XML

XML was designed to transport and store data. HTML was designed to display data.

What is XML?

XML stands for EXtensible Markup Language. XML is a markup language much like HTML. XML was designed to carry data, not to display data. XML tags are not predefined. You must define your own tags. XML is designed to be self-descriptive. XML is a W3C Recommendation.

The Difference Between XML and HTML

XML is not a replacement for HTML. XML and HTML were designed with different goals. XML was designed to transport and store data, with focus on what data is. HTML was designed to display data, with focus on how data looks. HTML is about displaying information, while XML is about carrying information.

XML Does Not DO Anything

Maybe it is a little hard to understand, but XML does not DO anything. XML was created to structure, store, and transport information. The following example is a note to Tove, from Jani, stored as XML:

```
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

The note above is quite self descriptive. It has sender and receiver information, it also has a heading and a message body. But still, this XML document does not DO anything. It is just information wrapped in tags. Someone must write a piece of software to send, receive or display it.

With XML You Invent Your Own Tags

The tags in the example above (like <to> and <from>) are not defined in any XML standard. These tags are "invented" by the author of the XML document. That is because the XML language has no predefined tags. The tags used in HTML are predefined. HTML documents can only use tags defined in the HTML standard (like <p>, <h1>, etc.). XML allows the author to define his/her own tags and his/her own document structure. XML is Not a Replacement for HTML XML is a complement to HTML. It is important to understand that XML is not a replacement for HTML. In most web applications, XML is used to transport data, while HTML is used to format and display the data.

My best description of XML is this:

XML is a software- and hardware-independent tool for carrying information. XML is a W3C Recommendation. XML became a W3C Recommendation February 10, 1998.

XML is Everywhere

XML is now as important for the Web as HTML was to the foundation of the Web.
XML is the most common tool for data transmissions between all sorts of applications.

XML Separates Data from HTML

If you need to display dynamic data in your HTML document, it will take a lot of work to edit the HTML each time the data changes. With XML, data can be stored in separate XML files. This way you can concentrate on using HTML for layout and display, and be sure that changes in the underlying data will not require any changes to the HTML. With a few lines of JavaScript code, you can read an external XML file and update the data content of your web page.

XML Simplifies Data Sharing

In the real world, computer systems and databases contain data in incompatible formats. XML data is stored in plain text format. This provides a software- and hardware-independent way of storing data. This makes it much easier to create data that can be shared by different applications.

XML Simplifies Data Transport

One of the most time-consuming challenges for developers is to exchange data between incompatible systems over the Internet. Exchanging data as XML greatly reduces this complexity, since the data can be read by different incompatible applications.

XML Simplifies Platform Changes

Upgrading to new systems (hardware or software platforms), is always time consuming. Large amounts of data must be converted and incompatible data is often lost. XML data is stored in text format. This makes it easier to expand or upgrade to new operating systems, new applications, or new browsers, without losing data.

XML Makes Your Data More Available

Different applications can access your data, not only in HTML pages, but also from XML data sources. With XML, your data can be available to all kinds of "reading machines" (Handheld computers, voice machines, news feeds, etc), and make it more available for blind people, or people with other disabilities.

XML Tree

XML documents form a tree structure that starts at "the root" and branches to "the leaves".

An Example XML Document
XML documents use a self-describing and simple syntax:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

The first line is the XML declaration. It defines the XML version (1.0) and the encoding used (ISO-8859-1 = Latin-1/West European character set). The next line describes the root element of the document (like saying: "this document is a note"):

```
<note>
```

The next 4 lines describe 4 child elements of the root (to, from, heading, and body):

```
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
```

And finally the last line defines the end of the root element:

```
</note>
```

You can assume, from this example, that the XML document contains a note to Tove from Jani. Don't you agree that XML is pretty self-descriptive?

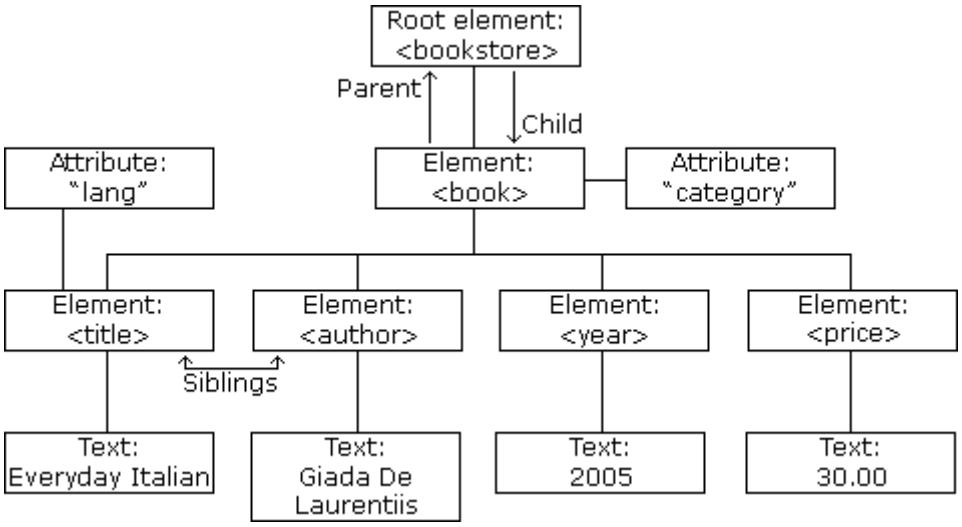
XML Documents Form a Tree Structure

XML documents must contain a root element. This element is "the parent" of all other elements. The elements in an XML document form a document tree. The tree starts at the root and branches to the lowest level of the tree. All elements can have sub elements (child elements):

```
<root>
  <child>
    <subchild>.... </subchild>
  </child>
</root>
```

The terms parent, child, and sibling are used to describe the relationships between elements. Parent elements have children. Children on the same level are called siblings (brothers or sisters). All elements can have text content and attributes (just like in HTML).

Example:



The image above represents one book in the XML below:

```
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

The root element in the example is <bookstore>. All <book> elements in the document are contained within <bookstore>. The <book> element has 4 children: <title>,< author>, <year>, <price>.

XML Syntax Rules

The syntax rules of XML are very simple and logical. The rules are easy to learn, and easy to use. All XML Elements Must Have a Closing Tag. In HTML, elements do not have to have a closing tag:

```
<p>This is a paragraph
<p>This is another paragraph
```

In XML, it is illegal to omit the closing tag. All elements must have a closing tag:

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

Note: You might have noticed from the previous example that the XML declaration did not have a closing tag. This is not an error. The declaration is not a part of the XML document itself, and it has no closing tag.

XML tags are case sensitive. The tag <Letter> is different from the tag <letter>. Opening and closing tags must be written with the same case:

```
<Message>This is incorrect</message>
<message>This is correct</message>
```

Note: "Opening and closing tags" are often referred to as "Start and end tags". Use whatever you prefer. It is exactly the same thing.

XML Elements Must be Properly Nested

In HTML, you might see improperly nested elements:

```
<b><i>This text is bold and italic</b></i>
```

In XML, all elements must be properly nested within each other:

```
<b><i>This text is bold and italic</i></b>
```

In the example above, "Properly nested" simply means that since the <i> element is opened inside the <b> element, it must be closed inside the <b> element.

XML Documents Must Have a Root Element

XML documents must contain one element that is the parent of all other elements. This element is called the root element.

```
<root>
  <child>
    <subchild>.... </subchild>
  </child>
</root>
```

XML Attribute Values Must be Quoted

XML elements can have attributes in name/value pairs just like in HTML. In XML, the attribute values must always be quoted. Study the two XML documents below. The first one is incorrect, the second is correct:

```
<note date=12/11/2007>
  <to>Tove</to>
  <from>Jani</from>
</note>
```

```
<note date="12/11/2007">
  <to>Tove</to>
  <from>Jani</from>
</note>
```

The error in the first document is that the date attribute in the note element is not quoted.

Entity References

Some characters have a special meaning in XML. If you place a character like "<" inside an XML element, it will generate an error because the parser interprets it as the start of a new element. This will generate an XML error:

```
<message>if salary < 1000 then</message>
```

To avoid this error, replace the "<" character with an entity reference:

```
<message>if salary &lt; 1000 then</message>
```

There are 5 predefined entity references in XML:

<	<	less than
>	>	greater than
&	&	ampersand
'	'	Apostrophe
"	"	quotation mark

Note: Only the characters "<" and "&" are strictly illegal in XML. The greater than character is legal, but it is a good habit to replace it.

Comments in XML

The syntax for writing comments in XML is similar to that of HTML.

```
<!-- This is a comment -->
```

White-space is Preserved in XML

HTML truncates multiple white-space characters to one single white-space:

HTML:	Hello Tove
Output:	Hello Tove

With XML, the white-space in a document is not truncated.

