**XML DOM Parser**

[« Previous](http://www.w3schools.com/dom/dom_nodetree.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_loadxmldoc.asp)

Most browsers have a built-in XML parser to read and manipulate XML.

The parser converts XML into a JavaScript accessible object (the XML DOM).

**XML Parser**

The XML DOM contains methods (functions) to traverse XML trees, access, insert, and delete nodes.

However, before an XML document can be accessed and manipulated, it must be loaded into an XML DOM object.

An XML parser reads XML, and converts it into an XML DOM object that can be accessed with JavaScript.

Most browsers have a built-in XML parser.

**Load an XML Document**

The following JavaScript fragment loads an XML document ("[books.xml](http://www.w3schools.com/dom/books.xml)"):

**Example**

if (window.XMLHttpRequest)  
  {  
  xhttp=new XMLHttpRequest();  
  }  
else // IE 5/6  
  {  
  xhttp=new ActiveXObject("Microsoft.XMLHTTP");  
  }  
xhttp.open("GET","books.xml",false);  
xhttp.send();  
xmlDoc=xhttp.responseXML;

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_loadxml)

**Code explained:**

* Create an XMLHTTP object
* Open the XMLHTTP object
* Send an XML HTTP request to the server
* Set the response as an XML DOM object

**Load an XML String**

The following code loads and parses an XML string:

**Example**

if (window.DOMParser)  
  {  
  parser=new DOMParser();  
  xmlDoc=parser.parseFromString(text,"text/xml");  
  }  
else // Internet Explorer  
  {  
  xmlDoc=new ActiveXObject("Microsoft.XMLDOM");  
  xmlDoc.async=false;  
  xmlDoc.loadXML(text);   
  }

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_loadxmltext)

**Note:** Internet Explorer uses the loadXML() method to parse an XML string, while other browsers use the DOMParser object.

**Access Across Domains**

For security reasons, modern browsers do not allow access across domains.

This means, that both the web page and the XML file it tries to load, must be located on the same server.

The examples on W3Schools all open XML files located on the W3Schools domain.

If you want to use the example above on one of your web pages, the XML files you load must be located on your own server.

[« Previous](http://www.w3schools.com/dom/dom_nodetree.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_loadxmldoc.asp)

**XML DOM Load Functions**

[« Previous](http://www.w3schools.com/dom/dom_parser.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_methods.asp)

The code for loading XML documents can be stored in a function.

**The loadXMLDoc() Function**

To make the code from the previous page simpler to maintain (and check for older browsers), it should be written as a function:

function loadXMLDoc(dname)  
{  
if (window.XMLHttpRequest)  
  {  
  xhttp=new XMLHttpRequest();  
  }  
else  
  {  
  xhttp=new ActiveXObject("Microsoft.XMLHTTP");  
  }  
xhttp.open("GET",dname,false);  
xhttp.send();  
return xhttp.responseXML;  
}

The function above can be stored in the <head> section of an HTML page, and called from a script in the page.

lamp**The function described above, is used in all XML document examples in this tutorial!**

**An External JavaScript for loadXMLDoc()**

To make the code above even easier to maintain, and to make sure the same code is used in all pages, we store the function in an external file.

The file is called "loadxmldoc.js", and will be loaded in the head section of an HTML page. Then, the loadXMLDoc() function can be called from a script in the page.

The following example uses the loadXMLDoc() function to load [books.xml](http://www.w3schools.com/dom/books.xml):

**Example**

<html>  
<head>  
<script src="loadxmldoc.js">  
</script>  
</head>  
<body>  
  
<script>  
xmlDoc=loadXMLDoc("books.xml");  
  
*code goes here.....*  
</script>  
  
</body>  
</html>

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_parsertest)

How to get the data from the XML file, will be explained in the next chapters.

**The loadXMLString() Function**

To make the code from the previous page simpler to maintain (and check for older browsers), it should be written as a function:

function loadXMLString(txt)   
{  
if (window.DOMParser)  
  {  
  parser=new DOMParser();  
  xmlDoc=parser.parseFromString(txt,"text/xml");  
  }  
else // Internet Explorer  
  {  
  xmlDoc=new ActiveXObject("Microsoft.XMLDOM");  
  xmlDoc.async=false;  
  xmlDoc.loadXML(txt);   
  }  
return xmlDoc;  
}

The function above can be stored in the <head> section of an HTML page, and called from a script in the page.

lamp**The function described above, is used in all XML string examples in this tutorial!**

**An External JavaScript for loadXMLString()**

We have stored the loadXMLString() function in a file called "loadxmlstring.js".

**Example**

<html>  
<head>  
<script src="loadxmlstring.js"></script>  
</head>  
<body>  
<script>  
text="<bookstore>"  
text=text+"<book>";  
text=text+"<title>Everyday Italian</title>";  
text=text+"<author>Giada De Laurentiis</author>";  
text=text+"<year>2005</year>";  
text=text+"</book>";  
text=text+"</bookstore>";  
  
xmlDoc=loadXMLString(text);  
  
*code goes here.....*</script>  
</body>  
</html>

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_parsertest2)

[« Previous](http://www.w3schools.com/dom/dom_parser.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_methods.asp)

**XML DOM - Properties and Methods**

[« Previous](http://www.w3schools.com/dom/dom_loadxmldoc.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_access.asp)

Properties and methods define the programming interface to the XML DOM.

**Programming Interface**

The DOM models XML as a set of node objects. The nodes can be accessed with JavaScript or other programming languages. In this tutorial we use JavaScript.

The programming interface to the DOM is defined by a set standard properties and methods.

**Properties** are often referred to as something that is (i.e. nodename is "book").

**Methods** are often referred to as something that is done (i.e. delete "book").

**XML DOM Properties**

These are some typical DOM properties:

* x.nodeName - the name of x
* x.nodeValue - the value of x
* x.parentNode - the parent node of x
* x.childNodes - the child nodes of x
* x.attributes - the attributes nodes of x

Note: In the list above, x is a node object.

**XML DOM Methods**

* x.getElementsByTagName(*name*) - get all elements with a specified tag name
* x.appendChild(*node*) - insert a child node to x
* x.removeChild(*node*) - remove a child node from x

Note: In the list above, x is a node object.

