click here for more resources

XML –eXtensible Markup Language

- **XML** stands for extensible markup language, a document formatting language used for some World Wide Web pages. A markup language is a set of codes, or tags, that describes the text in a digital document. Both **XML** and HTML contain markup symbols to describe page or file contents.
- **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 MathML (a mathematical markup language) etc.
- **XML** was designed to store and transport data. **XML** was designed to be both human- and machine-readable.

- **XML** is **easily** comprehensible to anyone who understands HTML, but it is much more powerful. More than just a markup language, **XML** is a metalanguage -- a language used to define new markup languages.

- **HTML and XML** are related to each other, where **HTML** displays data and describes the structure of a webpage, whereas **XML** stores and transfers data. **HTML** is a simple predefined language, while **XML** is a standard language that defines other languages.

Goals of development :-

- View documents easily
- Support wide variety of applications
- Compatible with existing
- Easy to write programs to process XML Documents
- Optional features absolute minimum
- Clear, legible, easy and quick

Features of XML :-A basic summary of the main features of XML follows:

- Excellent for handling data with a complex structure or a typical data.
- Data described using markup language.
- Text data description.
- Human- and computer-friendly format.
- Handles data in a tree structure having one-and only one-root element.
- Excellent for long-term data **storage** and data reusability.

XML Simplifies Things

- It simplifies data sharing
- It simplifies data transport
- It simplifies platform changes
- It simplifies data availability

XML –eXtensible Markup Language

XML Defined by :- **XML**, or Extensible Markup Language, is a markup language that you can use to **create** your own tags. It was **created** by the World Wide Web Consortium (W3C) to overcome the limitations of HTML, the Hypertext Markup Language that is the basis for all Web pages. **XML was designed** with the Web in mind.

XML – defines the syntax. /specification
XPointer - address a resource
Xlink - associate resources
XSL – stylesheet language

XML Used :- **XML** has a variety of **uses** for Web, e-business, and portable applications. The following are some of the many applications for which **XML** is useful: Web publishing: **XML** allows you to create interactive pages, allows the customer to customize those pages, and makes creating e-commerce applications. i.e.

- Exchange Data
- Share data
- Store Data
- Make Data more useful
- Create new languages

Difference between HTML and XML: There are many differences between HTML and XML. These important differences are given below:

HTML	XML
Is a markup language.	Is a standard markup language that defines other markup languages.
Is not case sensitive.	Is case sensitive.
Doubles up as a presentation language.	Is not a presentation language nor a programming language.
Has its own predefined tags.	Tags are defined as per the need of the programmer. XML is flexible as tags can be defined when needed.
Closing tags are not necessarily needed.	Closing tags are used mandatorily.
White spaces are not preserved.	Capable of preserving white spaces.
Showcases the design of a web page in the way it is displayed on client-side.	Enables transportation of data from database and related applications.
Used for displaying data.	Used for transferring data.
Static in nature.	Dynamic in nature.
Offers native support.	With the help of elements and attributes, objects are expressed by conventions.
Null value is natively recognised.	Xsi:nil on elements is needed in an XML instance document.
Extra application code is not needed to parse text.	XML DOM application and implementation code is needed to map text back into JavaScript objects.

XML Importance :-

XML allows the flexible development of user-defined document types. It provides a robust, non-proprietary, persistent, and verifiable file format for the storage and transmission of text and data both on and off the Web; and it removes the more complex options of SGML, making it easier to program for.

XML Message :-

A collection of data fields sent or received together between software applications. A **message** contains a header (which stores control information about the **message**) and a payload (the actual content of **message**). **Messaging** uses **messages** to communicate with different systems to perform some kind of function.

