

X

eXtensible

Extend our notion.

M

Markup

Write using Markups

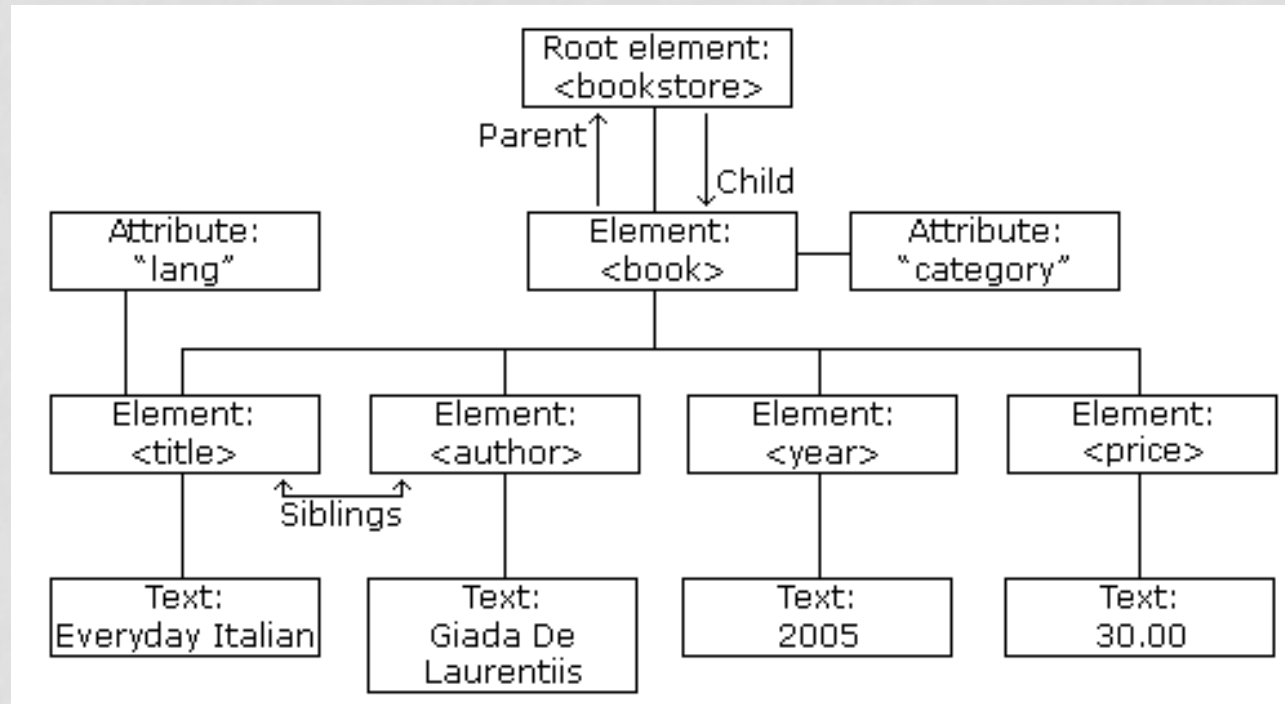
L

Language

Create your own document.

USING XML WITH JAVASCRIPT

XML IS HIERARCHICAL



THE DOCUMENT OBJECT MODEL

XML elements can be
accessed through the
DOM.

THE XML DOM

XML DOM is:

- A standard object model for XML
- A standard programming interface for XML
- Platform- and language-independent
- A W3C standard

THE XML DOCUMENT (THE ACTUAL DATA)

```
1  <bookstore>
2    <book category="cooking">
3      <title lang="en">Everyday Italian</title>
4      <author>Giada De Laurentiis</author>
5      <year>2005</year>
6      <price>30.00</price> </book>
7    <book category="children">
8      <title lang="en">Harry Potter</title>
9      <author>J K. Rowling</author>
10     <year>2005</year>
11     <price>29.99</price> </book>
12   <book category="web">
13     <title lang="en">XQuery Kick Start</title>
14     <author>James McGovern</author>
15     <author>Per Bothner</author>
16     <author>Kurt Cagle</author>
17     <author>James Linn</author>
18     <author>Vaidyanathan Nagarajan</author>
19     <year>2003</year>
20     <price>49.99</price> </book>
21   <book category="web" cover="paperback">
22     <title lang="en">Learning XML</title>
23     <author>Erik T. Ray</author>
24     <year>2003</year>
25     <price>39.95</price> </book>
26 </bookstore>
```

USING FETCH

Same folder/location
as the HTML



```
<script>
  fetch("books.xml")
    .then(result => {return result.text();})
    .then(xml => {
      const domParser = new DOMParser();
      const xmlDocument = domParser.parseFromString(xml, "text/xml");
      document.getElementById("demo").innerHTML = xmlDocument.getElementsByTagName("title")[0].childNodes[0].nodeValue;
    });
</script>
```

USING XHR

Object:
XMLDocument



```
<script>
const xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        const xmlDoc = xml.responseXML;
        const target = document.getElementById("demo");

        target.innerHTML = xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;
    }
};
xhttp.open("GET", "books.xml", true);
xhttp.send();
```