**Example**

The JavaScript code to get the text from the first <title> element in books.xml:

txt=xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue

After the execution of the statement, txt will hold the value "Everyday Italian"

Explained:

* **xmlDoc** - the XML DOM object created by the parser.
* **getElementsByTagName("title")[0]** - the first <title> element
* **childNodes[0]** - the first child of the <title> element (the text node)
* **nodeValue** - the value of the node (the text itself)

[« Previous](http://www.w3schools.com/dom/dom_loadxmldoc.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_access.asp)

**WEB HOSTING**

[Best Web Hosting](http://www.lunarpages.com/id/w3schools/goto/w3schools) [eUK Web Hosting](http://www.eukhost.com) [UK Reseller Hosting](http://www.heartinternet.co.uk) [Domain, Hosting & Email](http://www.webhosting.uk.com/cpanel-hosting.php) [Lowest Price Hosting](http://www.justhost.com/track/w3schools/BestPrice)

**WEB BUILDING**

[Download XML Editor](http://www.altova.com/ref/?s=w3s_text&q=xmlspy) [FREE Website BUILDER](http://www.wix.com/eteamhtml/400?utm_campaign=ma_w3schools.com&experiment_id=ma_w3schools.comlink1_400Html&embed_tags=camp&utm_camp=hOUdCO-AmAMQm__j8AM) [FREE Website Creator](http://www.wix.com/eteamhtml/400?utm_campaign=ma_w3schools.com&experiment_id=ma_w3schools.comlink2_400) [Best Website Templates](http://www.dreamtemplate.com/?ref=w3txtfree)

**STATISTICS**

[Browser Statistics](http://www.w3schools.com/browsers/browsers_stats.asp) [OS Statistics](http://www.w3schools.com/browsers/browsers_os.asp) [Display Statistics](http://www.w3schools.com/browsers/browsers_display.asp)

With the DOM, you can access every node in an XML document.

Examples

## Try it Yourself - Examples

The examples below use the XML file [books.xml](http://www.w3schools.com/dom/books.xml).  
A function, [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp), in an external JavaScript is used to load the XML file.

[Access a node using its index number in a node list](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_index)  
This example uses the getElementsByTagname() method to get the third <title> element in "books.xml"

[Loop through nodes using the length property](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_list_loop)  
This example uses the length property to loop through all <title> elements in "books.xml"

[See the node type of an element](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_root)  
This example uses the nodeType property to get node type of the root element in "books.xml".

[Loop through element nodes](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_loop)  
This example uses the nodeType property to only process element nodes in "books.xml".

[Loop through element nodes using node relationships](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_navigate)  
This example uses the nodeType property and the nextSibling property to process element nodes in "books.xml".

## Accessing Nodes

You can access a node in three ways:

1. By using the getElementsByTagName() method

2. By looping through (traversing) the nodes tree.

3. By navigating the node tree, using the node relationships.

## The getElementsByTagName() Method

getElementsByTagName() returns all elements with a specified tag name.

### Syntax

*node*.getElementsByTagName(*"tagname"*);

### Example

The following example returns all <title> elements under the x element:

x.getElementsByTagName("title");

Note that the example above only returns <title> elements under the x node. To return all <title> elements in the XML document use:

xmlDoc.getElementsByTagName("title");

where xmlDoc is the document itself (document node).

## DOM Node List

The getElementsByTagName() method returns a node list. A node list is an array of nodes.

The following code loads "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp) and stores a list of <title> nodes (a node list) in the variable x:

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("title");

The <title> elements in x can be accessed by index number. To access the third <title> you can write::

y=x[2];

**Note:** The index starts at 0.

You will learn more about node lists in a later chapter of this tutorial.

## DOM Node List Length

The length property defines the length of a node list (the number of nodes).

You can loop through a node list by using the length property:

## Example

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("title");  
  
for (i=0;i<x.length;i++)  
  {  
  document.write(x[i].childNodes[0].nodeValue);  
  document.write("  
");  
  }

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_list_loop)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Get all <title> element nodes
3. For each title element, output the value of its text node

## Node Types

The **documentElement** property of the XML document is the root node.

The **nodeName** property of a node is the name of the node.

The **nodeType** property of a node is the type of the node.

You will learn more about the node properties in the next chapter of this tutorial.

[Try it yourself](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_root)

## Traversing Nodes

The following code loops through the child nodes, that are also element nodes, of the root node:

## Example

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.documentElement.childNodes;  
  
for (i=0;i<x.length;i++)  
{  
  if (x[i].nodeType==1)  
  {//Process only element nodes (type 1)  
  document.write(x[i].nodeName);  
  document.write("  
");  
  }  
}

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_loop)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Get the child nodes of the root element
3. For each child node, check the node type of the node. If the node type is "1" it is an element node
4. Output the name of the node if it is an element node

## Navigating Node Relationships

The following code navigates the node tree using the node relationships:

## Example

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("book")[0].childNodes;  
y=xmlDoc.getElementsByTagName("book")[0].firstChild;  
  
for (i=0;i<x.length;i++)  
{  
if (y.nodeType==1)  
  {//Process only element nodes (type 1)  
  document.write(y.nodeName + "  
");  
  }  
y=y.nextSibling;  
}

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_navigate)

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Get the child nodes of the first book element
3. Set the "y" variable to be the first child node of the first book element
4. For each child node (starting with the first child node "y"):
5. Check the node type. If the node type is "1" it is an element node
6. Output the name of the node if it is an element node
7. Set the "y" variable to be the next sibling node, and run through the loop again

[« Previous](http://www.w3schools.com/dom/dom_methods.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_info.asp)

## WEB HOSTING

[Best Web Hosting](http://www.lunarpages.com/id/w3schools/goto/w3schools) [eUK Web Hosting](http://www.eukhost.com) [UK Reseller Hosting](http://www.heartinternet.co.uk) [Domain, Hosting & Email](http://www.webhosting.uk.com/cpanel-hosting.php) [Lowest Price Hosting](http://www.justhost.com/track/w3schools/BestPrice)

## WEB BUILDING

[Download XML Editor](http://www.altova.com/ref/?s=w3s_text&q=xmlspy) [FREE Website BUILDER](http://www.wix.com/eteamhtml/400?utm_campaign=ma_w3schools.com&experiment_id=ma_w3schools.comlink1_400Html&embed_tags=camp&utm_camp=hOUdCO-AmAMQm__j8AM) [FREE Website Creator](http://www.wix.com/eteamhtml/400?utm_campaign=ma_w3schools.com&experiment_id=ma_w3schools.comlink2_400) [Best Website Templates](http://www.dreamtemplate.com/?ref=w3txtfree)

## STATISTICS

[Browser Statistics](http://www.w3schools.com/browsers/browsers_stats.asp) [OS Statistics](http://www.w3schools.com/browsers/browsers_os.asp) [Display Statistics](http://www.w3schools.com/browsers/browsers_display.asp)

**XML DOM Node Information**

[« Previous](http://www.w3schools.com/dom/dom_nodes_access.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_nodelist.asp)

The nodeName, nodeValue, and nodeType properties contain information about nodes.