XML Stores New Line as LF

In Windows applications, a new line is normally stored as a pair of characters: carriage return (CR) and line feed (LF). In Unix applications, a new line is normally stored as an LF character. Macintosh applications also use an LF to store a new line. XML stores a new line as LF.

XML ELEMENTS

What is an XML Element?

An XML element is everything from (including) the element's start tag to (including) the element's end tag. An element can contain: other elements text, attributes or a mix of all of the above...

```
<bookstore>
  <book category="CHILDREN">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

In the example above, <bookstore> and <book> have element contents, because they contain other elements. <book> also has an attribute (category="CHILDREN"). <title>, <author>, <year>, and <price> have text content because they contain text.

XML Naming Rules

XML elements must follow these naming rules:

- Names can contain letters, numbers, and other characters
- Names cannot start with a number or punctuation character
- Names cannot start with the letters xml (or XML, or Xml, etc)
- Names cannot contain spaces
- Any name can be used, no words are reserved.

Best Naming Practices

- Make names descriptive. Names with an underscore separator are nice: <first_name>, <last_name>.
- Names should be short and simple, like this: <book_title> not like this: <the_title_of_the_book>.
- Avoid "-" characters. If you name something "first-name," some software may think you want to subtract name from first.
- Avoid "." characters. If you name something "first.name," some software may think that "name" is a property of the object "first."
- Avoid ":" characters. Colons are reserved to be used for something called namespaces (more later).
- XML documents often have a corresponding database. A good practice is to use the naming rules of your database for the elements in the XML documents.
- Non-English letters like éòá are perfectly legal in XML, but watch out for problems if your software vendor doesn't support them.

XML Elements are Extensible

XML elements can be extended to carry more information. Look at the following XML example:

```
<note>
<to>Tove</to>
<from>Jani</from>
<body>Don't forget me this weekend!</body>
</note>
```

Let's imagine that we created an application that extracted the <to>, <from>, and <body> elements from the XML document to produce this output:

```
MESSAGE
To: Tove
From: Jani
Don't forget me this weekend!
```

Imagine that the author of the XML document added some extra information to it:

```
<note>
<date>2008-01-10</date>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

Should the application break or crash?

No. The application should still be able to find the <to>, <from>, and <body> elements in the XML document and produce the same output.

One of the beauties of XML, is that it can be extended without breaking applications.

XML ATTRIBUTES

XML elements can have attributes, just like HTML. Attributes provide additional information about an element.

In HTML, attributes provide additional information about elements:

```
<img src="computer.gif">
<a href="demo.asp">
```

Attributes often provide information that is not a part of the data. In the example below, the file type is irrelevant to the data, but can be important to the software that wants to manipulate the element:

```
<file type="gif">computer.gif</file>
```

XML Attributes Must be Quoted

Attribute values must always be quoted. Either single or double quotes can be used. For a person's sex, the person element can be written like this:

```
<person sex="female">
```

or like this:

```
<person sex='female'>
```

If the attribute value itself contains double quotes you can use single quotes, like in this example:

```
<gangster name='George "Shotgun" Ziegler'>
```

or you can use character entities:

```
<gangster name="George &quot;Shotgun&quot; Ziegler">
```

XML Elements vs. Attributes

Take a look at these examples:

```
<person sex="female">
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>

<person>
  <sex>female</sex>
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

In the first example sex is an attribute. In the last, sex is an element. Both examples provide the same information. There are no rules about when to use attributes or when to use elements. Attributes are handy in HTML. In XML my advice is to avoid them. Use elements instead.

My Favorite Way

The following three XML documents contain exactly the same information:

A date attribute is used in the first example:

```
<note date="10/01/2008">
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

A date element is used in the second example:

```
<note>
  <date>10/01/2008</date>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

An expanded date element is used in the third: (THIS IS MY FAVORITE):

```
<note>
  <date>
    <day>10</day>
    <month>01</month>
    <year>2008</year>
  </date>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Avoid XML Attributes?

Some of the problems with using attributes are:

- attributes cannot contain multiple values (elements can)
- attributes cannot contain tree structures (elements can)
- attributes are not easily expandable (for future changes)
- Attributes are difficult to read and maintain. Use elements for data. Use attributes for information that is not relevant to the data.
- Don't end up like this:

```
<note day="10" month="01" year="2008"
to="Tove" from="Jani" heading="Reminder"
body="Don't forget me this weekend!">
</note>
```

XML Attributes for Metadata

Sometimes ID references are assigned to elements. These IDs can be used to identify XML elements in much the same way as the id attribute in HTML. This example demonstrates this:

```
<messages>
  <note id="501">
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </note>
  <note id="502">
    <to>Jani</to>
    <from>Tove</from>
    <heading>Re: Reminder</heading>
    <body>I will not</body>
  </note>
</messages>
```

The id attributes above are for identifying the different notes. It is not a part of the note itself.
What I'm trying to say here is that metadata (data about data) should be stored as attributes, and the data itself should be stored as elements.

XML Validation

XML with correct syntax is "Well Formed" XML. XML validated against a DTD is "Valid" XML.

Well Formed XML Documents

A "Well Formed" XML document has correct XML syntax. The syntax rules were described in the previous chapters:
XML documents must have a root element. XML elements must have a closing tag. XML tags are case sensitive. XML elements must be properly nested. XML attribute values must be quoted

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

Valid XML Documents

A "Valid" XML document is a "Well Formed" XML document, which also conforms to the rules of a Document Type Definition (DTD):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE note SYSTEM "Note.dtd">
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

The DOCTYPE declaration in the example above, is a reference to an external DTD file. The content of the file is shown in the paragraph below.

XML DTD

The purpose of a DTD is to define the structure of an XML document. It defines the structure with a list of legal elements:

```
<!DOCTYPE note
[
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
```

XML Schema

W3C supports an XML-based alternative to DTD, called XML Schema:

```
<xs:element name="note">

<xs:complexType>
  <xs:sequence>
    <xs:element name="to" type="xs:string"/>
    <xs:element name="from" type="xs:string"/>
    <xs:element name="heading" type="xs:string"/>
    <xs:element name="body" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

</xs:element>
```

Well Formed XML Documents:

A "Well Formed" XML document has correct XML syntax.
The syntax rules were described in the previous chapters:

- XML documents must have a root element
- XML elements must have a closing tag
- XML tags are case sensitive
- XML elements must be properly nested
- XML attribute values must be quoted

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

Valid XML Documents

A "Valid" XML document is a "Well Formed" XML document, which also conforms to the rules of a Document Type Definition (DTD):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE note SYSTEM "Note.dtd">
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

The DOCTYPE declaration in the example above, is a reference to an external DTD file. The content of the file is shown in the paragraph below.

XML DTD

The purpose of a DTD is to define the structure of an XML document. It defines the structure with a list of legal elements:

```
<!DOCTYPE note
[
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
```

XML Schema

W3C supports an XML-based alternative to DTD, called XML Schema:

```
<xs:element name="note">

<xs:complexType>
  <xs:sequence>
    <xs:element name="to" type="xs:string"/>
    <xs:element name="from" type="xs:string"/>
    <xs:element name="heading" type="xs:string"/>
    <xs:element name="body" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

</xs:element>
```

Viewing XML Files

```
<?xml version="1.0" encoding="ISO-8859-1"?>
 - <note>
      <to>Tove</to>
      <from>Jani</from>
      <heading>Reminder</heading>
      <body>Don't forget me this weekend!</body>
   </note>
```

The XML document will be displayed with color-coded root and child elements. A plus (+) or minus sign (-) to the left of the elements can be clicked to expand or collapse the element structure. To view the raw XML source (without the + and - signs), select "View Page Source" or "View Source" from the browser menu. Note: In Chrome, Opera, and Safari, only the element text will be displayed. To view the raw XML, you must right click the page and select "View Source"

Why Does XML Display Like This?

XML documents do not carry information about how to display the data. Since XML tags are "invented" by the author of the XML document, browsers do not know if a tag like <table> describes an HTML table or a dining table. Without any information about how to display the data, most browsers will just display the XML document as it is. In the next chapters, we will take a look at different solutions to the display problem, using CSS, XSLT and JavaScript.

Displaying XML with CSS

With CSS (Cascading Style Sheets) you can add display information to an XML document.

