11. EXTENSIBLE MARKUP LANGUAGE (XML)

Introduction

Extensible Markup Language is a Meta language that describes the contents of the document. So these tags can be called as self-describing data tags. XML doesn't describe about the meaning of the tags and the correct usage of tags. It defines only user-defined tags that are not required of grammatical rules. XML was designed to describe data, not to display data.

Use of XML

1. Simplifies the data exchange procedure
2. Easy to organize the document
3. Tags or document elements are reusable
4. XML provides consistency in display of information

Applications of XML

1. Electronic Commerce (popularly known as E-Commerce)
2. Financial Funds Transfer
3. Multimedia Messages and Messaging exchange
4. Better environment for data transfer
5. Configuration of files, used in J2EE environment.

Differences between XML and HTML

S.NO.	XML	HTML
1	User defined tags	Predefined tags
2	User has control on tags	As predefined, no such control
3	XML separates content from presentation.	HTML specifies presentation
4	XML allows any king of tag names like <UNAME>…</UNAME>	HTML defines set of legal tags
6	XML allows users to create new tags	HTML doesn't allow users to create new tags
7	Self describing data can be possible	No possibility
8	You can generate new mark up languages using XML	No such possibility
9	XML is case sensitive	HTML is not case sensitive
10	Root element is user defined and only one root element allowed.	Root element is <HTML>

Features and Advantages of XML

XML is widely used in the era of web development. It is also used to simplify data storage and data sharing. The main features or advantages of XML are given below.

1) *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.

2) *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.

3) *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.

4) *XML simplifies Platform change*

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.

5) *XML increases data availability*

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.

6) XML can be used to create new internet languages

A lot of new Internet languages are created with XML. Here are some examples:

- XHTML
- WSDL for describing available web services
- WAP and WML as markup languages for handheld devices
- RSS languages for news feeds
- RDF and OWL for describing resources and ontology
- SMIL for describing multimedia for the web

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 describe data, while HTML is used to format and display the data. XML is a software- and hardware-independent tool for carrying information.

XML is a W3C Recommendation

XML became a W3C Recommendation on February 10, 1998.

Simple XML Example

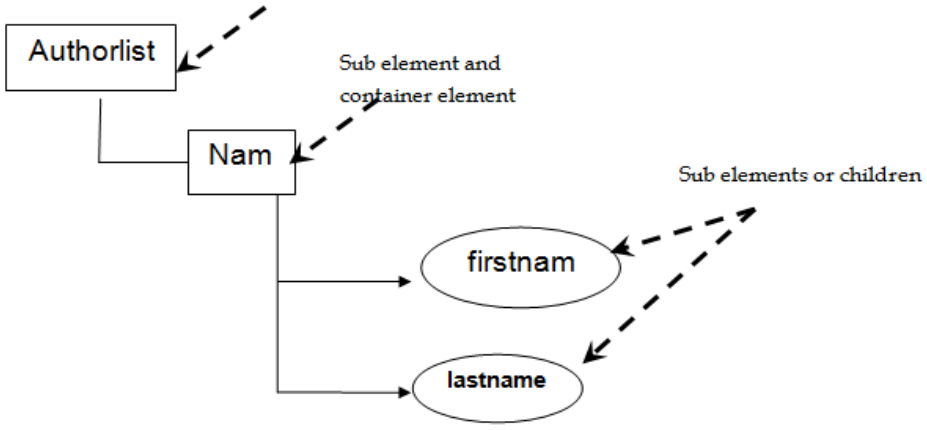
XML tags and attributes depend on the user. XML declaration always starts with xml key word. **<?xml version = "1.0"?>**

Here you can give version along with the xml key word. Version is attribute and value 1.0 indicates that xml 1.0 version you are using.