Examples

**Try it Yourself - Examples**

The examples below use the XML file [books.xml](http://www.w3schools.com/dom/books.xml).   
A function, [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp), in an external JavaScript is used to load the XML file.

[Get the node name of an element node](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nodename3)  
This example uses the nodeName property to get the node name of the root element in "books.xml".

[Get the text from a text node](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_getelement)  
This example uses the nodeValue property to get the text of the first <title> element in "books.xml".

[Change the text in a text node](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_changeelement)  
This example uses the nodeValue property to change the text of the first <title> element in "books.xml".

[Get the node name and type of an element node](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_root)  
This example uses the nodeName and nodeType property to get node name and type of the root element in "books.xml".

**Node Properties**

In the XML DOM, each node is an **object**.

Objects have methods and properties, that can be accessed and manipulated by JavaScript.

Three important node properties are:

* nodeName
* nodeValue
* nodeType

**The nodeName Property**

The nodeName property specifies the name of a node.

* nodeName is read-only
* nodeName of an element node is the same as the tag name
* nodeName of an attribute node is the attribute name
* nodeName of a text node is always #text
* nodeName of the document node is always #document

[Try it yourself.](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nodename3)

**The nodeValue Property**

The nodeValue property specifies the value of a node.

* nodeValue for element nodes is undefined
* nodeValue for text nodes is the text itself
* nodeValue for attribute nodes is the attribute value

**Get the Value of an Element**

The following code retrieves the text node value of the first <title> element:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];  
txt=x.nodeValue;

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_getelement)

Result:  txt = "Everyday Italian"

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Get text node of the first <title> element node
3. Set the txt variable to be the value of the text node

**Change the Value of an Element**

The following code changes the text node value of the first <title> element:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];  
x.nodeValue="Easy Cooking";

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_changeelement)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Get text node of the first <title> element node
3. Change the value of the text node to "Easy Cooking"

**The nodeType Property**

The nodeType property specifies the type of node.

nodeType is read only.

The most important node types are:

|  |  |
| --- | --- |
| **Node type** | **NodeType** |
| Element | 1 |
| Attribute | 2 |
| Text | 3 |
| Comment | 8 |
| Document | 9 |

[Try it yourself.](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_root)

[« Previous](http://www.w3schools.com/dom/dom_nodes_access.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_nodelist.asp)

**XML DOM Node List**

[« Previous](http://www.w3schools.com/dom/dom_nodes_info.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_traverse.asp)

A list of nodes is returned by the getElementsByTagName() method and the childNodes property.

Examples

**Try it Yourself - Examples**

The examples below use the XML file [books.xml](http://www.w3schools.com/dom/books.xml).   
A function, [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp), in an external JavaScript is used to load the XML file.

[Get the text from the first <title> element](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nodelist)  
This example uses the getElementsByTagName() method to get the text from the first <title> element in "books.xml".

[Loop through nodes using the length property](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nodelist_length)  
This example uses node list and the length property to loop through all <title> elements in "books.xml"

[Get the attribute of an element](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_attributes)  
This example uses a attribute list to get attribute from the first <book> element in "books.xml".

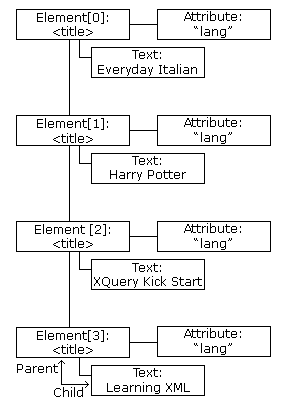
**DOM Node List**

When using properties or methods like childNodes or getElementsByTagName(), a node list object is returned.

A node list object represents a list of nodes, in the same order as in the XML.

Nodes in the node list are accessed with index numbers starting from 0.

The following image represents a node list of the <title> elements in "[books.xml](http://www.w3schools.com/dom/books.xml)":



The following code fragment loads "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp) and returns a node list of title elements in "books.xml":

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("title");

After the execution of the statement above, x is a node list object.

The following code fragment returns the text from the first <title> element in the node list (x):

**Example**

txt=x[0].childNodes[0].nodeValue;

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nodelist)

After the execution of the statement above, txt = "Everyday Italian".

**Node List Length**

A node list object keeps itself up-to-date. If an element is deleted or added, the list is automatically updated.

The length property of a node list is the number of nodes in the list.

The following code fragment loads "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp) and returns the number of <title> elements in "books.xml":

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName('title').length;

After the execution of the statement above, x = 4.

The length of the node list can be used to loop through all the elements in the list.

The following code fragment uses the length property to loop through the list of <title> elements:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
//the x variable will hold a node list  
x=xmlDoc.getElementsByTagName('title');  
  
for (i=0;i<x.length;i++)  
{  
document.write(x[i].childNodes[0].nodeValue);  
document.write("  
");  
}

Output:

Everyday Italian  
Harry Potter  
XQuery Kick Start  
Learning XML

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nodelist_length)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Set the x variable to hold a node list of all title elements
3. Output the value from the text node of all <title> elements

**DOM Attribute List (Named Node Map)**

The attributes property of an element node returns a list of attribute nodes.

This is called a named node map, and is similar to a node list, except for some differences in methods and properties.

A attribute list keeps itself up-to-date. If an attribute is deleted or added, the list is automatically updated.

