

Assignment 13

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

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 [11]: import csv
with open('username.csv','r') as file:
    csv_file = csv.reader(file,delimiter=',')
    for ele in csv_file:
        print(ele)
```

```
['Name', 'Branch', 'Year', 'CGPA']
[]
['Nikhil', 'COE', '2', '9.0']
[]
['Sanchit', 'COE', '2', '9.1']
[]
['Ravi', 'IT', '2', '9.3']
[]
```

In []:

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

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?

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 [10]: # 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("username.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?

Lets take the example of a csv file:

First Name, Branch, Year, CGPA

Nikhil, COE, 2, 9.0

Sanchit, COE, 2, 9.1

Ravi, IT, 2, 9.3

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 according to Requirement.

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

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

```
In [13]: # 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?

`dumps()` method takes a python data structure and returns a string of JSON data

```
In [16]: # Example of json.dumps() method
import json
my_details = {
    'Name': 'Nikhilesh Singh',
    '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': 'Nikhilesh Singh', 'Stream': 'Computer Science and Engineering', 'Qualification': 'Bachelor of Technology'}
Type of my_details is <class 'dict'>
{
    "Name": "Nikhilesh Singh",
    "Qualification": "Bachelor of Technology",
    "Stream": "Computer Science and Engineering"
}
Type of my_details_json is <class 'str'>
```

