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?

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

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

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?

Solutions :

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

**Ans:** The Advantages of Excel over CSV are:

1. Excel (XLS and XLSX) file formats are better for storing and analysing complex data.
2. An Excel not only stores data but can also do operations on the data using macros, formulas etc
3. CSV files are plain-text files, Does not contain formatting, formulas, macros, etc. It is also known as flat files

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

In [1]:

**import** csv

**with** open('text.csv','r') **as** file:

csv\_file **=** csv**.**reader(file,delimiter**=**',')

**for** ele **in** csv\_file:

print(ele)

['name,department,birthday month']

['John Smith,Accounting,November']

['Erica Meyers,IT,March']

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

**Ans:** For csv.reader(iterable\_file\_object), the file objects needed to be opened in read mode mode='r' Whereas for csv.writer(iterable\_file\_object) the file objects needed to be opened in write mode mode='w'

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

**Ans:** csv.writer class provides two methods for writing to CSV. They are writerow() and writerows(). writerow() method writes a single row at a time. Whereas writerows() method is used to write multiple rows at a time.

In [2]:

*# Example Program*

**import** csv

fields **=** ['Name', 'Branch', 'Year', 'CGPA'] *#column names*

rows **=** [

['Nikhil', 'COE', '2', '9.0'], *# data rows of csv file*

['Sanchit', 'COE', '2', '9.1'],

['Ravi', 'IT', '2', '9.3']

]

**with** open("university\_records.csv", 'w') **as** csvfile:

csvwriter **=** csv**.**writer(csvfile) *# creating a csv writer object*

csvwriter**.**writerow(fields) *# writing the fields*

csvwriter**.**writerows(rows) *# writing the data rows*

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

**Ans:** Lets take the example of a csv file:  
First Name, Last Name, Age  
Mano, Vishnu, 24  
Vishnu, Vardhan, 21  
Here ',' is Delimiter. We can use any Character as per our needs if required. Similarly Line Terminator comes at end of line by default it is newline and can be changed accourding to Requirement.

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

**Ans:** loads() method takes a string of JSON data and returns a Python data structure

In [3]:

*# Example of json.loads() method*

**import** json

my\_details\_json **=**'''{

"Name": "Mano Vishnu",

"Qualification": "Bachelor of Technology",

"Stream": "Computer Science and Engineering"

}'''

print(my\_details\_json)

print(f'Type of my\_details\_json is {type(my\_details\_json)}')

my\_details **=** json**.**loads(my\_details\_json)

print(my\_details)

print(f'Type of my\_details is {type(my\_details)}')

{

"Name": "Mano Vishnu",

"Qualification": "Bachelor of Technology",

"Stream": "Computer Science and Engineering"

}

Type of my\_details\_json is <class 'str'>

{'Name': 'Mano Vishnu', 'Qualification': 'Bachelor of Technology', 'Stream': 'Computer Science and Engineering'}

Type of my\_details is <class 'dict'>

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

**Ans:** dumps() method takes a python data structure and returns a string of JSON data

In [4]:

*# Example of json.dumps() method*

**import** json

my\_details **=** {

'Name':'Mano Vishnu',

'Stream':'Computer Science and Engineering',

'Qualification':'Bachelor of Technology'

}

print(my\_details)

print(f'Type of my\_details is {type(my\_details)}')

my\_details\_json **=** json**.**dumps(my\_details, indent**=**4, sort\_keys**=True**)

print(my\_details\_json)

print(f'Type of my\_details\_json is {type(my\_details\_json)}')

{'Name': 'Mano Vishnu', 'Stream': 'Computer Science and Engineering', 'Qualification': 'Bachelor of Technology'}

Type of my\_details is <class 'dict'>

{

"Name": "Mano Vishnu",

"Qualification": "Bachelor of Technology",

"Stream": "Computer Science and Engineering"

}

Type of my\_details\_json is <class 'str'>