It is possible to use CSS to format an XML document. Below is an example of how to use a CSS style sheet to format an XML document: Below is a fraction of the XML file. The second line links the XML file to the CSS file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/css" href="cd_catalog.css"?>
<CATALOG>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
  <CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bonnie Tyler</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS Records</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1988</YEAR>
  </CD>
.
.
.
</CATALOG>
```

Formatting XML with CSS is not the most common method.

Displaying XML with XSLT

With XSLT you can transform an XML document into HTML.

Parse an XML Document
The following code fragment parses an XML document into an XML DOM object:

```
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
else
  {// code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.open("GET","books.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;
```

Parse an XML String
The following code fragment parses an XML string into an XML DOM object:

```
txt="<bookstore><book>";
txt=txt+"<title>Everyday Italian</title>";
txt=txt+"<author>Giada De Laurentiis</author>";
txt=txt+"<year>2005</year>";
txt=txt+"</book></bookstore>";

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);
  }
```

XML DOM

A DOM (Document Object Model) defines a standard way for accessing and manipulating documents. The XML DOM defines a standard way for accessing and manipulating XML documents. The XML DOM views an XML document as a tree-structure. All elements can be accessed through the DOM tree. Their content (text and attributes) can be modified or deleted, and new elements can be created. The elements, their text, and their attributes are all known as nodes.

The HTML DOM

The HTML DOM defines a standard way for accessing and manipulating HTML documents. All HTML elements can be accessed through the HTML DOM. The following example parses an XML document ("note.xml") into an XML DOM object and then extracts some info from it with a JavaScript:

Example

```
<html>
<body>
<h1>W3Schools Internal Note</h1>
<p><b>To:</b> <span id="to"></span><br />
<b>From:</b> <span id="from"></span><br />
<b>Message:</b> <span id="message"></span>
<script type="text/javascript">
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
```

```
else
  {// code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.open("GET","note.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;

document.getElementById("to").innerHTML=
xmlDoc.getElementsByTagName("to")[0].childNodes[0].nodeValue;
document.getElementById("from").innerHTML=
xmlDoc.getElementsByTagName("from")[0].childNodes[0].nodeValue;
document.getElementById("message").innerHTML=
xmlDoc.getElementsByTagName("body")[0].childNodes[0].nodeValue;
</script>
</body>
</html>
```

Load an XML String - Cross-browser Example

The following example parses an XML string into an XML DOM object and then extracts some info from it with a JavaScript:

Example

```
<html>
<body>
<h1>W3Schools Internal Note</h1>
<p><b>To:</b> <span id="to"></span><br />
<b>From:</b> <span id="from"></span><br />
<b>Message:</b> <span id="message"></span></p>

<script>
txt="<note>";
txt=txt+"<to>Tove</to>";
txt=txt+"<from>Jani</from>";
txt=txt+"<heading>Reminder</heading>";
txt=txt+"<body>Don't forget me this weekend!</body>";
txt=txt+"</note>";

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);
  }

document.getElementById("to").innerHTML=
xmlDoc.getElementsByTagName("to")[0].childNodes[0].nodeValue;
document.getElementById("from").innerHTML=
xmlDoc.getElementsByTagName("from")[0].childNodes[0].nodeValue;
document.getElementById("message").innerHTML=
xmlDoc.getElementsByTagName("body")[0].childNodes[0].nodeValue;
</script>
</body>
</html>
```

XML to HTML

In the following example, we loop through an XML file ("cd_catalog.xml"), and display the contents of each CD element as an HTML table row:

Example

```
<html>
<body>

<script type="text/javascript">
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
else
  {// code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.open("GET","cd_catalog.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;

document.write("<table border='1'>");
var x=xmlDoc.getElementsByTagName("CD");
for (i=0;i<x.length;i++)
  {
  document.write("<tr><td>");
  document.write(x[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue);
  document.write("</td><td>");
  document.write(x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue);
  document.write("</td></tr>");
  }
document.write("</table>");
</script>

</body>
</html>
```

PCDATA - Parsed Character Data

XML parsers normally parse all the text in an XML document. When an XML element is parsed, the text between the XML tags is also parsed:

```
<message>This text is also parsed</message>
```

The parser does this because XML elements can contain other elements, as in this example, where the <name> element contains two other elements (first and last):

```
<name><first>Bill</first><last>Gates</last></name>
```

and the parser will break it up into sub-elements like this:

```
<name>
  <first>Bill</first>
  <last>Gates</last>
</name>
```

Parsed Character Data (PCDATA) is a term used about text data that will be parsed by the XML parser.

CDATA - (Unparsed) Character Data

The term CDATA is used about text data that should not be parsed by the XML parser. Characters like "<" and "&" are illegal in XML elements. "<" will generate an error because the parser interprets it as the start of a new element. "&" will generate an error because the parser interprets it as the start of an character entity. Some text, like JavaScript code, contains a lot of "<" or "&" characters. To avoid errors script code can be defined as CDATA. Everything inside a CDATA section is ignored by the parser.

EXAMPLES:

1. View a simple XML file (note.xml)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

2. View the same XML file with an error

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
        <to>Tove</to>
        <from>Jani</Ffrom>
        <heading>Reminder</heading>
        <body>Don't forget me this weekend!</body>
</note>
```

3. View an XML CD catalog

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CATALOG>
        <CD>
                <TITLE>Empire Burlesque</TITLE>
                <ARTIST>Bob Dylan</ARTIST>
                <COUNTRY>USA</COUNTRY>
                <COMPANY>Columbia</COMPANY>
                <PRICE>10.90</PRICE>
                <YEAR>1985</YEAR>
        </CD>
        <CD>
                <TITLE>Hide your heart</TITLE>
                <ARTIST>Bonnie Tyler</ARTIST>
                <COUNTRY>UK</COUNTRY>
                <COMPANY>CBS Records</COMPANY>
                <PRICE>9.90</PRICE>
                <YEAR>1988</YEAR>
        </CD>
        <CD>
                <TITLE>Greatest Hits</TITLE>
                <ARTIST>Dolly Parton</ARTIST>
                <COUNTRY>USA</COUNTRY>
                <COMPANY>RCA</COMPANY>
                <PRICE>9.90</PRICE>
                <YEAR>1982</YEAR>
        </CD>
        <CD>
                <TITLE>Still got the blues</TITLE>
                <ARTIST>Gary Moore</ARTIST>
                <COUNTRY>UK</COUNTRY>
                <COMPANY>Virgin records</COMPANY>
                <PRICE>10.20</PRICE>
                <YEAR>1990</YEAR>
        </CD>
        <CD>
                <TITLE>Eros</TITLE>
                <ARTIST>Eros Ramazzotti</ARTIST>
                <COUNTRY>EU</COUNTRY>
                <COMPANY>BMG</COMPANY>
                <PRICE>9.90</PRICE>
                <YEAR>1997</YEAR>
        </CD>
        <CD>
                <TITLE>One night only</TITLE>
                <ARTIST>Bee Gees</ARTIST>
                <COUNTRY>UK</COUNTRY>
                <COMPANY>Polydor</COMPANY>
                <PRICE>10.90</PRICE>
                <YEAR>1998</YEAR>
</CD>
<CD>
                <TITLE>Sylvias Mother</TITLE>
                <ARTIST>Dr.Hook</ARTIST>
                <COUNTRY>UK</COUNTRY>
                <COMPANY>CBS</COMPANY>
                <PRICE>8.10</PRICE>
                <YEAR>1973</YEAR>
</CD>
<CD>
                <TITLE>Maggie May</TITLE>
                <ARTIST>Rod Stewart</ARTIST>
                <COUNTRY>UK</COUNTRY>
                <COMPANY>Pickwick</COMPANY>
                <PRICE>8.50</PRICE>
                <YEAR>1990</YEAR>
</CD>
<CD>
                <TITLE>Romanza</TITLE>
                <ARTIST>Andrea Bocelli</ARTIST>
                <COUNTRY>EU</COUNTRY>
                <COMPANY>Polydor</COMPANY>
                <PRICE>10.80</PRICE>
                <YEAR>1996</YEAR>
</CD>
<CD>
                <TITLE>When a man loves a woman</TITLE>
                <ARTIST>Percy Sledge</ARTIST>
                <COUNTRY>USA</COUNTRY>
                <COMPANY>Atlantic</COMPANY>
                <PRICE>8.70</PRICE>
                <YEAR>1987</YEAR>
</CD>
<CD>
                <TITLE>Tupelo Honey</TITLE>
                <ARTIST>Van Morrison</ARTIST>
                <COUNTRY>UK</COUNTRY>
                <COMPANY>Polydor</COMPANY>
                <PRICE>8.20</PRICE>
                <YEAR>1971</YEAR>       </CD>
<CD>
                <TITLE>Soulsville</TITLE>
                <ARTIST>Jorn Hoel</ARTIST>
                <COUNTRY>Norway</COUNTRY>
                <COMPANY>WEA</COMPANY>
                <PRICE>7.90</PRICE>
                <YEAR>1996</YEAR>
</CD>
<CD>
                <TITLE>The very best of</TITLE>
                <ARTIST>Cat Stevens</ARTIST>
                <COUNTRY>UK</COUNTRY>
                <COMPANY>Island</COMPANY>
                <PRICE>8.90</PRICE>
                <YEAR>1990</YEAR>
</CD>
<CD>
                <TITLE>Red</TITLE>
                <ARTIST>The Communards</ARTIST>
                <COUNTRY>UK</COUNTRY>
                <COMPANY>London</COMPANY>
                <PRICE>7.80</PRICE>
                <YEAR>1987</YEAR>
</CD>
<CD>
```

