XML

XML is a software- and hardware-independent tool for storing and transporting data.

What is XML?

* XML stands for eXtensible Markup Language
* XML is a markup language much like HTML
* XML was designed to store and transport data
* XML was designed to be self-descriptive
* XML is a W3C Recommendation

XML Does Not DO Anything

Maybe it is a little hard to understand, but XML does not DO anything.

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

The XML above is quite self-descriptive:

* It has sender information
* It has receiver information
* It has a heading
* It has a message body

But still, the XML above does not DO anything. XML is just information wrapped in tags.

**The Difference Between XML and HTML**

XML and HTML were designed with different goals:

* XML was designed to carry data - with focus on what data is
* HTML was designed to display data - with focus on how data looks
* XML tags are not predefined like HTML tags are

Thousands of XML formats exist, in many different industries, to describe day-to-day data transactions:

* Stocks and Shares
* Financial transactions
* Medical data
* Mathematical data
* Scientific measurements
* News information
* Weather services

Let’s look at an example of XML data: An example XMLNews document:

<?xml version="1.0" encoding="UTF-8**"**?>  
<nitf>  
  <head>  
    <title>Colombia Earthquake</title>  
  </head>  
  <body>  
    <headline>  
      <hl1>143 Dead in Colombia Earthquake</hl1>  
    </headline>  
    <byline>  
      <bytag>By Jared Kotler, Associated Press Writer</bytag>  
    </byline>  
    <dateline>  
      <location>Bogota, Colombia</location>  
      <date>Monday January 25 1999 7:28 ET</date>  
    </dateline>  
  </body>  
</nitf>

**XML Tree Structure:**



The image above represents books in this XML:

<?xml version="1.0" encoding="UTF-8**"**?>  
<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>

XML documents are formed as **element trees**.

An XML tree starts at a **root element** and branches from the root to **child elements**.

All elements can have sub elements (child elements):

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

A prolog defines the XML version and the character encoding:

<?xml version="1.0" encoding="UTF-8**"**?>

**XML Syntax:**

1. XML Document must have root element: XML documents must contain one **root** element that is the **parent** of all other elements:

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

1. The XML prolong:

This line is called the XML **prolog**: <?xml version="1.0" encoding="UTF-8**"**?>

The XML prolog is optional. If it exists, it must come first in the document. XML documents can contain international characters, like Norwegian øæå or French êèé. To avoid errors, you should specify the encoding used, or save your XML files as UTF-8.

1. All XML elements must have a closing tag.

**Note:** The XML prolog does not have a closing tag! This is not an error. The prolog is not a part of the XML document.

1. XML tags are case-sensitive. The tag <Letter> is different from the tag <letter>.
2. XML Attributes must always be quoted. <note date="12/11/2007">.
3. 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>salary < 1000</message>

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

<message>salary &lt; 1000</message>

Comments in XML

<!-- This is a comment -->

**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.

<price>29.99</price>

An element can contain:

* text
* attributes
* other elements
* or a mix 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:

<title>, <author>, <year>, and <price> have **text content** because they contain text (like 29.99).

<bookstore> and <book> have **element contents**, because they contain elements.

<book> has an **attribute** (category="children").

<element></element> empty element.

<element /> self closing tag:

**XML Attributes:**

XML elements can have attributes, just like HTML. Attributes are designed to contain data related to a specific element.

<person gender="female"> OR <person gender='female'>

**XML Namespaces:**

XML Namespaces provide a method to avoid element name conflicts.

**Name Conflicts**

In XML, element names are defined by the developer. This often results in a conflict when trying to mix XML documents from different XML applications.

This XML carries HTML table information:

<table>  
  <tr>  
    <td>Apples</td>  
    <td>Bananas</td>  
  </tr>  
</table>

This XML carries information about a table (a piece of furniture):

<table>  
  <name>African Coffee Table</name>  
  <width>80</width>  
  <length>120</length>  
</table>

If these XML fragments were added together, there would be a name conflict. Both contain a <table> element, but the elements have different content and meaning.

A user or an XML application will not know how to handle these differences.

**Solving the Name Conflict Using a Prefix**

Name conflicts in XML can easily be avoided using a name prefix. This XML carries information about an HTML table, and a piece of furniture:

<h:table>  
  <h:tr>  
    <h:td>Apples</h:td>  
    <h:td>Bananas</h:td>  
  </h:tr>  
</h:table>  
  
<f:table>  
  <f:name>African Coffee Table</f:name>  
  <f:width>80</f:width>  
  <f:length>120</f:length>  
</f:table>

XML Namespaces - The **xmlns** Attribute:

When using prefixes in XML, a **namespace** for the prefix must be defined. The namespace can be defined by an **xmlns** attribute in the start tag of an element. The namespace declaration has the following syntax. xmlns:*prefix*="*URI*".

