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

Ans:-

* Formatting: Excel allows you to format cells and data in a variety of ways, including font size and color, cell borders, and conditional formatting. This can make the data easier to read and analyze.
* Formulas and Functions: Excel has a built-in library of formulas and functions that can be used to perform calculations and manipulate data. This can save time and effort when working with large datasets.
* Charts and Graphs: Excel makes it easy to create charts and graphs from your data, which can help to visualize trends and patterns.
* Macros and VBA: Excel allows you to create macros and use VBA (Visual Basic for Applications) to automate tasks and customize the software to your needs.
* Collaboration: Excel allows multiple users to work on a spreadsheet at the same time, which can be useful for collaborative projects.
* Data Validation: Excel allows you to set rules for data entry, which can help to ensure data accuracy and consistency.
* Security: Excel allows you to protect your spreadsheet with a password and control who can view or edit it. This can be useful for sensitive data or confidential information.

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

Ans:-

To create a reader object in Python's csv module, you pass a file object containing the CSV data to the csv.reader() function. The file object can be obtained by opening the CSV file using the built-in open() function, for example:

import csv

with open('example.csv', 'r') as csv\_file:

reader = csv.reader(csv\_file)

To create a writer object in Python's csv module, you pass a file object and specify the file mode as w or a (for write or append mode) to the csv.writer() function. The file object can also be obtained by opening a CSV file using the built-in open() function, for example:

import csv

with open('output.csv', 'w', newline='') as csv\_file:

writer = csv.writer(csv\_file)

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

Ans:-

For a reader object, you would need to open the File object in "read" mode, which is specified by the letter 'r' as the second argument to the open() function. For example:

file\_obj = open('filename.txt', 'r')

For a writer object, you would need to open the File object in "write" mode, which is specified by the letter 'w' as the second argument to the open() function. For example:

file\_obj = open('filename.txt', 'w')

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

Ans:-

The csv.writer() method in Python can be used to write a list of values to a CSV file. Here is an example code snippet that demonstrates this:

import csv

data = [['John', 25], ['Mary', 30], ['David', 35]]

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

writer = csv.writer(file)

writer.writerows(data)

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

Ans:-

The keyword argument delimiter is used in the csv module in Python to specify the character that separates fields in a CSV file. By default, the delimiter is set to a comma (','), but it can be changed to any other character, such as a semicolon (';') or a tab character ('\t'), depending on the specific requirements of the CSV file.

import csv

data = [['John', 'Doe', '25'], ['Mary', 'Smith', '30'], ['David', 'Jones', '35']]

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

writer = csv.writer(file, delimiter=';')

writer.writerows(data)

The keyword argument lineterminator is used to specify the character sequence that should be used to terminate lines in the CSV file. By default, the lineterminator is set to '\r\n', which represents the standard Windows line ending. However, this can be changed to any other sequence of characters, such as '\n' for Unix-style line endings.

import csv

data = [['John', 'Doe', '25'], ['Mary', 'Smith', '30'], ['David', 'Jones', '35']]

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

writer = csv.writer(file, lineterminator='\n')

writer.writerows(data)

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 JSON string and convert it into a Python data structure.

Here is an example code snippet that demonstrates the use of json.loads():

import json

json\_string = '{"name": "John", "age": 30, "city": "New York"}'

python\_dict = json.loads(json\_string)

print(python\_dict)

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

Ans:-

The json.dumps() function in Python takes a Python data structure (such as a dictionary, list, or tuple) and returns a string of JSON data. Here's an example:

import json

data = {

"name": "John",

"age": 30,

"city": "New York"

}

json\_string = json.dumps(data)

print(json\_string)