Loren Ormerod
August 26, 2020
IT FDN 110 B
Assignment07

Purpose:

Document actions taken and knowledge gained while working through Module07 course content.

Weekly Content:

File Types

Files are saved in one of two formats – text or binary. Both text and binary files store data in sequential bytes, but each bit in a text file represents a human readable character. Text files are saved as plain or rich text files and, in general, are less prone to corruption. The human readable nature of text files makes error spotting and corrections easy.

Binary files on the other hand, are files that store data in sequential bytes, grouped into eight or sixteen bits, and can be saved as various file types including audio, images, text, etc. Bits in binary files represent custom data that is machine readable. Binary files are often only readable by the intended file format. For example, if you were to open a .PNG file in a simple text editor you would see a list of unintelligible characters.[1]

Saving data in a binary format in Python is referred to as pickling; Python has a pickle module that takes data associated with an object and converts it as easily storable and loadable binary information[2].[3]

Exceptions

Exceptions are a type of error that does not unconditionally result in a program fail. Exceptions can be handled within a program to provide information about the error back to the user. For example, it is possible to display customized error messages or instructions when an exception error occurs. This is accomplished by wrapping functions in try/except statements that are expected to encounter erroneous inputs/handling. We reduce the likelihood of lost data, when a program quits unexpectedly, all data not saved to file is lost. It is much better to build error handling into the program beforehand, than lose time on the user end. [4]

New classes of exceptions may be derived by off of a base exception class. To do this, you create the base class as defined by the module and then create a subclass that has specific exception classes for different error conditions.

```
class Error(Exception):
    """Base class for exceptions in this module."""
    pass

class InputError(Error):
    """Exception raised for errors in the input.

    Attributes:
        expression -- input expression in which the error occurre
        message -- explanation of the error
    """

    def __init__(self, expression, message):
        self.expression = expression
        self.message = message
```

[1] https://www.thecrazyprogrammer.com/2018/05/difference-between-text-file-and-binary-file.html
[2] https://wiki.python.org/moin/UsingPickle
[3] FDN_Py_Module_07.pdf, page 14
[4] https://docs.python.org/3/tutorial/errors.html

Loren Ormerod
August 26, 2020
IT FDN 110 B
Assignment07

Markdown Language

Markdown is a plain-text format syntax language that is intended for easy reading and writing of structured documentation.[5] The idea is that markdown documents should be publishable as originally written, without appearing to have tags/formatting instructions.[6]

Summary

This week we covered the differences between text & binary files and how to save a Python script to either type. We also learned the benefits of structured error handling and practiced using this in the assignment. Appendix I holds screenshots of my executed code in Anaconda Prompt, the full script is available on GitHub.