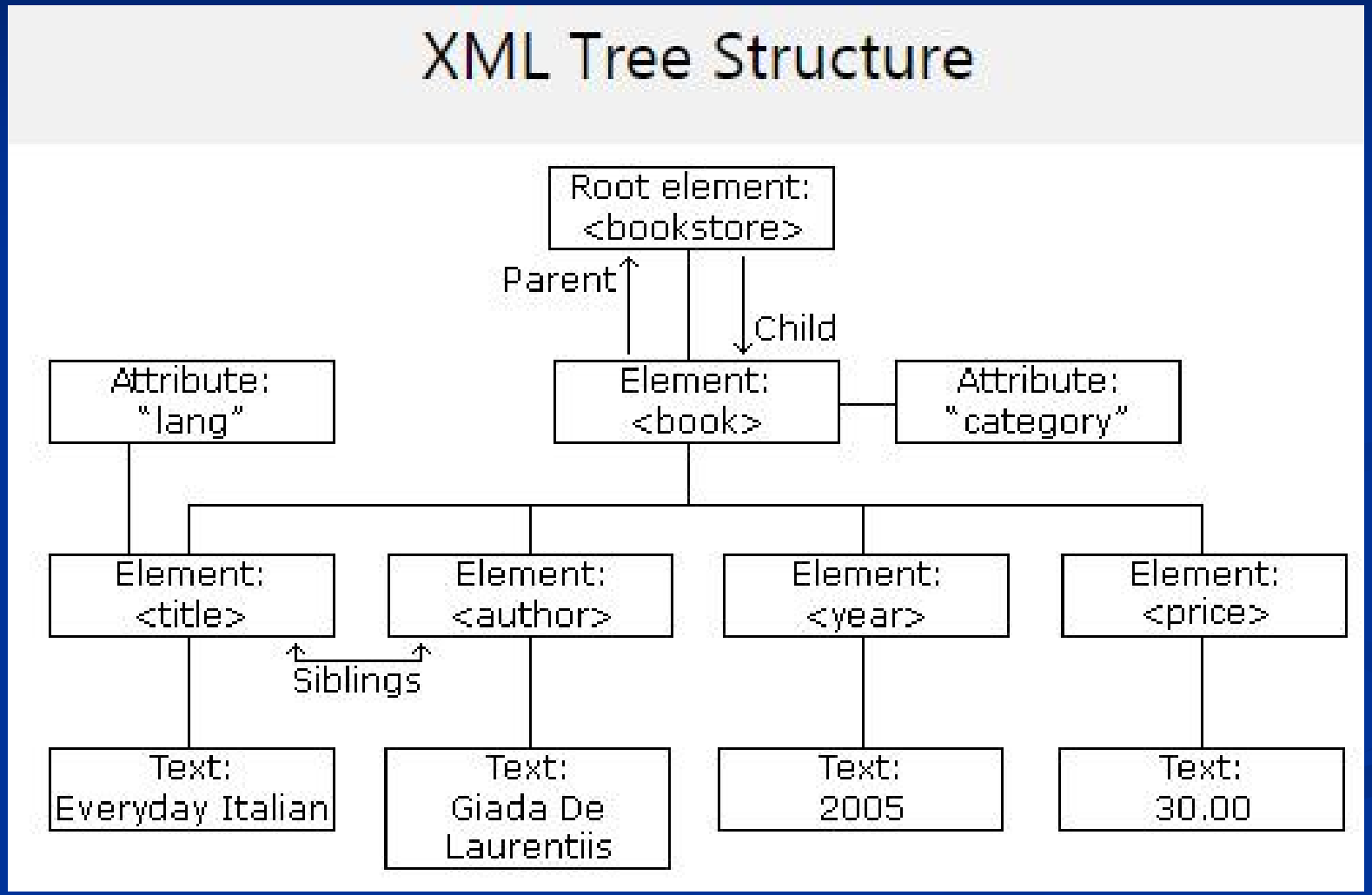
XML – Syntax and Structure Rules

XML documents have a hierarchical structure and can conceptually be interpreted as a tree structure, called an XML tree. XML documents must contain a **root** element (one that is the parent of all other elements). All elements in an XML document can contain sub elements, text and attributes.