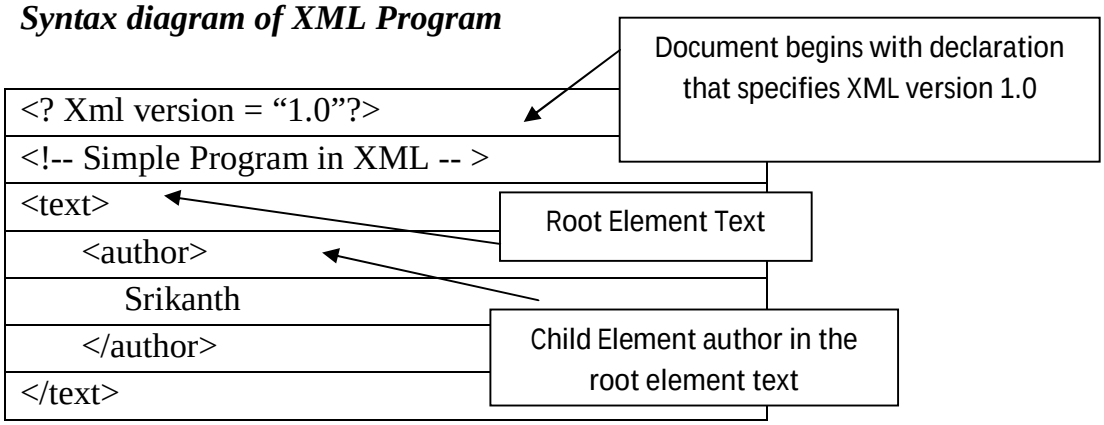
- First declaration tag starts with left angular bracket along with question mark (<?), ending with question mark followed by the right angular bracket (?>).
- Second statement is comment and then we start with actual xml program. First element <authorlist> called as root element.
  - A root element contains other sub elements. Here we used <name> as sub element and it is also known container element, because it contains other sub elements <firstname>, <lastname>.
  - Sub element is also called as children.

```
<? xml version = "1.0" ?>
<! - - First Example on XML - - >
<authorlist>
        <name>
                <firstname>Venkatesh</firstname>
                <lastname>Mahipal</lastname>
        </name>
        <name>
                <firstname>Madhupal</firstname>
                <lastname>Dhanvari</lastname>
        </name>
</authorlist>
```

Pictorial representation of above xml program is as follows:



Syntax diagram of XML Program



XML Documents Form a Tree Structure

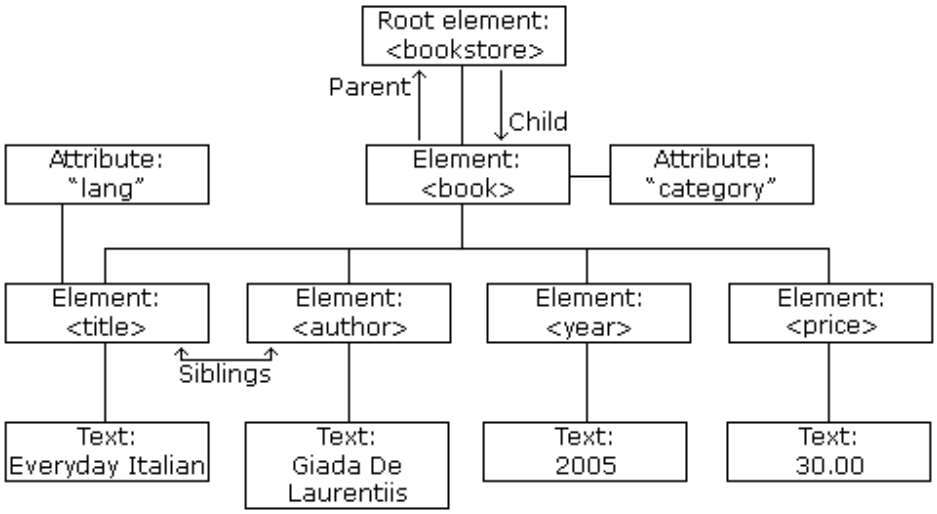
XML documents must contain a **root element**. This element is "the parent" of all other elements. The elements in an XML document form a document tree. The tree starts at the root and branches to the lowest level of the tree.

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
- XML Tags are Case Sensitive. (Opening and closing tags must be written with the same case)
- XML Elements Must be Properly Nested
- XML Documents Must Have a Root Element
- XML Attribute Values Must be Quoted (XML elements can have attributes in name/value pairs just like in HTML.)