```
        <TITLE>Unchain my heart</TITLE>
        <ARTIST>Joe Cocker</ARTIST>
        <COUNTRY>USA</COUNTRY>
        <COMPANY>EMI</COMPANY>
        <PRICE>8.20</PRICE>
        <YEAR>1987</YEAR>
    </CD> </CATALOG>
```

4. View an XML plant catalog

```
<?xml version="1.0" encoding="ISO-8859-1"?>

 <CATALOG>
    <PLANT>
        <COMMON>Marsh Marigold</COMMON>
        <BOTANICAL>Caltha palustris</BOTANICAL>
        <ZONE>4</ZONE>
        <LIGHT>Mostly Sunny</LIGHT>
        <PRICE>$6.81</PRICE>
        <AVAILABILITY>051799</AVAILABILITY>
    </PLANT>
    <PLANT>
        <COMMON>Cowslip</COMMON>
        <BOTANICAL>Caltha palustris</BOTANICAL>
        <ZONE>4</ZONE>
        <LIGHT>Mostly Shady</LIGHT>
        <PRICE>$9.90</PRICE>
        <AVAILABILITY>030699</AVAILABILITY>
    </PLANT>
    <PLANT>
        <COMMON>Dutchman's-Breeches</COMMON>
        <BOTANICAL>Dicentra cucullaria</BOTANICAL>
        <ZONE>3</ZONE>
        <LIGHT>Mostly Shady</LIGHT>
        <PRICE>$6.44</PRICE>
        <AVAILABILITY>012099</AVAILABILITY>
    </PLANT>
    <PLANT>
        <COMMON>Ginger, Wild</COMMON>
        <BOTANICAL>Asarum canadense</BOTANICAL>
        <ZONE>3</ZONE>
        <LIGHT>Mostly Shady</LIGHT>
        <PRICE>$9.03</PRICE>
        <AVAILABILITY>041899</AVAILABILITY>
    </PLANT>
    <PLANT>
        <COMMON>Trillium</COMMON>
        <BOTANICAL>Trillium grandiflorum</BOTANICAL>
        <ZONE>5</ZONE>
        <LIGHT>Sun or Shade</LIGHT>
        <PRICE>$3.90</PRICE>
        <AVAILABILITY>042999</AVAILABILITY>
    </PLANT>
    <PLANT>
        <COMMON>Wake Robin</COMMON>
        <BOTANICAL>Trillium grandiflorum</BOTANICAL>
        <ZONE>5</ZONE><LIGHT>Sun or Shade</LIGHT>
        <PRICE>$3.20</PRICE>
        <AVAILABILITY>022199</AVAILABILITY>
    </PLANT>
    <PLANT>
        <COMMON>Anemone</COMMON>
        <BOTANICAL>Anemone blanda</BOTANICAL>
        <ZONE>6</ZONE>
        <LIGHT>Mostly Shady</LIGHT>
        <PRICE>$8.86</PRICE>
        <AVAILABILITY>122698</AVAILABILITY>
    </PLANT>
    <PLANT>
        <COMMON>Grecian Windflower</COMMON>
        <BOTANICAL>Anemone blanda</BOTANICAL>
        <ZONE>6</ZONE>
        <LIGHT>Mostly Shady</LIGHT>
        <PRICE>$9.16</PRICE>
        <AVAILABILITY>071099</AVAILABILITY>
    </PLANT>

    <PLANT>
        <COMMON>Bee Balm</COMMON>
        <BOTANICAL>Monarda didyma</BOTANICAL>
        <ZONE>4</ZONE>
        <LIGHT>Shade</LIGHT>
        <PRICE>$4.59</PRICE>
        <AVAILABILITY>050399</AVAILABILITY>
    </PLANT>
    <PLANT>
        <COMMON>Cinquefoil</COMMON>
        <BOTANICAL>Potentilla</BOTANICAL>
        <ZONE>Annual</ZONE>
        <LIGHT>Shade</LIGHT>
        <PRICE>$7.06</PRICE>
        <AVAILABILITY>052599</AVAILABILITY>
    </PLANT>
    <PLANT>
        <COMMON>Primrose</COMMON>
        <BOTANICAL>Oenothera</BOTANICAL>
        <ZONE>3 - 5</ZONE>
        <LIGHT>Sunny</LIGHT>
        <PRICE>$6.56</PRICE>
        <AVAILABILITY>013099</AVAILABILITY>
    </PLANT>
    <PLANT>
        <COMMON>Gentian</COMMON>
        <BOTANICAL>Gentiana</BOTANICAL>
        <ZONE>4</ZONE>
        <LIGHT>Sun or Shade</LIGHT>
        <PRICE>$7.81</PRICE>
        <AVAILABILITY>051899</AVAILABILITY>
    </PLANT>
    <PLANT>
        <COMMON>Blue Gentian</COMMON>
        <BOTANICAL>Gentiana</BOTANICAL>
        <ZONE>4</ZONE>
        <LIGHT>Sun or Shade</LIGHT>
        <PRICE>$8.56</PRICE>
        <AVAILABILITY>050299</AVAILABILITY>
    </PLANT>
</CATALOG>
```

5. View an XML food menu

```
<?xml version="1.0" encoding="ISO-8859-1"?>

 <breakfast_menu>
    <food>
        <name>Belgian Waffles</name>
        <price>$5.95</price>
        <description>two of our famous Belgian Waffles
with plenty of real maple syrup</description>
        <calories>650</calories>
    </food>
    <food>
        <name>Strawberry Belgian Waffles</name>
```

```
        <price>$7.95</price>
        <description>light Belgian waffles covered with
strawberries and whipped cream</description>
        <calories>900</calories>
    </food>
    <food>
        <name>Berry-Berry Belgian Waffles</name>
        <price>$8.95</price>
        <description>light Belgian waffles covered with an
assortment of fresh berries and whipped cream</description>
        <calories>900</calories>
    </food>
    <food>
        <name>French Toast</name>
        <price>$4.50</price>
        <description>thick slices made from our
homemade sourdough bread</description>
        <calories>600</calories>
    </food>
    <food>
        <name>Homestyle Breakfast</name>
        <price>$6.95</price>
        <description>two eggs, bacon or sausage, toast,
and our ever-popular hash browns</description>
        <calories>950</calories>
    </food>
</breakfast_menu>
```

XML and CSS

1. View an XML CD catalog

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CATALOG>
    <CD>
        <TITLE>Empire Burlesque</TITLE>
        <ARTIST>Bob Dylan</ARTIST>
        <COUNTRY>USA</COUNTRY>
        <COMPANY>Columbia</COMPANY>
        <PRICE>10.90</PRICE>
        <YEAR>1985</YEAR>
    </CD>
    <CD>
        <TITLE>Hide your heart</TITLE>
        <ARTIST>Bonnie Tyler</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>CBS Records</COMPANY>
        <PRICE>9.90</PRICE>
        <YEAR>1988</YEAR>
    </CD>
    <CD>
        <TITLE>Greatest Hits</TITLE>
        <ARTIST>Dolly Parton</ARTIST>
        <COUNTRY>USA</COUNTRY>
        <COMPANY>RCA</COMPANY>
        <PRICE>9.90</PRICE>
        <YEAR>1982</YEAR>
    </CD>
    <CD>
        <TITLE>Still got the blues</TITLE>
        <ARTIST>Gary Moore</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>Virgin records</COMPANY>
        <PRICE>10.20</PRICE>
        <YEAR>1990</YEAR>
    </CD>
    <CD>
        <TITLE>Eros</TITLE>
        <ARTIST>Eros Ramazzotti</ARTIST>
        <COUNTRY>EU</COUNTRY>
        <COMPANY>BMG</COMPANY>
        <PRICE>9.90</PRICE>
        <YEAR>1997</YEAR>
    </CD>
    <CD>
        <TITLE>One night only</TITLE>
        <ARTIST>Bee Gees</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>Polydor</COMPANY>
        <PRICE>10.90</PRICE>
        <YEAR>1998</YEAR>
    </CD>
    <CD>
        <TITLE>Sylvias Mother</TITLE>
        <ARTIST>Dr.Hook</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>CBS</COMPANY>
        <PRICE>8.10</PRICE>
        <YEAR>1973</YEAR>
    </CD>
    <CD>
        <TITLE>Maggie May</TITLE>
        <ARTIST>Rod Stewart</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>Pickwick</COMPANY>
        <PRICE>8.50</PRICE>
        <YEAR>1990</YEAR>
    </CD>
    <CD>
        <TITLE>Tupelo Honey</TITLE>
        <ARTIST>Van Morrison</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>Polydor</COMPANY>
        <PRICE>8.20</PRICE>
        <YEAR>1971</YEAR>
    </CD>
    <CD>
        <TITLE>Pavarotti Gala Concert</TITLE>
        <ARTIST>Luciano Pavarotti</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>DECCA</COMPANY>
        <PRICE>9.90</PRICE>
        <YEAR>1991</YEAR>
    </CD>
    <CD>
        <TITLE>The dock of the bay</TITLE>
        <ARTIST>Otis Redding</ARTIST>
        <COUNTRY>USA</COUNTRY>
        <COMPANY>Atlantic</COMPANY>
        <PRICE>7.90</PRICE>
        <YEAR>1987</YEAR>
    </CD>
    <CD>
        <TITLE>Picture book</TITLE>
        <ARTIST>Simply Red</ARTIST>
        <COUNTRY>EU</COUNTRY>
        <COMPANY>Elektra</COMPANY>
        <PRICE>7.20</PRICE>
        <YEAR>1985</YEAR>
    </CD>
    <CD>
        <TITLE>Red</TITLE>
        <ARTIST>The Communards</ARTIST>
```

```
		<COUNTRY>UK</COUNTRY>
		<COMPANY>London</COMPANY>
		<PRICE>7.80</PRICE>
		<YEAR>1987</YEAR>
	</CD>
	<CD>
		<TITLE>Unchain my heart</TITLE>
		<ARTIST>Joe Cocker</ARTIST>
		<COUNTRY>USA</COUNTRY>
		<COMPANY>EMI</COMPANY>
		<PRICE>8.20</PRICE>
		<YEAR>1987</YEAR>
	</CD>
</CATALOG>
```

2. View the corresponding CSS file

```
CATALOG
{
background-color: #ffffff;
width: 100%;
}
CD
{
display: block;
margin-bottom: 30pt;
margin-left: 0;
}
TITLE
{
color: #FF0000;
font-size: 20pt;
}
ARTIST
{
color: #0000FF;
font-size: 20pt;
}
COUNTRY,PRICE,YEAR,COMPANY
{
display: block;
color: #000000;
margin-left: 20pt;
}
```

3. Display the CD catalog formatted with the CSS file

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/css" href="cd_catalog.css"?>
<CATALOG>
	<CD>
		<TITLE>Empire Burlesque</TITLE>
		<ARTIST>Bob Dylan</ARTIST>
		<COUNTRY>USA</COUNTRY>
		<COMPANY>Columbia</COMPANY>
		<PRICE>10.90</PRICE>
		<YEAR>1985</YEAR>
	</CD>
	<CD>
		<TITLE>Hide your heart</TITLE>
		<ARTIST>Bonnie Tyler</ARTIST>
		<COUNTRY>UK</COUNTRY>
		<COMPANY>CBS Records</COMPANY>
		<PRICE>9.90</PRICE>
		<YEAR>1988</YEAR>
	</CD>
	<CD>
		<TITLE>Greatest Hits</TITLE>
		<ARTIST>Dolly Parton</ARTIST>
		<COUNTRY>USA</COUNTRY>
		<COMPANY>RCA</COMPANY>
		<PRICE>9.90</PRICE>
		<YEAR>1982</YEAR>	</CD>
	<CD>
		<TITLE>Still got the blues</TITLE>
		<ARTIST>Gary Moore</ARTIST>
		<COUNTRY>UK</COUNTRY>
		<COMPANY>Virgin records</COMPANY>
		<PRICE>10.20</PRICE>
		<YEAR>1990</YEAR>
	</CD>
	<CD>
		<TITLE>Eros</TITLE>
		<ARTIST>Eros Ramazzotti</ARTIST>
		<COUNTRY>EU</COUNTRY>
		<COMPANY>BMG</COMPANY>
		<PRICE>9.90</PRICE>
		<YEAR>1997</YEAR>
	</CD>
	<CD>
		<TITLE>One night only</TITLE>
		<ARTIST>Bee Gees</ARTIST>
		<COUNTRY>UK</COUNTRY>
		<COMPANY>Polydor</COMPANY>
		<PRICE>10.90</PRICE>
		<YEAR>1998</YEAR>
	</CD>
	<CD>
		<TITLE>Sylvias Mother</TITLE>
		<ARTIST>Dr.Hook</ARTIST>
		<COUNTRY>UK</COUNTRY>
		<COMPANY>CBS</COMPANY>
		<PRICE>8.10</PRICE>
		<YEAR>1973</YEAR>
	</CD>
	<CD>
		<TITLE>Maggie May</TITLE>
		<ARTIST>Rod Stewart</ARTIST>
		<COUNTRY>UK</COUNTRY>
		<COMPANY>Pickwick</COMPANY>
		<PRICE>8.50</PRICE>
		<YEAR>1990</YEAR>
	</CD>
	<CD>
		<TITLE>Romanza</TITLE>
		<ARTIST>Andrea Bocelli</ARTIST>
		<COUNTRY>EU</COUNTRY>
		<COMPANY>Polydor</COMPANY>
		<PRICE>10.80</PRICE>
		<YEAR>1996</YEAR>
	</CD>
	<CD>
		<TITLE>Big Willie style</TITLE>
		<ARTIST>Will Smith</ARTIST>
		<COUNTRY>USA</COUNTRY>
		<COMPANY>Columbia</COMPANY>
		<PRICE>9.90</PRICE>
		<YEAR>1997</YEAR>
```

```
</CD>
<CD>
        <TITLE>Tupelo Honey</TITLE>
        <ARTIST>Van Morrison</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>Polydor</COMPANY>
        <PRICE>8.20</PRICE>
        <YEAR>1971</YEAR>
</CD>
<CD>
        <TITLE>Soulsville</TITLE>
        <ARTIST>Jorn Hoel</ARTIST>
        <COUNTRY>Norway</COUNTRY>
        <COMPANY>WEA</COMPANY>
        <PRICE>7.90</PRICE>
        <YEAR>1996</YEAR>
</CD>
<CD>
        <TITLE>The very best of</TITLE>
        <ARTIST>Cat Stevens</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>Island</COMPANY>
        <PRICE>8.90</PRICE>
        <YEAR>1990</YEAR>
</CD>


<CD>
        <TITLE>Stop</TITLE>
        <ARTIST>Sam Brown</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>A and M</COMPANY>
        <PRICE>8.90</PRICE>
        <YEAR>1988</YEAR>
</CD>
<CD>
        <TITLE>Bridge of Spies</TITLE>
        <ARTIST>T`Pau</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>Siren</COMPANY>
        <PRICE>7.90</PRICE>
        <YEAR>1987</YEAR>
</CD>
<CD>
        <TITLE>Private Dancer</TITLE>
        <ARTIST>Tina Turner</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>Capitol</COMPANY>
        <PRICE>8.90</PRICE>
        <YEAR>1983</YEAR>
</CD>
<CD>
        <TITLE>Midt om natten</TITLE>
        <ARTIST>Kim Larsen</ARTIST>
        <COUNTRY>EU</COUNTRY>
        <COMPANY>Medley</COMPANY>
        <PRICE>7.80</PRICE>
        <YEAR>1983</YEAR>       </CD>
<CD>
        <TITLE>Pavarotti Gala Concert</TITLE>
        <ARTIST>Luciano Pavarotti</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>DECCA</COMPANY>
        <PRICE>9.90</PRICE>
        <YEAR>1991</YEAR>
</CD>
<CD>
        <TITLE>The dock of the bay</TITLE>
        <ARTIST>Otis Redding</ARTIST>
        <COUNTRY>USA</COUNTRY>
        <COMPANY>Atlantic</COMPANY>
        <PRICE>7.90</PRICE>
        <YEAR>1987</YEAR>
</CD>
<CD>
        <TITLE>Picture book</TITLE>
        <ARTIST>Simply Red</ARTIST>
        <COUNTRY>EU</COUNTRY>
        <COMPANY>Elektra</COMPANY>
        <PRICE>7.20</PRICE>
        <YEAR>1985</YEAR>
</CD>
<CD>
        <TITLE>Red</TITLE>
        <ARTIST>The Communards</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>London</COMPANY>
        <PRICE>7.80</PRICE>
        <YEAR>1987</YEAR>
</CD>
<CD>
        <TITLE>Unchain my heart</TITLE>
        <ARTIST>Joe Cocker</ARTIST>
        <COUNTRY>USA</COUNTRY>
        <COMPANY>EMI</COMPANY>
        <PRICE>8.20</PRICE>
        <YEAR>1987</YEAR>
</CD> </CATALOG>
```

XML and XSLT

1. View an XML food menu

```
<?xml version="1.0" encoding="ISO-8859-1"?>
 <breakfast_menu>
        <food>
                <name>Belgian Waffles</name>
                <price>$5.95</price>
                <description>two of our famous Belgian Waffles
        with plenty of real maple syrup</description>
                <calories>650</calories>
        </food>
        <food>
                <name>Strawberry Belgian Waffles</name>
                <price>$7.95</price>
                <description>light Belgian waffles covered with
strawberries and whipped cream</description>
                <calories>900</calories>
        </food>
        <food>
                <name>Berry-Berry Belgian Waffles</name>
                <price>$8.95</price>
                <description>light Belgian waffles covered with an
assortment of fresh berries and whipped cream</description>
                <calories>900</calories>
        </food>
        <food>
                <name>French Toast</name>
                <price>$4.50</price>
                <description>thick slices made from our
homemade sourdough bread</description>
                <calories>600</calories>
```

```
        </food>
</breakfast_menu>
```

2. View the corresponding XSLT stylesheet

```
<?xml version="1.0" encoding="ISO-8859-1"?>

 <html xsl:version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns="http://www.w3.org/1999/xhtml">
  <body style="font-family:Arial;font-size:12pt;background-
color:#EEEEEE">
    <xsl:for-each select="breakfast_menu/food">
      <div style="background-color:teal;color:white;padding:4px">
        <span style="font-weight:bold"><xsl:value-of
select="name"/></span>
        - <xsl:value-of select="price"/>
      </div>
      <div style="margin-left:20px;margin-bottom:1em;font-
size:10pt">
        <xsl:value-of select="description"/>
        <span style="font-style:italic">
          <xsl:value-of select="calories"/> (calories per serving)
        </span>
      </div>
    </xsl:for-each>
  </body>
</html>
```

3. Display the food menu styled with the XSLT stylesheet

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="simple.xsl" ?>
 <breakfast_menu>
        <food>
                <name>Belgian Waffles</name>
                <price>$5.95</price>
                <description>two of our famous Belgian Waffles
with plenty of real maple syrup</description>
                <calories>650</calories>
        </food>
        <food>
                <name>Strawberry Belgian Waffles</name>
                <price>$7.95</price>
                <description>light Belgian waffles covered with
strawberries and whipped cream</description>
                <calories>900</calories>
        </food>
        <food>
                <name>Berry-Berry Belgian Waffles</name>
                <price>$8.95</price>
                <description>light Belgian waffles covered with an
assortment of fresh berries and whipped cream</description>
                <calories>900</calories> </food>
        <food>
                <name>French Toast</name>
                <price>$4.50</price>
                <description>thick slices made from our
homemade sourdough bread</description>
                <calories>600</calories>
        </food>
        <food>
                <name>Homestyle Breakfast</name>
                <price>$6.95</price>
                <description>two eggs, bacon or sausage, toast,
and our ever-popular hash browns</description>
                <calories>950</calories>
        </food>
</breakfast_menu>
```

Parsing XML and the XML DOM

1. View a simple XML file (note.xml)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
 <note>
        <to>Tove</to>
        <from>Jani</from>
        <heading>Reminder</heading>
        <body>Don't forget me this weekend!</body>
</note>
```

2. Parse the XML file - Crossbrowser example

```
<html>
<body>
<h1>W3Schools Internal Note</h1>
<p><b>To:</b> <span id="to"></span><br />
<b>From:</b> <span id="from"></span><br />
<b>Message:</b> <span id="message"></span>
<script type="text/javascript">
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
else
  {// code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.open("GET","note.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;

document.getElementById("to").innerHTML=
xmlDoc.getElementsByTagName("to")[0].childNodes[0].nodeValue;
document.getElementById("from").innerHTML=
xmlDoc.getElementsByTagName("from")[0].childNodes[0].nodeVal
ue;
document.getElementById("message").innerHTML=
xmlDoc.getElementsByTagName("body")[0].childNodes[0].nodeVal
ue;
</script>

</body>
</html>
```

3. Parse an XML string - Crossbrowser example

```
<html>
<body>
<h1>W3Schools Internal Note</h1>
```

```
<p><b>To:</b> <span id="to"></span><br />
<b>From:</b> <span id="from"></span><br />
<b>Message:</b> <span id="message"></span></p>

<script>
txt="<note>";
txt=txt+"<to>Tove</to>";
txt=txt+"<from>Jani</from>";
txt=txt+"<heading>Reminder</heading>";
txt=txt+"<body>Don't forget me this weekend!</body>";
txt=txt+"</note>";
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);
  }
document.getElementById("to").innerHTML=xmlDoc.getElementsB
yTagName("to")[0].childNodes[0].nodeValue;
document.getElementById("from").innerHTML=xmlDoc.getElemen
tsByTagName("from")[0].childNodes[0].nodeValue;
document.getElementById("message").innerHTML=xmlDoc.getEle
mentsByTagName("body")[0].childNodes[0].nodeValue;
</script>
</body>
</html>
```

XML to HTML

1. View an XML CD catalog

```
<?xml version="1.0" encoding="ISO-8859-1"?>
 <CATALOG>
        <CD>
                <TITLE>Empire Burlesque</TITLE>
                <ARTIST>Bob Dylan</ARTIST>
                <COUNTRY>USA</COUNTRY>
                <COMPANY>Columbia</COMPANY>
                <PRICE>10.90</PRICE>
                <YEAR>1985</YEAR>
        </CD>
        <CD>
                <TITLE>Hide your heart</TITLE>
                <ARTIST>Bonnie Tyler</ARTIST>
                <COUNTRY>UK</COUNTRY>
                <COMPANY>CBS Records</COMPANY>
                <PRICE>9.90</PRICE>
                <YEAR>1988</YEAR>
        </CD>
        <CD>
                <TITLE>Greatest Hits</TITLE>
                <ARTIST>Dolly Parton</ARTIST>
                <COUNTRY>USA</COUNTRY>
                <COMPANY>RCA</COMPANY>
                <PRICE>9.90</PRICE>
                <YEAR>1982</YEAR>
        </CD>
        <CD>
                <TITLE>Still got the blues</TITLE>
                <ARTIST>Gary Moore</ARTIST>
                <COUNTRY>UK</COUNTRY>
                <COMPANY>Virgin records</COMPANY>
                <PRICE>10.20</PRICE>
                <YEAR>1990</YEAR>
        </CD>
        <CD>
                <TITLE>Eros</TITLE>
                <ARTIST>Eros Ramazzotti</ARTIST>
                <COUNTRY>EU</COUNTRY>
                <COMPANY>BMG</COMPANY>
                <PRICE>9.90</PRICE>
                <YEAR>1997</YEAR>
        </CD>
        <CD>
                <TITLE>One night only</TITLE>
                <ARTIST>Bee Gees</ARTIST>
                <COUNTRY>UK</COUNTRY>
                <COMPANY>Polydor</COMPANY>
                <PRICE>10.90</PRICE>
                <YEAR>1998</YEAR>
        </CD>
        <CD>
                <TITLE>Sylvias Mother</TITLE>
                <ARTIST>Dr.Hook</ARTIST>
                <COUNTRY>UK</COUNTRY>
                <COMPANY>CBS</COMPANY>
                <PRICE>8.10</PRICE>
                <YEAR>1973</YEAR>
        </CD>
        <CD>
                <TITLE>Maggie May</TITLE>
                <ARTIST>Rod Stewart</ARTIST>
                <COUNTRY>UK</COUNTRY>
                <COMPANY>Pickwick</COMPANY>
                <PRICE>8.50</PRICE>
                <YEAR>1990</YEAR>
        </CD>
        <CD>
                <TITLE>Romanza</TITLE>
                <ARTIST>Andrea Bocelli</ARTIST>
                <COUNTRY>EU</COUNTRY>
                <COMPANY>Polydor</COMPANY>
                <PRICE>10.80</PRICE>
                <YEAR>1996</YEAR>
        </CD>
        <CD>
                <TITLE>When a man loves a woman</TITLE>
                <ARTIST>Percy Sledge</ARTIST>
                <COUNTRY>USA</COUNTRY>
                <COMPANY>Atlantic</COMPANY>
                <PRICE>8.70</PRICE>
```

```
        <YEAR>1987</YEAR>
</CD>
<CD>
        <TITLE>Black angel</TITLE>
        <ARTIST>Savage Rose</ARTIST>
        <COUNTRY>EU</COUNTRY>
        <COMPANY>Mega</COMPANY>
        <PRICE>10.90</PRICE>
        <YEAR>1995</YEAR>
</CD>
<CD>
        <TITLE>1999 Grammy Nominees</TITLE>
        <ARTIST>Many</ARTIST>
        <COUNTRY>USA</COUNTRY>
        <COMPANY>Grammy</COMPANY>
        <PRICE>10.20</PRICE>
        <YEAR>1999</YEAR>
</CD>
<CD>
        <TITLE>For the good times</TITLE>
        <ARTIST>Kenny Rogers</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>Mucik Master</COMPANY>
        <PRICE>8.70</PRICE>
        <YEAR>1995</YEAR>
</CD>
<CD>
        <TITLE>Big Willie style</TITLE>
        <ARTIST>Will Smith</ARTIST>
        <COUNTRY>USA</COUNTRY>
        <COMPANY>Columbia</COMPANY>
        <PRICE>9.90</PRICE>
        <YEAR>1997</YEAR>
</CD>
<CD>
        <TITLE>Tupelo Honey</TITLE>
        <ARTIST>Van Morrison</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>Polydor</COMPANY>
        <PRICE>8.20</PRICE>
        <YEAR>1971</YEAR>
</CD>
<CD>
        <TITLE>Soulsville</TITLE>
        <ARTIST>Jorn Hoel</ARTIST>
        <COUNTRY>Norway</COUNTRY>
        <COMPANY>WEA</COMPANY>
        <PRICE>7.90</PRICE>
        <YEAR>1996</YEAR>
</CD>
<CD>
        <TITLE>The very best of</TITLE>
        <ARTIST>Cat Stevens</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>Island</COMPANY>
        <PRICE>8.90</PRICE>
        <YEAR>1990</YEAR>
</CD>
<CD>
        <TITLE>The dock of the bay</TITLE>
        <ARTIST>Otis Redding</ARTIST>
        <COUNTRY>USA</COUNTRY>
        <COMPANY>Atlantic</COMPANY>
        <PRICE>7.90</PRICE>
        <YEAR>1987</YEAR>
</CD>
<CD>
        <TITLE>Picture book</TITLE>
        <ARTIST>Simply Red</ARTIST>
        <COUNTRY>EU</COUNTRY>
        <COMPANY>Elektra</COMPANY>
        <PRICE>7.20</PRICE>
        <YEAR>1985</YEAR>
</CD>
<CD>
        <TITLE>Red</TITLE>
        <ARTIST>The Communards</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>London</COMPANY>
        <PRICE>7.80</PRICE>
        <YEAR>1987</YEAR>
</CD>
<CD>
        <TITLE>Unchain my heart</TITLE>
        <ARTIST>Joe Cocker</ARTIST>
        <COUNTRY>USA</COUNTRY>
        <COMPANY>EMI</COMPANY>
        <PRICE>8.20</PRICE>
        <YEAR>1987</YEAR>
</CD> </CATALOG>
```

2. Display XML data in an HTML table

```
<html>
<body>
<script type="text/javascript">
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }        else
  {// code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.open("GET","cd_catalog.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;
document.write("<table border='1'>");
var x=xmlDoc.getElementsByTagName("CD");
for (i=0;i<x.length;i++)
  {
  document.write("<tr><td>");

document.write(x[i].getElementsByTagName("ARTIST")[0].childN
odes[0].nodeValue);
  document.write("</td><td>");

document.write(x[i].getElementsByTagName("TITLE")[0].childNod
es[0].nodeValue);
  document.write("</td></tr>");
  }
document.write("</table>");
```

```
</script>
</body>
</html>
```

Output:

Bob Dylan	Empire Burlesque
Bonnie Tyler	Hide your heart
Dolly Parton	Greatest Hits
Gary Moore	Still got the blues
Eros Ramazzotti	Eros
Bee Gees	One night only
Dr.Hook	Sylvias Mother
Rod Stewart	Maggie May
Andrea Bocelli	Romanza
Percy Sledge	When a man loves a woman
Savage Rose	Black angel
Many	1999 Grammy Nominees
Kenny Rogers	For the good times
Will Smith	Big Willie style
Van Morrison	Tupelo Honey
Jorn Hoel	Soulsville
Cat Stevens	The very best of
Sam Brown	Stop
T'Pau	Bridge of Spies
Tina Turner	Private Dancer
Kim Larsen	Midt om natten
Luciano Pavarotti	Pavarotti Gala Concert
Otis Redding	The dock of the bay
Simply Red	Picture book
The Communards	Red
Joe Cocker	Unchain my heart

XML Applications

View an XML CD catalog

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<CATALOG>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
  <CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bonnie Tyler</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS Records</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1988</YEAR>
  </CD>
  <CD>
    <TITLE>Maggie May</TITLE>
    <ARTIST>Rod Stewart</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>Pickwick</COMPANY>
    <PRICE>8.50</PRICE>
    <YEAR>1990</YEAR>
  </CD>
  <CD>
    <TITLE>Romanza</TITLE>
    <ARTIST>Andrea Bocelli</ARTIST>
    <COUNTRY>EU</COUNTRY>
    <COMPANY>Polydor</COMPANY>
    <PRICE>10.80</PRICE>
    <YEAR>1996</YEAR>
  </CD>
  <CD>
    <TITLE>When a man loves a woman</TITLE>
    <ARTIST>Percy Sledge</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Atlantic</COMPANY>
    <PRICE>8.70</PRICE>
    <YEAR>1987</YEAR>
  </CD>
  <CD>
    <TITLE>Black angel</TITLE>
    <ARTIST>Savage Rose</ARTIST>
    <COUNTRY>EU</COUNTRY>
    <COMPANY>Mega</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1995</YEAR>
  </CD>
  <CD>
    <TITLE>1999 Grammy Nominees</TITLE>
    <ARTIST>Many</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Grammy</COMPANY>
    <PRICE>10.20</PRICE>
    <YEAR>1999</YEAR>
  </CD>
  <CD>
    <TITLE>For the good times</TITLE>
    <ARTIST>Kenny Rogers</ARTIST>
    <COUNTRY>UK</COUNTRY>
```

```
        <COMPANY>Mucik Master</COMPANY>
        <PRICE>8.70</PRICE>
        <YEAR>1995</YEAR>
</CD>
<CD>
        <TITLE>Big Willie style</TITLE>
        <ARTIST>Will Smith</ARTIST>
        <COUNTRY>USA</COUNTRY>
        <COMPANY>Columbia</COMPANY>
        <PRICE>9.90</PRICE>
        <YEAR>1997</YEAR>
</CD>
<CD>
        <TITLE>Tupelo Honey</TITLE>
        <ARTIST>Van Morrison</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>Polydor</COMPANY>
        <PRICE>8.20</PRICE>
        <YEAR>1971</YEAR>
</CD>
<CD>
        <TITLE>Soulsville</TITLE>
        <ARTIST>Jorn Hoel</ARTIST>
        <COUNTRY>Norway</COUNTRY>
        <COMPANY>WEA</COMPANY>
        <PRICE>7.90</PRICE>
        <YEAR>1996</YEAR>
</CD>
<CD>
        <TITLE>The very best of</TITLE>
        <ARTIST>Cat Stevens</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>Island</COMPANY>
        <PRICE>8.90</PRICE>
        <YEAR>1990</YEAR>
</CD>
<CD>
        <TITLE>Stop</TITLE>
        <ARTIST>Sam Brown</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>A and M</COMPANY>
        <PRICE>8.90</PRICE>
        <YEAR>1988</YEAR>
</CD>
<CD>
        <TITLE>Bridge of Spies</TITLE>
        <ARTIST>T'Pau</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>Siren</COMPANY>
        <PRICE>7.90</PRICE>
        <YEAR>1987</YEAR>
</CD>
<CD>
        <TITLE>Private Dancer</TITLE>
        <ARTIST>Tina Turner</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>Capitol</COMPANY>
        <PRICE>8.90</PRICE>
        <YEAR>1983</YEAR>
</CD>
<CD>
        <TITLE>Midt om natten</TITLE>
        <ARTIST>Kim Larsen</ARTIST>
        <COUNTRY>EU</COUNTRY>
        <COMPANY>Medley</COMPANY>
        <PRICE>7.80</PRICE>
        <YEAR>1983</YEAR>
</CD>
<CD>
        <TITLE>Red</TITLE>
        <ARTIST>The Communards</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>London</COMPANY>
        <PRICE>7.80</PRICE>
        <YEAR>1987</YEAR>
</CD>
<CD>
        <TITLE>Unchain my heart</TITLE>
        <ARTIST>Joe Cocker</ARTIST>
        <COUNTRY>USA</COUNTRY>
        <COMPANY>EMI</COMPANY>
        <PRICE>8.20</PRICE>
        <YEAR>1987</YEAR>
</CD>
</CATALOG>
```

Show XML data inside an HTML div element

```
<html>
<head>
<script type="text/javascript">
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
else
{// code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.open("GET","cd_catalog.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;
x=xmlDoc.getElementsByTagName("CD");
i=0;
function displayCD()
{
artist=(x[i].getElementsByTagName("ARTIST")[0].childNodes[0].n
odeValue);
title=(x[i].getElementsByTagName("TITLE")[0].childNodes[0].node
Value);
year=(x[i].getElementsByTagName("YEAR")[0].childNodes[0].node
Value);
txt="Artist: " + artist + "<br />Title: " + title + "<br />Year: "+ year;
document.getElementById("showCD").innerHTML=txt;
}
</script>
</head>
<body onload="displayCD()">
<div id='showCD'></div>
</body>
</html>
```

Output:
Artist: Bob Dylan
Title: Empire Burlesque
Year: 1985

Navigate through XML nodes

```
<html>
<head>

<script type="text/javascript">
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
else
  {// code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.open("GET","cd_catalog.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;

x=xmlDoc.getElementsByTagName("CD");
i=0;

function displayCD()
{
artist=(x[i].getElementsByTagName("ARTIST")[0].childNodes[0].n
odeValue);
title=(x[i].getElementsByTagName("TITLE")[0].childNodes[0].node
Value);
year=(x[i].getElementsByTagName("YEAR")[0].childNodes[0].node
Value);
txt="Artist: " + artist + "<br />Title: " + title + "<br />Year: "+ year;
document.getElementById("showCD").innerHTML=txt;
}

function next()
{
if (i<x.length-1)
  {
  i++;
  displayCD();
  }
}

function previous()
{
if (i>0)
  {
  i--;
  displayCD();
  }
}
</script>
</head>
<body onload="displayCD()">

<div id='showCD'></div><br />
<input type="button" onclick="previous()" value="<<" />
<input type="button" onclick="next()" value=">>" />

</body>
</html>
```

Output:
Artist: Bob Dylan
Title: Empire Burlesque
Year: 1985

A simple CD catalog application

```
<html>
<head>
<script type="text/javascript">
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
else
  {// code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.open("GET","cd_catalog.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;
x=xmlDoc.getElementsByTagName("CD");

function displayCDInfo(i)
{
artist=(x[i].getElementsByTagName("ARTIST")[0].childNodes[0].n
odeValue);
title=(x[i].getElementsByTagName("TITLE")[0].childNodes[0].node
Value);
year=(x[i].getElementsByTagName("YEAR")[0].childNodes[0].node
Value);
country=(x[i].getElementsByTagName("COUNTRY")[0].childNode
s[0].nodeValue);
company=(x[i].getElementsByTagName("COMPANY")[0].childNo
des[0].nodeValue);
price=(x[i].getElementsByTagName("PRICE")[0].childNodes[0].nod
eValue);
txt="Artist: "+artist+"<br />Title: "+title+"<br />Year: "+year+"<br
/>Country: "+country+"<br />Company: "+company+"<br />Price:
"+price ;
document.getElementById("showCD").innerHTML=txt;
}
</script>
</head>

<body>
<div id='showCD'>Click on a CD to display album
information.</div><br />
<script type="text/javascript">
document.write("<table border='1'>");
for (var i=0;i<x.length;i++)
  {
  document.write("<tr onclick='displayCDInfo(" + i + ")'>");
  document.write("<td>");

document.write(x[i].getElementsByTagName("ARTIST")[0].childN
odes[0].nodeValue);
  document.write("</td><td>");
```

```
document.write(x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue);
  document.write("</td></tr>");
  }
document.write("</table>");
</script>

</body>
</html>
```

Output:
Click on a CD to display album information.

Bob Dylan	Empire Burlesque
Bonnie Tyler	Hide your heart
Dolly Parton	Greatest Hits
Gary Moore	Still got the blues
Eros Ramazzotti	Eros
Bee Gees	One night only
Dr.Hook	Sylvias Mother
Rod Stewart	Maggie May
Andrea Bocelli	Romanza
Percy Sledge	When a man loves a woman
Savage Rose	Black angel
Many	1999 Grammy Nominees
Kenny Rogers	For the good times
Will Smith	Big Willie style
Van Morrison	Tupelo Honey
Jorn Hoel	Soulsville
Cat Stevens	The very best of
Sam Brown	Stop
T'Pau	Bridge of Spies
Tina Turner	Private Dancer
Kim Larsen	Midt om natten
Luciano Pavarotti	Pavarotti Gala Concert
Otis Redding	The dock of the bay
Simply Red	Picture book
The Communards	Red
Joe Cocker	Unchain my heart

XML Output From a Server

1. See how ASP can return XML

```
<?xml version='1.0' encoding='ISO-8859-1'?><note><from>Jani</from><to>Tove</to><message>Remember me this weekend</message></note>
```

2. See how PHP can return XML

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<note>
<from>Jani</from>
<to>Tove</to>
<message>Remember me this weekend</message>
</note>
```

3. View XML output from a database

```
<?xml version='1.0' encoding='ISO-8859-1'?><guestbook><guest><fname>Terje</fname><lname>Beck</lname></guest><guest><fname>Jan</fname><lname>Refsnes</lname></guest><guest><fname>Torleif</fname><lname>Rasmussen</lname></guest><guest><fname>anton</fname><lname>chek</lname></guest><guest><fname>stale</fname><lname>refsnes</lname></guest><guest><fname>hari</fname><lname>prawin</lname></guest><guest><fname>Hege</fname><lname>Refsnes</lname></guest></guestbook>
```

XML DOM Advanced

1. Get the value of an XML element

```
<html>
<body>
<script type="text/javascript">
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
else
  {// code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.open("GET","books.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;
txt=xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;
document.write(txt);
</script>
</body>
</html>
```

Output:
Everyday Italian

2. Get the value of an XML attribute

```
<html>
<body>
<script type="text/javascript">
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
else
  {// code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.open("GET","books.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;

txt=xmlDoc.getElementsByTagName("title")[0].getAttribute("lang");
document.write(txt);
</script>
```

```
</body>
</html>
```
Output: en

3. Change the value of an XML element

```
<html>
<body>
<script type="text/javascript">
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
else
  {// code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.open("GET","books.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;

x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];
x.nodeValue="Easy Cooking";
x=xmlDoc.getElementsByTagName("title")[0].childNodes[0];
txt=x.nodeValue;
document.write(txt);
</script>
</body>
</html>
```
Output: Easy Cooking

4. Add a new attribute to an XML element

```
<html>
<body>
<script type="text/javascript">
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
else
  {// code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.open("GET","books.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;

x=xmlDoc.getElementsByTagName("book");
for(i=0;i<x.length;i++)
  {
  x[i].setAttribute("edition","first");
  }

//Output all attribute values
for (i=0;i<x.length;i++)
{
document.write("Category: " + x[i].getAttribute('category') + "<br
/>");
document.write("Edition: " + x[i].getAttribute('edition') + "<br />");
}
</script>
</body>
</html>
```
Output:
```
Category: COOKING
Edition: first
Category: CHILDREN
Edition: first
Category: WEB
Edition: first
Category: WEB
Edition: first
```

5. Create a new XML element

```
<html>
<body>
<script type="text/javascript">
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
else
  {// code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.open("GET","books.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;

newel=xmlDoc.createElement("edition");
newtext=xmlDoc.createTextNode("First");
newel.appendChild(newtext);
x=xmlDoc.getElementsByTagName("book");
x[0].appendChild(newel);

for (i=0;i<x[0].childNodes.length;i++)
{
if (x[0].childNodes[i].nodeType==1)
  {
  document.write(x[0].childNodes[i].nodeName);
  document.write(": ");
  document.write(x[0].childNodes[i].childNodes[0].nodeValue);
  document.write("<br />");
  }
}
</script>
</body>
</html>
```
Output:
```
title: Everyday Italian
author: Giada De Laurentiis
year: 2005
price: 30.00
edition: Firs
```

6. Remove an XML element

```
<html>
<body>
<script type="text/javascript">
if (window.XMLHttpRequest)
  {// code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
```

```
else
  {// code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.open("GET","books.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;

var x=xmlDoc.getElementsByTagName("book")[0];
document.write("Child nodes before removal: ");
document.write(x.childNodes.length);

x.removeChild(x.childNodes[0]);

document.write("<br />Child nodes after removal: ");
document.write(x.childNodes.length);

</script>
</body>
</html>
```

Output:

Child nodes before removal: 9

Child nodes after removal: 8