James Sharma

August 28, 2024

Foundations of Programming: Python

Assignment 5

jimsharma206/IntroToProg-Python: This page will be used to review homework files (github.com)

Introduction

Assignment 5 examined various forms of error handling, ensuring the user inserts correctly formatted data. Specifically, we applied try-except-finally statements, as well as raise ValueError("custom message"). Personally, I had multiple obstacles. For instance, I had to modify the code to allow for spaces in the course name. Similarly, I had to modify the code to allow for hyphenated first names. Finally, we learned how to upload our script to GitHub.

Obstacles

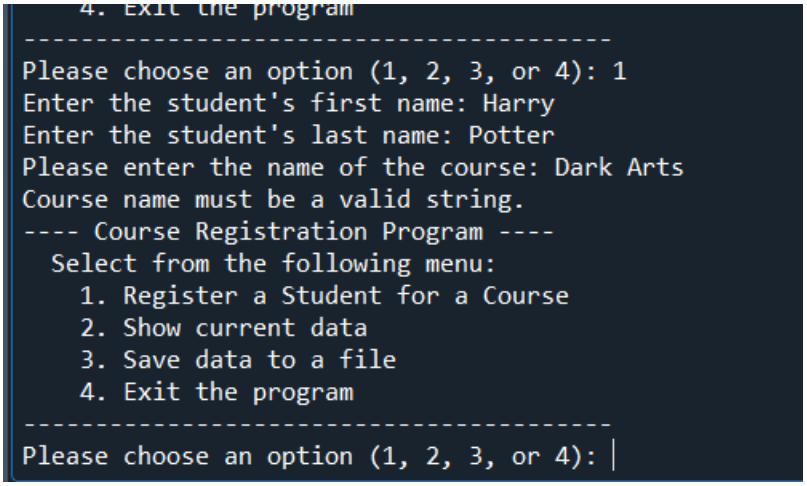

Figure 1: Error with Course Name

For a long time, I was wondering why my input data wasn't saved. Then, I found out that the program wouldn't permit me to enroll Harry Potter to "Dark Arts." The reason for this was because of the space between "Dark" and "Arts." In conclusion, I did not realize that I had to specify that a space between words was permittable.

```
        course_name = input("Please enter the name of the course: ")
        if not course_name.isalpha():
            raise ValueError("Course name must be a valid string.")
```

Figure 2: Source of Course Name Error

The above code signals that “course_name” is alphabetical. This means spaces and integers aren’t permitted.

```
course_name = input("Please enter the name of the course: ")
if not all(x.isalpha() or x.isspace() for x in course_name):
    raise ValueError("Course name must only contain letters and spaces.")
```

Figure 3: Solution of Course Name Error

By adding “x.isspace(),” I was finally permitted to enroll Harry Potter into the “Dark Arts” course. However, even with this new code, I couldn’t enroll Hermione Granger into a course called “100.” Essentially, this program only permits course names with letters and spaces as indicated int the warning message.

```
----------------------------------------
Please choose an option (1, 2, 3, or 4): 1
Enter the student's first name: Mary-Louise
First name must be a valid string.
---- Course Registration Program ----
  Select from the following menu:
    1. Register a Student for a Course
    2. Show current data
    3. Save data to a file
    4. Exit the program
----------------------------------------
Please choose an option (1, 2, 3, or 4):
```

Figure 4: Error for double first names.

Another issue I had was entering my mother’s name, which is Mary-Louise. Figure 4 reveals that the program wouldn’t let me because there is a hyphen. This revealed to me just how many social aspects we need to consider when creating these programs. In fact, simple programs that run smoothly contain a lot of complexity because there are several factors that we need to consider.

```
student_first_name = input("Enter the student's first name: ")
if not all(x.isalpha() or x == '-' for x in student_first_name):
    raise ValueError("First name must only contain letters and hyphens.")
```

Figure 5: Solution for hyphenated first names

As shown in Figure 5, I needed to add x==’-‘ to make sure the program permits hyphens. This is important because it is not “that” rare for people to have double names.

Exception Handling

```
    # Save the data to a file
    elif menu_choice == "3":
        try:
            file = open(FILE_NAME, "w")
            for student in students:
                csv_data = f"{student[0]},{student[1]},{student[2]}\n"
                file.write(csv_data)
            print("Data successfully saved to file.")
        except TypeError as e:
            print("Please check that the data is a valid string.")
            print("-- Technical Error Message --")
            print(e, e.__doc__, type(e), sep='\n')
        except Exception as e:
            print("-- Technical Error Message --")
            print("Built-In Python error info:")
            print(e, e.__doc__, type(e), sep='\n')
        finally:
            if file and not file.closed:
                file.close()
                print("File has been closed.")
```

Figure 6: Step 3 with Exceptions for Error Handling

Figure 6 illustrates “Option 3,” which saves the data. Essentially, it opens the file. Then, it inputs the csv data by placing the data in the correct column through comma separation. We use try-except-finally statements for error handling. The “try portion” starts the testing for exceptions. The “except” statements make sure the data has valid formatting. In other words, it doesn’t contain integers in the student’s name. The” finally” portion is always executed. In this script, it closes the file.

