Dominic Verretto

Aug 30, 2024

IT FDN 100 A

Assignment 05

https://github.com/dverretto/IntroToProg-Python/blob/main/assignment05.py

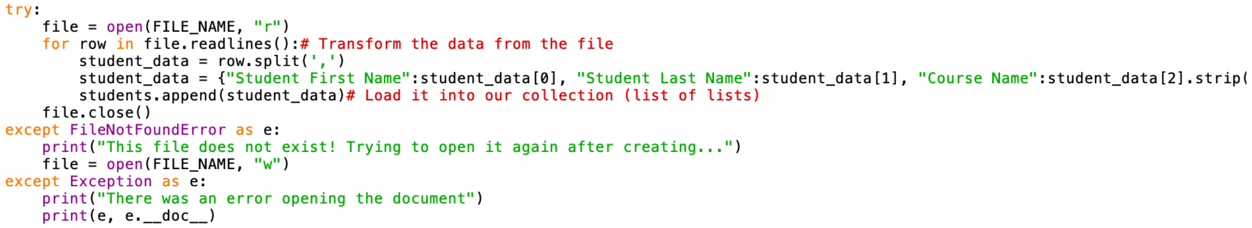
Exceptions, Dictionaries and GitHb

Intro

In this module I learned how to use exceptions to better inform a user how to operate my script. I also learned how to use dictionaries in python to format a table. Finally, I created a GitHub account and uploaded my work to a repository that is open source. This guide should offer insight into how to do each of these things, so that a reader can mimic the same steps I used to accomplish a task in python.

Exceptions

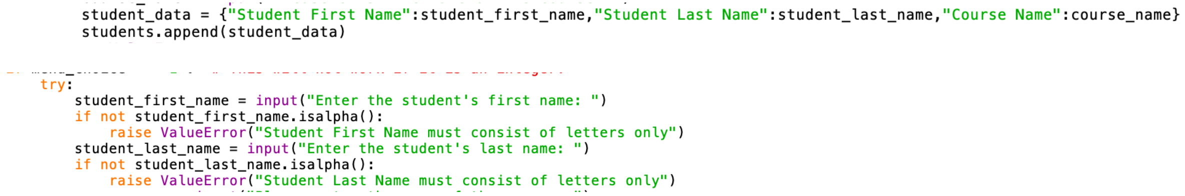
Exceptions are a way to provide a user with information regarding an error that occurred while running the script. When an exception is printed it gives a brief insight into what the error was. Relaying this to the user provides them with information on how to properly execute the script to get a desired result.
The general way to format exceptions in the script is to type the commad “try:” and then indent all of the following script that you are wanting to be included in this exception scan. In the same indentation as the “try:” command, typing the command “except Exception as e” and followed by print(e, e.__doc__) will print any errors that pop up to the user. If you are aware of a specific error that may arise, you can use the print() function to address this as well.



```
try:
    file = open(FILE_NAME, "r")
    for row in file.readlines():# Transform the data from the file
        student_data = row.split(',')
        student_data = {"Student First Name":student_data[0], "Student Last Name":student_data[1], "Course Name":student_data[2].strip(
        students.append(student_data)# Load it into our collection (list of lists)
    file.close()
except FileNotFoundError as e:
    print("This file does not exist! Trying to open it again after creating...")
    file = open(FILE_NAME, "w")
except Exception as e:
    print("There was an error opening the document")
    print(e, e.__doc__)
```

Figure 1: Bracketing a section of code with the “try:” and “except” functions

Another simple exception you can raise to a user is the valueerror. If you want a user to only type a certain kind of character you can raise a valueerror to the user, followed by a message describing the correct way to use the script.



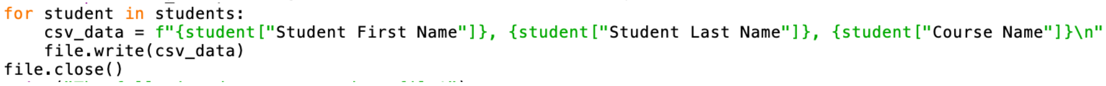
```
student_data = {"Student First Name":student_first_name,"Student Last Name":student_last_name,"Course Name":course_name}
students.append(student_data)
```

```
    try:
        student_first_name = input("Enter the student's first name: ")
        if not student_first_name.isalpha():
            raise ValueError("Student First Name must consist of letters only")
        student_last_name = input("Enter the student's last name: ")
        if not student_last_name.isalpha():
            raise ValueError("Student Last Name must consist of letters only")
```

Figure 2: Demonstrating the proper way to use the valueerror exception

Dictionaries

Dictionaries are another way to manipulate data in python. They are like lists but rather than referring to a location in the data by using numerical language [0], [1] etc., dictonaries use string data. In a dictionary table of first names, last names and course names, we would manipulate the data by referring to the column in the script.



```
for student in students:
    csv_data = f"{student["Student First Name"]}, {student["Student Last Name"]}, {student["Course Name"]}\n"
    file.write(csv_data)
file.close()
```

Figure 3: An example of how dictionaries can be used to write data to a file

In the example shown, the variable csv_data is being written to a file. The information stored in the columns "Student First Name", "Student Last Name" and "Course Name" have all been added to a .csv.

GitHub

GitHub is an open source platform for users to publish documents for review and use of other programmers. A user can create a repository that stores all relevant documentation in, for another user to look at or use in their own programming. If a user does not want others seeing what is in a repository they can make it private.

Citations

This document was built using information from the module originally developed by Randal Root, Luis Conejo-Alpizar and Anubhaw Arya. Python can be accessed at python.org and github can be accessed github.com