Darivanh Vlachos

March 3, 2021

IT FDN 110 B Wi 21: Foundations of Programming: Python

Assignment07

https://darivanhatuw.github.io/IntroToProg-Python-Mod07/

Boom! I'm a Pickle

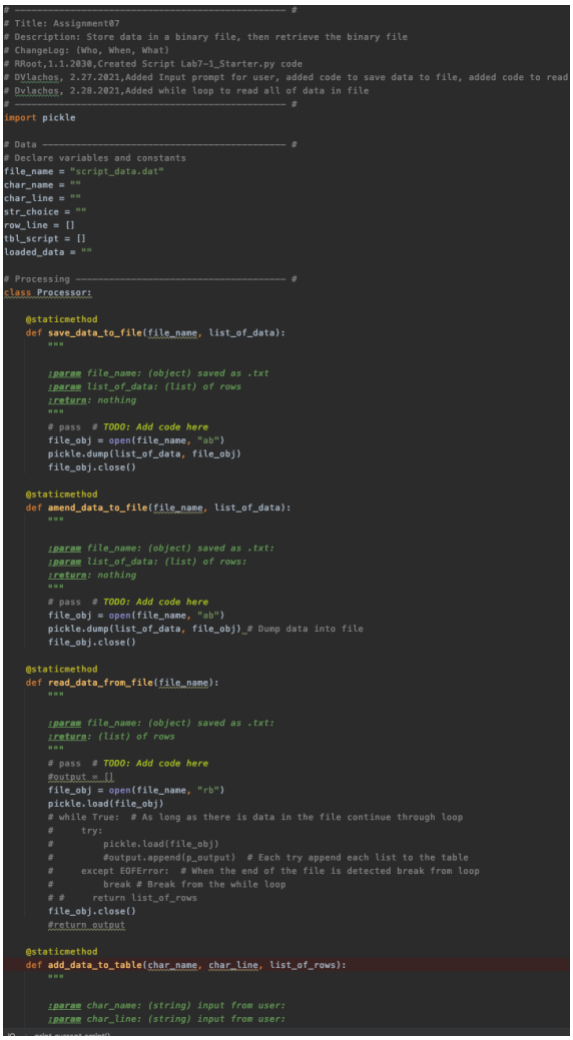
Introduction

There is a show on Adult Swim called "Rick and Morty" that follows the misadventures of a genius scientist, named Rick, and his grandson, Morty. Season 3, Episode 3 the show Rick's lands him into hot water as he has turned himself into a pickle to avoid attending a family therapy session with the family. Rick has a tendency to provide absolutely convoluted solutions to problems he encounters with absolutely complicated solutions, which of course complicates his situation even farther. Although I am far from brilliant, I fear I am burdened with the curse of making things far more complicated than they ought to be therefore making my situation more complicated including this assignment. The previous assignments may have been more complicated in regards to its challenge in programming but when provided freedom I will entangle myself in a grandiose scheme rather than keeping it simple and providing only what is outlined in the rubric.

Big Reveal: I'm a pickle

Better yet, I was in a pickle. I read the book and the web pages ahead of class in order to have a better understanding when attending class. It seemed easy enough. Most of the code was provided for us and our lecturer, Randal Root, ensured us that the assignment was "easy." Our assignment was to build a page that explains Python pickling and error exceptions. Pickling and unpickling is the serialization and de-serialization of data structures for saving and sending files. Error Exceptions, using the try statement and except clause allow programmers to handle errors in a controlled manner that in, my instance, would provide the user more insight into the error and a way to correct the action if needed. Both pickling and error exception, it is my understanding, is a means for humans and computers to communicate. I didn't want to just use the boring code from class and add a few elements to it to make it my own. Instead I created a whole world where the creators and writer of the show ask the user to provide script revisions for the Pickle Rick episode. This would be the background for learning the concepts previously mentioned. Unfortunately, I quickly learned that I did not fully understand the concepts of byte streams and pickling. Not enough to make a complicated program to work properly, that is. Below, Figure 1, is the screenshot of the program I built. Included in the program were 3 different options, error exceptions, appending, loops, as well as 211 lines of code. By the following Tuesday night I was able to run the program with a few hiccups in which I would gladly have sacrificed some points for. I then ran it in Terminal, and it failed miserably. I never