APPENDIX I

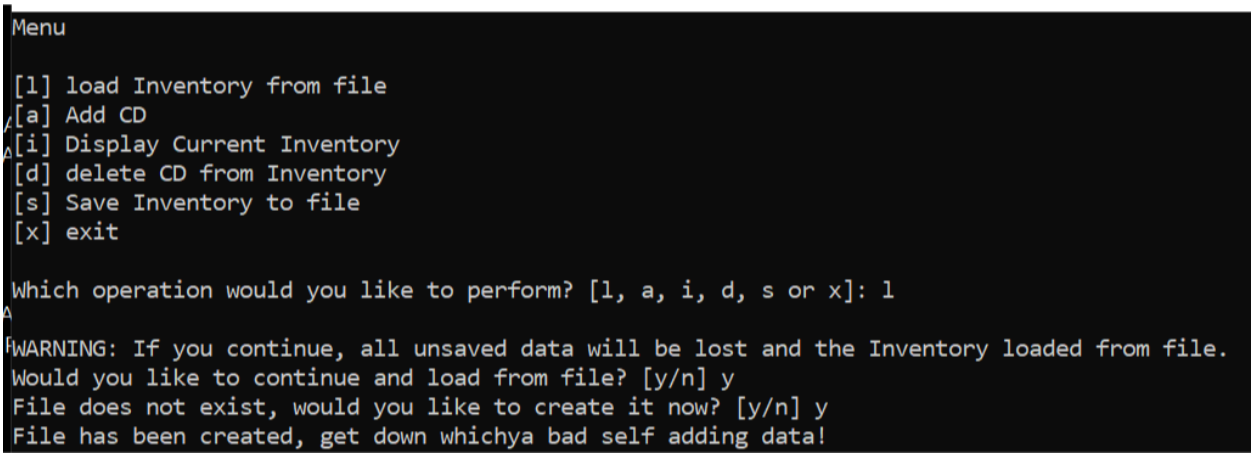

Figure 1 - FileNotFoundError

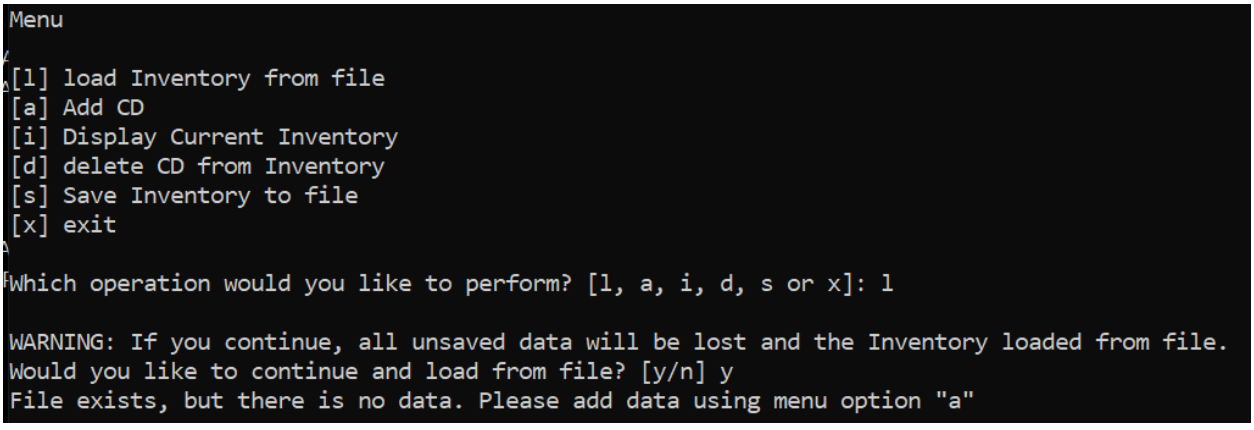

Figure 2- EOFError

[5] https://medium.com/hackernoon/say-yes-to-markdown-no-to-ms-word-be4692e7a8cd
[6] https://github.github.com/gfm/#what-is-github-flavored-markdown-

```
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, d, s or x]: a

Enter ID: H
ID must be a positive integer. Please try again.
```

Figure 3 - ValueError

```
Which operation would you like to perform? [l, a, i, d, s or x]: a

Enter ID: 1
What is the CD's title? Hello
What is the Artist's name? World

======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       Hello (by:World)
======================================
```

Figure 4 - Add data

```
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, d, s or x]: i


======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       Hello (by:World)
2       What's up yo (by:Crackin' Backs)
=======================================
```

Figure 5 - Display Inventory

```
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, d, s or x]: d

Enter the ID you would like to delete: T
Sorry, ID must be a positive integer. Please try again.
```

Figure 6 - ValueError

```
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, d, s or x]: d

Enter the ID you would like to delete: 1
The CD was removed

======= The Current Inventory: =======
ID      CD Title (by: Artist)

2       What's up yo (by:Crackin' Backs)
=======================================
```

Figure 7 - Delete CD

```
Menu

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit

Which operation would you like to perform? [l, a, i, d, s or x]: s


======= The Current Inventory: =======
ID      CD Title (by: Artist)

2       What's up yo (by:Crackin' Backs)
======================================

Save this inventory to file? [y/n] y
CD Inventory has been saved to "CDInventory.dat"
```

Figure 8 - Save data