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 < 1000 then</message>

There are 5 pre-defined entity references in XML:

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

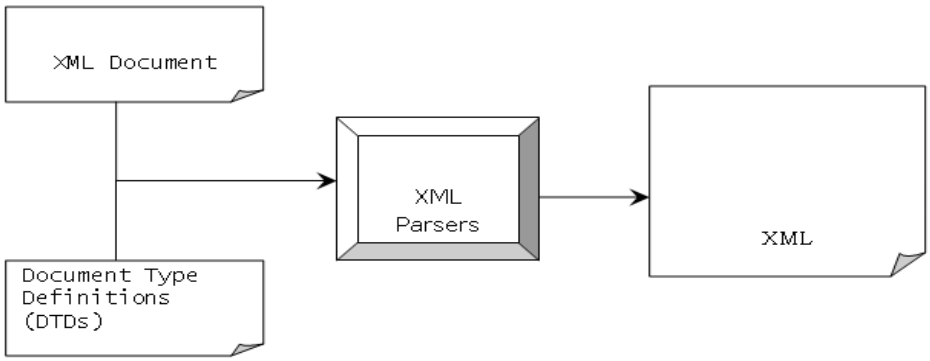
Comments in XML

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

<!-- This is a comment -->

PROCESSING OF XML OR XML PARSERS

To process the XML document content and structure XML processor or XML parsers are useful. XML parsers take XML document and DTD produces an XML application. DTD is not always necessary. The following diagram shows the usage of XML parsers.



XML Related Technologies

When people are talking about XML means it is speaking of XML and related technologies and those are:

No	Technology	Meaning	Description
1)	XHTML	Extensible html	It is a clearer and stricter version of XML. It belongs to the family of XML markup languages. It was developed to make html more extensible and increase inter-operability with other data.
2)	XML DOM	XML document object model	It is a standard document model that is used to access and manipulate XML. It defines the XML file in tree structure.
3)	XSL	Extensible	i) It transforms XML into other formats, like html.

	it contain three parts: i) XSLT (xsl transform) ii) XSL iii)XPath	style sheet language	ii) It is used for formatting XML to screen, paper etc. iii) It is a language to navigate XML documents.
4)	XQuery	XML query language	It is a XML based language which is used to query XML based data.
5)	DTD	Document type definition	It is an standard which is used to define the legal elements in an XML document.
6)	XSD	XML schema definition	It is an XML based alternative to dtd. It is used to describe the structure of an XML document.
7)	XLink	XML linking language	xlink stands for XML linking language. This is a language for creating hyperlinks (external and internal links) in XML documents.
8)	XPointer	XML pointer language	It is a system for addressing components of XML based internet media. It allows the xlink hyperlinks to point more specific parts in the XML document.
9)	SOAP	Simple object access protocol	It is an acronym stands simple object access protocol. It is XML based protocol to let applications exchange information over http. It is protocol used for accessing web services.
10)	WSDL	web services description languages	It is an XML based language to describe web services. It also describes the functionality offered by a web service.
11)	RDF	Resource description framework	RDF is an XML based language to describe web resources. It is a standard model for data interchange on the web. It is used to describe the title, author, content and copyright information of a web page.
12)	SVG	Scalable vector graphics	It is an XML based vector image format for two-dimensional images. It defines graphics in XML format. It also supports animation.
13)	RSS	Really simple syndication	RSS is a XML-based format to handle web content syndication. It is used for fast browsing for news and updates. It is generally used for news like sites.

XML Attributes

XML elements can have attributes. By the use of attributes we can add the information about the element. XML attributes enhance the properties of the elements.

Note: XML attributes must always be quoted. We can use single or double quote.

Let us take an example of a book publisher. Here, book is the element and publisher is the attribute.

<book publisher="Tata McGraw Hill"></book> OR
<book publisher='Tata McGraw Hill'></book>

Metadata should be stored as attribute and data should be stored as element.

<book>
<book category="computer">
<author> A & B </author>
</book>

Data can be stored in attributes or in child elements. But there are some limitations in using attributes, over child elements.

