About Serialization and Deserialization

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
For Seralization and Deserialization we can use:
    1.Pickle(Machine learning apps)  --> .pkl
    2.JSON   --> dictionary
    3.YAML
```

About JSON

In [ ]:
```
Json --> Javascript object notation
     --> Any programming language can understand json . hence json is the most commonly Used 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
```

WHAT IS JSON?

In [ ]:
```
WHAT IS JSON?

PYTHON                                                    JAVASCRIPT

INT                                                        NUMBER
FLOAT                                                      NUMBER
LIST                                                       ARRAY
TRUE                                                       true
False                                                      false
str                                                        string
None                                                       null
Dictionary                                                 object(JSON)

#time complexity of dictionary is o(1) constant time.
```

Why Json?

In [ ]:
```
Why json is more trending:
    1.Light weighted
    2.Human Readable
```

How to Work on Json Using Python

In [ ]:
```
--> In python if you want to use json then we need to import one module that is json.
```

For Serlization

In [ ]:
```
for serilization we are having two Functions:
    dumps() --> it serilizes the python dictionary object to json string
    dump()  --> it serilizes the python dictionary object to json file.
```

dumps() Function

In [ ]:
```
dumps()--> it serilizes the python dictionary object to json string
```

Example

In [9]:
```
#Using Dumps Function:
import json
employee = {"name":"Pratyush Srivastava","age":21,"address":"New Delhi","Qualification":"B.Tech","None":None,"True":True}
print(type(employee))
json_string = json.dumps(employee)
print(json_string)
print(type(json_string))
```

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

dump() Function

In [ ]:
```
dump()--> it serilizes the python dictionary object to json file.
```

Example

In [5]:
```
#with dump function
import json
employee = {"name":"Pratyush Srivastava","age":21,"address":"New Delhi","Qualification":"B.Tech","None":None,"True":True}
with open("json_employee.json","w") as f:
    json.dump(employee,f)
    print("Json Completed")
```

```
Json Completed
```

For Deserlization

In [ ]:
```
for Deserlization we are having Two Functions:
    1.loads --> converting the json object into python dict objec in form of string
    2.load --> converting the json object from a file into dict object
```

loads() Function

In [ ]:
```
loads() --> converting the json object into python dict objec in form of string
```

Example

In [7]:
```
#using loads function
import json
json_object = """{"name": "Pratyush Srivastava", "age": 21, "address": "New Delhi", "Qualification": "B.Tech", "None": null, "True": true}"""
json_string=json.loads(json_object)
print(json_string)
for k,v in json_string.items():
    print(k,v)
```

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

load() Function

In [ ]:
```
load() --> converting the json object from a file into dict object
```

Example

In [8]:
```
#load function
import json
with open("json_employee.json","r") as f:
    x=json.load(f)
    print("file loaded")
    print(x)
```

```
file loaded
{'name': 'Pratyush Srivastava', 'age': 21, 'address': 'New Delhi', 'Qualification': 'B.Tech', 'None': None, 'True': True}
```

XML File Handling

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
```

Example(XML to dict)

In [1]:
```
#python xml to dict
import xmltodict

my_xml = """
             <audience>
             <name>Pratyush</name>
             <Section>2</Section>
             </audience>
             """
mydict = xmltodict.parse(my_xml)
print(type(mydict))
print(mydict)
```

```
<class 'dict'>
{'audience': {'name': 'Pratyush', 'Section': '2'}}
```

Example(XML to dict)

In [23]:
```
#xml to json
import xmltodict
import json
my_xml = """
             <audience>
             <name>Pratyush</name>
             <Section>2</Section>
             <True>True</True>
             </audience>
             """
mydict = xmltodict.parse(my_xml)
x = json.dumps(mydict)
print(x)
```

```
{"audience": {"name": "Pratyush", "Section": "2", "True": "True"}}
```