Assignment 13 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?

ANS: You pass a File object, obtained from a call to open().

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 [1]:

```
# Example Program
import csv
fields = ['Name', 'Branch', 'Year', 'CGPA']
rows = [
            ['Abhi', 'COE', '2', '9.1'],
            ['Harish', 'COE', '2', '7.8'],
            ['Tejas', 'IT', '2', '8.9']
        ]
with open("university_records.csv", 'w') as csvfile:
    csvwriter = csv.writer(csvfile)
    csvwriter.writerow(fields)
    csvwriter.writerows(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
Gaurav, Ratan, 28
Krish, Naik, 30
```

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:method takes a string of `json.loads()` data and returns a Python data structure

In [6]:

```
# Example of json.dumps() method
import json
my_details = {
    "Name": "Abhi Rajk",
    "Qualification": "Computer Science And MBA",
    "Stream": "Marketing and Analytics"
}
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": "Abhi Rajk",
    "Qualification": "Computer Science And MBA",
    "Stream": "Marketing and Analytics"
}
Type of my_details_json is <class 'str'>
{'Name': 'Abhi Rajk', 'Qualification': 'Computer Science And MBA', 'Stream':
'Marketing and Analytics'}
Type of my_details is <class 'dict'>
```

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

ANS: method takes a python data structure and returns a string of json.dumps().

In [4]:

```python
# Example of json.dumps() method
import json
my_details = {
    "Name": "Abhi Rajk",
    "Qualification": "Computer Science And MBA",
    "Stream": "Marketing and Analytics"
}
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': 'Abhi Rajk', 'Qualification': 'Computer Science And MBA', 'Stream':
'Marketing and Analytics'}
Type of my_details is <class 'dict'>
{
    "Name": "Abhi Rajk",
    "Qualification": "Computer Science And MBA",
    "Stream": "Marketing and Analytics"
}
Type of my_details_json is <class 'str'>
```