**What is Ajax?**  
Ajax is an acronym for Asynchronous Javascript and XML. It is used to communicate with the server without refreshing the web page and thus increasing the user experience and better performance.

**Prerequisites:**  
There are no such pre-requisites required to understand the latter portion of the article. Only the basic knowledge of HTML, CSS, and Javascript are good to go.

**How does it work?**  
First, let us understand what does asynchronous actually mean. There are two types of requests synchronous as well as asynchronous. Synchronous requests are the one which follows sequentially i.e if one process is going on and in the same time another process wants to be executed, it will not be allowed that means the only one process at a time will be executed. This is not good because in this type most of the time CPU remains idle such as during I/O operation in the process which are the order of magnitude slower than the CPU processing the instructions. Thus to make the full utilization of the CPU and other resources use asynchronous calls. For more information visit this [link](https://en.wikipedia.org/wiki/Asynchronous_I/O). Why the word javascript is present here. Actually, the requests are made through the use of javascript functions. Now the term XML which is used to create **XMLHttpRequest object**.  
Thus the summary of the above explanation is that Ajax allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. Now discuss the important part and its implementation. For implementing Ajax, only be aware of XMLHttpRequest object. Now, what actually it is. It is an object used to exchange data with the server behind the scenes. Try to remember the paradigm of OOP which says that object communicates through calling methods (or in general sense message passing). The same case applied here as well. Usually, create this object and use it to call the methods which result in effective communication. All modern browsers support the XMLHttpRequest object.

**Basic Syntax:** The syntax of creating the object is given below

|  |
| --- |
| req = new XMLHttpRequest(); |

There are two types of methods open() and send(). Uses of these methods explained below.

|  |
| --- |
| req.open("GET", "abc.php", true);  req.send(); |

The above two lines described the two methods. req stands for the request, it is basically a reference variable. The GET parameter is as usual one of two types of methods to send the request. Use POST as well depending upon whether send the data through POST or GET method. The second parameter being the name of the file which actually handles the requests and processes them. The third parameter is true, it tells that whether the requests are processed asynchronously or synchronously. It is by default true which means that requests are asynchronous. The open() method prepares the request before sending it to the server. The send method is used to send the request to the server.  
Sending the parameter through getting or POST request. The syntax is given below

|  |
| --- |
| req.open("GET", "abc.php?x=25", true);  req.send(); |

In the above lines of code, the specified query in the form of URL followed by ? which is further followed by the name of the variable then = and then the corresponding value. If sending two or more variables use ampersand(&) sign between the two variables. The above method as shown applies for GET request. Sending the data through the POST, then send it in the send method as shown below.

|  |
| --- |
| req.send("name=johndoe&marks=99"); |

.

Use of setRequestHeader() method as shown below.

|  |
| --- |
| req.setRequestHeader("Content-type", "application/x-www-form-urlencoded"); |

**Events and handling mechanism:**  
Any action performs on the clicking button, hovering over elements, page loading etc all are termed as events. Also aware of fact that javascript can detect events. So bind the code of specific event with its action which can be implemented by javascript. These are basically event handlers.  
Implementing event handlers which actually hold the events. Events handlers are basically functions written in javascript which act on or set into action when an event is fired by the user. When sending the request through send method usually get the response from the server later. But getting of response time is not known. So track it.  
Therefore to keep a track of the response onreadystatechange event which is binding with the event handler(function) which will get executed when a response comes.  
When a request to the server is sending perform actions based on the response. The onreadystatechange event is triggered every time the readyState changes. So what actually a ready state is and when will the onreadystate event actually occur and how many times it will occur between the request and response?  
The XMLHttpRequest object has a property called as readyState whose value changes in the complete request-response journey i.e when a request is prepared, sent, resolves, processed and when the response comes. That’s why it is called os onreadystatechange.  
The onreadystatechange stores a function (or the name of the function)to be called automatically each time the readyState property changes.  
The readyState holds different values ranging from 0 to 4.

1. request not initialized
2. server connection established
3. request received
4. processing request
5. request finished and response is ready

XMLHttpRequest also has a property named as status. The status has following values

* 200: “OK”
* 404: “Page not found”

Now remember it always that when readyState is 4 and status is 200, the response is ready.  
The whole thing described above is implemented in coding as given below

|  |
| --- |
| <p id = "dis">< /p>  req.onreadystatechange = function(){    if(req.readyState == 4 && req.status == 200){        document.getElementById("dis").innerHTML = req.responseText;    }  } |

(Note: This is only a section of code and moreover describing the communication between client and server so the code described above will not show any effect if run on IDE)  
In the above code if the condition is true (i.e the response is ready) then the result is displayed.  
**Advantages:**

1. Speed is enhanced as there is no need to reload the page again.
2. AJAX make asynchronous calls to a web server, this means client browsers avoid waiting for all the data to arrive before starting of rendering.
3. Form validation can be done successfully through it.
4. Bandwidth utilization – It saves memory when the data is fetched from the same page.
5. More interactive.

**Disadvantages:**

1. Ajax is dependent on Javascript. If there is some Javascript problem with the browser or in the OS, Ajax will not support.
2. Ajax can be problematic in Search engines as it uses Javascript for most of its parts.
3. Source code written in AJAX is easily human readable. There will be some security issues in Ajax.
4. Debugging is difficult.
5. Problem with browser back button when using AJAX enabled pages.

## What is JSON

* JSON stands for JavaScript Object Notation.
* JSON is an open standard data-interchange format.
* JSON is lightweight and self-describing.
* JSON originated from JavaScript.
* JSON is easy to read and write.
* JSON is language independent.
* JSON supports data structures such as arrays and objects.

## Features of JSON

* Simplicity
* Openness
* Self-Describing
* Internationalization
* Extensibility
* Interoperability

## Why do we use JSON?

Since JSON is an easy-to-use, lightweight language data interchange format in comparison to other available options, it can be used for API integration. Following are the advantages of JSON:

* **Less Verbose**: In contrast to XML, JSON follows a compact style to improve its users' readability. While working with a complex system, JSON tends to make substantial enhancements.
* **Faster**: The JSON parsing process is faster than that of the XML because the DOM manipulation library in XML requires extra memory for handling large XML files. However, JSON requires less data that ultimately results in reducing the cost and increasing the parsing speed.
* **Readable**: The JSON structure is easily readable and straightforward. Regardless of the programming language that you are using, you can easily map the domain objects.
* **Structured Data**: In JSON, a map data structure is used, whereas XML follows a tree structure. The key-value pairs limit the task but facilitate the predictive and easily understandable model.

## JSON Data Types

Following are the most commonly used JSON data types.

|  |  |  |
| --- | --- | --- |
| **Data Type** | **Description** | **Example** |
| String | A string is always written in double-quotes. It may consist of numbers, alphanumeric and special characters. | "student", "name", "1234", "Ver\_1" |
| Number | Number represents the numeric characters. | 121, 899 |
| Boolean | It can be either True or False. | true |
| Null | It is an empty value. |  |

## JSON Objects

In JSON, objects refer to dictionaries, which are enclosed in curly brackets, i.e., { }. These objects are written in key/value pairs, where the key has to be a string and values have to be a valid JSON data type such as string, number, object, Boolean or null. Here the key and values are separated by a colon, and a comma separates each key/value pair.

For example:

1. {"name" : "Jack", "employeeid" : 001, "present" : false}

## JSON Arrays

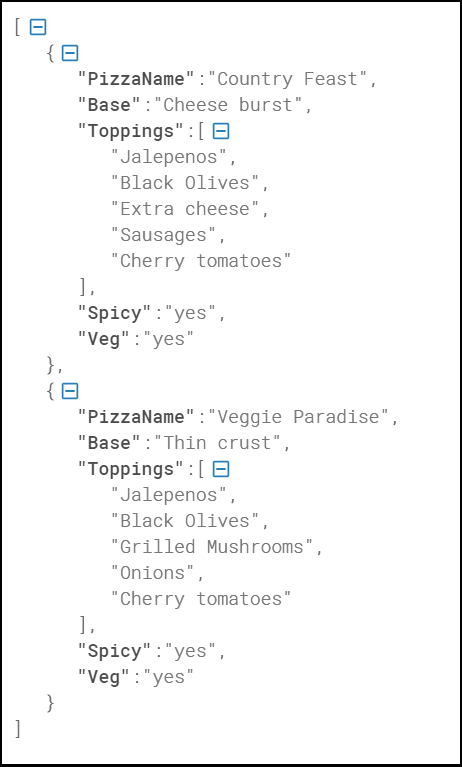
In JSON, arrays can be understood as a list of objects, which are mainly enclosed in square brackets [ ]. An array value can be a string, number, object, array, boolean or null.

For example:

1. [{
2. "PizzaName" : "Country Feast",
3. "Base" : "Cheese burst",
4. "Toppings" : ["Jalepenos", "Black Olives", "Extra cheese", "Sausages", "Cherry tomatoes"],
5. "Spicy" : "yes",
6. "Veg" : "yes"
7. },
9. {
10. "PizzaName" : "Veggie Paradise",
11. "Base" : "Thin crust",
12. "Toppings" : ["Jalepenos", "Black Olives", "Grilled Mushrooms", "Onions", "Cherry tomatoes"],
13. "Spicy" : "yes",
14. "Veg" : "yes"
15. }
16. ]

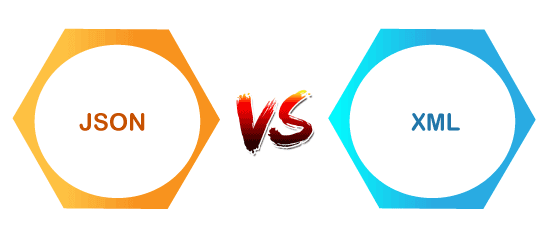
In the above example, the object "Pizza" is an array. It contains five objects, i.e., PizzaName, Base, Toppings, Spicy, and Veg.

Output:



Difference between JSON and XML

|  |  |
| --- | --- |
| JSON is easy to learn. | XML is quite more complex to learn than JSON. |
| It is simple to read and write. | It is more complex to read and write than JSON. |
| It is data-oriented. | It is document-oriented. |
| JSON is less secure in comparison to XML. | XML is highly secured. |
| It doesn't provide display capabilities. | It provides the display capability because it is a markup language. |
| It supports the array. | It doesn't support the array |
| Example :  [  {  "name" : "Peter",  "employed id" : "E231",  "present" : true,  "numberofdayspresent" : 29  },  {  "name" : "Jhon",  "employed id" : "E331",  "present" : true,  "numberofdayspresent" : 27  }  ] | Example :  <name>  <name>Peter</name>  </name> |



**The following are the differences between the json and xml:**

|  |  |
| --- | --- |
| **JSON** | **XML** |
| JSON stands for javascript object notation. | XML stands for an extensible markup language. |
| The extension of json file is .json. | The extension of xml file is .xml. |
| The internet media type is application/json. | The internet media type is application/xml or text/xml. |
| The type of format in JSON is data interchange. | The type of format in XML is a markup language. |
| It is extended from javascript. | It is extended from SGML. |
| It is open source means that we do not have to pay anything to use JSON. | It is also open source. |
| The object created in JSON has some type. | XML data does not have any type. |
| The data types supported by JSON are strings, numbers, Booleans, null, array. | XML data is in a string format. |
| It does not have any capacity to display the data. | XML is a markup language, so it has the capacity to display the content. |
| JSON has no tags. | XML data is represented in tags, i.e., start tag and end tag. |
|  | XML file is larger. If we want to represent the data in XML then it would create a larger file as compared to JSON. |
| JSON is quicker to read and write. | XML file takes time to read and write because the learning curve is higher. |
| JSON can use arrays to represent the data. | XML does not contain the concept of arrays. |
| It can be parsed by a standard javascript function. It has to be parsed before use. | XML data which is used to interchange the data, must be parsed with respective to their programming language to use that. |
| It can be easily parsed and little bit code is required to parse the data. | It is difficult to parse. |
| File size is smaller as compared to XML. | File size is larger. |
| JSON is data-oriented. | XML is document-oriented. |
| It is less secure than XML. | It is more secure than JSON. |

# What is xml

* **Xml** (eXtensible Markup Language) is a mark up language.
* XML is designed to store and transport data.
* Xml was released in late 90’s. it was created to provide an easy to use and store self describing data.
* XML became a W3C Recommendation on February 10, 1998.
* XML is not a replacement for HTML.
* XML is designed to be self-descriptive.
* XML is designed to carry data, not to display data.
* XML tags are not predefined. You must define your own tags.
* XML is platform independent and language independent.

|  |  |  |
| --- | --- | --- |
| **No.** | **HTML** | **XML** |
| 1) | HTML is used **to display data** and focuses on how data looks. | XML is a software and hardware independent tool used **to transport and store data**. It focuses on what data is. |
| 2) | HTML is a **markup language** itself. | XML provides a **framework to define markup languages**. |
| 3) | HTML is **not case sensitive**. | XML is **case sensitive**. |
| 4) | HTML is a presentation language. | XML is neither a presentation language nor a programming language. |
| 5) | HTML **has its own predefined tags**. | You **can define tags according to your need**. |
| 6) | In HTML, it is **not necessary to use a closing tag**. | XML **makes it mandatory to use a closing tag**. |
| 7) | HTML is **static** because it is used to display data. | XML is **dynamic** because it is used to transport data. |
| 8) | HTML **does not preserve whitespaces**. | XML **preserve whitespaces**. |

# XML Attributes

XML elements can have attributes. By the use of attributes we can add the information about the element.

XML attributes enhance the properties of the elements.

#### Note: XML attributes must always be quoted. We can use single or double quote.

Let us take an example of a book publisher. Here, book is the element and publisher is the attribute.

1. **<book** publisher="Tata McGraw Hill"**></book>**

## Why should we avoid XML attributes

* Attributes cannot contain multiple values but child elements can have multiple values.
* Attributes cannot contain tree structure but child element can.
* Attributes are not easily expandable. If you want to change in attribute's vales in future, it may be complicated.
* Attributes cannot describe structure but child elements can.
* Attributes are more difficult to be manipulated by program code.
* Attributes values are not easy to test against a DTD, which is used to define the legal elements of an XML document.

### ) What is the meaning of version in XML?

Version is a tag used to show which version of XML is used.

### 4) What are the benefits of XML?

These are the main benefits of using XML.

**Simplicity:** Very easy to read and understand the information coded in XML.

**Openness:** It is a W3C standard, endorsed by software industry market leaders.

**Extensibility:** It is extensible because it has no fixed set of tags. You can define them as you need.

**Self-descriptive:** XML documents do not need special schema set-up like traditional databases to store data. XML documents can be stored without such definitions, because they contain metadata in the form of tags and attributes.

**Scalable:** XML is not in binary format so you can create and edit files with anything and it is also easy to debug.

**Fast access:** XML documents are arranged in hierarchical form so it is comparatively faster.

### 5) What is XML DOM?

**DOM** stands for Document Object Model which is used to describe the logical structure of XML document. It is a hierarchical model that provides a way to access and manipulate an XML document.

DOM methods and objects can be used with any languages like C#, VB, JavaScript and VB Script.

### 6) What is SAX in XML?

**SAX** stands for Simple API for XML. It is a sequential access parser. It is a simple API for XML which provides a mechanism for reading data from an XML document. It is an alternative of DOM. DOM operates on the documents as whole, SAX parsers operate on each piece of the XML document sequentially.

SAX has no formal specification like DOM and consumes less memory. But it can be used to read the XML document only not write.

### 7) What is XSNL?

**XSNL** is an XML search neutral language. This language acts between the meta search interface and targeted system.

### 8) What is a well formed XML document?

A syntactically correct document is called well formed XML document. A well formed XML document must follow the XML?s basic rules of syntax:

* It must have a closing tag.
* The closing tag must exactly match the open tag: XML is case sensitive.
* All elements should be included within a single root tag.
* Child elements must be closed within parent tag.

### 9) What is a valid XML document?

A structurally correct element is called a valid XML document. It should follow some predefined rules of a specific type of document. These rules determine the type of data that each part of the document can contain. These rules can be written by the author of an XML document or someone other.

#### Note: A valid XML document may be well-formed but a well-formed XML document may not be valid.

### 10) What is DTD?

**DTD** stands for Document Type Definition. It defines a leading building block of an XML document. It defines:

* Names of elements
* How and where they can be used
* Element attributes
* Proper nesting

### 11) How can you apply a DTD to an XML document?

To apply a DTD to an XML document, you can:

* Use the DTD element definition within the XML document itself.
* Provide a DTD as a separate file and reference its name in XML document.

### 12) What are the basic rules to write XML document?

You should consider the following rules to write an XML document.

* It should have a root element.
* All tags must be closed.
* Spaces are not allowed in tag names.
* All tags must be nested properly.
* XML tags are case sensitive.
* Use the attribute values within quotes.
* Whitespace is preserved in XML.

### 13) What is XSL?

**XSL** stands for Extensible Stylesheet Language. It is a language for expressing stylesheets. These stylesheets are like CSS which describes how to display an XML document of a given type.

