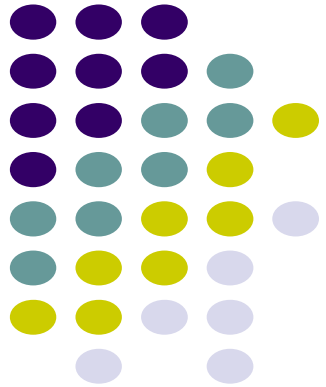
JSON

Javascript Object Notation

Dr. Arul Xavier V M



Javascript Object

- Real Life Objects, Properties, and Method

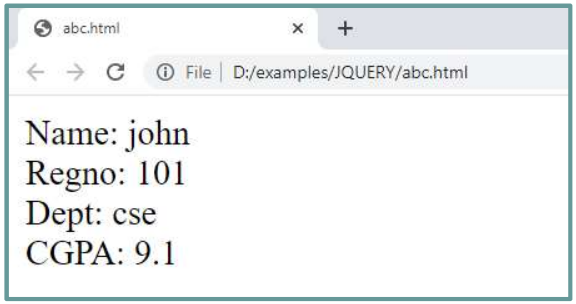
Object	Properties	Methods
	car.name = Fiat	car.start()
	car.model = 500	car.drive()
	car.weight = 850kg	car.brake()
	car.color = white	car.stop()

Creating Javascript Object

- Javascript object used to describe the properties and methods of real world objects.
- It is created using { } with name : value pairs
- For example: You want to store student details

```
<script>
      //example for javascript object (Student details)
      const student = {name:'john',regno:101,dept:'cse',cgpa:9.1};

      //accessing data from student object using dot operator
      document.write('Name: ' + student.name + '<br>');
      document.write('Regno: ' + student.regno + '<br>');
      document.write('Dept: ' + student.dept + '<br>');
      document.write('CGPA: ' + student.cgpa + '<br>');
 </script>
```



JSON - Introduction

- JSON stands for JavaScript Object Notation
- JSON is a text format for storing and transporting data
- JSON is "self-describing" and easy to understand
- JSON is a lightweight Data-Interchange Format.
- JSON Is Programming Language Independent

Why JSON?

- The JSON format is syntactically similar to the code for creating JavaScript objects.
- Because of this, a JavaScript program can easily convert JSON data into JavaScript objects.

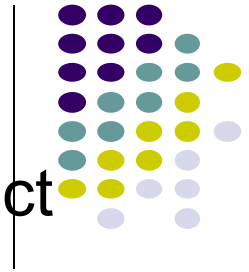
JavaScript has a built in function for converting JSON strings into JavaScript objects:

JSON.parse()

JavaScript also has a built in function for converting a JSON object into a JSON string:

JSON.stringify()

JSON Object Literal



- JSON object literal syntax is derived from JavaScript object notation syntax:
  - Data is in name/value pairs
  - Data is separated by commas
  - Curly braces hold objects
  - Square brackets hold arrays

JSON names/keys require double quotes.

```
<script>
    var student = {"name":"John","regno":101, "cgpa":9.1}
</script>
```

The name in the double quotes should be any valid string.
The value can be any valid data types

JSON Object Literal is it Object?

- It is a common mistake to call a JSON object literal "a JSON object".
- JSON cannot be an object.
- JSON is a string format.
- The data is only JSON when it is in a string format. When it is converted to a JavaScript variable, it becomes a JavaScript object

JSON String

- When we create JSON object literal inside single quotes it is called as JSON String

```
<script>
        let jsonstring = '{"name":"Rajiv","age":20,"salary":1000}';
</script>
```

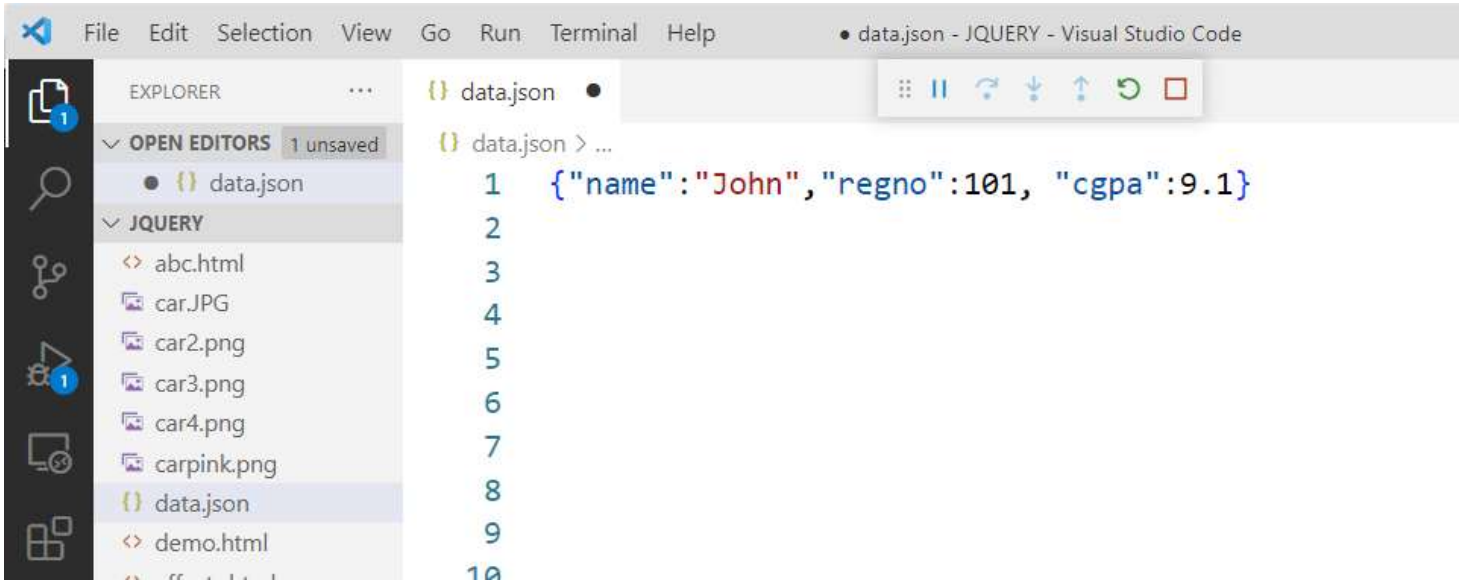
- It can be converted as JSON object literal or JavaScript object using JSON.parse() method

JSON Validation Tools

- https://jsoneditoronline.org/
- https://jsonlint.com/

JSON as document

- JSON data can be created in an external file with an extension as .json
- The advantage here is, the JSON data file can be exchanged between web application components such as Javascript, Jquery, Angular Js, React JS, Vu JS etc…



JSON Data Types

- The JSON name or key must be a string type.
- The JSON values can be any of the following
  - Number
    - Integer, Floating point number

```
{"regno":101, "cgpa":9.1}
```

  - String
    - Text with double quotes

```
{"name":"John"}
```

  - Boolean
    - true or false

```
{"sale":true}
```

  - Array
    - [ ]

```
{ "employees":["John", "Anna", "Peter"] }
```

  - Object
    - { }

```
{ "employee":{"name":"John", "age":30} }
```

  - Null
    - null

```
{"middlename":null}
```

Number, String, Boolean Values(Literals)

- Example: Representing shoe data

```
{ "brand": "Crocs", "color": "pink", "size": 9, "hasLaces": false }
```

- It can be written vertically

```
{
    "brand": "Crocs",
    "color": "pink",
    "size": 9,
    "hasLaces": false
}
```

JSON Array Values(Literals)

- Arrays in JSON are almost the same as arrays in JavaScript.

Example: Representing set of fruits

```
{
    "fruits":["apple","orange","mango","pineapple","grapes"]
}
```

Example: Representing set of scores

```
{
    "scores":[98,23,78,89,97]
}
```

JSON Object Literals

- The JSON value can be object data created using { }
- It means, the JSON value is another JSON String

```
{
    "book1":{"title" :"Life", "author":"Luther", "stock": 18, "price":2000}
}
```

It can be written vertically

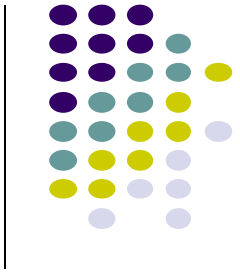
```
{
    "book1":{
        "title":"Life",
        "author":"Luther",
        "stock": 18,
        "price":2000
      }
}
```

Nested JSON Objects

College information

```
{
    "students": {
        "cse":{
            "URK20CS1001" : {"name":"John","cgpa":8.7},
            "URK20CS1002" : {"name":"Grace","cgpa":7.7},
            "URK20CS1003" : {"name":"Lincy","cgpa":9.7}
        }
    },
    "faculty":{
        "cse":{
            "678":{"name":"Ashok", "gender":"male","place":"Chennai"},
            "857":{"name":"Arun","gender":"male","place":"Coimbatore"}
        }
    }
}
```

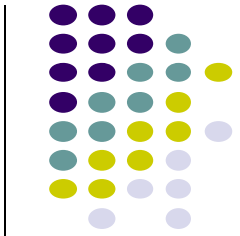
Null data value



- Null data value can be specified as null

```
{
    "person1":{
        "firstname":"John",
        "middlename":null,
        "lastname": "william"
    },
    "person2":{
        "firstname":"David",
        "middlename":"Wilson",
        "lastname": null
    }
}
```

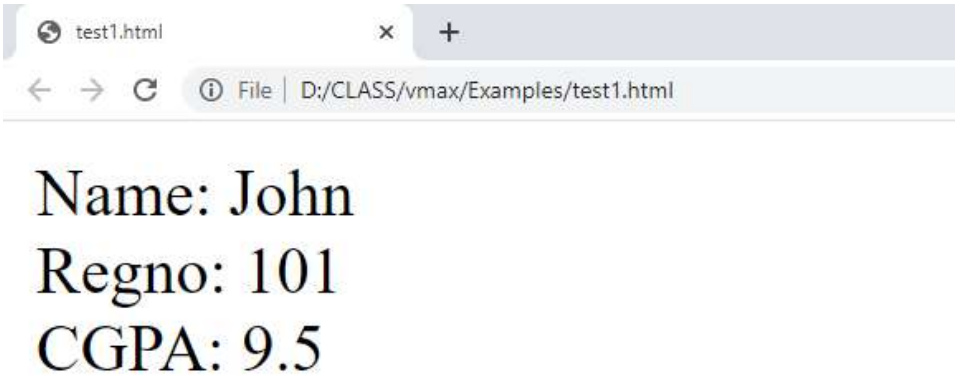
All Data Types



```
<script>
  var json= {
              "array": [1,2,3],
              "boolean": true,
              "color": "gold",
              "status": null,
              "number": 123,
              "object": {"a": "b", "c": "d"},
              "string": "Hello World"
        }
  </script>
```

Display JSON Data

```
<script>
      var jsondata = {"name":"John","regno":101,"cgpa":9.5};
      document.write("Name: " + jsondata.name + "<br>");
      document.write("Regno: " + jsondata.regno + "<br>");
      document.write("CGPA: " + jsondata.cgpa + "<br>");
</script>
```



Creating an Array of JSON data

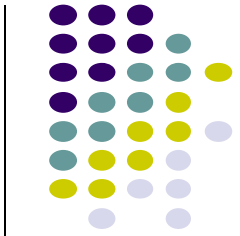
Consider the Table Data

Name	Regno	CGPA
John	101	7.4
Vmax	102	6.4
Vijay	103	9.4
Grace	104	8.4
Lincy	105	7.4

JSON.stringify()
- Used to convert the data as JSON string

```
var data =[
          { "name": "John", "regno": 101, "cgpa": 7.4 },
          { "name": "Vmax", "regno": 102, "cgpa": 6.4 },
          { "name": "Vijay", "regno": 103, "cgpa": 9.4 },
          { "name": "Grace", "regno": 104, "cgpa": 8.4 },
          { "name": "Lincy", "regno": 105, "cgpa": 7.4 }
    ];
var jsonstring = JSON.stringify(data);    //JSON String
```

Convert JSON String to JavaScript Object



- JSON.parse()
  - Converts the JSON string data to JavaScript object.

```
var data =[
            { "name": "John", "regno": 101, "cgpa": 7.4 },
            { "name": "Vmax", "regno": 102, "cgpa": 6.4 },
            { "name": "Vijay", "regno": 103, "cgpa": 9.4 },
            { "name": "Grace", "regno": 104, "cgpa": 8.4 },
            { "name": "Lincy", "regno": 105, "cgpa": 7.4 }
      ];

var jsonstring = JSON.stringify(data);

var javascriptobject = JSON.parse(jsonstring);
```

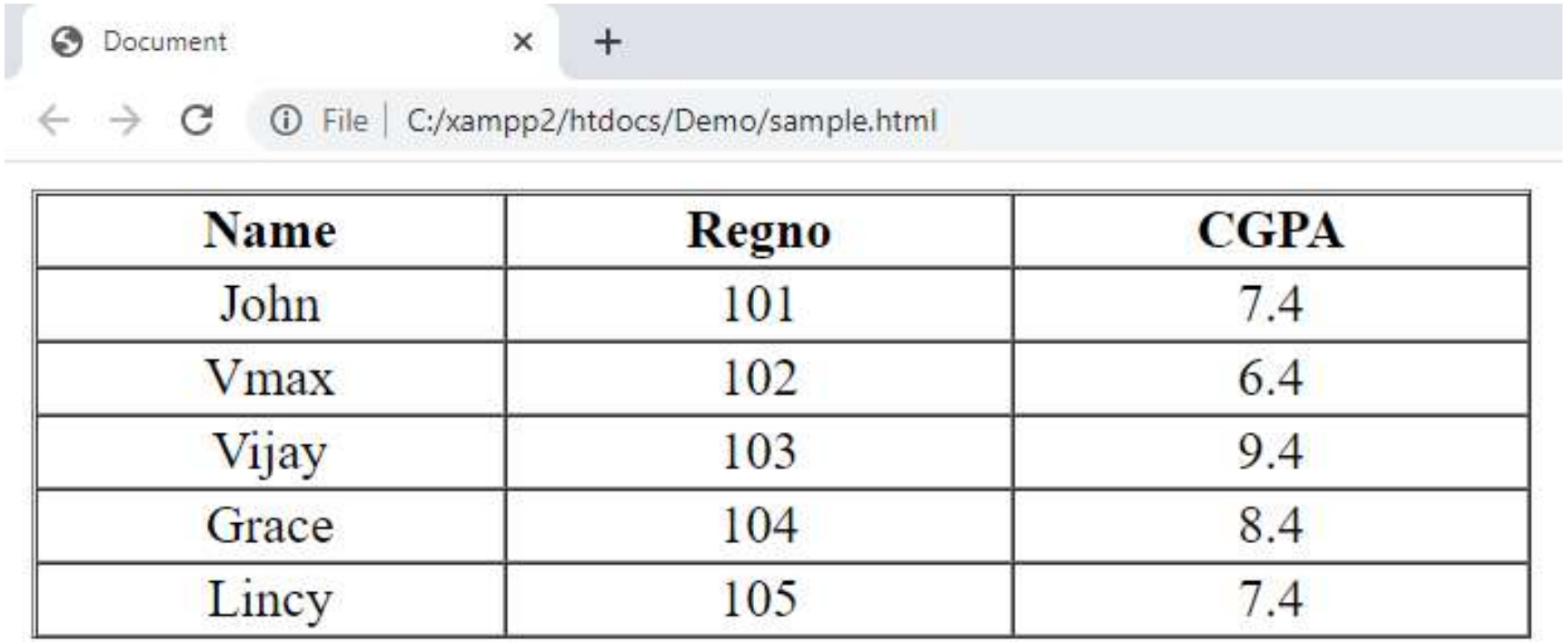
```
<script>
 var data =[
             {"name": "John", "regno": 101, "cgpa": 7.4 },
             {"name": "Vmax", "regno": 102, "cgpa": 6.4 },
             {"name": "Vijay", "regno": 103, "cgpa": 9.4 },
             {"name": "Grace", "regno": 104, "cgpa": 8.4 },
             {"name": "Lincy", "regno": 105, "cgpa": 7.4 }
         ];
  var json = JSON.stringify(data);
  var jsobj = JSON.parse(json);

  //display
  function display(){
    var element = document.getElementById('view')
    var output = "<table border=1 cellspacing=0 width=40%>";
    output += "<tr><th>Name</th><th>Regno</th><th>CGPA</th></tr>";
    jsobj.forEach((item)=>{
       output += "<tr><td>"+item.name+"</td><td>"+item.regno+"</td><td>"+item.cgpa+"</td></tr>";
      })
    output += "</table>"
    element.innerHTML = output;
  }
 </script>
```

```
<body onload="display()">
    <div id="view"></div>
</body>
```

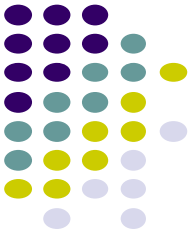
Display as Table Format

Output



Name	Regno	CGPA
John	101	7.4
Vmax	102	6.4
Vijay	103	9.4
Grace	104	8.4
Lincy	105	7.4

