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

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

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

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

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

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

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

**Answers**

1. **Advantages of Excel Spreadsheets over CSV Spreadsheets**:
   * Excel spreadsheets offer more advanced features and functionality compared to CSV files.
   * Advantages of Excel:
     + Formatting options (colors, fonts, cell styles).
     + Data analysis tools (pivot tables, charts, formulas).
     + Calculations and formulas.
     + Graphical representation of data.
     + Multiple sheets within a single file.
   * [CSV files, on the other hand, are lightweight and simple, suitable for basic tabular data storage without any advanced features](https://www.howtoexcel.org/csv-vs-excel/)[1](https://www.howtoexcel.org/csv-vs-excel/).
2. **Creating Reader and Writer Objects with csv.reader() and csv.writer()**:
   * To create a reader object, use csv.reader(csvfile, dialect='excel', \*\*fmtparams).
   * To create a writer object, use csv.writer(csvfile, dialect='excel', \*\*fmtparams).
   * Example:
   * import csv
   * with open('data.csv', 'r') as csvfile:
   * reader = csv.reader(csvfile)
   * for row in reader:
   * print(row)
3. **Modes for File Objects**:
   * File objects for reader and writer need to be opened in specific modes:
     + r: Read mode (for reading data from the file).
     + w: Write mode (for writing data to the file, truncating existing content).
     + a: Append mode (for adding data to the end of the file).
     + r+: Read and write mode (for both reading and writing).
4. **Method to Write a List to a CSV File**:
   * You can use the writerow() method of the writer object to write a list (or multiple lists) to a CSV file.
   * Example:
   * import csv
   * data = [[1, 'Alice', 25], [2, 'Bob', 30]]
   * with open('output.csv', 'w', newline='') as csvfile:
   * writer = csv.writer(csvfile)
   * writer.writerows(data)
5. **Keyword Arguments delimiter and line terminator**:
   * delimiter: Specifies the character used to separate fields in the CSV file (default is a comma ,).
   * line terminator: Specifies the character(s) used to terminate lines (default is newline \n).
   * Example:
   * with open('data.csv', 'w', newline='') as csvfile:
   * writer = csv.writer(csvfile, delimiter=';', lineterminator='\r\n')
   * writer.writerow(['Name', 'Age'])
   * writer.writerow(['Alice', 25])
6. **Function to Convert JSON Data to Python Data Structure**:
   * Use json.loads(json\_string) to convert a JSON string to a Python data structure (dictionary, list, etc.).
   * Example:
   * import json
   * json\_data = '{"name": "Alice", "age": 25}'
   * python\_dict = json.loads(json\_data)
7. **Function to Convert Python Data Structure to JSON String**:
   * Use json.dumps(python\_object) to convert a Python data structure to a JSON string.
   * Example:
   * python\_dict = {'name': 'Alice', 'age': 25}
   * json\_string = json.dumps(python\_dict)