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

Formatting and Styling: Excel supports rich formatting and styling options, allowing you to apply various font styles, colors, cell borders, conditional formatting, and more. This makes it easier to create visually appealing and organized spreadsheets compared to plain-text CSV files.

Formulas and Functions: Excel provides a wide range of built-in formulas and functions that can be used for calculations, data analysis, and automation. You can perform complex calculations, create formulas that reference other cells or sheets, and leverage functions like SUM, AVERAGE, COUNT, IF statements, and more. CSV files, on the other hand, contain only plain text data and do not support formulas or functions.

Multiple Sheets and Workbooks: Excel allows you to organize data into multiple sheets within a single workbook and link data between sheets. This enables you to create complex data structures, build relationships, and perform comprehensive analysis across different sheets. CSV files, in contrast, represent a single table of data without the concept of sheets or workbooks.

Data Validation and Protection: Excel provides features for data validation, where you can define rules to restrict input values, enforce data types, and prevent errors. Additionally, you can protect cells, sheets, or the entire workbook with passwords to control access and prevent unauthorized modifications. These data validation and protection features are not available in CSV files.

Charts and Graphs: Excel offers powerful charting capabilities, allowing you to create various types of charts and graphs to visualize data. You can customize chart styles, labels, legends, and other chart elements. This makes it easier to present data in a visually appealing and understandable manner. CSV files do not have built-in charting capabilities.

Data Sorting and Filtering: Excel provides convenient tools for sorting and filtering data based on specific criteria. You can sort data in ascending or descending order and apply filters to display specific data subsets. These data manipulation features are not available directly in CSV files, requiring additional programming or data processing steps.

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

To create reader and writer objects using the csv.reader() and csv.writer() functions from the csv module in Python, you need to pass file objects as arguments. Here's how you can create reader and writer objects:

1. Creating a reader object:
   * Pass a file object opened in text mode as the argument to csv.reader().
   * The file object can be opened using the open() function or obtained from an existing file.

Creating a writer object:

* Pass a file object opened in text mode as the argument to csv.writer().
* The file object can be opened using the open() function in write or append mode.

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

When working with reader and writer objects from the csv module in Python, you need to open the File objects in specific modes depending on whether you intend to read or write to the CSV file. Here are the recommended modes for File objects used with reader and writer objects:

Reader object:

* + The File object used with a reader object should be opened in text mode with the "r" mode.
  + This allows the reader to read data from the file.

Writer object:

* The File object used with a writer object should be opened in text mode with the "w" or "a" mode.
* Use "w" mode if you want to create a new CSV file or overwrite an existing file.
* Use "a" mode if you want to append data to an existing CSV file.

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

The writerow() method is used to write a list argument to a CSV file using a writer object from the csv module in Python.

Here's how you can use the writerow() method to write a list to a CSV file:

import csv

# Open the CSV file in write mode

with open("output.csv", "w", newline="") as file:

writer = csv.writer(file)

# Write a list to the CSV file

data = ["John", 25, "example@example.com"]

writer.writerow(data)

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

The keyword arguments delimiter and line\_terminator are used in the csv.writer() function from the csv module in Python to control the formatting of the CSV file.

delimiter:

* + The delimiter keyword argument specifies the character used to separate fields (columns) within each row of the CSV file.
  + By default, the delimiter is set to a comma (,), which is why CSV stands for "Comma-Separated Values".
  + You can specify a different delimiter character if desired, such as a semicolon (;), a tab (\t), or any other character.

line\_terminator:

* The line\_terminator keyword argument specifies the character sequence used to terminate each line (row) of the CSV file.
* By default, the line\_terminator is set to "\r\n" (CRLF), which is the standard line ending sequence for CSV files on most systems.
* You can specify a different line terminator if needed, such as "\n" (LF) for Unix-like systems or "\r" (CR) for older Mac OS systems.

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

The json.loads() function in Python is used to parse a string of JSON data and convert it into a Python data structure. The loads() function stands for "load string" and is part of the json module.

Here's how you can use the json.loads() function:

import json

json\_data = '{"name": "John", "age": 25, "city": "New York"}'

python\_data = json.loads(json\_data)

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

The json.dumps() function in Python is used to convert a Python data structure into a string of JSON data. The dumps() function stands for "dump string" and is part of the json module.

Here's how we can use the json.dumps() function:

import json

python\_data = {"name": "John", "age": 25, "city": "New York"}

json\_data = json.dumps(python\_data)