The following code fragment loads "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp) and returns a list of attribute nodes from the first <book> element in "books.xml":

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName('book')[0].attributes;

After the execution of the code above, x.length = is the number of attributes and x.getNamedItem() can be used to return an attribute node.

The following code fragment displays the value of the "category" attribute, and the number of attributes, of a book:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("book")[0].attributes;  
  
document.write(x.getNamedItem("category").nodeValue);  
document.write("  
" + x.length);

Output:

cooking  
1

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_attributes)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Set the x variable to hold a list of all attributes of the first <book> element
3. Output the value from the "category" attribute
4. Output the length of the attribute list

[« Previous](http://www.w3schools.com/dom/dom_nodes_info.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_traverse.asp)

**WEB HOSTING**

[Best Web Hosting](http://www.lunarpages.com/id/w3schools/goto/w3schools) [eUK Web Hosting](http://www.eukhost.com) [UK Reseller Hosting](http://www.heartinternet.co.uk) [Domain, Hosting & Email](http://www.webhosting.uk.com/cpanel-hosting.php) [Lowest Price Hosting](http://www.justhost.com/track/w3schools/BestPrice)

**WEB BUILDING**

[Download XML Editor](http://www.altova.com/ref/?s=w3s_text&q=xmlspy) [FREE Website BUILDER](http://www.wix.com/eteamhtml/400?utm_campaign=ma_w3schools.com&experiment_id=ma_w3schools.comlink1_400Html&embed_tags=camp&utm_camp=hOUdCO-AmAMQm__j8AM) [FREE Website Creator](http://www.wix.com/eteamhtml/400?utm_campaign=ma_w3schools.com&experiment_id=ma_w3schools.comlink2_400) [Best Website Templates](http://www.dreamtemplate.com/?ref=w3txtfree)

**STATISTICS**

[Browser Statistics](http://www.w3schools.com/browsers/browsers_stats.asp) [OS Statistics](http://www.w3schools.com/browsers/browsers_os.asp) [Display Statistics](http://www.w3schools.com/browsers/browsers_display.asp)

**XML DOM Traverse Node Tree**

[« Previous](http://www.w3schools.com/dom/dom_nodes_nodelist.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_mozilla_vs_ie.asp)

Traversing means looping through or traveling across the node tree.

**Traversing the Node Tree**

Often you want to loop an XML document, for example: when you want to extract the value of each element.

This is called "Traversing the node tree"

The example below loops through all child nodes of <book>, and displays their names and values:

**Example**

<html>  
<head>  
<script src="loadxmlstring.js"></script>  
</head>  
<body>  
<script>  
text="<book>";  
text=text+"<title>Everyday Italian</title>";  
text=text+"<author>Giada De Laurentiis</author>";  
text=text+"<year>2005</year>";  
text=text+"</book>";  
  
xmlDoc=loadXMLString(text);  
  
// documentElement always represents the root node  
x=xmlDoc.documentElement.childNodes;  
for (i=0;i<x.length;i++)  
{  
document.write(x[i].nodeName);  
document.write(": ");  
document.write(x[i].childNodes[0].nodeValue);  
document.write("  
");  
}  
</script>  
</body>  
</html>

Output:

title: Everyday Italian  
author: Giada De Laurentiis  
year: 2005

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_parsertest3)

Example explained:

1. [loadXMLString()](http://www.w3schools.com/dom/dom_loadxmldoc.asp) loads the XML string into xmlDoc
2. Get the child nodes of the root element
3. For each child node, output the node name and the node value of the text node

[« Previous](http://www.w3schools.com/dom/dom_nodes_nodelist.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_mozilla_vs_ie.asp)

**XML DOM Browser Differences**

[« Previous](http://www.w3schools.com/dom/dom_nodes_traverse.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_navigate.asp)

**Browser Differences in DOM Parsing**

All modern browsers support the W3C DOM specification.

However, there are some differences between browsers. One important difference is:

* The way they handle white-spaces and new lines

**DOM - White Spaces and New Lines**

XML often contains new line, or white space characters, between nodes. This is often the case when the document is edited by a simple editor like Notepad.

The following example (edited by Notepad) contains CR/LF (new line) between each line and two spaces in front of each child node:

<book>  
  <title>Everyday Italian</title>  
  <author>Giada De Laurentiis</author>  
  <year>2005</year>  
  <price>30.00</price>  
</book>

Internet Explorer will NOT treat empty white-spaces, or new lines as text nodes, while other browsers will.

The following code fragment displays how many child nodes the root element (of books.xml) has:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.documentElement.childNodes;  
document.write("Number of child nodes: " + x.length);

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nodetype_ievsmozilla)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Get the child nodes of the root element
3. Output the number of child nodes
4. The result is different depending on which browser you use. IE9 and earlier will alert 4 child nodes, while IE10 and other browsers will alert 9 child nodes

[« Previous](http://www.w3schools.com/dom/dom_nodes_traverse.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_navigate.asp)

**XML DOM - Navigating Nodes**

[« Previous](http://www.w3schools.com/dom/dom_mozilla_vs_ie.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_get.asp)

Nodes can be navigated using node relationships.

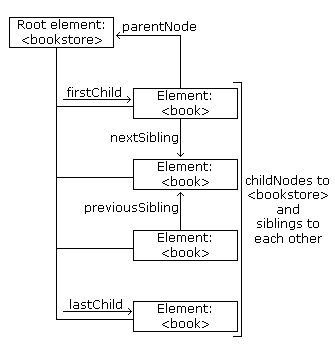
**Navigating DOM Nodes**

Accessing nodes in the node tree via the relationship between nodes, is often called "navigating nodes".

In the XML DOM, node relationships are defined as properties to the nodes:

* parentNode
* childNodes
* firstChild
* lastChild
* nextSibling
* previousSibling

The following image illustrates a part of the node tree and the relationship between nodes in [books.xml](http://www.w3schools.com/dom/books.xml):



**DOM - Parent Node**

All nodes has exactly one parent node. The following code navigates to the parent node of <book>:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("book")[0];  
document.write(x.parentNode.nodeName);

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nav_parentnode)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Get the first <book> element
3. Output the node name of the parent node of "x"

**Avoid Empty Text Nodes**

Firefox, and some other browsers, will treat empty white-spaces or new lines as text nodes, Internet Explorer will not.

