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?

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

When working with reader and writer objects in the `csv` module of Python, the File objects associated with these objects need to be opened in specific modes. The modes for File objects used with reader and writer objects are as follows:

1. Reader Objects:
   - The File object used with a reader object should be opened in **'r'** mode, which stands for read mode. This allows the reader object to read data from the file. Therefore, the file should be opened with `open('filename.csv', 'r')` or `open('filename.csv', mode='r')`.
2. Writer Objects:
   - The File object used with a writer object should be opened in **'w'** mode, which stands for write mode. This enables the writer object to write data to the file. Therefore, the file should be opened with `open('filename.csv', 'w')` or `open('filename.csv', mode='w')`.

Additionally, when writing to CSV files, it is recommended to specify `newline=''` as an argument when opening the file. This ensures that the proper line endings are used in the CSV file across different platforms.

Here's an example of opening files in the required modes for reader and writer objects:

Remember to adjust the filenames and file paths according to your specific scenario.

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

```
import csv

data = ['John', 'Doe', 'john.doe@example.com']

# Open the CSV file in write mode
with open('output.csv', 'w', newline='') as file:
    csv_writer = csv.writer(file)

    # Write the list as a single row
    csv_writer.writerow(data)
```

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?

```python
import json

data = {
    "name": "John",
    "age": 30,
    "city": "New York"
}

# Convert the Python data structure to a JSON string
json_string = json.dumps(data)

# Print the JSON string
print(json_string)
```

```
{"name": "John", "age": 30, "city": "New York"}
```

```json
{"name": "John", "age": 30, "city": "New York"}
```

```
{'name': 'John', 'age': 30, 'city': 'New York'}
```