

Assignment 07

Introduction

I've encountered more challenges this week but have learned many distinctions in how Python processes data. From file handling to naming confusions, it's been an interesting week of writing code.

Reading and writing custom objects to JSON

It took me several watches of the Assignment 07 review video to understand what exactly we needed to do with the custom object of "Student" we created in order to read existing data from a file to that object, and then write that object back to the JSON file.

For the former, we have to convert the list of dictionaries from JSON to our object (Figure 1). For the latter, we have to do the opposite: convert our object to a list of dictionaries to write to JSON properly (Figure 2). I'm not sure why it took me a few watches to really understand this concept (I'll blame it on the head cold), but I'm happy to finally appreciate it.

```
117 @staticmethod
118 def read_data_from_file(file_name: str, student_data: list[Student]) -> list[Student]:
119     """ This function reads data from a json file and loads it into a list of dictionary rows,
120     then converts the dictionary rows into the custom Student object
121
122     ChangeLog: (Who, When, What)
123     RRoot, 1.1.2030, Created function
124     RebeccaBergh, 8/12/2024, Loaded json into dictionary rows; added FileNotFoundError
125
126     :param file_name: string data with name of file to read from
127     :param student_data: list of dictionary rows to be filled with file data
128
129     :return: list
130     """
131     file_data = []
132     file = None
133
134     try:
135         file = open(file_name, "r")
136         file_data = json.load(file)
137         file.close()
138     except FileNotFoundError as e:
139         IO.output_error_messages(message="Error: The file does not exist.", e)
140         file = open(file_name, 'w')
141         print("Creating the file...")
142         file.close()
143     except Exception as e:
144         IO.output_error_messages(message="Error: There was a problem with reading the file.", e)
145     finally:
146         if not file.closed and file is not None:
147             file.close()
148     for row in file_data:
149         student_data.append(
150             Student(first_name=row['FirstName'],
151                     last_name=row['LastName'],
152                     course_name=row['CourseName'])
153         )
154     return student_data
```

Figure 1: The modified documentation line explains how this function works, step by step.

```

156     @staticmethod
157     def write_data_to_file(file_name: str, student_data: list[Student]):
158         """ This function first converts the custom Student object into a list
159             of dictionary rows, then writes those rows to a json file
160
161             ChangeLog: (Who, When, What)
162             RRoot,1.1.2030, Created function
163
164             :param file_name: string data with name of file to write to
165             :param student_data: list of dictionary rows to be written to the file
166
167             :return: None
168         """
169         # Convert Student object data into dictionary rows
170         file_data: list[dict[str, str]] = []
171         for student in student_data:
172             file_data.append({'FirstName': student.first_name,
173                             'LastName': student.last_name,
174                             'CourseName': student.course_name})
175         file = None
176         # write dictionary rows into JSON and display to end user
177         try:
178             file = open(file_name, "w")
179             json.dump(file_data, file, indent=1)
180             file.close()
181             IO.output_student_and_course_names(student_data=student_data)
182         except Exception as e:
183             message = "Error: There was a problem with writing to the file.\n"
184             message += "Please check that the file is not open by another program."
185             IO.output_error_messages(message=message, error=e)
186         # close file if not already
187         finally:
188             if not file.closed and file is not None:
189                 file.close()

```

Figure 2: The opposite step-by-step procedure is done to write data to the JSON file.

Challenges in naming

One of the biggest challenges for me when using Python is the “reuse” of certain names (attributes, properties, parameters). I’m used to programs that don’t allow duplicate names, even for things that are different in nature (parameter vs. variable for example), or if they do allow duplicate names, there’s a tooltip to indicate what kind of object it is. In Python, the same string (i.e. `student_data`) can be used as a parameter and as a variable. For me, I will probably need to append my parameter names with “p_” to ensure I don’t get confused.

Summary

Despite being ill this week, I did feel like I made a breakthrough in my learning when I realized exactly how the custom object (Student) was being used in relation to the JSON file. I also became a lot more familiar with how different functions relate to each other, and at what point an error gets called while running a program (i.e. either as the input is received, or when the function is finished running).