Why should we avoid XML attributes

- Attributes cannot contain multiple values but child elements can have multiple values.
- Attributes cannot contain tree structure but child element can.
- Attributes are not easily expandable. If you want to change in attribute's vales in future, it may be complicated.
- Attributes cannot describe structure but child elements can.
- Attributes are more difficult to be manipulated by program code.
- Attributes values are not easy to test against a DTD, which is used to define the legal elements of an XML document.

Difference between attribute and sub-element

In the context of documents, attributes are part of markup, while sub elements are part of the basic document contents.

In the context of data representation, the difference is unclear and may be confusing.

Same information can be represented in two ways:

1st way: <book publisher="Tata McGraw Hill"> </book>

2nd way:

```
<book>
    <publisher> Tata McGraw Hill </publisher>
</book>
```

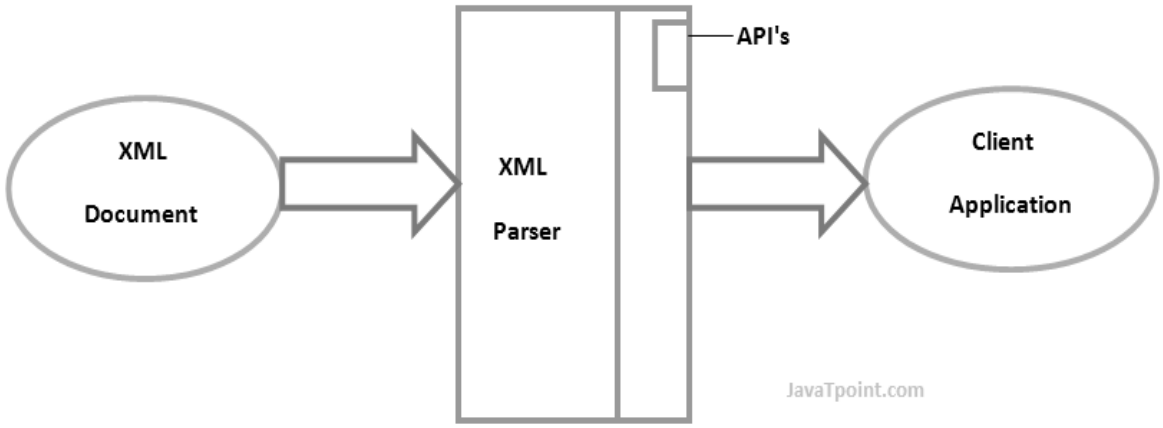
In the first example publisher is used as an attribute and in the second example publisher is an element.

Both examples provide the same information but it is good practice to avoid attribute in XML and use elements instead of attributes.

XML PARSERS

- An XML parser is a software library or package that provides interfaces for client applications to work with an XML document. The XML Parser is designed to read the XML and create a way for programs to use XML.
- XML parser validates the document and check that the document is well formatted.

Let's understand the working of XML parser by the figure given below:



Types of XML Parsers

These are the two main types of XML Parsers:

1. DOM
2. SAX

DOM (Document Object Model)

A DOM document is an object which contains all the information of an XML document. It is composed like a tree structure. The DOM Parser implements a DOM API. This API is very simple to use.

Features of DOM Parser

- A DOM Parser creates an internal structure in memory which is a DOM document object and the client applications get information of the original XML document by invoking methods on this document object.
- DOM Parser has a tree based structure.

Advantages

1) It supports both read and write operations and the API is very simple to use.
2) It is preferred when random access to widely separated parts of a document is required.

Disadvantages

1) It is memory inefficient. (consumes more memory because the whole XML document needs to loaded into memory).
2) It is comparatively slower than other parsers.

SAX (Simple API for XML)

A SAX Parser implements SAX API. This API is an event based API and less intuitive.

Features of SAX Parser

- It does not create any internal structure.
- Clients does not know what methods to call, they just overrides the methods of the API and place his own code inside method.
- It is an event based parser, it works like an event handler in Java.

Advantages

1) It is simple and memory efficient.
2) It is very fast and works for huge documents.

Disadvantages

1) It is event-based so its API is less intuitive.
2) Clients never know the full information because the data is broken into pieces.