DOM PARSER

The DOMParser parses XML (or HTML) from a string into a DOM Document.

```
1 | var parser = new DOMParser();  
2 | var doc = parser.parseFromString(stringContainingXMLSource, "application/xml");
```


DOMPARSER() IS USEFUL.

```
1  var parser = new DOMParser();
2  var doc = parser.parseFromString(stringContainingXMLSource, "application/xml");
3  // returns a Document, but not a SVGDocument nor a HTMLDocument
4
5  parser = new DOMParser();
6  doc = parser.parseFromString(stringContainingSVGSource, "image/svg+xml");
7  // returns a SVGDocument, which also is a Document.
8
9  parser = new DOMParser();
10 doc = parser.parseFromString(stringContainingHTMLSource, "text/html");
11 // returns a HTMLDocument, which also is a Document.
```

LOADING XML - STRINGS

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <title>XML Parsing Example</title>
5   <meta charset="utf-8"/>
6   <meta name="viewport" content="width=device-width, initial-scale=1"/>
7   <link href="https://fonts.googleapis.com/css?family=Mali" rel="stylesheet">
8   <style>
9     * { font-family: 'Mali', cursive; }
10  </style>
11 </head>
12 <body>
13   <p id="demo"></p>
14   <script>
15     const xmlAsString = "<bookstore><book><title>Everyday Ethiopian</title><author>Giada De Laurentiis</author><year>2005</year></book></bookstore>";
16     const parser = new DOMParser();
17     const xmlDoc = parser.parseFromString(xmlAsString, "text/xml");
18
19     document.getElementById("demo").innerHTML = xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;
20   </script>
21 </body>
22 </html>
```

INTERNET EXPLORER

This approach is old,
but works in IE 5,6,7,8..

```
xmlDoc = new ActiveXObject("Microsoft.XMLDOM");  
xmlDoc.async = false;  
xmlDoc.loadXML(txt);
```

ACTIVEXObject IS DEAD

```
var parseXml;
if (typeof window.DOMParser != "undefined") {
    parseXml = function(xmlStr) {
        return ( new window.DOMParser() ).parseFromString(xmlStr, "text/xml");
    };
} else if (typeof window.ActiveXObject != "undefined" &&
    new window.ActiveXObject("Microsoft.XMLDOM")) {
    parseXml = function(xmlStr) {
        var xmlDoc = new window.ActiveXObject("Microsoft.XMLDOM");
        xmlDoc.async = "false";
        xmlDoc.loadXML(xmlStr);
        return xmlDoc;
    };
} else {
    throw new Error("No XML parser found");
}
```

```
var xml = parseXml("<foo>Stuff</foo>");
```

PARSING W/JQUERY

```
var xml = "<music><album>Beethoven</album></music>";  
var result = $(xml).find("album").text();
```

XML PARSING

-ANOTHER EXAMPLE-

```
<fruits>
  <fruit name="Apple" colour="Green" />
  <fruit name="Banana" colour="Yellow" />
</fruits>
```

```
function getFruits(xml) {
  var fruits = xml.getElementsByTagName("fruits")[0];
  if (fruits) {
    var fruitsNodes = fruits.childNodes;
    if (fruitsNodes) {
      for (var i = 0; i < fruitsNodes.length; i++) {
        var name = fruitsNodes[i].getAttribute("name");
        var colour = fruitsNodes[i].getAttribute("colour");
        alert("Fruit " + name + " is coloured " + colour);
      }
    }
  }
}
```


JQUERY

jQuery.parseXML()

```
1  <!doctype html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8">
5    <title>jQuery.parseXML demo</title>
6    <script src="https://code.jquery.com/jquery-1.10.2.js"></script>
7  </head>
8  <body>
9
10 <p id="someElement"></p>
11 <p id="anotherElement"></p>
12
13 <script>
14 var xml = "<rss version='2.0'><channel><title>RSS Title</title></channel></rss>",
15     xmlDoc = $.parseXML( xml ),
16     $xml = $( xmlDoc ),
17     $title = $xml.find( "title" );
18
19 // Append "RSS Title" to #someElement
20 $( "#someElement" ).append( $title.text() );
21
22 // Change the title to "XML Title"
23 $title.text( "XML Title" );
24
25 // Append "XML Title" to #anotherElement
26 $( "#anotherElement" ).append( $title.text() );
27 </script>
28
29 </body>
30 </html>
```

LOADING XML FROM A FILE

```
$.ajax({  
  url: 'xmlfile.xml',  
  dataType: 'xml',  
  success: function(data){  
    // Extract relevant data from XML  
    var xml_node = $('Pages',data);  
    console.log( xml_node.find('Page[Name="test"] > controls > test').text() );  
  },  
  error: function(data){  
    console.log('Error loading XML data');  
  }  
});
```



Same folder as
the html resides

XHR OBJECT

The XMLHttpRequest (XHR) object has a XML property

```
xmlDoc = xmlhttp.responseXML;
```

