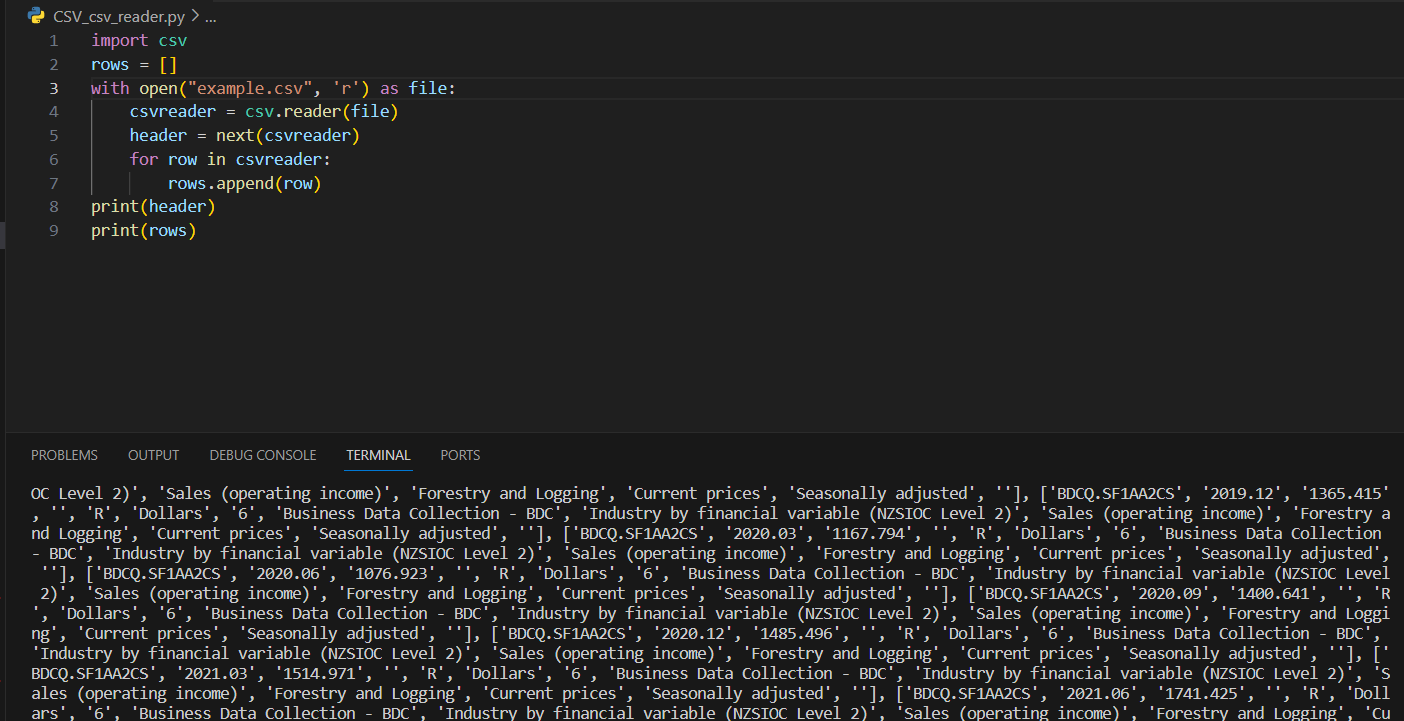
**Day 14 Assignment - 19/12/2023 - Vamsi Viswanadham**

CSV files, essential in data science, are plain text files used to store tabular data. Each line in a CSV file corresponds to a row in the table, and commas separate individual cells in that row.

**Reading CSV files:**

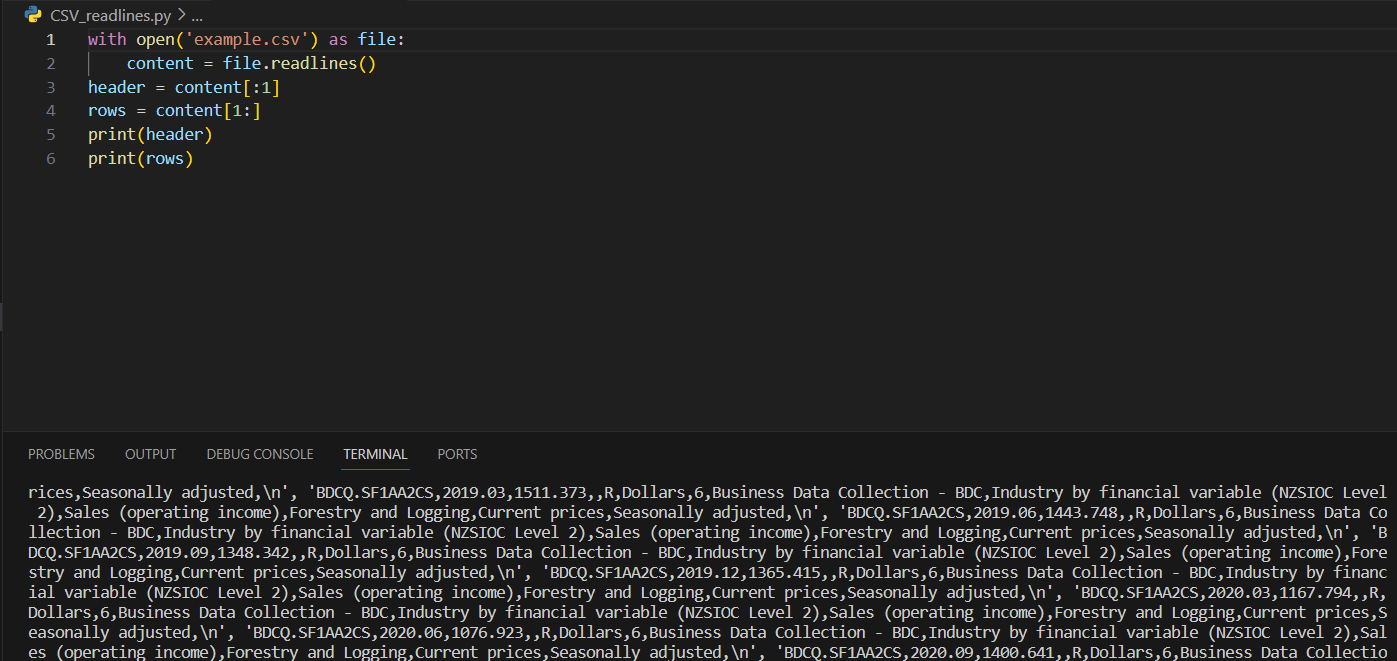
**Using csv.reader:** A built-in Python module, csv, provides the csv.reader method for reading CSV files. This method involves opening the file, creating a reader object, iterating over rows, and finally closing the file.

Here is a simple example:



