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

:- Multiple sheets: Excel allows you to create multiple sheets within a single workbook, which makes it easy to organize and analyze large amounts of data. CSV files store only one sheet of data per file.

Data types: Excel allows you to store data in various formats, including text, numbers, dates, and formulas, whereas CSV files store only plain text.

Formatting: Excel allows you to apply formatting to your data, such as font styles, colors, borders, and alignment, whereas CSV files have no formatting options.

Data validation: Excel allows you to define rules and constraints for your data, such as requiring certain values or preventing duplicate entries, whereas CSV files have no data validation options.

Charts and graphs: Excel allows you to create charts and graphs to visualize your data, whereas CSV files require external software to create visualizations.

Built-in functions: Excel has a wide range of built-in functions for performing calculations and analysis on your data, whereas CSV files require external software or programming skills to perform the same tasks.

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

:- To create a reader object writer object in the csv module, you pass a file object and optionally specify the delimiter character, quote character, and other parameters.

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

:- To create a reader object, we need to open the file in read mode ('r')

To create a writer object, we need to open the file in write mode ('w'):

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

:- To write a list of data to a CSV file in Python's csv module, we can use the writerow() method of a writer object. This method takes a list as an argument, where each element of the list represents a value to be written to a separate column in the CSV file.

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

:- The delimiter argument specifies the character that separates individual fields in each row of the CSV file. By default, the csv.reader and csv.writer classes use a comma (,) as the delimiter character. However, we can use the delimiter keyword argument to specify a different delimiter character, such as a tab (\t), a semicolon (;), or a pipe (|)

The lineterminator argument specifies the character(s) that should be used to terminate each row of the CSV file. By default, the csv.writer class uses the platform-specific line terminator (\r\n on Windows, \n on Unix-like systems) as the line terminator character. However, we can use the lineterminator keyword argument to specify a different line terminator character, such as a carriage return (\r), a line feed (\n), or a combination of both (\r\n).

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

:- The json.loads() function is used to parse a JSON-formatted string and convert it into a Python data structure.

We can use the json.loads() function to parse this string and convert it into a Python dictionary.

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

:- The json.dumps() function is used to convert a Python data structure into a JSON-formatted string and then we can use the json.dumps() function to convert this dictionary into a JSON-formatted string.