Armand Feuba Baweu

Aug 2, 2023

IT FDN 100 A

Assignment 06

https://github.com/armand-cesti/IntroToProg-Python

Intro to Programming (Python)

**Intro**

In this module, it is questioned to lean the benefits of putting built-in python command into function and using structured error handling. In other hand, it’s questioned to learn the difference between a text file and binary file. In additional how Exception can be used and how I can derive a new class from the Exception class, and when I might create a class derived from Exception class. At the end learning Markdown Language and how using in GitHub Webpage.

**Working with Text files**

Over preview module, i learned how to open, write and closed data into the file. In this module 7 I leaned how to customize with the function and combine to those 3 within one function. Refer to listing 1 in module 7, it demonstrates how open, write and close were combined in function save\_data. Follow the below screenshot.

A screenshot of a computer

Description automatically generated

**Read Mode**

same as write into file, Read mode can also be customized with the function. Followed below screenshot referring from listing2 of module 7 how the function was created.

A screenshot of a computer

Description automatically generated

**Append Mode.**

Same as other, I can add data in to existing file by adding “a” when we open the file. Refer to listing3 of Module7 show, Screenshot below show us how to add data into existing file.

As you can see in screenshot, within add\_more\_data() function, “a” was added to:

file = open(file\_name, "a")

A screenshot of a computer

Description automatically generated

**Reading Data Options**

In this section, I learned that python has other options to create to read data from a file such that :

* The readline() function: this method is associated with files objects, and is used to read a single line from a text file. Refer to list4 of Module7, within read\_data\_row(): with data = file.readline(), it can only read one row of data in the file.
* Using a while loop: the while loop is used to call readline() so it can repeat itself. Check listing list5 by the end of program, show how while look is used to read data.
* The readlines() function: just “s” added at the end of the lines. Made us learn that this method read many lines from a text file. Check listing 6 in modules for demo.
* For loop: it’s also another option. Check listing 7 in module 7 for Demo.

**Combining Reading and Writing**

Combine write and read it’s one of simple useful applications. Refer to listing8 of module 7 for demo how write and read are combined to make application useful.

**Working with Binary Files**

Working with binary files in Python involves reading and writing files that contain non-text data. Binary files are different from text files in that they contain data in a format that isn't human-readable, and they are typically processed differently. That’s what pickling is used as technique in python. Refer to listing 9 of module 7, we learned that we used “import” to call pickle that help us to import code from another code file. Mode deep in listing9 , the pickle.dump() is used to store the data, and pickle.load().

**LAB 7-1: Working with binary files**

In this lab is question to create a function save data to a binary file, by adding code to the stater file and Test the script and write down how the code works.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Structured Error Handling (Try-Except)**

Past couple module, we introduce Try-Except. But now we leaned little deeper how Try-Except can be more used. However, the Try-Except help us to handle Error. As see into listing11 of Module, Try-Except is used to avoid error of 5/0 because 5/0 is Indefinite number.

* Excepted: it’s used to hold information about an error. Check listing 12 to see how Excepted working.

**Catching Specific Exceptions**

Catching specific exceptions in Python allows you to handle different types of errors gracefully, providing appropriate responses or actions for each error scenario. As you see in listing13 of module7, catching is used to correct ZeroDivisionError.

**Raising Custom Errors**

In Python, you can raise custom exceptions to signal specific error conditions in your code. This can be useful when you want to handle exceptional situations that aren't covered by built-in exceptions. Check listing 14 in module 7 for demo.

**Summary**

It was question in module 7 to understand the benefits of putting built-in Python command in to function and benefits of using structured error handling. In other side, I learned the difference between text file and a binary file. I also learned how to use the Exception class. In addition, I learned how to derive a new class from Exception class and when I might create a class derived from the exception class. At the end understand the markdown language and how to use it GitHub webpage. My only difficult in this module was understanding how derive class from Exception class and learned Markdown language and how using it into GitHub.

**Assignement07**

**This assignment was to create a demonstration of how Pickling and Structure errors handling. the following screenshot below show how the code was written by using IF-statement.**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated