In [ ]:

```
#Csv File Handling
Csv --> csv stand for comma seperated file.
If we want to store datain form seprated with the  commas
121,name,100,ash
for using csv file you need to import one module that is named as csv
```

In [2]:

```
#writer method --> it ensures in which file you need to write the data
#write row method --> it is used to write the data in the first row that will used as a coloumn for
#rest of the data. In this function you need to pass argument as a list elemnt.
#Writing Data in CSV File
import csv
with open("student.csv","w",newline="") as f:
    w=csv.writer(f)
    w.writerow(["StudentRollno","StudentName","StudentMarks","StudentAddress"])
    n=int(input("Enter number of student")) #5
    for i in range(n):
        StudentRollno=int(input("Enter student roll no"))
        StudentName=input("Enter student name")
        StudentMarks=int(input("Enter student marks"))
        StudentAddress=input("Enter address")
        w.writerow([StudentRollno,StudentName,StudentMarks,StudentAddress])
print("Total Student data is stored")
```

```
Enter number of student4
Enter student roll no1
Enter student namePavan
Enter student marks99
Enter addressMumbai
Enter student roll no2
Enter student nameBharat
Enter student marks99
Enter addressKolkata
Enter student roll no3
Enter student nameShubham
Enter student marks96
Enter addressDelhi
Enter student roll no5
Enter student nameTwarita
Enter student marks94
Enter addressPune
Total Student data is stored
```

In [10]:

```
#Reading the data
#Reader method is used to define which file is to be read
import csv
f=open("student.csv","r")
r=csv.reader(f)  #return the reader object
data=list(r)
for line in data:
    for word in line:
        print(word,"\t\t",end=" ")
    print()
```

```
StudentRollno            StudentName             StudentMarks            StudentAddress
1                Pavan           99              Mumbai
2                Bharat                  99              Kolkata
3                Shubham                 96              Delhi
5                Twarita                 94              Pune
```

In [12]:

```
import pandas as pd
df=pd.read_csv("student.csv")
print(df.head(5))
```

```
   StudentRollno StudentName  StudentMarks StudentAddress
0              1       Pavan            99         Mumbai
1              2      Bharat            99        Kolkata
2              3     Shubham            96          Delhi
3              5     Twarita            94           Pune
```

In [ ]:

```
Object Serialization: The process of converting an object from python to any other supported file
    over the network supported from is known as Object serialization.
Object Deserialization: The process of converting an object of any supported file
    to any python objectover the network supported from is known as Object desserialization
    2.By using Json
    3.By using YAML
```

In [ ]:

```
#Json: Javascript object notation
Any programming language can understand json . hence json is the most commonly
ised message format for applications irrespective of programming languages and platform.It is
very important to provide interportability between the application.
Json is also very useful to store the data
```

In [ ]:

```
What is Json
Python                                       JavaScript
int                                           Number
float                                         Number
list                                          arrays
True                                          true
False                                         false
str                                           string
None                                          null
Dictionary                                    object (JSON)
#time complexity of dictionary is o(1) constant time.
```

In [ ]:

```
Why json is more trending?
1.Light weighted
2.Human Readable
```

In [ ]:

```
In python if you want to use json then you need to use one module that is json
```

In [ ]:

```
For serialization :
    dumps()--> it serilizes the python dictionary object to json string
    dump()--> it serilies the python dictionary object to json file.
```

In [36]:

```
#Example:
#Using dumps() function
import json
employee={"name":"Pratyush","age":21,"address":"Delhi","Qualification":"B.Tech","None":None,"True":True}
print(type(employee))
json_string=json.dumps(employee,indent=4)
print(json_string)
print(type(json_string))
#Generally we are using dump function for storing json object
```

```
<class 'dict'>
{
    "name": "Pratyush",
    "age": 21,
    "address": "Delhi",
    "Qualification": "B.Tech",
    "None": null,
    "True": true
}
<class 'str'>
```

In [20]:

```
#using dump function
with open("emp.json","w") as f:
    json.dump(employee,f,indent=4)
    print("Json file generated")
```

```
Json file generated
```

In [ ]:

```
#Deserilization:
loads --> converting json object into python dictionary in form of string
load -->  reading json object from a file and converting it into python dictionary.
```

In [32]:

```
#Example:
#Using loads Function
import json
json_object = '''{"name": "Pratyush","age": 21,"address": "Delhi","Qualification": "B.Tech","None": null}'''
json_string=json.loads(json_object)
print(json_string)
for k,v in json_string.items():
    print(k,v)
```

```
{'name': 'Pratyush', 'age': 21, 'address': 'Delhi', 'Qualification': 'B.Tech', 'None': None}
name Pratyush
age 21
address Delhi
Qualification B.Tech
None None
```

In [35]:

```
#Using load Function
import json
with open("emp.json","r")as f:
    x = json.load(f)
    print("File readed")
    print(x)
for k,v in x.items():
    print(k,v)
```

```
File readed
{'name': 'Pratyush', 'age': 21, 'address': 'Delhi', 'Qualification': 'B.Tech', 'None': None}
name Pratyush
age 21
address Delhi
Qualification B.Tech
None None
```

In [23]:

```
d={1:2,3:4,6:5,6:6}
x=str(d)
print(type(x))
```

```
<class 'str'>
```

In [ ]:

```
#XML--> will store the data in the form of tags. Full form of xml is Extensible markup language
For working with xml in python you need to use module which is xmltodict
```

In [44]:

```
#Python xml to dict
#Parse will convert the xml into mpython dict
import xmltodict
import json
import pprint

my_xml="""
        <audience>
        <id what="attribute">123</id>
        <name>Suraj</name>
        </audience>
    """
mydict =xmltodict.parse(my_xml)
print(mydict)
print(mydict['audience']["id"])
print(mydict['audience']["name"])
```

```
{'audience': {'id': {'@what': 'attribute', '#text': '123'}, 'name': 'Suraj'}}
{'@what': 'attribute', '#text': '123'}
Suraj
```

In [48]:

```
#Python xml to json
#PrettyPrinter will convert the xml into json
import xmltodict
import json
import pprint

my_xml="""
        <audience>
        <id what="attribute">123</id>
        <name>Suraj</name>
        </audience>
    """
mydict =pprint.PrettyPrinter(indent=4)
print(type(mydict))
mydict.pprint(json.dumps(xmltodict.parse(my_xml)))
```

```
<class 'pprint.PrettyPrinter'>
'{"audience": {"id": {"@what": "attribute", "#text": "123"}, "name": "Suraj"}}'
```

In [57]:

```
import requests
response=requests.get("https://api.coindesk.com/v1/bpi/currentprice.json")
info=response.json()  #provides python dict object
print(type(info))
#print(info)
print("Bitcoin price as on ",info['time']['updated'])
print("1 Bitcoin is $",info['bpi']['USD']['rate'])
```

```
<class 'dict'>
Bitcoin price as on  Sep 13, 2022 16:12:00 UTC
1 Bitcoin is $ 20,829.6909
```

In [ ]: