This document is for private circulation only, For Third Year BTech. Computer 2020-21 CCOEW students.

- **Subject: Database Management Systems (DMS)**
- **Faculty: Prof. Jyoti Bangare**

ACKNOWLEDGEMENT

- This presentation contains pictures, contents taken from multiple sources, authors and sites.
- We would like to appreciate and thank the authors, artists and do not claim any of following work to be ours, but it has been compiled here to use for academic purpose only.

JSON

JSON

- JSON stands for JavaScript Object Notation.
- JSON objects are used for transferring data between server and client, XML serves the same purpose.
- However JSON objects have several advantages over XML and we are going to discuss them in this tutorial along with JSON concepts and its usages

JSON

- Let’s have a look at the piece of a JSON data: It basically has key-value pairs.

```
var chaitanya = {
        "firstName" : "Chaitanya",
        "lastName" : "Singh",
        "age" :  "28"
};
```

Features of JSON:

- It is light-weight
- It is language independent
- Easy to read and write
- Text based, human readable data exchange format

Why use JSON?

- **Standard Structure**: As we have seen so far that JSON objects are having a standard structure that makes developers job easy to read and write code, because they know what to expect from JSON.
- **Light weight**: When working with AJAX, it is important to load the data quickly and asynchronously without requesting the page re-load. Since JSON is light weighted, it becomes easier to get and load the requested data quickly.
- **Scalable**: JSON is language independent, which means it can work well with most of the modern programming language. Let's say if we need to change the server side language, in that case it would be easier for us to go ahead with that change as JSON structure is same for all the languages.

Comparison of JSON and XML

- Similarities:
  - Both are human readable
  - Both have very simple syntax
  - Both are hierarchical
  - Both are language independent
  - Both can be used by Ajax
  - Both supported in APIs of many programming languages
- Differences:
  - Syntax is different
  - JSON is less verbose
  - JSON can be parsed by JavaScript's eval method
  - JSON includes arrays
  - Names in JSON must not be JavaScript reserved words
  - XML can be validated

JSON vs. XML

- XML style:

```
<students>
 <student>
<name>John</name> <age>23</age> <city>Agra</city>
</student>
 <student>
<name>Steve</name> <age>28</age> <city>Delhi</city>
</student>
 <student>
 <name>Peter</name> <age>32</age> <city>Chennai</city>
 </student>
<student>
<name>Chaitanya</name> <age>28</age> <city>Bangalore</city>
 </student>
</students>
```

JSON style:

- JSON style:

{"students":[
 {"name":"John", "age":"23", "city":"Agra"},
  {"name":"Steve", "age":"28", "city":"Delhi"},
  {"name":"Peter", "age":"32", "city":"Chennai"},
   {"name":"Chaitanya", "age":"28", "city":"Bangalore"}
]}

As you can clearly see JSON is much more light-weight compared to XML. Also, in JSON we take advantage of arrays that is not available in XML.

JSON data structure types and how to read them:

1) JSON objects

2) JSON objects in array

3) Nesting of JSON objects

1) JSON objects:

```
var chaitanya = {
        "name" : "Chaitanya Singh",
        "age" : "28",
        "website" : "beginnersbook"
       };
```

The above text creates an object that we can access using the variable chaitanya. Inside an object we can have any number of key-value pairs like we have above. We can access the information out of a JSON object like this:

```
document.writeln("The name is:  " +chaitanya.name);
document.writeln("his age is: " + chaitanya.age);
document.writeln("his website is: "+ chaitanya.website);
```

2) JSON objects in array

In the above example we have stored the information of one person in a JSON object suppose we want to store the information of more than one person; in that case we can have an array of objects.

```
var students = [{
   "name" : "Steve",
  "age" :  "29",
   "gender" : "male"
},
{
  "name" : "Peter",
   "age" : "32",
   "gender" : "male"
}];
```

To access the information out of this array, we do write the code like this:

```
document.writeln(students[0].age); //output would be: 29
document.writeln(students[1].name); //output: Peter
```

3) Nesting of JSON objects:

```
var students = {
  "steve" : {
 "name" : "Steve",
  "age" :  "29",
  "gender" : "male" },
"pete" : {
"name" : "Peter",
 "age" : "32",
  "gender" : "male"},
"sop" : {
  "name" : "Sophie",
  "age" : "27",
  "gender" : "female"}}
document.writln(students.steve.age); //output: 29
document.writeln(students.sop.gender); //output: female
```

JSON & JavaScript:

- JSON is considered as a subset of JavaScript but that does not mean that JSON cannot be used with other languages. In fact it works well with PHP, Perl, Python, Ruby, Java, Ajax and many more.
- Just to demonstrate how JSON can be used along with JavaScript, here is an example:
  If you have gone though the above tutorial, you are familiar with the JSON structures. A JSON file type is .json
- **How to read data from json file and convert it into a JavaScript object?**
  We have two ways to do this.
  1) Using eval function, but this is not suggested due to security reasons (malicious data can be sent from the server to the client and then eval in the client script with harmful effects).
  2) Using JSON parser: No security issues plus it is faster than eval. Here is how to use it:

JSON & JavaScript:

```
var chaitanya = {
"name" : "Chaitanya Singh",
"age" : "28",
"website" : "beginnersbook"
};
```

We are converting the above JSON object to javascript object using JSON parser:

```
var myJSObject = JSON.parse(chaitanya);
```

How to convert JavaScript object to JSON text?

By using method stringify

```
var jsonText= JSON.stringify(myJSObject);
```

JSON Example

```
<html>
<head>
<title>Creating Object JSON with JavaScript</title>
<script language="javascript" >
var JSONObj = { "name" : "tutorialspoint.com", "year": 2005 };
document.write("<h1>JSON with JavaScript example</h1>");
document.write("<br>");
document.write("<h3>Website Name="+JSONObj.name+"</h3>");
document.write("<h3>Year="+JSONObj.year+"</h3>");
</script>
</head>
<body>
</body>
</html>
```

json_object.htm