Hao Liang
November 28, 2021
FDN101
Assignment 07

Introducing to Python Programming

Introduction

Module/Assignment_07 introduces how to handle txt files and the concepts of pickle and binary files. The module further expands the uses of try-except for structured error handling. Creating custom error classes and markdown code are also introduced.

Topic_1 – with statement

with statement in Python is used in exception handling to make the code cleaner and much more readable. It simplifies the management of common resources like file streams. Observe the following code example on how the use of with statement makes code cleaner1[1].

Topic_2 – pickle

To work with binary file we need to understand the pickling and unpickling processes. Two examples from this module,
pickle.load(file) equals unpickler(file).load();
pickle.dump(data, file) equals Pickler(file).dump(data).

Topic_3 – pickle.load()

pickle.load(file_name) will only read the first object in the file. We can use while loop to read all content, below is a screenshot from Assignment07,

```
        table.clear()  # this clears existing data and allows to load data from file
        with open (file_name, 'rb') as objFile:
            while True:
                try:
                    line = pickle.load(objFile)
                    data = line.strip().split(',')
                    dicRow = {'ID': int(data[0]), 'Title': data[1], 'Artist': data[2]}
                    table.append(dicRow)
                except:
                    break
        return table
```

[1] with Statement in Python, Retrieved 2021-Nov-29

Topic_4 – Structured Error Handling in Assignment 07

In order to process the code until getting a right value type from user's input I separated the ID input as a function and used structured error handling to return back an integer,

```
@staticmethod
def Num_input():
    """Ask user's inputs of ID number
       Separate ID input as function for structured error handling

    Args:
        None

    Returns:
        ID

    """
    Val0 = input('Enter ID: ').strip()
    return Val0

@staticmethod
def user_input():
    """Ask user's inputs of ID, Title, and Artist

    Args:
        Val1 for user's ID
        Val2 for user's Title
        Val3 for user's Artist

    Returns:
        a table of 3 values

    """
    Val0 = IO.Num_input()
    Val1 = Error.e_value(Val0) # Through structured error handling to get an integer
    Val2 = input('Enter Title: ').strip()
    Val3 = input('Enter Artist: ').strip()
    return Val1, Val2, Val3
```

List 1 key I/O functions in Assignment07

```
# -- Structured Error Hanlding  -- #
class Error:
    @staticmethod
    def e_value(Num):
        while True:
            try:
                int(Num)
                return int(Num) #returning a integer
                break
            except ValueError as e:
                print('Please input an integer!')
                print('Python built in error:')
                print(type(e), e, e.__doc__, sep='\n')
                print('Please input a valid ID number!')
                Num = IO.Num_input()
            except Exception as e:
                print('General error!')
                print('Python built in error:')
                print(type(e),e,e.__doc__,sep ='\n')
                print('Please input a valid ID number!')
                Num = IO.Num_input()

    @staticmethod
    def e_file(file_name):
        try:
            with open(file_name) as objFile:
                objFile.close()
        except FileNotFoundError as e:
            print('Text file not found!')
            print('Python built in error:')
            print(type(e), e, e.__doc__,sep='\n')
            print('A new file has been created as', file_name)
        except Exception as e:
            print('General error!')
            print('Python built in error:')
            print(type(e), e, e.__doc__,sep ='\n')
```

List 2 Structured Error Handling in Assignment07

Topic_5 – Assignment 7

```
In [2]: runfile('/Users/HAO/_FDNProgramming/Mod_07/Assignment07/
CDInventory.py', wdir='/Users/HAO/_FDNProgramming/Mod_07/Assignment07')
Text file not found!
Python built in error:
<class 'FileNotFoundError'>
[Errno 2] No such file or directory: 'CDInventory.txt'
File not found.
A new file has been created as CDInventory.txt
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: a
Please input an integer!
Python built in error:
<class 'ValueError'>
invalid literal for int() with base 10: 'a'
Inappropriate argument value (of correct type).
Please input a valid ID number!

Enter ID: #
Please input an integer!
Python built in error:
<class 'ValueError'>
invalid literal for int() with base 10: '#'
Inappropriate argument value (of correct type).
Please input a valid ID number!

Enter ID: 111

Enter Title: AAA

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

111 AAA (by:BBB)
======================================
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)

111 AAA (by:BBB)
======================================

Save this inventory to file? [y/n] y
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

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

111 AAA (by:BBB)
======================================

Enter ID: s
Please input an integer!
Python built in error:
<class 'ValueError'>
invalid literal for int() with base 10: 's'
Inappropriate argument value (of correct type).
Please input a valid ID number!

Enter ID: 1
Could not find this CD!
======= The Current Inventory: =======
ID  CD Title (by: Artist)

111 AAA (by:BBB)
======================================
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]: x
```

Figure 1 screenshot – running assignment 7 script in Spyder

```
(base) Liangs-MBP:Assignment07 HAO$ python CDInventory.py
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: 222
Enter Title: CCC
Enter Artist: DDD
======= The Current Inventory: =======
ID      CD Title (by: Artist)

111     AAA (by:BBB)
222     CCC (by:DDD)
======================================
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

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

111     AAA (by:BBB)
222     CCC (by:DDD)
======================================
Enter ID: abc
Please input an integer!
Python built in error:
<class 'ValueError'>
invalid literal for int() with base 10: 'abc'
Inappropriate argument value (of correct type).
Please input a valid ID number!
Enter ID: 1
Could not find this CD!
======= The Current Inventory: =======
ID      CD Title (by: Artist)

111     AAA (by:BBB)
222     CCC (by:DDD)
======================================
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)

111     AAA (by:BBB)
222     CCC (by:DDD)
======================================
Save this inventory to file? [y/n] y
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]: x
```

Figure 2 Screenshot – running assignment7 script in Terminal

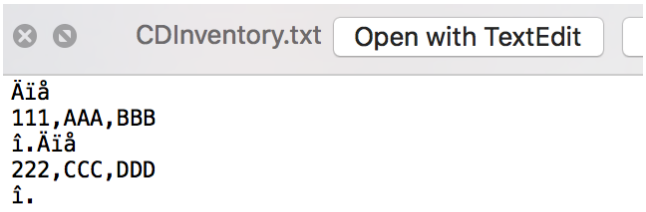

Figure 3 Screenshot – binary file after running the code

GitHub

https://github.com/seahao/Assignment_07

Summary

Text file and binary file encode data differently. They show us that computer reads, writes, and stores data in different ways. By using structured error handling, we can avoid file crashing and safeguard the unsaved data in runtime.