

JSON vs XML

JSON vs XML

A list of differences between JSON and XML are given below.

No.	JSON	XML
1)	JSON stands for JavaScript Object Notation.	XML stands for eXtensible Markup Language.
2)	JSON is simple to read and write.	XML is less simple than JSON.
3)	JSON is easy to learn.	XML is less easy than JSON.
4)	JSON is data-oriented.	XML is document-oriented.
5)	JSON doesn't provide display capabilities.	XML provides the capability to display data because it is a markup language.
6)	JSON supports array.	XML doesn't support array.
7)	JSON is less secured than XML.	XML is more secured.
8)	JSON files are more human readable than XML.	XML files are less human readable.
9)	JSON supports only text and number data type.	XML support many data types such as text, number, images, charts, graphs etc. Moreover, XML offers options for transferring the format or structure of the data with actual data.

JSON Example

```
1. {"employees":[
2.   {"name":"A", "email":"a@a.com"},
3.   {"name":"B", "email":"b@b.com"},
4.   {"name":"C", "email":"c@c.com"}
5. ]}
```

XML Example

```
1. <employees>
2.   <employee>
3.     <name>A</name>
4.     <email>a@a.com</email>
5.   </employee>
6.   <employee>
7.     <name>B</name>
8.     <email>b@b.com</email>
9.   </employee>
10.  <employee>
11.    <name>C</name>
12.    <email>c@c.com</email>
13.  </employee>
14. </employees>
```

Similarities between JSON and XML

- Both are simple and open.
- Both supports unicode. So internationalization is supported by JSON and XML both.
- Both represents self describing data.
- Both are interoperable or language-independent.

JSON Example

JSON example can be created by object and array. Each object can have different data such as text, number, boolean etc. Let's see different JSON examples using object and array.

JSON Object Example

A JSON object contains data in the form of key/value pair. The keys are strings and the values are the JSON types. Keys and values are separated by colon. Each entry (key/value pair) is separated by comma.

The { (curly brace) represents the JSON object.

```
1. {  
2.     "employee": {  
3.         "name":      "A",  
4.         "salary":    56000,  
5.         "married":   true  
6.     }  
7. }
```

JSON Array example

The [(square bracket) represents the JSON array. A JSON array can have values and objects.

Let's see the example of JSON array having values.

```
1. ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",  
   "Friday", "Saturday"]
```

Let's see the example of JSON array having objects.

```
1. [  
   {  
     "name": "John",  
     "age": 30,  
     "married": true  
   },  
   {  
     "name": "Jane",  
     "age": 25,  
     "married": false  
   }  
]
```

```
2.     {"name": "A", "email": "a@a.com"},
3.     {"name": "B", "email": "b@b.com"}
4. ]
```

JSON Example 1

```
1. {"employees": [
2.     {"name": "A", "email": "a@a.com"},
3.     {"name": "B", "email": "b@b.com"},
4.     {"name": "C", "email": "c@c.com"}
5. ]}
```

The XML representation of above JSON example is given below.

```
1. <employees>
2.   <employee>
3.     <name>A</name>
4.     <email>a@a.com</email>
5.   </employee>
6.   <employee>
7.     <name>B</name>
8.     <email>b@b.com</email>
9.   </employee>
10.  <employee>
11.    <name>C</name>
12.    <email>c@c.com</email>
13.  </employee>
14. </employees>
```

JSON Example 2

```
1. {"menu": {
2.   "id": "file",
3.   "value": "File",
```

```
4.  "popup": {
5.    "menuitem": [
6.      {"value": "New", "onclick": "CreateDoc()"},
7.      {"value": "Open", "onclick": "OpenDoc()"},
8.      {"value": "Save", "onclick": "SaveDoc()"}
9.    ]
10.  }
11. }
```

The XML representation of above JSON example is given below.

```
1. <menu id="file" value="File">
2.   <popup>
3.     <menuitem value="New" onclick="CreateDoc()" />
4.     <menuitem value="Open" onclick="OpenDoc()" />
5.     <menuitem value="Save" onclick="SaveDoc()" />
6.   </popup>
7. </menu>
```

JSON Object

JSON object holds key/value pair. Each key is represented as a string in JSON and value can be of any type. The keys and values are separated by colon. Each key/value pair is separated by comma.

The curly brace { represents JSON object.

Let's see an example of JSON object.

```
1. {
2.   "employee": {
3.     "name":      "A",
4.     "salary":    56000,
5.     "married":   true
6.   }
7. }
```

In the above example, employee is an object in which “name”, “salary” and “married” are the key. In this example, there are string, number and boolean value for the keys.

JSON Object with Strings

The string value must be enclosed within double quote.

```
1. {  
2.     "name":      "A",  
3.     "email":    "a@a.com"  
4. }
```

JSON Object with Numbers

JSON supports numbers in double precision floating-point format. The number can be digits (0-9), fractions (.33, .532 etc) and exponents (e, e+, e-, E, E+, E-).

```
1. {  
2.     "integer": 34,  
3.     "fraction": .2145,  
4.     "exponent": 6.61789e+0  
5. }
```

JSON Object with Booleans

JSON also supports boolean values *true* or *false*.

```
1. {  
2.     "first": true,  
3.     "second": false  
4. }
```

JSON Nested Object Example

A JSON object can have another object also. Let's see a simple example of JSON object having another object.

```
1. {  
2.     "firstName": "A",  
3.     "lastName": "A",  
4.     "age": 27,  
5.     "address" : {  
6.         "streetAddress": "Street",  
7.         "city": "City",  
8.         "state": "State",  
9.         "postalCode": "123456"  
10.    }  
11. }
```

JSON Array

JSON array represents ordered list of values. JSON array can store multiple values. It can store string, number, boolean or object in JSON array.

In JSON array, values must be separated by comma.

The [(square bracket) represents JSON array.

JSON Array of Strings

Let's see an example of JSON arrays storing string values.

```
1. ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",  
   "Friday", "Saturday"]
```

JSON Array of Numbers

Let's see an example of JSON arrays storing number values.

```
1. [12, 34, 56, 43, 95]
```

JSON Array of Booleans

Let's see an example of JSON arrays storing boolean values.

```
1. [true, true, false, false, true]
```

JSON Array of Objects

Let's see a simple JSON array example having 4 objects.

```
1. {"employees":[
2.   {"name":"A", "email":"a@a.com", "age":23},
3.   {"name":"B", "email":"b@b.com", "age":24},
4.   {"name":"C", "email":"c@c.com", "age":25},
5.   {"name":"D", "email":"d@dl.com", "age":26}
6. ]}
```

JSON Multidimensional Array

We can store array inside JSON array, it is known as array of arrays or multidimensional array.

```
1. [
2.  [ "a", "b", "c" ],
3.  [ "m", "n", "o" ],
4.  [ "x", "y", "z" ]
5. ]
```


JSON Comments

JSON doesn't support comments. It is not a standard.

But you can do some tricks such as adding extra attribute for comment in JSON object, for example:

```
1. {  
2.   "employee": {  
3.     "name":      "Bob",  
4.     "salary":   56000,  
5.     "comments": "He is a nice man"  
6.   }  
7. }
```

Here, "comments" attribute can be treated as comment.