- Document structure is like a tree
- Contains Elements – Topmost is Root
- Other elements are nested under the root element – child elements
- All paired tags
- Tags are case sensitive
- Created by text editor
- Saved with extension .xml
- Both tags and tag content seen on browser

XML Tree

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



XML Tree Structure

- 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>
```

The terms parent, child, and sibling are used to describe the relationships between elements. Parents have children. Children have parents. Siblings are children on the same level (brothers and sisters).
All elements can have text content (Harry Potter) and attributes (category="cooking").

Self-Describing Syntax

XML uses a much self-describing syntax.
A prolog defines the XML version and the character encoding:

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

The next line is the **root element** of the document:

<bookstore>

The next line starts a <book> element:

<book category="cooking">

The <book> elements have **4 child elements**: <title>, <author>, <year>, <price>.

<title lang="en">Everyday Italian</title>
<author>Giada De Laurentiis</author>
<year>2005</year>
<price>30.00</price>

The next line ends the book element:
</book>

that the XML document contains information about books in a bookstore.

```
<?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 Syntax Rules

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

XML Documents must have a Root Element

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

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

In this example **<note>** is the root element:

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

The XML Prolog

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.

UTF-8 is the default character encoding for XML documents.

Note : UTF-8 is also the default encoding for HTML5, CSS, JavaScript, PHP, and SQL.

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:
This is a paragraph.

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.

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 correct</message>

XML Elements Must be Properly Nested

In HTML, you might see improperly nested elements:
This text is bold and italic
In XML, all elements **must** be properly nested within each other:
This text is bold and italic

XML Attribute Values Must Always be Quoted

XML elements can have attributes in name/value pairs just like in HTML.
In XML, the attribute values must always be quoted:

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

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 < 1000</message>

Comments in XML

The syntax for writing comments in XML is similar to that of HTML:
<!-- This is a comment -->

Two dashes in the middle of a comment are not allowed:
<!-- This is an invalid -- comment -->

Well Formed XML

XML documents that conform to the syntax rules above are said to be "Well Formed" XML documents.

XML – Components

An **XML document** consists of three parts, in the order given: An **XML** declaration (which is technically optional, but recommended in most normal cases) A **document** type declaration that refers to a DTD (which is optional, but required if you want validation). A body or **document** instance (which is required)

The XML elements are the basic building block of the XML document. It is used as a container to store text elements, **attributes**, media objects etc. Every XML documents contain at least one element whose scopes are delimited by start and end tags or in **case** of empty elements it is delimited by an empty tag

XML – Syntax and Structure Rules

Example

```
<?xml version = “1.0”?>
<root>
    <one>
       This is the first line
     </one>
     <second>
        This is the second line
      </second>
 </root>
```

XML – Syntax and Structure Rules

XML documents have two components :-

Markup
Character data – physical structure of a document.

Markup – skeleton of document
instructs XML processors
instructs browsers for tag content

It consists of
prolog – call to XML processor
DTD (Document type definition) – start and end tag, empty elements, comments, etc.

XML – Syntax and Structure Rules

Processing instructions –

<? ?>

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

Markup –

< >

keywords – ELEMENT, !DOCTYPE

CDATA –

<![CDATA[ ]]>

is passed directly to the application, without interpretation by the XML parser

EMPTY ELEMENTS –

no end tags

< />

No page formatting information included in the XML file

XML – Syntax and Structure Rules

```
<? Xml version="1.0" encoding="UTF-8" standalone="yes"?>
 <itemdetails>
     <item>
        <description>
        </description>
        <price>
        </price>
     </item>
 </itemdetails>
```

```
 <? Xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <itemdetails>
      <item>
         <description> table
         </description>
         <price> 8000
         </price>
      </item>
  </itemdetails>
```

XML – Syntax and Structure Rules

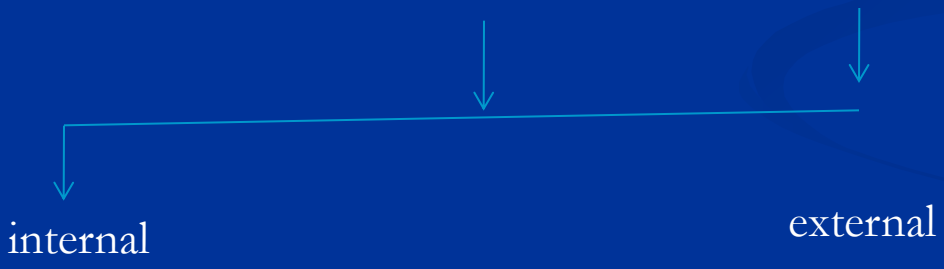
```
<? Xml version="1.0" encoding="UTF-8" standalone="yes"?>
 <itemdetails>
     <item ID="I001">
        <description color="black"> table
        </description>
        <price> 8000
        </price>
     </item>
      <item ID="I002">
        <description color="brown> chair
        </description>
        <price> 5000
        </price>
     </item>

 </itemdetails>
```

