

Introduction to XML

Objectives

Welcome to Introduction to XML. The purpose of this tutorial is to expose you to some of the basic ideas behind XML, and to show you how you should expect to see XML used in the future.

Before we begin, it's important for you to understand that XML is similar on the HyperText Markup Language – that is, HTML. This tutorial assumes a basic understanding of HTML. If you don't know what HTML is, or if you need a refresher course, check out our [Introduction to HTML](#) tutorial.

What will this tutorial teach you?

It's also important to understand that this tutorial will not teach you how to “do” anything with XML – that is, you will not learn how to develop XML applications or documents. Instead, you will learn why XML is important, and you will see some examples of existing XML documents and applications on the World Wide Web. If you're interested in actually working with XML, the last section of this tutorial will highlight some tools and resources you can use to get started.

So why should you know about XML?

Simply put, XML is likely to become a much more common tool for sharing data, developing documents and enhancing applications on the World Wide Web. While HTML probably won't disappear from the Internet, newer sites and systems may be XML-based. To effectively analyze and develop Web applications, some basic familiarity with XML will be very helpful.

What should you I know RIGHT NOW?

For now, it's important that you code your HTML correctly, so that your Web documents will display correctly in the future when XML is used more frequently.

The global transition from HTML to XML is having a significant impact on Web document coding and markup. The standards for developing HTML-based documents are getting "stricter" – that is, HTML-based documents and applications are being written in more specific ways.

To prepare for this transition, there is a new markup standard known as XHTML. XHTML is almost identical to HTML 4.01, except the syntax and organization of code is more precise. For example:

- 1) XHTML requires that tag names must be lowercase (e.g. <body> instead of <BODY>)
- 2) XHTML requires that code must be "well-formed" - that is, tags must be properly nested. You must provide the appropriate number of open and closed parentheses in the correct order.
- 3) XHTML requires that all tags must be closed - that is, each non-empty tag must have a closing tag.

Many Web page editors - including Dreamweaver - automatically apply "strict" HTML formatting. However, you should always check your code to make sure it is correctly written. One way to make sure that your code is correctly written is to use the W3 Markup Validation Service available at:

<http://validator.w3.org/>

This utility will automatically review your HTML and notify you if your code is not correctly written. Your code doesn't necessarily have to be written in XHTML to be correct - correct HTML documents will do just as well.

If you'd like more information about XHTML requirements, consult the section of this tutorial entitled "additional resources".

Just What Is XML?

XML stands for the Extensible Markup Language. It is a markup language for documents, much like HTML.

However, XML differs from HTML because it is extensible, meaning that users can define their own tags or other structural aspects of the document.

XML allows users to specify the type or form of a document, the elements of a document, the values of those elements, and the relationship of the document and elements to other documents and elements.

IMPORTANT: XML does not display data in web browsers!

Although XML does share many similarities with HTML, a very important difference between these languages is that XML was not designed to describe page layout or functionality in the same way that HTML was. One way to understand this is to remember that XML does not create or modify the appearance of a web document – instead, it can be used to store data within a document.

XML is a tool to develop, store and share data. XML can communicate structured information to other users. In other words, if a group of users agree to implement the same kinds of tags to describe a certain kind of information, XML applications can assist these users in communicating their information in an more robust and efficient manner. XML can make it easier to exchange information between cooperating entities.

Where did XML come from?

In the past, when someone wanted to design a document for use on the World Wide Web, they were generally limited to using basic HTML tags. Occasionally, a new feature would be provided by browser designers – for instance, Netscape offered a <BLINK> element that allowed users to produce flashing text.

Unfortunately, such improvements were limited to specific browsers – that is “blinking” text designed for the Netscape browser would not display properly on Microsoft Internet Explorer browsers. The feature was not adequately “standardized”, and people who used the new feature could only reach a limited audience.

The W3 (also known as the World Wide Web consortium) is the premier nonprofit organization that determines and promotes standards for interoperable technologies such as HTML and XML. When Web developers wanted to introduce a new HTML feature to a large audience, they had to submit a request to the W3.

Unfortunately, submitting a recommendation for change to the W3 was a long and tedious process, often involving the approval of many organizations. Getting the public-at-large to accept these changes was even more difficult and time-consuming, since it involved the commitment and cooperation of large corporations such as Microsoft and Netscape.

With XML, Web developers can introduce new tags, elements, and features to a Web document almost immediately. Currently, major browsers such as Microsoft Internet Explorer and Netscape can support XML. A good way of thinking of XML is to remember that the features of XML can be extended or “added to” - that is, the user can specify new features and add them to a document.

What does XML look like?

