Sebastian Quesada

11/20/19

Foundations of Python

Assignment 07

Files and Exceptions

Introduction

This assignment demonstrates the ability for one to use functions to read and manipulate files and error handling. As part of the error handling process, this assignment demonstrates how to pickle and un-pickle data. In this script the user is asked for an ID and a name. These two items are stored in a list and then manipulated. A user can create a new file or load an existing one.

Operation and Error Handling

The images on the left show the GUI of the script. The user is asked for an ID and a name. The first figure is reading from an already existing file. The second figure is demonstrating the error handling when there is no current file available.

```
Hi, please enter your ID and name.
ID: 888
Name: test
Save Data or Read from a file? (S/R)r
Pickling lists.

Unpickling lists.
Enter the name of the file you'd wish to READ: TEsting.txt

Reading the entire file into a list.
length of line is:  1
[[('5555', 'Seeebas')]]
Do you wish to exit? (Y/N):
```

```
Hi, please enter your ID and name.
ID: 6545
Name: Seba
Save Data or Read from a file? (S/R)r
Pickling lists.

Unpickling lists.
Enter the name of the file you'd wish to READ: Sebas.txy

Reading the entire file into a list.
Error - Data could not be read.
Do you wish to exit? (Y/N):
```

The Script and Functions

The while loop below is where the main program is located. There are a few functions that are ran, each for a specific purpose. The while loop contains the welcome screen where the user is asked for data, the menu giving the user a choice, picking functions (stu_pickles) and unpicking functions (tommy_pickles). When a user is asked to enter a choice, that answer is used by the if loop to determine if it is going to save to a file or read from existing data.

```
while True: #While loop with main functions
    welcome()
    menu()
    stu_pickles(lstCustomer)
    tommy_pickles()
    if "s" in choice:
        file_name = input(str("Enter the name of the file you'd wish to SAVE: "))
        save_data_to_file(file_name, list_of_data)
    elif "r" in choice:
        file_name = input(str("Enter the name of the file you'd wish to READ: "))
        read_data_from_file(file_name)
    else:
        wrong_entry()
    exit()
    if quit_choice:
        break #Exiting
```

```
def menu(): #menu function
    global choice #declaring variable for menu choice
    choice = input(str("Save Data or Read from a file? (S/R)"))
    choice = choice.lower() #Converting input to lower
    return choice #returning menu choice
```

```
def stu_pickles(lstCustomer): #pickling function passing names and IDs
    print("Pickling lists.")
    try: #error handling
        f = open(strFileName, "wb") #Writes binary files
        pickle.dump(lstCustomer, f) #dumps the list
        f.close() #closing
    except:
        print("Pickling was unsuccessful </3")
    return lstCustomer


def tommy_pickles(): #unpickling function with no passing variables
    print("\nUnpickling lists.")
    try: #error handling
        f = open(strFileName, "rb")  #Reading the binary file
        global list_of_data #delaring global variable for new list
        list_of_data = [pickle.load(f)] #writing new list
        f.close() # closing
    except:
        print("Unpickling was unsuccessful :-(")
    return list_of_data
```

Saving and Reading Data

The user will be prompted to save or read to a file. This function will pass a file name and the list of data. It will use the file name to name the file and write the data. The figure on the right will read the data using a file name that the user is asked for. If there are errors they will be caught by the exceptions.

```
def save_data_to_file(file_name, list_of_data): #passing file name and unpickled data
    try:
        text_file = open(file_name, "w+") #writing to file
        text_file.writelines(str(list_of_data)) #wriningLines
        text_file.close()
    except:
        print("Error - Data was not saved")
```

```
def read_data_from_file(file_name):
    try:
        print("\nReading the entire file into a list.") # Reading the file
        text_file = open(file_name, "r") #Opening the file
        lines = text_file.readlines() #Reading the lines
        print("length of line is: ", len(lines))
        for line in lines:
            print(line)
        text_file.close()
    except:
        print("Error - Data could not be read.")
```

Summary
In summary, we created a script that will open and manipulate files chosen by the user. This assignment demonstrates the ability and the usefulness of pickling and unpickling. Also, this assignment demonstrates how useful and essential error handling is.