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

* It is a binary file that holds information about all the worksheets in a workbook
* An Excel not only stores data but can also do operations on the data
* Files saved in excel cannot be opened or edited by text editors
* Large files user is much easier in Excel for the end user. Also, you can have additional functions like selecting individual cells for import, convert dates and time automatically, reading formulas and their results, filters, sorting, etc
* Apart from text, data can also be stored in form of charts and graphs
* Excel can connect to external data sources to fetch data. You can use custom add-in in Excel to increase its functionality.
* Excel allows for Review of Data with detailed tracking and commenting feature
* In Excel, spreadsheets can have values of data types other than strings; cells can have different fonts, sizes, or color settings; cells can have varying widths and heights; adjacent cells can be merged

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

To read a CSV file in Python, we can use the csv.reader() function. We pass a File object, obtained from a call to open().

**Code:**

import csv

file\_csv = open('abc.csv')

reader = csv.reader(file\_csv)

for row in reader:

print(row)

The csv. writer() function returns a writer object that converts the user's data into a delimited string. This string can later be used to write into CSV files using the writerow() function.

**Code:**

import csv

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

writer = csv.writer(file)

writer.writerow(["SN", "Movie", "Protagonist"])

writer.writerow([1, "Lord of the Rings", "Frodo Baggins"])

writer.writerow([2, "Harry Potter", "Harry Potter"])

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

r: Opens a file for reading only

r+: Opens a file for both reading and writing

w: Opens a file for writing only

w+: Open a file for writing and reading.

a: Opens a file for appending

a+: Opens a file for both appending and reading

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

The most common method to write data from a list to CSV file is the writerow() method of writer and DictWriter class.

**Code:**

import csv

# data to be written row-wise in csv file

data = [['a', ‘b’, ‘c’], [1, 2, 3]]

# opening the csv file in 'w+' mode

file = open('abc.csv', 'w+', newline ='')

# writing the data into the file

with file:

write = csv.writer(file)

write.writerows(data)

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

The delimiter is the character that appears between cells on a row. By default, the delimiter for a CSV file is a comma. The line terminator is the character that comes at the end of a row. By default, the line terminator is a newline.

The delimiter argument is used to change the string used to separate cells in a row.

The line terminator argument is used to change the string used to separate rows.

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

If you have a JSON string, you can parse it by using the json.loads() method. The result will be a Python dictionary.

**Code:**

import json

# some JSON:

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

# parse x:

y = json.loads(x)

# the result is a Python dictionary:

print(y["age"])

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

If you have a Python object, you can convert it into a JSON string by using the json.dumps() method.

**Code:**

import json

# a Python object (dict):

x = {

"name": "John",

"age": 30,

"city": "New York"

}

# convert into JSON:

y = json.dumps(x)

# the result is a JSON string:

print(y)