UNITED
GROUP OF INSTITUTIONS

Web Technology
(KCS-602)
Unit 2
XML

Prepared By

Mr. Abhishek Kesharwani

Assistant Professor,UCER Naini,Allahabad

Introduction to 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
- XML is a W3C Recommendation
- XML is platform independent and language independent.

Features and Advantages of XML

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. This way you can focus on using HTML/CSS for display and layout, 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.

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.

This makes it much easier to create data that can be shared by different applications.

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.

XML is Not a Replacement for HTML

- It is important to understand that XML is not a replacement for HTML. In most web applications, XML is used to transport data, while HTML is used to format and display the data.

With XML You Invent Your Own Tags

- 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.

HTML	XML
HTML stands for Hyper Text Markup Language.	XML stands for extensible Markup Language.
HTML is static in nature.	XML is dynamic in nature.
HTML is a markup language.	XML provides framework to define markup languages.
HTML can ignore small errors.	XML does not allow errors.
HTML is not Case sensitive.	XML is Case sensitive.
HTML tags are predefined tags.	XML tags are user defined tags.
There are limited number of tags in HTML.	XML tags are extensible.

HTML does not preserve white spaces.	White space can be preserved in XML.
HTML tags are used for displaying the data.	XML tags are used for describing the data not for displaying.
In HTML, closing tags are not necessary.	In XML, closing tags are necessary.
HTML is used to display the data.	XML is used to store data.
HTML does not carry data it just display it.	XML carries the data to and from database.

Example XML Document

```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="web">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

Overall structure

- An XML document may start with one or more processing instructions (PIs) or directives:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="ss.css"?>
```

- Following the directives, there must be exactly *one* root element containing all the rest of the XML:

```
<bookstore>
   ...
</bookstore>
```

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>
```

Elements and attributes

- Attributes and elements are somewhat interchangeable
- Example using just elements:

```
<name>
    <first>David</first>
    <last>Matuszek</last>
</name>
```

- Example using attributes:

```
<name first="David" last="Matuszek"></name>
```

- You will find that elements are easier to use in your programs--this is a good reason to prefer them
- Attributes often contain metadata, such as unique IDs

Well-formed XML

- Every element must have *both* a start tag and an end tag, e.g. <name> … </name>
  - But empty elements can be abbreviated: <break />.
  - XML tags are case sensitive
  - XML tags may not begin with the letters xml, in any combination of cases
- Elements must be properly nested, e.g. *not* <b><i>bold and italic</b></i>
- Every XML document must have one and only one root element
- The values of attributes must be enclosed in single or double quotes, e.g. <time unit="days">
- Character data cannot contain < or &

Entities

- Five special characters must be written as entities:

&amp;	for	&	(almost always necessary)
&lt;	for	<	(almost always necessary)
&gt;	for	>	(not usually necessary)
&quot;	for	"	(necessary inside double quotes)
&apos;	for	'	(necessary inside single quotes)

- These entities can be used even in places where they are not absolutely required
- These are the *only* predefined entities in XML

XML declaration

- The XML declaration looks like this:
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  - The XML declaration is not required by browsers, but *is* required by most XML processors (so include it!)
  - If present, the XML declaration must be first--*not even whitespace* should precede it
  - Note that the brackets are <? and ?>
  - version="1.0" is required (this is the *only* version so far)
  - encoding can be "UTF-8" (ASCII) or "UTF-16" (Unicode), or something else, or it can be omitted
  - standalone tells whether there is a separate DTD

XML Encoding

- To avoid errors, you should specify the encoding used, or save your XML files as UTF-8.
- Unicode is an industry standard for character encoding of text documents. It defines (nearly) every possible international character by a name and a number.
- Unicode has two variants: UTF-8 and UTF-16.
- UTF = **U**niversal character set **T**ransformation **F**ormat.

XML Encoding

- UTF-8 uses a single byte (8-bits) to represent commonly used characters and two (or three) bytes for the rest.
- UTF-16 uses two bytes (16 bits) for most characters, and three bytes for the rest.
- UTF-8 Web Standard
- UTF-8 is the standard character encoding on the web.
- UTF-8 is the default character encoding for HTML-5, CSS, JavaScript, PHP, SQL, and XML.

Comments

<!-- This is a comment in both HTML and XML -->

- Comments can be put anywhere in an XML document
- Comments are useful for:
  - Explaining the structure of an XML document
  - Commenting out parts of the XML during development and testing