This causes a problem when using the properties: firstChild, lastChild, nextSibling, previousSibling.

To avoid navigating to empty text nodes (spaces and new-line characters between element nodes), we use a function that checks the node type:

function get\_nextSibling(n)  
{  
y=n.nextSibling;  
while (y.nodeType!=1)  
  {  
  y=y.nextSibling;  
  }  
return y;  
}

The function above allows you to use get\_nextSibling(*node*) instead of the property *node*.nextSibling.

Code explained:

Element nodes are type 1. If the sibling node is not an element node, it moves to the next nodes until an element node is found. This way, the result will be the same in both Internet Explorer and Firefox.

**Get the First Child Element**

The following code displays the first element node of the first <book>:

**Example**

<html>  
<head>  
<script src="loadxmldoc.js">  
</script>  
<script>  
//check if the first node is an element node  
function get\_firstChild(n)  
{  
y=n.firstChild;  
while (y.nodeType!=1)  
  {  
  y=y.nextSibling;  
  }  
return y;  
}  
</script>  
</head>  
  
<body>  
<script>  
xmlDoc=loadXMLDoc("books.xml");  
  
x=get\_firstChild(xmlDoc.getElementsByTagName("book")[0]);  
document.write(x.nodeName);  
</script>  
</body>  
</html>

Output:

title

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nav_firstchild)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Use the get\_firstChild function on the first <book> element node to get the first child node that is an element node
3. Output the node name of first child node that is an element node

Examples

**More Examples**

[lastChild()](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nav_lastchild)  
This example uses the lastChild() method and a custom function to get the last child node of a node

[nextSibling()](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nav_nextsibling)  
This example uses the nextSibling() method and a custom function to get the next sibling node of a node

[previousSibling()](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_nav_prevsibling)  
This example uses the previousSibling() method and a custom function to get the previous sibling node of a node

[« Previous](http://www.w3schools.com/dom/dom_mozilla_vs_ie.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_get.asp)

**WEB HOSTING**

[Best Web Hosting](http://www.lunarpages.com/id/w3schools/goto/w3schools) [eUK Web Hosting](http://www.eukhost.com) [UK Reseller Hosting](http://www.heartinternet.co.uk) [Domain, Hosting & Email](http://www.webhosting.uk.com/cpanel-hosting.php) [Lowest Price Hosting](http://www.justhost.com/track/w3schools/BestPrice)

**WEB BUILDING**

[XML Editor - Free Trial!](http://www.altova.com/ref/?s=w3s_text2&amp;q=xmlspy) [FREE Website BUILDER](http://www.wix.com/eteamhtml/400?utm_campaign=ma_w3schools.com&experiment_id=ma_w3schools.comlink1_400Html&embed_tags=camp&utm_camp=hOUdCO-AmAMQm__j8AM) [FREE Website Creator](http://www.wix.com/eteamhtml/400?utm_campaign=ma_w3schools.com&experiment_id=ma_w3schools.comlink2_400) [Best Website Templates](http://www.dreamtemplate.com/?ref=w3txtfree)

**STATISTICS**

[Browser Statistics](http://www.w3schools.com/browsers/browsers_stats.asp) [OS Statistics](http://www.w3schools.com/browsers/browsers_os.asp) [Display Statistics](http://www.w3schools.com/browsers/browsers_display.asp)

**XML DOM Get Node Values**

[« Previous](http://www.w3schools.com/dom/dom_nodes_navigate.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_set.asp)

The nodeValue property is used to get the text value of a node.

The getAttribute() method returns the value of an attribute.

**Get the Value of an Element**

In the DOM, everything is a node. Element nodes do not have a text value.

The text of an element node is stored in a child node. This node is called a text node.

The way to get the text of an element is to get the value of the child node (text node).

**Get an Element Value**

The getElementsByTagName() method returns a node list containing all elements with the specified tag name in the same order as they appear in the source document.

The following code loads "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp) and retrieves the first <title> element:

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("title")[0];

The childNodes property returns a list of child nodes. The <title> element has only one child node. It is a text node.

The following code retrieves the text node of the <title> element:

x=xmlDoc.getElementsByTagName("title")[0];  
y=x.childNodes[0];

The nodeValue property returns the text value of the text node:

**Example**

x=xmlDoc.getElementsByTagName("title")[0];  
y=x.childNodes[0];  
txt=y.nodeValue;

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_getelementsbytagname)

Result:  txt = "Everyday Italian"

Loop through all <title> elements: [Try it yourself](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_getelementsbytagname1)

**Get the Value of an Attribute**

In the DOM, attributes are nodes. Unlike element nodes, attribute nodes have text values.

The way to get the value of an attribute, is to get its text value.

This can be done using the getAttribute() method or using the nodeValue property of the attribute node.

**Get an Attribute Value - getAttribute()**

The getAttribute() method returns an attribute **value**.

The following code retrieves the text value of the "lang" attribute of the first <title> element:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
txt=xmlDoc.getElementsByTagName("title")[0].getAttribute("lang");

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_getattribute)

Result:  txt = "en"

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Set the txt variable to be the value of the "lang" attribute of the first title element node

Loop through all <book> elements and get their "category" attributes: [Try it yourself](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_getattribute1)

**Get an Attribute Value - getAttributeNode()**

The getAttributeNode() method returns an attribute **node**.

The following code retrieves the text value of the "lang" attribute of the first <title> element:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("title")[0].getAttributeNode("lang");  
txt=x.nodeValue;

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_getattributenode)

Result:  txt = "en"

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Get the "lang" attribute node of the first <title> element node
3. Set the txt variable to be the value of the attribute

Loop through all <book> elements and get their "category" attributes: [Try it yourself](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_getattributenode2)

[« Previous](http://www.w3schools.com/dom/dom_nodes_navigate.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_set.asp)

**XML DOM Change Node Values**

[« Previous](http://www.w3schools.com/dom/dom_nodes_get.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_remove.asp)

The nodeValue property is used to change a node value.

The setAttribute() method is used to change an attribute value.

Examples

**Try it Yourself - Examples**

The examples below use the XML file [books.xml](http://www.w3schools.com/dom/books.xml).   
A function, [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp), in an external JavaScript is used to load the XML file.