XML – DTD

Creating a valid XML Document :-

- Correct nesting of Root and child elements
- Paired tags
- Correctly used empty tags
- Contains DTD

DTD – defines the elements, attributes, entities, and rules for creating one or more documents in a markup language



XML – DTD

Internal DTD

```
<?xml version="1.0"  standalone="yes"?>
<!DOCTYPE doc
[
    <!ELEMENT  doc (#PCDATA)>
]>
<doc>

</doc>
```

External DTD

```
<?xml version="1.0"  standalone="no"?>
<!DOCTYPE doc SYSTEM/PUBLIC ".....">
<doc>
</doc>

<!ELEMENT  doc (#PCDATA)>
```

XML – DTD

DECLARING ELEMENTS

The content type specification –

- EMPTY
- ANY
- MIXED
- CHILDREN

For ex.

```
<?xml version="1.0"  standalone="yes"?>
<!DOCTYPE doc
[
    <!ELEMENT  doc EMPTY>
]>

<doc>

</doc>
```

XML – DTD

Using CHILD ELEMENTS

Strict order - ,
any order - |

For ex.

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE doc
[
    <!ELEMENT doc ANY
    <!ELEMENT name (firstname, lastname>
    <!ELEMENT firstname (#PCDATA)
    <!ELEMENT lastname (#PCDATA)
]>

<doc>
   <name>
      <firstname> kumud</firstname>
      <lastname> saxena </lastname>
   </name>
</doc>
```

XML – DTD

Some special character symbols to control each element

? – once or not at all

* - Not at all or as many times

+ - Once or as many times

For ex.

```
<?xml version="1.0"  standalone="yes"?>
<!DOCTYPE doc
[
    <!ELEMENT  doc ANY
    <!ELEMENT  name (firstname?, lastname+>
    <!ELEMENT  firstname (#PCDATA)
    <!ELEMENT  lastname (#PCDATA)
]>

<doc>
   <name>
      <lastname>  saxena </lastname>
   </name>
</doc>
```

XML – DTD

Metadata
Data about data
W3C supports XML data as

-Resource Description Framework (RDF) model
- Resource Description Framework (RDF) Schema Specification

-RDF section looks like

```
<RDF>
  <Description about="……">
    <Creator> ……</Creator>
     <Email>……</Email>
   </Description>
</RDF>
```

XML AND DATA BINDING

- Data islands – DSO(Data source Object)
- Binds XML in HTML
- Client side data manipulation – data is updated., modified on client side and send back to the server
- Display the recordsets and uses methods

XML AND DATA BINDING

```
<HTML>
<HEAD>
  <title>   using XML</title>
</HEAD>
<BODY>

<XML ID="Stud">
 <?xml version="1.0"?>
   <COLLEGE>
      <STUDENT>
         <NAME>A</NAME>
         <CLASS>MCA</CLASS>
          <MARKS>70</MARKS>
       </STUDENT>
       <STUDENT>
         <NAME>B</NAME>
         <CLASS>MBA</CLASS>
          <MARKS>60</MARKS>
       </STUDENT>
    </COLLEGE>
 </XML>
```

XML AND DATA BINDING

```
<TABLE  DATASRC="#stud">
      <tr>
         <th>name</th>
         <th>class</th>
         <th>marks</th>
      </tr>
      <tr>
         <td><DIV datafld="NAME"></DIV></td>
         <td><DIV datafld="CLASS"></DIV></td>
         <td><DIV datafld="MARKS"></DIV></td>
      </tr>
</TABLE>

</BODY>

</HTML>
```