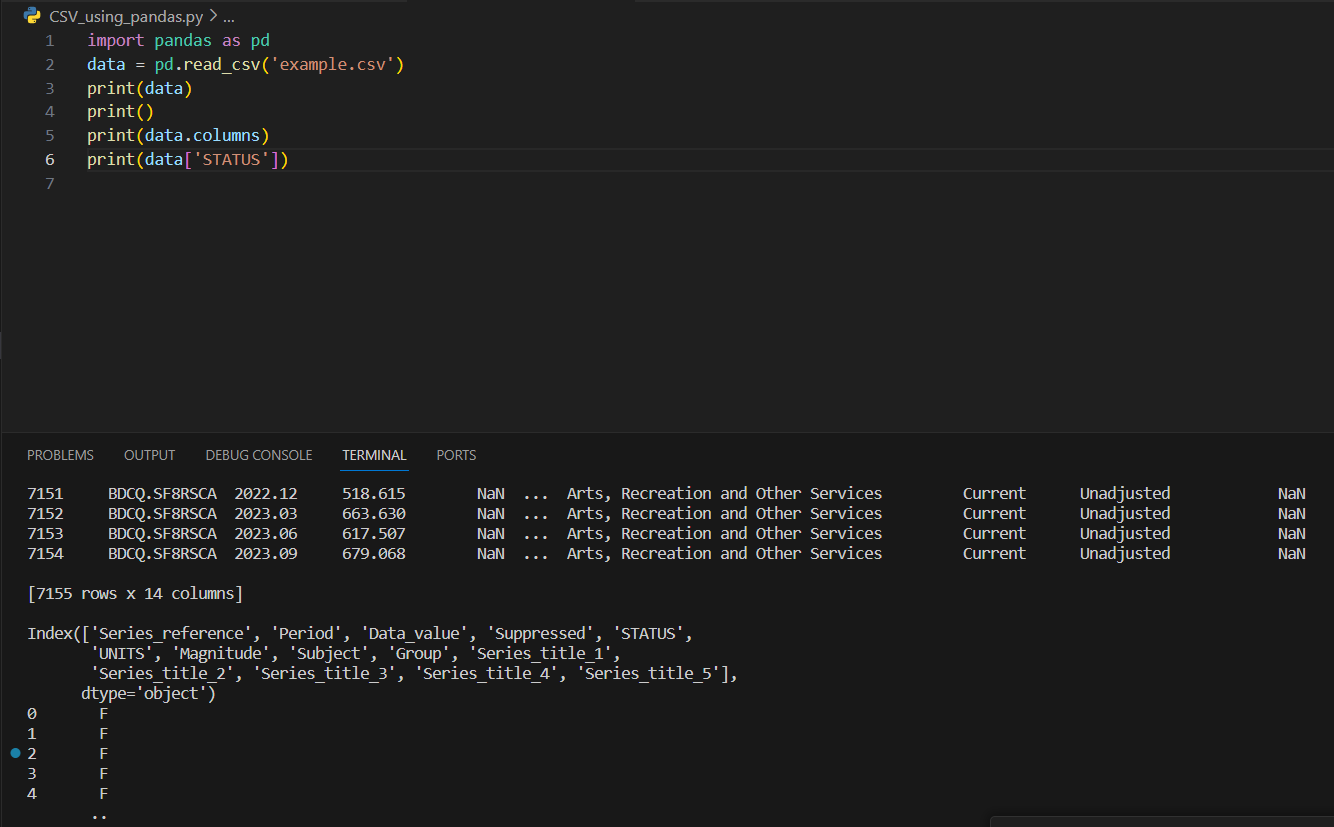
**Using .readlines() Method:** For simpler operations, the .readlines() method can be employed. It reads the entire file as a list of strings, where each string is a line in the CSV file.

Here is an example:



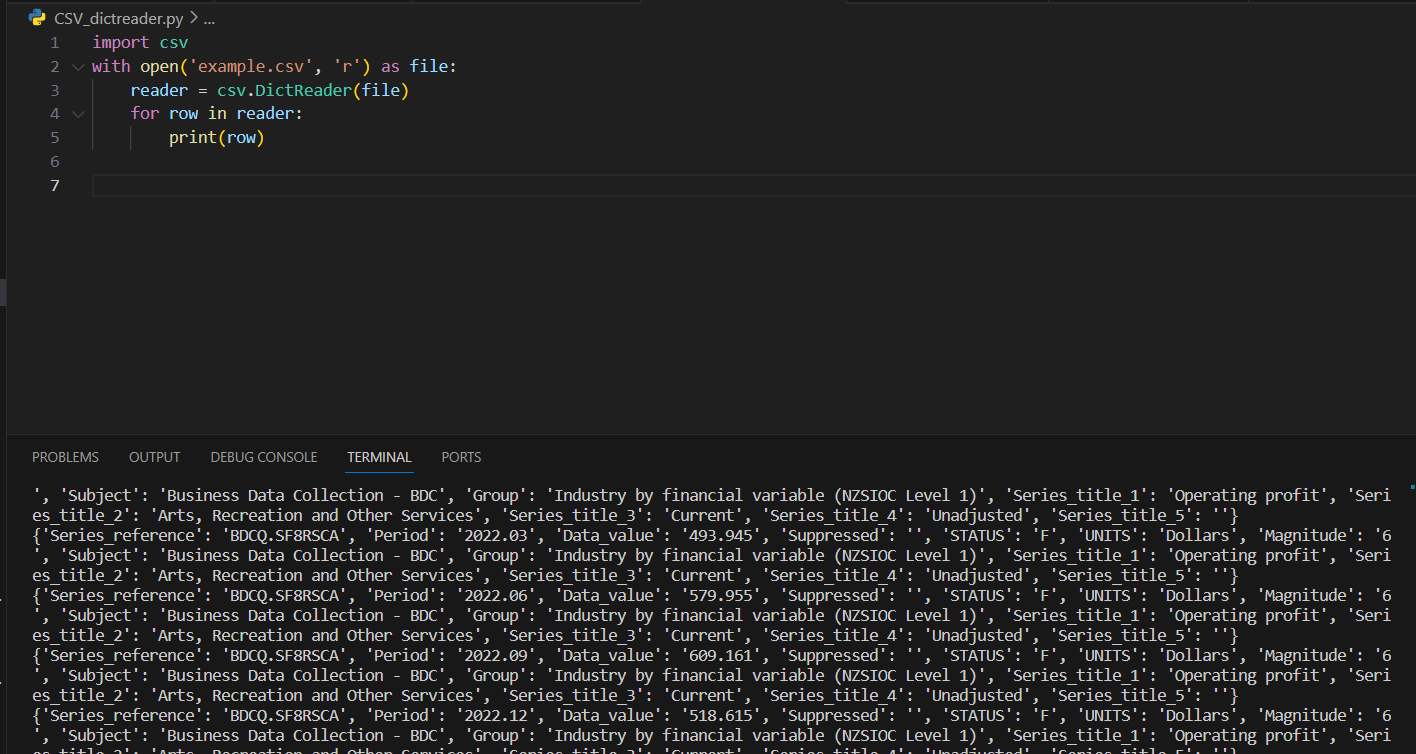
**Using Pandas:** For more complex data handling, the Pandas library is preferred. It provides the read\_csv function, which is highly efficient for reading large datasets and offers extensive data manipulation capabilities.