Custom Error Messaging

```
if menu_choice == "1":
    try:
        student_first_name = input("Enter the student's first name: ")
        if not all(x.isalpha() or x == '-' for x in student_first_name):
            raise ValueError("First name must only contain letters and hyphens.")

        student_last_name = input("Enter the student's last name: ")
        if not student_last_name.isalpha():
            raise ValueError("Last name must be a valid string.")

        course_name = input("Please enter the name of the course: ")
        if not all(x.isalpha() or x.isspace() for x in course_name):
            raise ValueError("Course name must only contain letters and spaces.")
```

Figure 7: raise ValueError() to inform user the issue

One of the error handling methods we learned this week was to use raise ValueError(“Custom Message”) to inform the user of the issues with the input data. Specifically, the if not statement,

which is used with “isalpha” and “isspace” to inform the program of the acceptance inputted replies. If the input data is not formatted to these restrictions, the raise ValueError() will state a specific customized message to inform the user of the issues.

GitHub

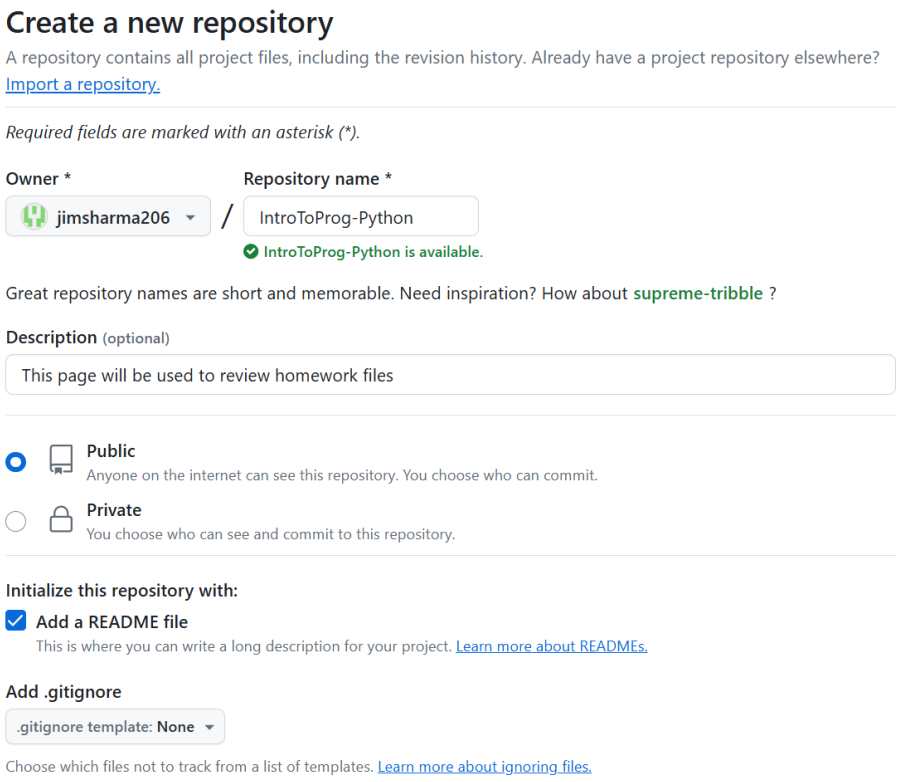

Figure 8: Create Repository on GitHub

Figure 8 demonstrates the creation of a repository on GitHub. GitHub is a platform that programmers use to ask questions and show their code, as well as projects. I had to create one for the Business Intelligence certificate and I have posted projects on there. These projects include quantitative text analysis and electoral maps.

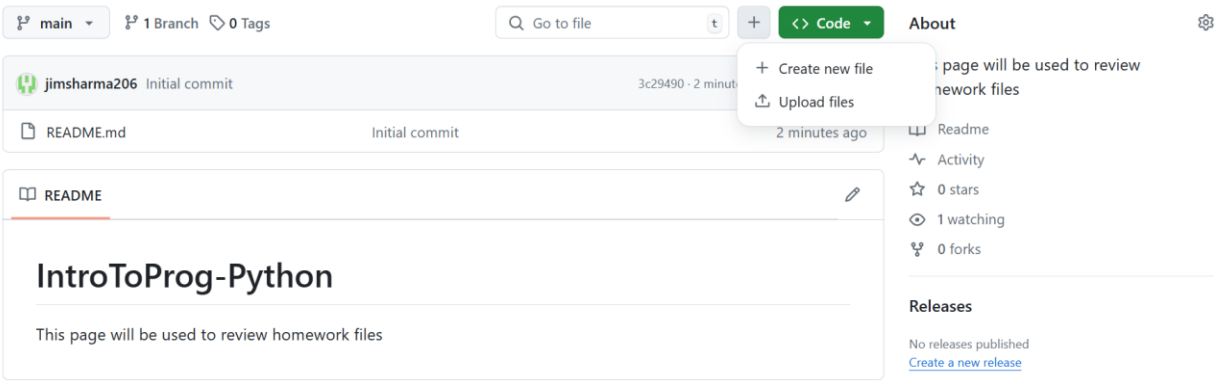

Figure 9: Upload Files to GitHub

Figure 9 illustrates how to upload files to the repository. For the rest of the assignments for this course, we will be required to upload our script and summary of steps and obstacles. This is great for showing to a potential employer and for recounting our steps, as well as the material we have learned in the course.

Conclusion

Assignment 5 focused on different techniques for error handling to ensure that users enter properly formatted data. We utilized try-except-finally blocks and raised ValueError exceptions with custom messages to handle specific errors. I encountered several challenges while working on the assignment. For example, I needed to adjust the code to accept spaces in course names and to accommodate hyphenated first names. Ultimately, we learned how to upload our script to GitHub.