[Change an elements text node](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_change_nodevalue)  
This example uses the nodeValue property to change the text node of the first <title> element in "books.xml".

[Change an attributes value using setAttribute](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_setattribute1)  
This example uses the setAttribute() method to change the value of the "category" attribute of the first <book>.

[Change an attributes value using nodeValue](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_att_nodevalue)  
This example use the nodeValue property to change the value of the "category" attribute of the first <book>.

**Change the Value of an Element**

In the DOM, everything is a node. Element nodes do not have a text value.

The text of an element node is stored in a child node. This node is called a text node.

The way to change the text of an element, is to change the value of the child node (text node).

**Change the Value of a Text Node**

The nodeValue property can be used to change the value of a text node.

The following code changes the text node value of the first <title> element:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];  
x.nodeValue="Easy Cooking";

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_change_nodevalue)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Get the text node of the first <title> element
3. Change the node value of the text node to "Easy Cooking"

Loop through and change the text node of all <title> elements: [Try it yourself](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_change_nodevalue2)

**Change the Value of an Attribute**

In the DOM, attributes are nodes. Unlike element nodes, attribute nodes have text values.

The way to change the value of an attribute, is to change its text value.

This can be done using the setAttribute() method or using the nodeValue property of the attribute node.

**Change an Attribute Using setAttribute()**

The setAttribute() method changes the value of an existing attribute, or creates a new attribute.

The following code changes the category attribute of the <book> element:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName('book');  
x[0].setAttribute("category","food");

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_setattribute1)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Get the first <book> element
3. Change the "category" attribute value to "food"

Loop through all <title> elements and add a new attribute: [Try it yourself](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_setattribute2)

**Note:** If the attribute does not exist, a new attribute is created (with the name and value specified).

**Change an Attribute Using nodeValue**

The nodeValue property can be used to change the value of a attribute node:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("book")[0]  
y=x.getAttributeNode("category");  
y.nodeValue="food";

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_att_nodevalue)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Get the "category" attribute of the first <book> element
3. Change the attribute node value to "food"

**XML DOM Remove Nodes**

[« Previous](http://www.w3schools.com/dom/dom_nodes_set.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_replace.asp)

The removeChild() method removes a specified node.

The removeAttribute() method removes a specified attribute.

Examples

**Try it Yourself - Examples**

The examples below use the XML file [books.xml](http://www.w3schools.com/dom/books.xml).   
A function, [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp), in an external JavaScript is used to load the XML file.

[Remove an element node](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_removechild)  
This example uses removeChild() to remove the first <book> element.

[Remove the current element node](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_removecurrent)  
This example uses parentNode and removeChild() to remove the current <book> element.

[Remove a text node](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_removetextnode)  
This example uses removeChild() to remove the text node from the first <title> element.

[Clear the text of a text node](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_remove_nodevalue)  
This example uses the nodeValue() property to clear the text node of the first <title> element.

[Remove an attribute by name](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_removeattribute)  
This example uses removeAttribute() to remove the "category" attribute from the first <book> element.

[Remove attributes by object](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_removeattributenode)  
This example uses removeAttributeNode() to remove all attributes from all <book> elements.

**Remove an Element Node**

The removeChild() method removes a specified node.

When a node is removed, all its child nodes are also removed.

The following code fragment will remove the first <book> element from the loaded xml:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
y=xmlDoc.getElementsByTagName("book")[0];  
  
xmlDoc.documentElement.removeChild(y);

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_removechild)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Set the variable y to be the element node to remove
3. Remove the element node by using the removeChild() method from the parent node

**Remove Myself - Remove the Current Node**

The removeChild() method is the only way to remove a specified node.

When you have navigated to the node you want to remove, it is possible to remove that node using the parentNode property and the removeChild() method:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("book")[0];  
  
x.parentNode.removeChild(x);

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_removecurrent)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Set the variable y to be the element node to remove
3. Remove the element node by using the parentNode property and the removeChild() method

**Remove a Text Node**

The removeChild() method can also be used to remove a text node:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("title")[0];  
  
y=x.childNodes[0];  
x.removeChild(y);

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_removetextnode)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Set the variable x to be the first title element node
3. Set the variable y to be the text node to remove
4. Remove the element node by using the removeChild() method from the parent node

It is not very common to use removeChild() just to remove the text from a node. The nodeValue property can be used instead. See next paragraph.

**Clear a Text Node**

The nodeValue property can be used to change or clear the value of a text node:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];  
x.nodeValue="";

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_remove_nodevalue)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Set the variable x to be the text node of the first title element
3. Use the nodeValue property to clear the text from the text node

Loop through and change the text node of all <title> elements: [Try it yourself](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_remove_nodevalue2)

**Remove an Attribute Node by Name**

The removeAttribute(*name*) method is used to remove an attribute node by its name.

Example: removeAttribute('category')

The following code fragment removes the "category" attribute in the first <book> element:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("book");  
x[0].removeAttribute("category");

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_removeattribute)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Use getElementsByTagName() to get book nodes
3. Remove the "category" attribute form the first book element node

Loop through and remove the "category" attribute of all <book> elements: [Try it yourself](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_removeattribute2)

**Remove Attribute Nodes by Object**

The removeAttributeNode(*node*) method is used to remove an attribute node, using the node object as parameter.

