

1.9 Introduction to **XML** (Text Book Chapter 18)

➤ 1.0 Client Side Application Development

HTML, XHTML, CSS, Document Object Model(DOM), JavaScript , DHTML, Ajax, jQuery & XML

1.9 XML (EXtensible Markup Language)

1.9.1 Introduction

The essence of XML is in its name: Extensible Markup Language.

Extensible

XML is extensible. It lets you define your own tags, the order in which they occur, and how they should be processed or displayed. Another way to think about extensibility is to consider that XML allows all of us to extend our notion of what a document is: it can be a file that lives on a file server, or it can be a transient piece of data that flows between two computer systems (as in the case of Web Services).

Markup

The most recognizable feature of XML is its tags, or elements (to be more accurate). In fact, the elements you'll create in XML will be very similar to the elements you've already been creating in your HTML documents. However, XML allows you to define your own set of tags.

Language

XML is a language that's very similar to HTML. It's much more flexible than HTML because it allows you to create your own custom tags. However, it's important to realize that XML is not just a language. XML is a meta-language: a language that allows us to create or define other languages. For example, with XML we can create other languages, such as RSS, MathML (a mathematical markup language), and even tools like XSLT. More on this later.

So in short, 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

➤ 1.0 Client Side Application Development

HTML, XHTML, CSS, Document Object Model(DOM), JavaScript , DHTML, Ajax, jQuery & XML

1.9.2 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 independent of software and hardware** whereas HTML was designed to layout and display data.

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

1.9.3 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 to pickup the material</body>
</note>
```

Example 1.9-1

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.

1.9.3 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.

➤ 1.0 Client Side Application Development

HTML, XHTML, CSS, Document Object Model(DOM), JavaScript , DHTML, Ajax, jQuery & XML

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.

1.9.4 How Can XML be Used

XML is used in many aspects of web development, often to simplify data storage and sharing. Here are some good points to remember.

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.

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.

Simplifies Data Sharing:

In the real world, computer systems and databases contain data in incompatible formats. Since XML data is stored in plain text format, it provides a software- and hardware-independent way of storing and sharing data between computer systems. 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.

➤ 1.0 Client Side Application Development

HTML, XHTML, CSS, Document Object Model(DOM), JavaScript , DHTML, Ajax, jQuery & XML

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 is Used to Create New Internet Languages:

A lot of new Internet languages are created with XML.

Here are some examples:

- **XHTML**
- **WSDL** [Web Services Description Language] for describing available web services
- **WAP** [Wireless Application Protocol] and **WML** [Wireless Markup Language] as markup languages for handheld devices
- **RSS** [Really Simple Syndication] languages for news feeds
- **RDF** [Resource Description Framework] and **OWL** [Web Ontology Language] for describing resources and ontology
- **SMIL** [Synchronized Multimedia Integration Language] for describing multimedia for the web.

➤ 1.0 Client Side Application Development

HTML, XHTML, CSS, Document Object Model(DOM), JavaScript , DHTML, Ajax, jQuery & XML

1.9.5 XML Documents Form a Tree Structure

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

Example 1.9-2

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)

See a full example next page.

➤ 1.0 Client Side Application Development

HTML, XHTML, CSS, Document Object Model(DOM), JavaScript , DHTML, Ajax, jQuery & XML

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

Example 1.9-3

In the above example, 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 rest can be put into a tree like this.

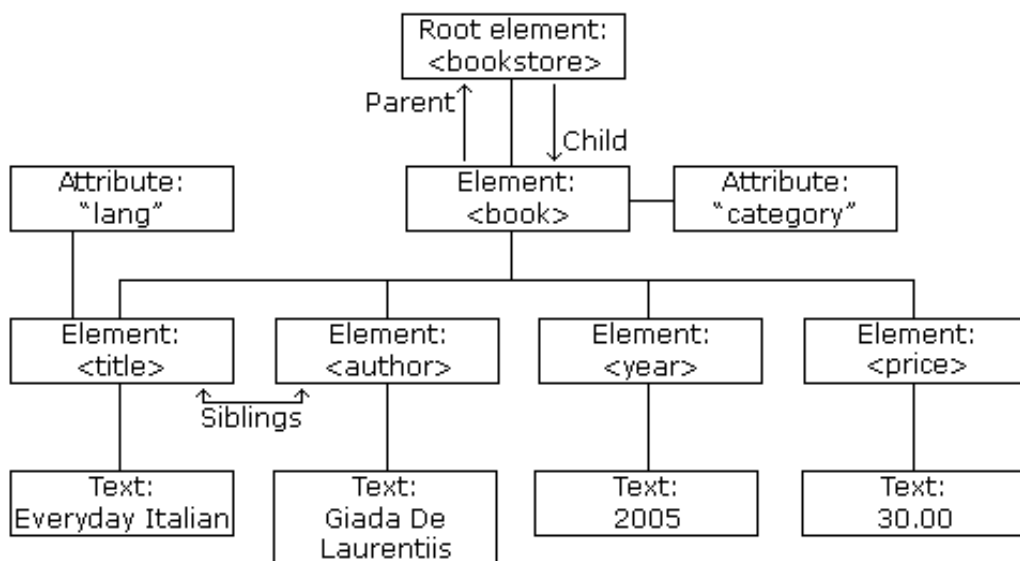


Figure 1.9-1

➤ 1.0 Client Side Application Development

HTML, XHTML, CSS, Document Object Model(DOM), JavaScript , DHTML, Ajax, jQuery & XML

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>**.

1.9.6 XML Syntax

The syntax rules of XML are very simple and logical. The rules are easy to learn, and easy to use.

1.9.6.1 XML Tags are Case Sensitive

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.

1.9.6.2 All XML Elements Must Have a Closing Tag

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>**

1.9.6.3 XML Elements Must be Properly Nested

In HTML, you might see improperly nested elements:

<i>This text is bold and italic**</i>**

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

<i>This text is bold and italic**</i>**

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

➤ 1.0 Client Side Application Development

HTML, XHTML, CSS, Document Object Model(DOM), JavaScript , DHTML, Ajax, jQuery & XML

1.9.6.4 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>
```

