1. What advantages do Excel spreadsheets have over CSV spreadsheets?

Ans:

In summary, the advantages of Excel spreadsheets over CSV spreadsheets include:

1. Rich formatting and styling options for visually appealing reports.
2. Built-in formulas and functions for complex calculations and data analysis.
3. Data validation and protection features for secure data management.
4. Support for macros and automation using VBA (Visual Basic for Applications).
5. Capability to manage multiple sheets and workbooks, enabling efficient organization of large datasets.
6. Powerful tools like pivot tables, sorting, filtering, and data analysis add-ins for in-depth data analysis and insights.

2.What do you pass to csv.reader() and csv.writer() to create reader and writer objects?

Ans: To create reader and writer objects in the csv module in Python, you pass file objects as arguments to csv.reader() and csv.writer(). These file objects should be opened in the appropriate mode (e.g., 'r' for reading and 'w' for writing) and should be associated with CSV files.

3. What modes do File objects for reader and writer objects need to be opened in?

Ans: File objects for reader and writer objects in the csv module need to be opened in the following modes:

1. **Reader Object (csv.reader):** Open the file in the 'r' mode (read mode).
2. **Writer Object (csv.writer):** Open the file in the 'w' mode (write mode).

Additionally, it's crucial to handle the newline parameter appropriately, setting it to '' to ensure proper handling of line endings when working with CSV files.

4. What method takes a list argument and writes it to a CSV file?

Ans: The writerow method in the csv.writer object is used to write a list of values to a row in a CSV file. This method takes a list as an argument, where each element of the list represents a field in the CSV row. The writerow method automatically formats the list elements according to the CSV formatting rules and writes them to the file as a new row.

5. What do the keyword arguments delimiter and line terminator do?

Ans: In the context of the csv module in Python, the keyword arguments delimiter and lineterminator provide control over the delimiters used between fields and the line termination characters, respectively, when reading from or writing to a CSV file.

1. **Delimiter:** The delimiter argument specifies the character used to separate fields in the CSV file. By default, it is set to a comma (','), but you can specify a different delimiter character if the CSV file uses a different separator, such as a tab ('\t') or a semicolon (';').
2. **Lineterminator:** The lineterminator argument controls the character sequence used to terminate each line in the CSV file. By default, it is set to '\r\n' (carriage return + line feed), which is the standard line terminator for CSV files on Windows. You can specify a different line terminator, such as '\n' for Unix-based systems or '\r' for older Mac systems, depending on the target platform or specific requirements.

These keyword arguments allow you to customize the formatting of the CSV file based on the specific delimiters and line termination characters needed for your use case.

6. What function takes a string of JSON data and returns a Python data structure?

Ans: The json.loads() function in Python is used to parse a string containing JSON (JavaScript Object Notation) data and convert it into a corresponding Python data structure. The loads() function is part of the json module and is specifically designed for deserializing JSON data into Python objects.

7. What function takes a Python data structure and returns a string of JSON data?

Ans: The json.dumps() function in Python is used to serialize a Python data structure into a JSON formatted string. The dumps() function is part of the json module and is specifically designed to convert Python objects to their corresponding JSON string representations.