[more details...](https://www.javatpoint.com/xml-schema)

### 14) What is XPath in XML?

**XPath** is a technology used in XML. It is used to retrieve elements from XML documents. XPath expressions can be used to locate and retrieve elements, attributes and values from XML files because XML documents are structured. It is similar to SQL.

As SQL is used to retrieve data from database, XPath is used to retrieve data from XML.

### 15) What is XSLT?

**XSLT** is a popular XML technology which is used to transform one XML file to other format like HTML etc. XSLT is like a language which has its own syntax, functions and operator to transform XML documents. XSLT is also used to display data present in XML files as HTML pages.

### 16) What is the difference between XML DTD and XML schema or XSD?

XSL stands for Extensible Stylesheet Language. It is a language for expressing stylesheets. These stylesheets are like CSS which describes how to display an XML document of a given type.

* DTD stands for Document Type Definition whereas XSD stands for XML Schema Definition.
* DTD does not support namespaces. It has its own set of keywords to define a schema whereas XSD uses its own set of namespaces and elements to define the schema.
* DTD is the predecessor of XML schema whereas XML schema is a new technology, some XML parser do not support it yet.

[more details...](https://www.javatpoint.com/dtd-vs-xsd)

### 17) Is XML meant to be a replacement of HTML?

No, both languages have their own specification and used for different purposes. XML is used to describes data while HTML focus on how to display data.

### 18) Define element and attribute in an XML document.

From a start tag to end tag everything you see in a XML document, is XML element. It can contain other elements, attributes and texts.

An Attribute is a value that is used to define additional information about the element.

Let's take an example to understand it well:

1. **<librarymembers>**
2. **<student** id="001"**>**
3. **<name>**Vimal**</name>**
4. **<age>**25**</age>**
5. **</student>**
6. **</librarymembers>**

In the above example, <student> is an element and id is an attribute of the <student> element.

### 19) What is the difference between simple element and complex element?

In XML, simple elements are text-based elements. It contains less attributes, child elements, and cannot be left empty.

But, complex elements can hold multiple attributes and elements. It can contain additional sub elements and empty element also.

### 20) Define the concept of XPOINTER.

XPOINTER is used to point data within XML document. It is used to locate the particular part of the XML document. It is a W3C recommendation.

See this example:

1. address.xml#pointer(/ descendant ::streetnumber[@id =9])

In this example, XPOINTER points "streetnumber = 9" in "address.xml".

### 21) What is XML data binding? Why is it used?

XML data binding is the process of representing the information in an XML document as an object in computer memory.

XML data binding is used to short your development effort, simplify maintenance, increase reliability. It saves your development time and money. It makes working with XML data very intuitive.

### 22) What is XML encoding error?

There are two types of XML encoding errors:

1. An invalid character was found in text content.
2. Switching from current encoding to specified encoding not supported.

These errors occur because XML document can contain non ASCII characters like Norwegian and French. These errors can be avoided by specifying the XML encoding Unicode.

### 23) What are the different XML API's?

**Tree-based API:** It compiles an XML document in a tree like structure and loads it into the memory. You can traverse and change the tree structure. Tree based API's are useful for a wide range of applications. Example of tree-based API is DOM parser.

**Event-based API:** An event based API provides the reports to an application about the parsing event. It uses a set of built-in call back functions. Example of event-based API is SAX parser.

### 24) What is XmlReader class? Explain.

The XmlrReader class represents a reader that provides fast, noncached, forward-only access to XML data. You need to import the following namespaces to work with XmlReader class in .NET.

**In VB:**

Imports System.Xml

**In C#:**

using System.Xml;

### 25) What is the difference between CDATA and PCDATA?

CDATA means unparsed character data whereas PCDATA means parsed character data.

[more details...](https://www.javatpoint.com/cdata-pcdata)

### 26) What is XQuery?

XQuery is a query language that is used to retrieve data from XML document.

### 27) What is XML Namespace?

A namespace is a qualified name that is associated with the DTD/Schema location .

A document may have duplicate elements and attributes. So, the namespace defines a way to compare duplicate elements and attribute names.

### 28) What is SGML?

SGML is a vast and powerful generalized markup language that is used to define descriptions for the structure of several electronic documents.

### 29) Can we use graphics in XML?

Yes, graphics can be stored in XML file by using XLink and XPointer. It supports graphics like GIF, JPG, TIFF, PNG, CGM, EPS, SVG.

### 30) Is XML case sensitive?

Yes, XML is a case sensitive language.