UNIT 3 : JSON (JAVASCRIPT OBJECT NOTATION)

3.1 Concept and Features of JSON

3.2 Similarities and difference among JSON and XML

3.3 JSON objects(with string and Numbers))

3.4 JSON Arrays and their examples :

3.4.1 Array of string, Array of Numbers, Array of Booleans

3.4.2 Array of objects, Multi-Dimensional Arrays

3.4.3 JSON comments

Introduction

- JSON stands for JavaScript Object Notation
- The format was specified by Douglas Crockford in April 2001.
- JSON is a text format for storing and transporting data
- JSON is a lightweight data-interchange format
- JSON is plain text written in JavaScript object notation
- JSON is used to send data between computers
- JSON is language independent
- JSON is "self-describing" and easy to understand

- It was designed for human-readable data interchange.
- It has been extended from the JavaScript scripting language.
- The filename extension is **.json**.
- JSON Internet Media type is **application/json**.
- The Uniform Type Identifier is public.json.
- XML is an alternative to JSON. However, JSON objects have several advantages instead of XML.
- JSON has extended from JavaScript scripting language.

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.
- Since the format is text only, JSON data can easily be sent between computers, and used by any programming language.

- It is used while writing JavaScript based applications that includes browser extensions and websites.
- JSON format is used for serializing and transmitting structured data over network connection.
- It is primarily used to transmit data between a server and web applications.
- Web services and APIs use JSON format to provide public data.
- It can be used with modern programming languages.

Advantages/Characteristics of JSON

1. It is a lightweight text-based interchange and portable program.
2. JSON is easy to read and write.
3. JSON is language independent.
4. It is available in 55 different language platforms like javascript, PHP, Ruby, and Python.
5. It can be used by direct invocation in the HTML pages.

What you can do with JSON?

- JSON is mainly used to transmit data between servers and web applications.
- Currently, the NoSQL database is also using JSON for storing information.

JSON code looks like

```
{"employees":[
  { "firstName":“Virat", "lastName":“Kohli" },
  { "firstName":“RohIT", "lastName":"Sharma" },
  { "firstName":“Ravindra", "lastName":“Jadeja" }
]}
```

XML code looks like

```
<employees>
  <employee>
    <firstName>Virat</firstName> <lastName>Kohli</lastName>
  </employee>
  <employee>
    <firstName>RohIT</firstName> <lastName>Sharma</lastName>
  </employee>
  <employee>
    <firstName>Ravindra</firstName> <lastName>Jadeja</lastName>
  </employee>
</employees>
```

3.1 Concept and Features of JSON

- JSON is Scalable. Because of language-independent, it works with most of the modern programming language.
- JSON is lightweight.
- JSON is easy to read and write.
- JSON is a text-based, human-readable data exchange format.

3.2 Similarities and difference among JSON and XML

Similarities

- Both JSON and XML are "self describing" (human readable)
- Both JSON and XML are hierarchical (values within values)
- Both JSON and XML can be parsed and used by lots of programming languages.
- Both JSON and XML can be fetched with an XMLHttpRequest.

Difference

- JSON doesn't use end tag
- JSON is shorter
- JSON is quicker to read and write
- JSON can use arrays

Why JSON is Better Than XML?

- XML is much more difficult to parse than JSON.
- JSON is parsed into a ready-to-use JavaScript object.

Syntax

- JSON 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

- Because JSON syntax is derived from JavaScript object notation, very little extra software is needed to work with JSON within JavaScript.
- Example:
  - person = {name:“Virat", age:32, city:“Delhi"};

3.3 JSON objects(with string and Numbers)

- JSON object literals are surrounded by curly braces {}.
- JSON object literals contains key/value pairs.
- Keys and values are separated by a colon.
- Keys must be strings, and values must be a valid JSON data type:
  - string
  - number
  - object
  - array
  - boolean
  - null
- Each key/value pair is separated by a comma.

JSON Objects with String

JSON.parse()

- A common use of JSON is to exchange data to/from a web server.
- When receiving data from a web server, the data is always a string.
- Parse the data with JSON.parse(), and the data becomes a JavaScript object.

3.4 JSON Arrays and their examples:

3.4.1 Array of string, Array of Numbers, Array of Booleans

3.4.2 Array of objects, Multi-Dimensional Arrays

3.4.3 JSON comments

JSON Arrays

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

- In JSON, array values must be of type string, number, object, array, boolean or null.

- In JavaScript, array values can be all of the above, plus any other valid JavaScript expression, including functions, dates, and undefined.

- You can create a JavaScript array by parsing a JSON string:
- player='["Virat","Rohit","Bumrah"]';

3.4.1 Array of string, Array of Numbers, Array of Booleans

```
player=
'{
        "name":"Virat",
        "age":32,
        "type":["Batsman","Bowler"]
}';
```

Arrays of String

player=
‘

        ["Virat","Rohit","Bumrah"]

';

Arrays of Number

player=

‘[54,48,50]’

;

Arrays of Object

```
player=
'{
        "name":"Virat",
        "age":32,
        "type":["Batsman","Bowler"]
}';
```

3.4.2 Array of objects, Multi-Dimensional Arrays

3.4.3 JSON comments

- JSON is a data-only format. Comments in the form //, #, or /* */, which are used
- in popular programming languages, are not allowed in JSON.
- You can add comments to JSON as custom JSON elements that will hold your comments, but these elements will still be data. To do this, you need to add an element to your JSON file, such as "_comment,"which will contain your comment as a value of it. The JSON API endpoint must ignore this particular JSON comment element.

```
{
"Id": 1007,
"Customer": "Thomas",
"Quantity": 5,
"Price": 100.00,
"Date":"12-11-21",
"//first_comment": "Customer Bill.",
"//second_comment": "generated with 5
main attributes."
}
```

- Douglas Crockford, who popularized the JSON data format, deliberately removed comments from JSON to prevent misuse of the JSON format and keep it as a data-only format. He describes the reason he removed the comments from the JSON as follows

- **”I removed comments from JSON because I saw people using them to store parsing directives, which would break compatibility.”**