<root>  
  
<h:table xmlns:h="http://www.w3.org/TR/html4/">  
  <h:tr>  
    <h:td>Apples</h:td>  
    <h:td>Bananas</h:td>  
  </h:tr>  
</h:table>  
  
<f:table xmlns:f="https://www.w3schools.com/furniture">  
  <f:name>African Coffee Table</f:name>  
  <f:width>80</f:width>  
  <f:length>120</f:length>  
</f:table>  
  
</root>

Namespaces can also be declared in the XML root element:

<root xmlns:h="http://www.w3.org/TR/html4/"  
xmlns:f="https://www.w3schools.com/furniture">  
  
<h:table>  
  <h:tr>  
    <h:td>Apples</h:td>  
    <h:td>Bananas</h:td>  
  </h:tr>  
</h:table>  
  
<f:table>  
  <f:name>African Coffee Table</f:name>  
  <f:width>80</f:width>  
  <f:length>120</f:length>  
</f:table>  
  
</root>

**Note:** The namespace URI is not used by the parser to look up information. The purpose of using an URI is to give the namespace a unique name.

However, companies often use the namespace as a pointer to a web page containing namespace information.

Uniform Resource Identifier (URI)

A **Uniform Resource Identifier** (URI) is a string of characters which identifies an Internet Resource.

The most common URI is the **Uniform Resource Locator** (URL) which identifies an Internet domain address. Another, not so common type of URI is the **Uniform Resource Name** (URN).

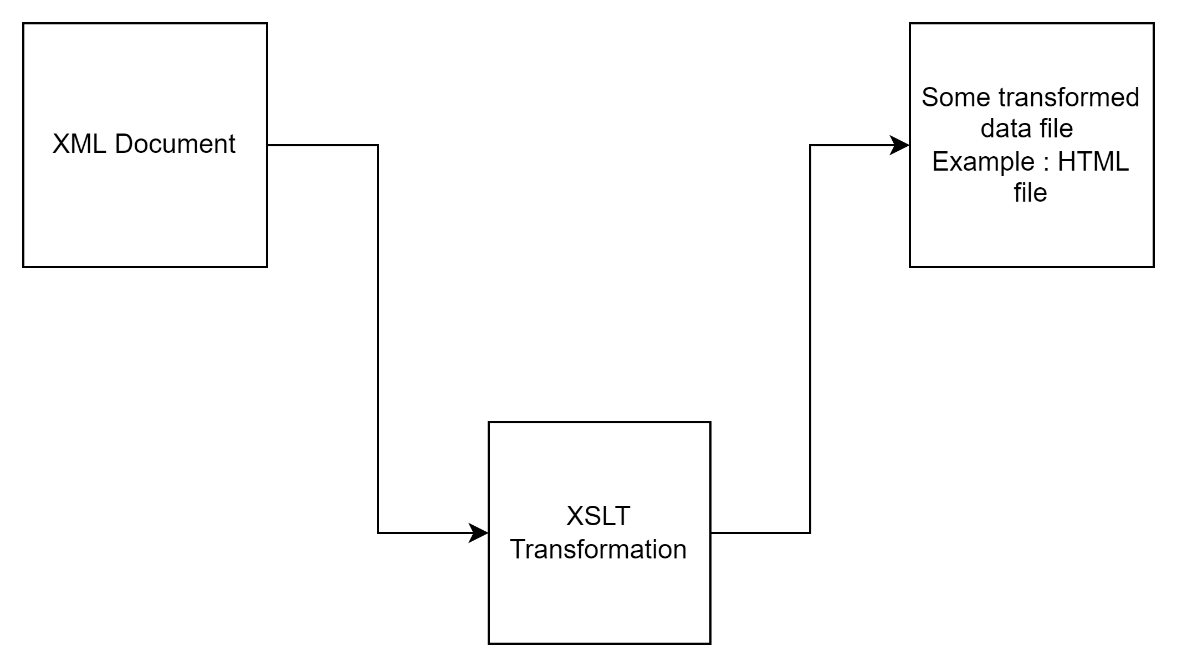
Default Namespaces

Defining a default namespace for an element saves us from using prefixes in all the child elements. It has the following syntax: xmlns="*namespaceURI*"

<table xmlns="http://www.w3.org/TR/html4/">  
  <tr>  
    <td>Apples</td>  
    <td>Bananas</td>  
  </tr>  
</table>

Namespaces in Real Use

XSLT is a language that can be used to transform XML documents into other formats. The XML document below, is a document used to transform XML into HTML.



<?xml version="1.0" encoding="UTF-8**"**?>  
  
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">  
  
<xsl:template match="/">  
<html>  
<body>  
  <h2>My CD Collection</h2>  
  <table border="1">  
    <tr>  
      <th style="text-align:left">Title</th>  
      <th style="text-align:left">Artist</th>  
    </tr>  
    <xsl:for-each select="catalog/cd">  
    <tr>  
      <td><xsl:value-of select="title"/></td>  
      <td><xsl:value-of select="artist"/></td>  
    </tr>  
    </xsl:for-each>  
  </table>  
</body>  
</html>  
</xsl:template>  
  
</xsl:stylesheet>