Example: removeAttributeNode(x)

The following code fragment removes all the attributes of all <book> elements:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("book");  
  
for (i=0;i<x.length;i++)  
{  
while (x[i].attributes.length>0)  
  {  
  attnode=x[i].attributes[0];  
  old\_att=x[i].removeAttributeNode(attnode);  
  }  
}

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_removeattributenode)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Use getElementsByTagName() to get all book nodes
3. For each book element check if there are any attributes
4. While there are attributes in a book element, remove the attribute

[« Previous](http://www.w3schools.com/dom/dom_nodes_set.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_replace.asp)

**WEB HOSTING**

[Best Web Hosting](http://www.lunarpages.com/id/w3schools/goto/w3schools) [eUK Web Hosting](http://www.eukhost.com) [UK Reseller Hosting](http://www.heartinternet.co.uk) [Domain, Hosting & Email](http://www.webhosting.uk.com/cpanel-hosting.php) [Lowest Price Hosting](http://www.justhost.com/track/w3schools/BestPrice)

**WEB BUILDING**

[XML Editor - Free Trial!](http://www.altova.com/ref/?s=w3s_text2&amp;q=xmlspy) [FREE Website BUILDER](http://www.wix.com/eteamhtml/400?utm_campaign=ma_w3schools.com&experiment_id=ma_w3schools.comlink1_400Html&embed_tags=camp&utm_camp=hOUdCO-AmAMQm__j8AM) [FREE Website Creator](http://www.wix.com/eteamhtml/400?utm_campaign=ma_w3schools.com&experiment_id=ma_w3schools.comlink2_400) [Best Website Templates](http://www.dreamtemplate.com/?ref=w3txtfree)

**STATISTICS**

[Browser Statistics](http://www.w3schools.com/browsers/browsers_stats.asp) [OS Statistics](http://www.w3schools.com/browsers/browsers_os.asp) [Display Statistics](http://www.w3schools.com/browsers/browsers_display.asp)

**XML DOM Replace Nodes**

[« Previous](http://www.w3schools.com/dom/dom_nodes_remove.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_create.asp)

The replaceChild() method replaces a specified node.

The nodeValue property replaces text in a text node.

Examples

**Try it Yourself - Examples**

The examples below use the XML file [books.xml](http://www.w3schools.com/dom/books.xml).   
A function, [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp), in an external JavaScript is used to load the XML file.

[Replace an element node](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_replacechild)  
This example uses replaceChild() to replace the first <book> node.

[Replace data in a text node](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_replacedata2)  
This example uses the nodeValue property to replace data in a text node.

**Replace an Element Node**

The replaceChild() method is used to replace a node.

The following code fragment replaces the first <book> element:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.documentElement;  
  
//create a book element, title element and a text node  
newNode=xmlDoc.createElement("book");  
newTitle=xmlDoc.createElement("title");  
newText=xmlDoc.createTextNode("A Notebook");  
  
//add the text node to the title node,  
newTitle.appendChild(newText);  
//add the title node to the book node  
newNode.appendChild(newTitle);  
  
y=xmlDoc.getElementsByTagName("book")[0]  
//replace the first book node with the new node  
x.replaceChild(newNode,y);

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_replacechild)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Create a new element node <book>
3. Create a new element node <title>
4. Create a new text node with the text "A Notebook"
5. Append the new text node to the new element node <title>
6. Append the new element node <title> to the new element node <book>
7. Replace the first <book> element node with the new <book> element node

**Replace Data In a Text Node**

The replaceData() method is used to replace data in a text node.

The replaceData() method has three parameters:

* offset - Where to begin replacing characters. Offset value starts at zero
* length - How many characters to replace
* string - The string to insert

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];  
  
x.replaceData(0,8,"Easy");

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_replacedata)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Get the text node of the first <title> element node
3. Use the replaceData method to replace the eight first characters from the text node with "Easy"

**Use the nodeValue Property Instead**

It is easier to replace the data in a text node using the nodeValue property.

The following code fragment will replace the text node value in the first <title> element with "Easy Italian":

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];  
  
x.nodeValue="Easy Italian";

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_replacedata2)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Get the text node of the first <title> element node
3. Use the nodeValue property to change the text of the text node

You can read more about changing node values in the [Change Node chapter](http://www.w3schools.com/dom/dom_nodes_set.asp).

[« Previous](http://www.w3schools.com/dom/dom_nodes_remove.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_create.asp)

# XML DOM Create Nodes

[« Previous](http://www.w3schools.com/dom/dom_nodes_replace.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_add.asp)

Examples

## Try it Yourself - Examples

The examples below use the XML file [books.xml](http://www.w3schools.com/dom/books.xml).   
A function, [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp), in an external JavaScript is used to load the XML file.

[Create an element node](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createelement1)  
This example uses createElement() to create a new element node, and appendChild() to add it to a node.

[Create an attribute node using createAttribute](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createattribute)   
This example uses createAttribute() to create a new attribute node, and setAttributeNode() to insert it to an element.

[Create an attribute node using setAttribute](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createattribute3)   
This example uses setAttribute() to create a new attribute for an element.

[Create a text node](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createelement1)  
This example uses createTextNode() to create a new text node, and appendChild() to add it to an element.

[Create a CDATA section node](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createcdatasection1)  
This example uses createCDATAsection() to create a CDATA section node, and appendChild() to add it to an element.

[Create a comment node](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createcomment1)  
This example uses createComment() to create a comment node, and appendChild() to add it to an element.

## Create a New Element Node

The createElement() method creates a new element node:

## Example

xmlDoc=loadXMLDoc("books.xml");  
  
newel=xmlDoc.createElement("edition");  
  
x=xmlDoc.getElementsByTagName("book")[0];  
x.appendChild(newel);

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createelement2)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Create a new element node <edition>
3. Append the element node to the first <book> element

Loop through and add an element to all <book> elements: [Try it yourself](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createelement)

## Create a New Attribute Node

The createAttribute() is used to create a new attribute node:

## Example

xmlDoc=loadXMLDoc("books.xml");  
  
newatt=xmlDoc.createAttribute("edition");  
newatt.nodeValue="first";  
  
x=xmlDoc.getElementsByTagName("title");  
x[0].setAttributeNode(newatt);

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createattribute)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Create a new attribute node "edition"
3. Set the value of the attribute node to "first"
4. Add the new attribute node to the first <title> element

Loop through all <title> elements and add a new attribute node: [Try it yourself](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createattribute2)

**Note:** If the attribute already exists, it is replaced by the new one.

## Create an Attribute Using setAttribute()

Since the setAttribute() method creates a new attribute if the attribute does not exist, it can be used to create a new attribute.