1.9.6.5 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=07/29/2014>
  <to>Tove</to>
  <from>Jani</from>
</note>
```

Example 1.9-4a

```
<note date="07/29/2014">
  <to>Tove</to>
  <from>Jani</from>
</note>
```

Example 1.9-4b

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

1.9.6.6 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>
```

➤ 1.0 Client Side Application Development

HTML, XHTML, CSS, Document Object Model(DOM), JavaScript , DHTML, Ajax, jQuery & XML

There are 5 predefined entity references in XML. They are shown in figure 1.9-2

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

Figure 1.9-2 XML Entity References

1.9.6.7 White-space is Preserved in XML

Unlike in HTML , XML does not truncate multiple white-space characters to one single white-space:

HTML: Hello Tove

Output: Hello Tove

XML: Hello Tove

Output : Hello Tove

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

1.9.6.8 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.

➤ 1.0 Client Side Application Development

HTML, XHTML, CSS, Document Object Model(DOM), JavaScript , DHTML, Ajax, jQuery & XML

1.9.7 XML Elements

An XML element is everything from (including) the element's **start** tag to (including) the element's **end** tag.

1.9.7.1 Elements Contents

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>
```

Example 1.9-5

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.

➤ 1.0 Client Side Application Development

HTML, XHTML, CSS, Document Object Model(DOM), JavaScript , DHTML, Ajax, jQuery & XML

1.9.7.2 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.

1.9.7.3 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.

➤ 1.0 Client Side Application Development

HTML, XHTML, CSS, Document Object Model(DOM), JavaScript , DHTML, Ajax, jQuery & XML

1.9.7.3 How is XML Elements Extensible?

XML **elements** can be extended to carry more information.

Look at the following XML example:

```
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget to pickup the material</body>
</note>
```

Example 1.9-6a

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

Figure 1.9-3

To: Tove

From: Jani

Don't forget to pickup the material

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

```
<note>
<date>03-29-2011</date>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget to pickup the material</body>
</note>
```

Example 1.9-6b

Should the application break or crash? 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.

➤ 1.0 Client Side Application Development

HTML, XHTML, CSS, Document Object Model(DOM), JavaScript , DHTML, Ajax, jQuery & XML

1.9.7 XML Attributes

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

```
  
<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>
```

1.9.7.1 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:

```
<college name='University of Houston – Clear Lake “UHCL” '>
```

or you can use character entities:

```
<gangster name="“University of Houston – Clear Lake &quot;UHCL&quot; ">
```

➤ 1.0 Client Side Application Development

HTML, XHTML, CSS, Document Object Model(DOM), JavaScript , DHTML, Ajax, jQuery & XML

1.9.8 XML Validation

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

The syntax rules were described in the previous sections.

1.9.8.1 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):

<pre><?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></pre>	Example 1.9-7
--	---------------

The DOCTYPE declaration in the example above, is a reference to an external DTD file.

Make a deliberate error in the above code and then validate your XML code here: <http://www.xmlvalidation.com/>

<pre><?xml version="1.0" ?> <note> <to>Tove</to> <from>Jani</Ffrom> <heading>Reminder</heading> <body>Don't forget me this weekend!</body> </note></pre>	Example 1.9-8
--	---------------