python
import pickle

file_name = "Name_ID.txt"
data = []
name_id_list = []

try:
    file = open(file_name, "rb")
    while(True):
        try:
            data.append(pickle.load(file))
        except:
            break
    file.close()
    print("\n Current List: ")
    print(data, "\n")

except:
    pass
```

Figure 3: try/except for error handling

Completed Script

The finished code and its text file is shown in figure 7.

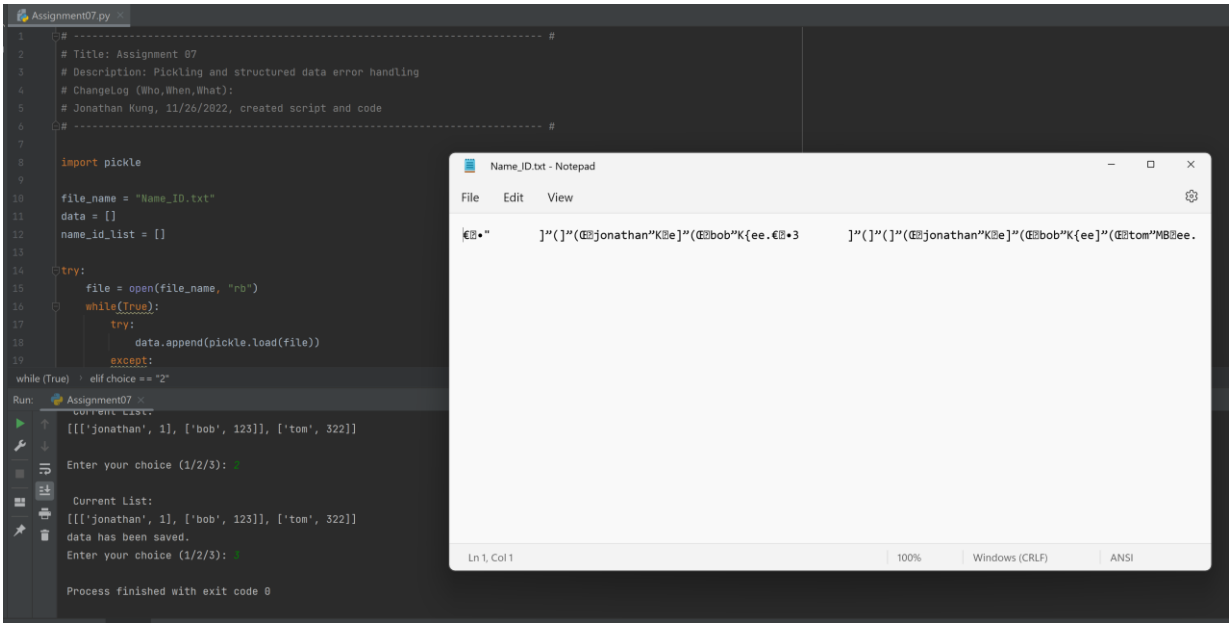

Figure 4: Output from code