Unit 2: HTML5, JQuery and Ajax

XML Vs JSON

## XML Vs JSON

JSON, XML, are Text-file formats that can be used to store structured data that can be handy for embedded and Web applications.

**XML** (Extensible Markup Language) has been around for more than 3 decades now and it is an integral part of every web application. Be it a configuration file, mapping document or a schema definition, XML made life easier for data interchange by giving a clear structure to data and helping in dynamic configuration and loading of variables!

**JSON** stores all of its data in a map format (key/value pairs) that was neat and easier to comprehend. JSON is said to be slowly replacing XML because of several benefits like ease of data modeling or mapping directly to domain objects, more predictability and easy to understand the structure. JSON is just a data format whereas XML is a markup language.

Structure of XML vs JSON :



# Difference between XML and JSON:

|  |  |
| --- | --- |
| **XML**  **(Extensible Markup Language)** | **JSON**  **(JavaScript Object Notation)** |
| XML is a markup language, not a programming language, that has tags to define elements. | JSON is just a format written in JavaScript. |
| XML data is stored as a tree structure. Example –  2001  Varsha  2002  Akash | Data is stored like a map with key value pairs. Example –  **{“employees”: [**  {"id":"2001", “name":"Varsha"},  {"id":"2002", "name":"Akash"}  ]} |
| Can perform processing and formatting documents and objects. | It does not do any processing or computation |
| Bulky and slow in parsing, leading to slower data transmission | Very fast as the size of file is considerably small, faster parsing by the JavaScript engine and hence faster transfer of data |
| Supports namespaces, comments and metadata | There is no provision for namespace, adding comments or writing metadata |
| Document size is bulky and with big files, the tag structure makes it huge and complex to read. | Compact and easy to read, no redundant or empty tags or data, making the file look simple. |
| Doesn’t support array directly. To be able to use array, one has to add tags for each item.  science  maths  computers | Supports array which can be accessed as –  x = student.subjects[i];  **where** “subjects” **is** an array **as** –  “subjects”: [“science”, “math”, “computers”] |
| Supports many complex data types including charts, images and other non-primitive data types. | JSON supports only strings, numbers, arrays Boolean and object. Even object can only contain primitive types. |
| XML supports UTF-8 and UTF-16 encodings. | JSON supports UTF as well as ASCII encodings. |
| XML structures are prone to some attacks as external entity expansion and DTD validation are enabled by default. When these are disabled, XML parsers are safer. | JSON parsing is safe almost all the time except if JSONP is used, which can lead to Cross-Site Request Forgery (CSRF) attack. |
| Though the X is AJAX stands for XML, because of the tags in XML, a lot of bandwidth is unnecessarily consumed, making AJAX requests slow. | As data is serially processed in JSON, using it with AJAX ensures faster processing and hence preferable. Data can be easily manipulated using eval() method. |