## Example

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName('book');  
x[0].setAttribute("edition","first");

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createattribute3)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Set (create) the attribute "edition" with the value "first" for the first <book> element

Loop through all <title> elements and add a new attribute: [Try it yourself](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createattribute4)

## Create a Text Node

The createTextNode() method creates a new text node:

## Example

xmlDoc=loadXMLDoc("books.xml");  
  
newel=xmlDoc.createElement("edition");  
newtext=xmlDoc.createTextNode("first");  
newel.appendChild(newtext);  
  
x=xmlDoc.getElementsByTagName("book")[0];  
x.appendChild(newel);

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createelement1)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Create a new element node <edition>
3. Create a new text node with the text "first"
4. Append the new text node to the element node
5. Append the new element node to the first <book> element

Add an element node, with a text node, to all <book> elements: [Try it yourself](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createelement)

## Create a CDATA Section Node

The createCDATASection() method creates a new CDATA section node.

## Example

xmlDoc=loadXMLDoc("books.xml");  
  
newCDATA=xmlDoc.createCDATASection("Special Offer & Book Sale");  
  
x=xmlDoc.getElementsByTagName("book")[0];  
x.appendChild(newCDATA);

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createcdatasection1)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Create a new CDATA section node
3. Append the new CDATA node to the first <book> element

Loop through, and add a CDATA section, to all <book> elements: [Try it yourself](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createcdatasection)

## Create a Comment Node

The createComment() method creates a new comment node.

## Example

xmlDoc=loadXMLDoc("books.xml");  
  
newComment=xmlDoc.createComment("Revised March 2008");  
  
x=xmlDoc.getElementsByTagName("book")[0];  
x.appendChild(newComment);

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createcomment1)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Create a new comment node
3. Append the new comment node to the first <book> element

Loop through, and add a comment node, to all <book> elements: [Try it yourself](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createcomment)

[« Previous](http://www.w3schools.com/dom/dom_nodes_replace.asp)

**XML DOM Add Nodes**

[« Previous](http://www.w3schools.com/dom/dom_nodes_create.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_clone.asp)

Examples

**Try it Yourself - Examples**

The examples below use the XML file [books.xml](http://www.w3schools.com/dom/books.xml).   
A function, [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp), in an external JavaScript is used to load the XML file.

[Add a node after the last child node](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createelement2)  
This example uses appendChild() to add a child node to an existing node.

[Add a node before a specified child node](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_insertbefore)  
This example uses insertBefore() to insert a node before a specified child node.

[Add a new attribute](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createattribute3)  
This example uses the setAttribute() method to add a new attribute.

[Add data to a text node](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_insertdata)  
This example uses insertData() to insert data into an existing text node.

**Add a Node - appendChild()**

The appendChild() method adds a child node to an existing node.

The new node is added (appended) after any existing child nodes.

**Note:** Use insertBefore() if the position of the node is important.

The following code fragment creates an element (<edition>), and adds it after the last child of the first <book> element:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
newel=xmlDoc.createElement("edition");  
  
x=xmlDoc.getElementsByTagName("book")[0];  
x.appendChild(newel);

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createelement2)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Create a new node <edition>
3. Append the node to the first <book> element

Loop through and append an element to all <book> elements:[Try it yourself](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createelement" \t "_blank)

**Insert a Node - insertBefore()**

The insertBefore() method is used to insert a node before a specified child node.

This method is useful when the position of the added node is important:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
newNode=xmlDoc.createElement("book");  
  
x=xmlDoc.documentElement;  
y=xmlDoc.getElementsByTagName("book")[3];  
  
x.insertBefore(newNode,y);

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_insertbefore)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Create a new element node <book>
3. Insert the new node in front of the last <book> element node

If the second parameter of insertBefore() is null, the new node will be added after the last existing child node.

**x.insertBefore(newNode,null)** and **x.appendChild(newNode)** will both append a new child node to x.

**Add a New Attribute**

There is no method called addAtribute().

The setAttribute() method creates a new attribute if the attribute does not exist:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName('book');  
x[0].setAttribute("edition","first");

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_createattribute3)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Set (create) the attribute "edition" with the value "first" for the first <book> element

**Note:** If the attribute already exists, the setAttribute() method will overwrite the existing value.

**Add Text to a Text Node - insertData()**

The insertData() method inserts data into an existing text node.

The insertData() method has two parameters:

* offset - Where to begin inserting characters (starts at zero)
* string - The string to insert

The following code fragment will add "Easy" to the text node of the first <title> element of the loaded XML:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];  
  
x.insertData(0,"Easy ");

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_insertdata)

[« Previous](http://www.w3schools.com/dom/dom_nodes_create.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_nodes_clone.asp)

**XML DOM Clone Nodes**

[« Previous](http://www.w3schools.com/dom/dom_nodes_add.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_httprequest.asp)

Examples

**Try it Yourself - Examples**

The examples below use the XML file [books.xml](http://www.w3schools.com/dom/books.xml).   
A function, [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp), in an external JavaScript is used to load the XML file.

[Copy a node and append it to an existing node](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_clonenode)  
This example uses cloneNode() to copy a node and append it to the root node of the XML document

**Copy a Node**

The cloneNode() method creates a copy of a specified node.

The cloneNode() method has a parameter (true or false). This parameter indicates if the cloned node should include all attributes and child nodes of the original node.

The following code fragment copies the first <book> node and appends it to the root node of the document:

**Example**

xmlDoc=loadXMLDoc("books.xml");  
  
oldNode=xmlDoc.getElementsByTagName('book')[0];  
newNode=oldNode.cloneNode(true);  
xmlDoc.documentElement.appendChild(newNode);  
  
//Output all titles  
y=xmlDoc.getElementsByTagName("title");  
for (i=0;i<y.length;i++)  
{  
document.write(y[i].childNodes[0].nodeValue);  
document.write("  
");  
}

Output:

Everyday Italian  
Harry Potter  
XQuery Kick Start  
Learning XML  
Everyday Italian

[Try it yourself »](http://www.w3schools.com/dom/tryit.asp?filename=try_dom_clonenode)

Example explained:

1. Load "[books.xml](http://www.w3schools.com/dom/books.xml)" into xmlDoc using [loadXMLDoc()](http://www.w3schools.com/dom/dom_loadxmldoc.asp)
2. Get the node to copy
3. Copy the node into "newNode" using the cloneNode method
4. Append the new node to the the root node of the XML document
5. Output all titles for all books in the document

[« Previous](http://www.w3schools.com/dom/dom_nodes_add.asp)

[Next Chapter »](http://www.w3schools.com/dom/dom_httprequest.asp)