JSON (Java Script Object Notation)

- **Need for JSON**: The data exchanging between a browser and a server, the data can only be text format. Need a conversion method to convert JSON to java script object and vice versa.
- **Purpose of JSON**:
  a. JSON is a format for storing and transporting data.
  b. JSON is often used when data is sent from a server to a web page.
  c. Define key value pairs (every key must have a value)
- **Characteristics of JSON**:
  1. JSON is a lightweight data interchange format (text-based interchange format)
  2. JSON is language independent
  3. JSON is "self-describing" and easy to understand
  4. No complicated parsing and translations in JSON conversions
  5. It is used to understand
  6. It allows to store any kind of data (audio, video and image)
  7. It is faster than other – structured data
- **Format:** Text (*.json)
- ***JSON Syntax rules:***
  1. Data is in name/value pairs
  2. Data is separated by commas
  3. Curly braces hold objects
  4. Square brackets hold arrays
  5. No NaN or Infinity is used
- **Example**: s = {name : “CSE”, College: “GCE”};
- **Different data types supported by JSON:**
  1. Number – double precision floating point
     (Octal and hexadecimal format is not used)
  2. String
  3. Array – It defines the order collection of key value pairs.
     - Defined within []
     - The values are separated by ,
     - Array index starts with an index from 0
  4. Object – unordered collection of key / value pairs
     - Objects are enclosed between {}
     - Each name is followed by :

- The pairs are separated by comma
- It is preferred to use when the key vales are strings

5. Whitespace – Defined between any pair of token
6. Null (empty)

❖ **Method used to convert JSON into javascript object**:

```
var d = JSON.parse(JSON_File_name);
```

❖ **Method used to convert javascript object into JSON:**

```
var   x = JSON.stringify(java_object_name);
```

❖ **Code for Sending JSON Data**

```
<html>
    <head>
        <script>
            function d()
            {
                var y = '{ "ss" : [ {"a": "karthik"}, {"a":"Arun"}]}';
                const obj = JSON.parse(y);
                alert(obj.ss.length);
            }
        </script>
    </head>
    <body onload="d()">
            <p id="ss"></p>
    </body>
</html>
```

❖ **Receiving Data:**

```
function s()
{
var myObj = {name: "John", age: 31, city: "New York"};
var myJSON = JSON.stringify(myObj);
document.write(myJSON);
var d=JSON.parse(myJSON); // convert JSON into Java script Object
document.write(d.name);
}
```

- **Functions for storing and accessing JSON into the memory**
  - *Code for storing JSON object in memory*: localStorage.setItem("testJSON", myJSON);
  - *Code for retrieving from memory*: text = localStorage.getItem("testJSON");
- **Sample program to access the JSON file from the secondary memory:**

Content of JSON File: abc.json

```
y = '{ "ss" : [ {"a": "karthik"}, {"a":"Arun"}]}';
```

HTML program:

```
<html>
  <head>
    <script type="text/javascript" src="abc.json"></script>
      <script type="text/javascript" >
        function load() {
          var mydata = JSON.parse(y);
            alert(mydata.ss.length);
        }
    </script>
  </head>
  <body onload="load()">
  <div id= "data">
  </div>
  </body>
</html>
```

- **Disadvantages of JSON:**
  1. No error handling mechanism in JSON (slight mistake in the JSON script → unable to get the structured data)
  2. JSON has limited supported tools