Here is an example:



**Using csv.DictReader:** This method reads the CSV file into a dictionary, making the data handling more intuitive, especially when dealing with files with headers.

Here is an example:



**Writing CSV Files in Python**

Writing data into CSV format is a common task in data processing. Python offers various methods to write data to CSV files, making it an invaluable tool for data manipulation and storage.

**Methods for Writing CSV Files:**

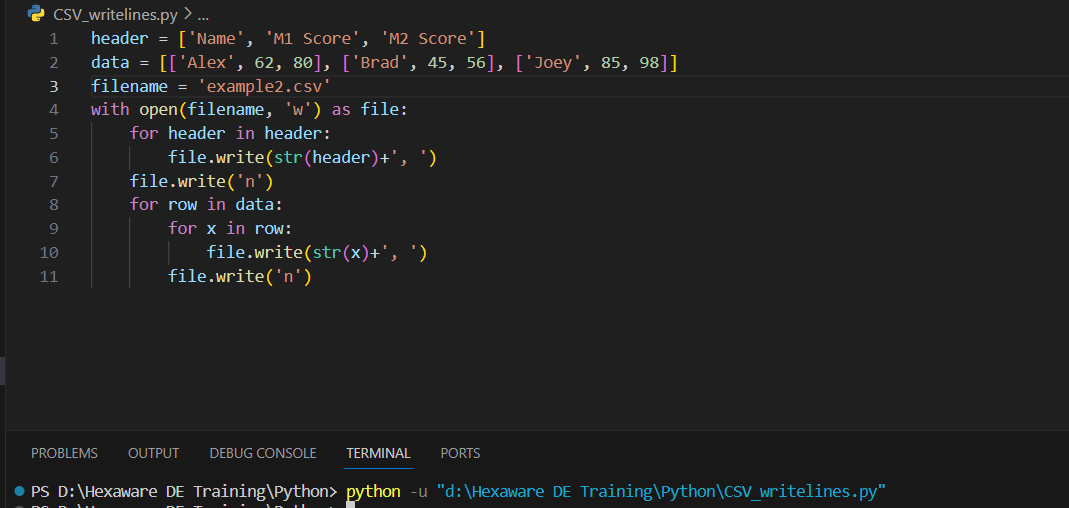
**Using csv.writer:** This method involves creating a writer object that converts your data into a CSV formatted string. It allows you to write rows one at a time.

**Using .writelines():** For a more direct approach, .writelines() takes a list of strings and writes them to a file. This method requires manual handling of delimiters and line terminators.

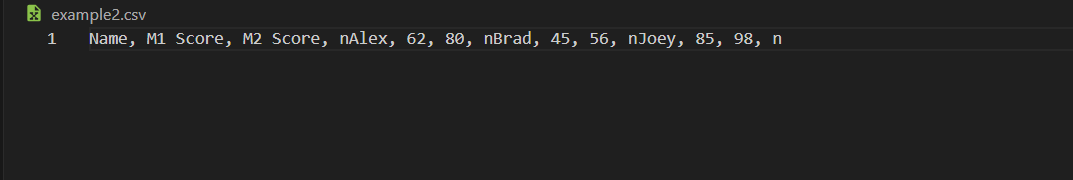
**Using Pandas:** Pandas provides a to\_csv method, part of the DataFrame class, which is useful for writing complex data structures to CSV files efficiently. This method offers flexibility in specifying delimiters, headers, and index options.