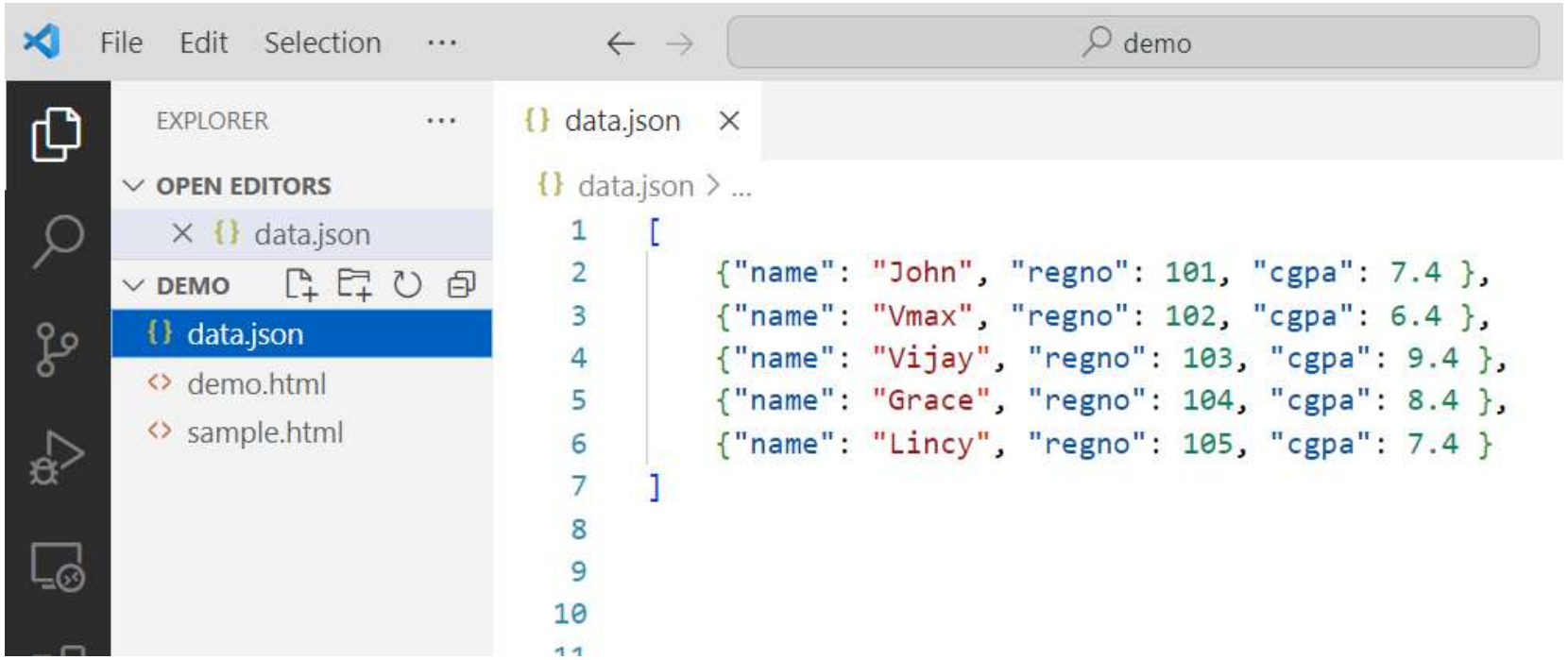
Filter the student details based on CGPA

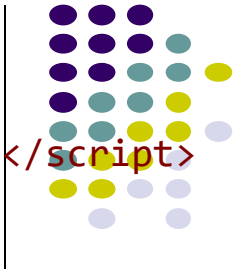


```
//display
function display(){
 var element = document.getElementById('view')
 var output = "<table border=1 cellspacing=0 width=40%>";
 output += "<tr><th>Name</th><th>Regno</th><th>CGPA</th></tr>";
jsobj.forEach((item)=>{
  if(item.cgpa > 7.5){
   output+="<tr><td>"+item.name+"</td><td>"+item.regno+"</td><td>"+item.cgpa+"</td></tr>";
     }
 })
 output += "</table>"
 element.innerHTML = output;
 }
```

JSON and JQuery

- JQuery also can be used to access JSON data.
- $.getJSON() method used to load the JSON data file





```
<!DOCTYPE html>
<html lang="en">
<head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
   <script>
       $(document).ready(function(){
           $.getJSON('data.json',function(data){
               data.forEach(item => {

 var temp = "<tr><td>"+item.name+"</td><td>"+item.regno+"</td><td>"+item.cgpa+"</td></tr>"
              $('#mytable').append(temp)

               });
           })
       })
   </script>
</head>
<body>
    <div>
        <table border="1" cellspacing="0" width="40%" id="mytable">
            <tr><th>Name</th><th>Regno</th><th>CGPA</th></tr>
        </table>
    </div>
</body>
</html>
```