Let's look at some examples of XML text:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<letter>
<to>Buddy</to>
<from>Fran</from>
<greeting>Howdy!</greeting>
<body>Don't forget me this weekend!</body>
</letter>
```

The first line in the document - the XML declaration - defines the XML version and the character encoding used in the document. In this case the document conforms to the 1.0 specification of XML and uses the ISO-8859-1 (Latin-1/West European) character set. Note that there is no closing tag for this declaration. Unlike most elements of an XML document, the declaration does not require a closing tag.

The next line describes the root element of the document (as if it were saying, "this document is a letter"). Note that this element has a closing tag.

The next 4 lines describe 4 child elements of the root (to, from, greeting, and body). Note that all of these elements also have closing tags.

How Can XML be used?

In the previous example, we saw five user-defined elements: “letter”, “to”, “from”, “greeting”, and “body”. Unlike HTML, these values will not display in a web browser if we load our XML document. If you want to display XML values, XML must be “embedded” in an HTML document. For instance, XML can be embedded in a “data island” which looks like this:

```
<xml id="letter">
<letter>
<to>Buddy</to>
<from>Fran</from>
<greeting>Howdy!</greeting>
<body>Don't forget me this weekend!</body>
</letter>
</xml>
```

This XML data can then be “bound” to HTML elements to display in a web browser by building a table in HTML and populating it with XML elements, like this:

```
<html>
<body>

<xml id="letter">
<letter>
<to>Buddy</to>
<from>Fran</from>
<greeting>Howdy!</greeting>
<body>Don't forget me this weekend!</body>
</letter>
</xml>

<table border="1" datasrc="#letter">
<tr>
<td><span datafld="to"></span></td>
<td><span datafld="from"></span></td>
</tr>
</table>

</body>
</html>
```

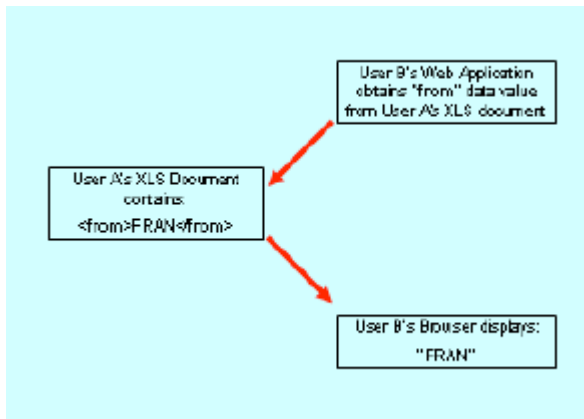
When you display this HTML document in a browser, you will see this:



You may ask, “So what good is that? Why not simply use HTML to display these values?”

This is where the advantages of XML really become apparent. If two user’s applications use the same elements, they can exchange information between each other.

For instance, if another user's application also has an element labeled "letter" with a child element labeled "from", they could also display the value "Howdy!" by referring to the internet address of User A's XML document:



In this sense, XML is a tool for exchanging data. XML documents with the same elements can "pass" data to each other, and multiple users can form a sort of "data network" that can be used to store, update, and display information from multiple sites and addresses.

Example: RSS

I'm going to use a real-life example to show what I mean.

The lightweight XML application RSS is especially popular these days. RSS is a web-based tool for sharing news headlines and other Web content.

RSS users take advantage of the same user-defined elements. By using the same XML elements, RSS users can build software and website applications that consistently update and display "headlines".

Here's what the basic RSS elements would look like for a news aggregator site:

```
<item>
<title>Item Headline</title>
<link>Link to story</link>
<description>Description of story</description>
</item>
```

By making XML RSS documents available online and syndicating them through RSS aggregator sites, a news organization can notify its readers every time a new headline is posted to the news site. Also, individual users can install special software on their computers that allows them to view other users' RSS XML documents.

Here's an example of an [aggregator site](#).

Here's an example of an [RSS software application](#).

Both the news aggregator sites and the software use the same RSS data elements to share information. These applications "refresh" periodically to locate new headlines. With individual software applications, users can specify which RSS news services they would like to subscribe to.

RSS is just one of the many applications that have been developed using XML. XML markup is also heavily used in B2B data transfers, and will continue to play a significant role in online business transactions.

Additional resources

Here's some of the introductory resources I used to prepare this presentation:

<http://www.xml.com/pub/a/98/10/guide1.html>

<http://www.ariadne.ac.uk/issue15/what-is/#ref-09>

<http://www.w3schools.com/xml/default.asp>

Also, there is an open-source XML tool available called Cooktop. It is available from:

<http://www.xmlcooktop.com/>

There are also many proprietary software packages available for purchase.