**Using csv.DictWriter:** This method is particularly useful when your data is in the form of a list of dictionaries. It allows you to write each dictionary to a row in the CSV, with keys as column headers.

Here is a simple example of writing Data into CSV File.



Here is the file in which the data is written.



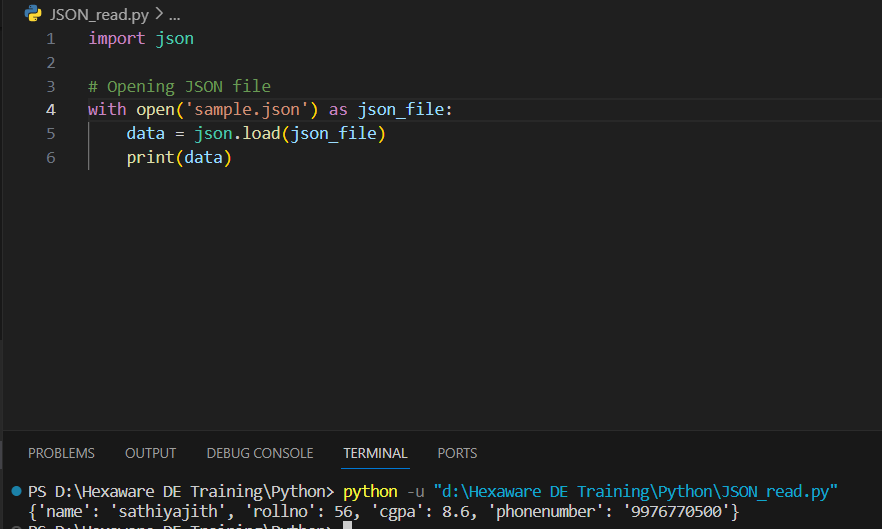
JSON:

JSON, which is an acronym for JavaScript Object Notation, refers to a text format used in programming for storing and transferring data. This format, resembling executable script files, is fully supported in Python through a built-in module named 'json'. To utilize this functionality, one must import the 'json' package into their Python script.

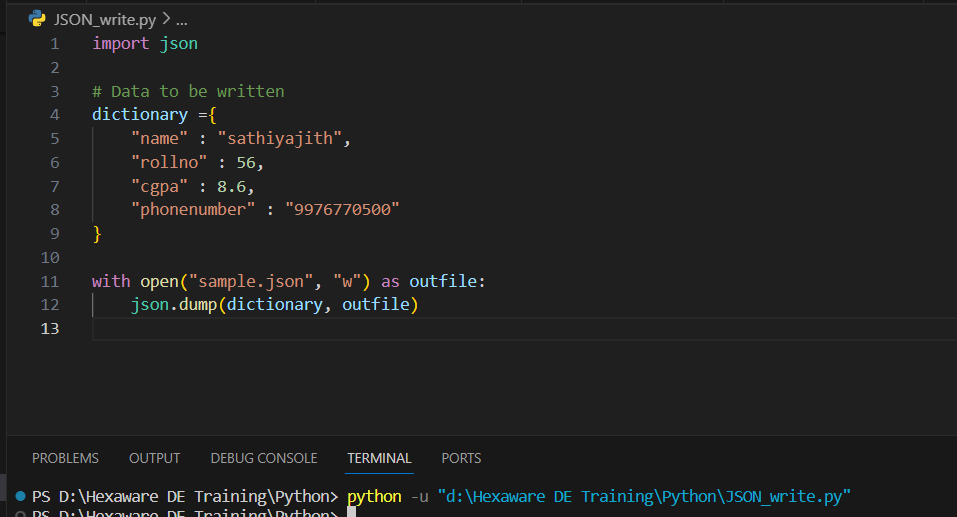
**json.load():** json.load() function is present in Python built-in ‘JSON’ module. This function is used to parse the JSON string.

**json.loads():** json.loads() function is present in Python built-in ‘json’ module. This function is used to parse the JSON string.

Here is a simple example for reading JSON data:



Here is a simple example for writing JSON data to the file:



Here is the data that is written:

