|  |  |
| --- | --- |
| Danielle Lilleston  Python 100  14 May 2024 | Assignment 05 |
| A black rectangular object with white text  Description automatically generated | Introduction In this article, I will be using what was learned during assignment 04 but instead include the use of data processing using dictionaries and exception handling. The use of dictionaries in python is to store data in a collection that is ordered, changeable, and does not allow duplicates. They can also be of any data type! The purpose of using exception is to disrupt the programs execution to bring up any errors found along the way. This allows for issues to be resolved early on. Step One: Script Header Always start your script off with a header!  A screen shot of a computer  Description automatically generated  Figure 1. This is my header for assignment 05. Step two: Importing your file When using json files, python requires that you import the json module first in order to utilize the functionality that is provided. Step Three: Define the Constants There are two constants in this script. The first is the MENU, The menu is a string, which is what provides the users with the options at the beginning when running the file. This is the exact same constant used in assignment 04. The second and last constant is the FILE\_NAME. This is the named that will be given to the file in option 3 when all the data is saved and stored. However, in this scenario, the file name is “Enrollments.json” to let the user know what file type we will be using. |
| A black rectangular object with white text  Description automatically generated | A screen shot of a computer  Description automatically generated Figure 2. This shows both the callout to import the json module and the constants I defined in my script. Step Three: Define the Variables **A screen shot of a computer  Description automatically generated**In this code, almost all the variables are the same as in assignment 04. However, student\_data is now set to a dict data type rather than a list. I also included json\_data which is a string that holds information in the necessary format style for json flies.  Figure 3. These are the lists of the variables in the script. Step Four: Checking the File Before we begin, we need to check the file and see if it can be read. First, we use “try” to attempt to open the file. Once loaded, it will then close the file. However, if it does not load, the script will call out to the except block that will print an error message indicating there was a problem. Then, the script utilizes the “Finally” block to execute the closure of the file with or without any exception errors that occurred. The reason we do this is to ensure that the file will run smoothly and can successfully be open and read.    Figure 4. This shows different blocks used to stop a script from running and to check to see if there are any errors that may occur. In this case, we wanted to make sure the file would open and be read successfully. |
|  | Step Five: Processing the Data, Choice 1 This is the heart of the code. In this case, we are asking the user to input the students first name, last name, and course. However, since we are creating a dictionary, we need to make sure that the data is inputted correctly. To do this, when the user enters the students name, the script will verify that it contains only alphabetical characters. If there are other characters, it will raise an error message. If both names pass, a dictionary is created with keys ‘FirstName’, ‘LastName’, and ‘CourseName’ which will store the respective values.  A computer screen shot of a program  Description automatically generated  Figure 5. This shows the first option available to the user. This option allows the user to input the students first name, last name, and course name. This is the beginning of our dictionary as we have created our keys that will have data stored to it. Step 6: Processing the Data, Choice 2 This section threw me for a loop. For the longest time I kept getting an error that the key “LastName” was not a part of my dictionary. To resolve the issue, I used an “if” statement to verify the keys were indeed in the script and if so, to print all the respective data.  A screen shot of a computer program  Description automatically generated  Figure 6. This shows how I wrote my script. I wanted to show how to verify all the keys were defined and if not, to print an error message. Step 7: Processing the Data, choice 3 The last option given to the user is to take all the keys and their respective value and put it into a json file. This is achieved by using json.dump(students, file). The json file can then be accessed with all the data that has been stored. I have also utilized the except stops in the code so that if an error occurs, it will call out to it in the script and give the user an idea of what may causing an issue in the code.  A computer screen shot of a program code  Description automatically generated  Figure 7. This is the final option given to the user. In this part of the script, it saves all the values in the dictionary to a json file. Summary: This was one of my hardest scripts to write. I had a hard time understanding the error that “LastName” was not a key in my dictionary even when I felt I had called out to it. Overall it was a lot of fun to write and the coding is definitely getting longer and much more complex. The use of the exception breaks are extremely helpful too in trying to solve the issues in my script. |