tested in Terminal. Overly complicated program and was useless and made my situation more complicated. It was then I decided to start over.



```
# ------------------------------------------------- #
# Title: Assignment07
# Description: Store data in a binary file, then retrieve the binary file
# ChangeLog: (Who, When, What)
# RRoot,1.1.2030,Created Script Lab7-1_Starter.py code
# DVlachos, 2.27.2021,Added Input prompt for user, added code to save data to file, added code to read
# Dvlachos, 2.28.2021,Added while loop to read all of data in file
# ------------------------------------------------- #
import pickle

# Data -------------------------------------------- #
# Declare variables and constants
file_name = "script_data.dat"
char_name = ""
char_line = ""
str_choice = ""
row_line = []
tbl_script = []
loaded_data = ""

# Processing -------------------------------------- #
class Processor:

    @staticmethod
    def save_data_to_file(file_name, list_of_data):
        """

        :param file_name: (object) saved as .txt
        :param list_of_data: (list) of rows
        :return: nothing
        """
        # pass  # TODO: Add code here
        file_obj = open(file_name, "ab")
        pickle.dump(list_of_data, file_obj)
        file_obj.close()

    @staticmethod
    def amend_data_to_file(file_name, list_of_data):
        """

        :param file_name: (object) saved as .txt:
        :param list_of_data: (list) of rows:
        :return: nothing
        """
        # pass  # TODO: Add code here
        file_obj = open(file_name, "ab")
        pickle.dump(list_of_data, file_obj)_# Dump data into file
        file_obj.close()

    @staticmethod
    def read_data_from_file(file_name):
        """

        :param file_name: (object) saved as .txt:
        :return: (list) of rows
        """
        # pass  # TODO: Add code here
        #output = []
        file_obj = open(file_name, "rb")
        pickle.load(file_obj)
        # while True:  # As long as there is data in the file continue through loop
        #     try:
        #         pickle.load(file_obj)
        #         #output.append(p_output)  # Each try append each list to the table
        #     except EOFError:  # When the end of the file is detected break from loop
        #         break # Break from the while loop
        # #     return list_of_rows
        file_obj.close()
        #return output

    @staticmethod
    def add_data_to_table(char_name, char_line, list_of_rows):
        """

        :param char_name: (string) input from user:
        :param char_line: (string) input from user:
```

Figure 1 My failed, overly complicated program

K.I.S.S. (Keep It Simple Stupid)

The first time I heard these words were in Marine Corps boot camp as a young woman fresh out of high school. Marines aren't known for their intellect, but Marine Corps bravado definitely got the best of me this time. K.I.S.S. was what I needed to do to finish this project and turn it in. I used the Lab7-1 code as the skeleton for my final code. I pulled code from the first Assignment 7 program that I knew worked and built only the essential code needed to fulfill the requirement of the assignment and stopped there. Below was my the integration of the try statement and the except clause within a while loop to append the data into a list as it scanned through the byte stream until it no longer had data to load, or in other words when it encountered the end of the stream, Figure 2.

```
    # pass  # TODO: Add code here
    output = []
    file_obj = open(file_name, "rb")
    while True:  # As long as there is data in the file continue through loop
        try:
            output.append(pickle.load(file_obj))  # Each try append each list
        except EOFError:  # When the end of the file is detected break from l
            break # Break from the while loop
    #     return list_of_rows
    file_obj.close()
    return output
```

Figure 2 Error exception using a try statement nested inside a while loop to append to a list.

I kept the program simple and was still able to include two input prompts for the user and appended the list items to a table, which would be saved back into a binary file. When I tested this simpler program with some lines from another television show dear to my heart it worked. Not as fancy. Not as many bells and whistles, but it worked. Figure 3 is the screenshot from my test.

```
Here's the amazing script you have saved!

Would you like to add lines to the script? [1 = Yes, 0 = No] 1

Character: Jim
Enter Jim's line: Bears

Would you like to add lines to the script? [1 = Yes, 0 = No] 1

Character: Jim
Enter Jim's line: Beets

Would you like to add lines to the script? [1 = Yes, 0 = No] 1

Character: Jim
Enter Jim's line: Battlestar Galactica

Would you like to add lines to the script? [1 = Yes, 0 = No] n
Oops. You entered a letter. Please enter the number 1 or 0.

Would you like to add lines to the script? [1 = Yes, 0 = No] 0

Here's the amazing script you have saved!
Jim: Bears
Jim: Beets
Jim: Battlestar Galactica

Process finished with exit code 0
```

Figure 3 My simple program worked.

Summary

Rick Sanchez, the brilliant scientist character from the animated television show Rick and Morty, continually complicates his situation with his overly complicated schemes and I was no different, minus the brilliant part. Not to put a negative spin on wanting to deliver a magnificent product, but I will continue to work on keeping my projects simple and build from there continually testing to ensure that I can deliver my future assignment in a timely manner.

References