Joseph Watts
Feb 25, 2025
IT FDN 110 A Wi 25: Foundations Of Programming Python
Assignment05

# Basics of Python:JSON Files and Error Handling

## Introduction

So far we have been using csv files to store our data. They have worked for our needs, but as we move from using lists to dictionaries, we need a more robust file type. JSON files support dictionaries as well as allowing a smoother file handling experience. We will also be adding error handling in the form of try except statements.

## Dictionaries

A dictionary is similar to a table in that it is a list of lists. Dictionaries support multiple data types. Unlike lists, dictionaries store data as key value pairs. Take this example where student is out dictionary:
student = { "FirstName": "John", "LastName": "Doe", "Age": 20, "Course": "Computer Science" }

If this was a list and we wanted to call on the first name value within the list, you would have to use indexing as such: print(sutdent[0]). This works fine enough, but this method is not intuitive. As your data grows more and more complex, it gets harder to keep track of the index for each value. Using dictionaries you can call on a value using the key as such:
print(student[“FirstName”])

This is much more user friendly as the key gives you information on the value you are calling upon. Furthermore, if you are saving your data to a file it makes it much easier to read since each value key gives you information on the value, something a list lacks.

## JSON Files

A JavaScript Object Notation, or JSON file, are files that are used in many different applications. They are structured like python dictionaries while being a text based representation. This makes reading and writing dictionaries to JSON files very easy. Like dictionaries, they are human readable, meaning you can open one up and make sense of the data inside. For these reasons, they are used for a wide range of applications.

## Reading and Writing to JSON

Reading and writing csv files takes some work, while writing you need to add the escape character \n which you then need to remove when you read the file using split function. Importing the json module will allow you to read and write json files with ease. As long as you are reading a json file, you can use the json.load(file) function. Similarly, as long as you are using dictionaries you can write to json by using json.dump(data, filename). You still have to follow file reading and writing practices such as opening and closing your files, but using the json module greatly simplifies the process.

## Error Handling

As we have experienced working in this course, python does have its own error handling. However, the error messages are not very user friendly, and throwing these errors will cause your script to exit. Adding our own error handling will not only allow us to provide more information on what caused the error, but will also keep the script from exiting, allowing the user to keep running the script. To do this we use the try and except blocks. To start, you contain your script within a try block. If the code within runs without issues, it will pass over the except block and continue running the rest of the script. However, if the script encounters an error, it will jump to the except block and run whatever code is within the except loop. This allows us to give custom error messages, as well as preventing the code from breaking. You can call on specific errors, use a general catch all, or a combination of both.

## GitHub

An important aspect of coding is sharing your code. While sending your script as a file does work, a common way of sharing script is GitHub. GitHub is a platform for hosting and sharing code repositories, offering version control, collaboration tools, and documentation support.

## Running the Script

Running the script through either IDLE or the commands will first import the json module. Then it will read the file “Ennrolments.json” that sits within the Output folder. Using the json module, the data will be saved to the students dictionary. After the data is read a while loop starts. It will print a menu of options, and prompt the user to enter an option from the menu. If they select the first option, they will be asked to enter their first and last name as well as the course name. After that the while loop starts again. If they then pick the second option the script will print the names of all the registered students and their class, including any names you may have just entered. If the third option is selected, the data is written to a file and the data being saved is printed into the terminal. Again we are using the json module to write our dictionary to our json file. Selecting option four exits the program. You may choose any of these options as many times as you like. Entering a number not between 1 and 4 will remind the user that they must choose one of those options.

## Summary

This assignment explores the transition from CSV files to JSON for data storage, as JSON supports dictionaries and provides a more structured file format. Dictionaries store data as key-value pairs, making data retrieval more intuitive compared to lists. Unlike lists, dictionaries allow accessing values using descriptive keys rather than numerical indexes. JSON files are text-based and structured similarly to dictionaries, making them ideal for storing and exchanging data. The json module simplifies reading (json.load()) and writing (json.dump()) operations. Error handling using try-except prevents script crashes by catching and handling specific errors. Lastly, GitHub is introduced as a platform for version control and sharing code, offering features like repository management and collaboration. These concepts enhance efficient data handling and improve code reliability.