

## XML

(1)

XML stands for extensible Markup Language. XML was designed to add new facilities to existing web technologies. XML is a hierarchical data structure stored in text format files, using a markup language.

XML has following two main reasons for development:

1) Computers don't understand the info. stored in them. e.g. there is no way for a search engine or any other computer, to know that this page contains the introduction part of an XML. All it is a collection of letters & numbers, with HTML formatting around it. If a page or document is written in XML, a computer can understand exactly what it is about.

2) Web-Pages are not compatible across diff. devices. People are now accessing the web pages from a variety of different devices i.e. PCs, Macs, mobile phones, palm-top computers & even televisions. Bcoz of this, web designers must now either produce their pages in several diff. formats to cope with this. XML is used to define what data means & not how it is displayed, it makes it very easy to use the same data on several diff. platforms.

XML is not a way to design your web page. XML is to describe data.

HTML is used to describe how data is formatted. XML is used to describe what data actually means.

(A) XML tags: The tags used in XML, is similar in construction to HTML. XML tags are case-sensitive. XML tags are not pre-defined like HTML ones are. XML tags can be nested.

(B) XML Correctness: XML is that it should be independent of the platform it is running on. As XML does not actually do anything it is up to software developers to make it easy to use this data on a particular platform. If XML has not written in proper syntax, the error will be displayed.

(C) Declaring XML: The correct way how to declare an XML document

```
<?xml version="1.0"?>
```

This tells whatever software receives this data that you are writing XML and that it should match the specification for version 1.0. Two optional attributes can be used in this line

1) encoding: specifies the character set used in the document, possible values are "UTF-8", "UTF-16" etc.

2) standalone attribute is optional and specifies whether the XML document can be completely validated using DTD rules contained in the document. Possible values are Yes & No.

3) XML Document must have a Root element. Root element is the parent of all other elements.

## Advantages of XML

### 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 concentrate on using HTML for layout & display & be sure that change in the underlying data will not require any changes to HTML.

### 2) XML Simplifies Data Sharing

In the real world, computer systems & databases contain data in incompatible formats. XML is stored in plain text. This provides a S/w and H/w independent way of storing data.

### 3) XML Simplifies Data Transport

Exchanging data b/w incompatible systems over the Internet with XML reduces the complexity. XML is stored in plain text so it makes it easier to expand or upgrade to new O/S, new applications or new browsers without losing data.

## ~~WWW~~ (W3C) - World wide web (consortium).

~~HTML~~ started out as an informal specification used by only a few people as more & more authors begin to use the lang. it became obvious that more formal means were needed to define & manage to standardize HTML features making it easier for everyone to create & share documents.

~~W3C~~ In Oct. 1994, Tim Berners Lee founded an organization called WWW (World wide Web Consortium) W3C, devoted to develop non-proprietary, inter-operable technologies for the WWW. One of the W3C primary goal is to make the Web universal, accessible regardless of availability, language or culture.

The W3C is also a standardization organization.

Web technologies standardized by the W3C are called recommendations.

W3C recommendations include the XML, CSS, HTML, X-HTML.

A recommendation is not the software product but a document that specifies a technology's role, syntax rules of etc.

Members provide the primary financing for the W3C & help provide the strategy & direction of the consortium.

Members are responsible for drafting, circulating for review, & modifying the standard based on cross-Internet feedback.

The W3C homepage i.e. [www.w3.org](http://www.w3.org) provides extensive resources on Internet & web-technology.

for each internet tech with which W3C is involved, the site provides a description of technology & its beneficial to a web designer, the history of technology future goals of the W3C in developing the technology.

Another is IETF (Internet Engineering task force).

Even broader + reach than W3C.  
The IETF is responsible for defining + managing every aspect of Internet tech.

The IETF defines all of the tech. of the internet via official document known as request for comment or RFC's. (RFC's).

## \*RDF (Resource distribution framework)

It is a family of W3C specification originally designed as a meta-data model, what which has come to be used as a general method of modelling info through a variety of syntax format.

RDF is a standard model for data interchange on the web.

The RDF meta-data model is based upon the idea of making statements about resources in the form of subject-predicate-object.

called triples in RDF terminology.

The subject denotes the resource & the predicate denotes aspect or properties of the resource & express a relationship b/w subject & object.

For e.g.: if we want to represent the statement, the sky has the colour Blue in RDF where the subject is 'the Sky' predicate is '(has the colour)' & object is 'Blue'.

RDF is used for representing meta-data about web resources such as title, author, modification date of web page, copy right & licensing info about the web document for some 'share' resources.

RDF can be used for to represent info about things which can be identified on the web. Even when they can't be directly retrieved on the web.

e.g.: Include inf<sup>n</sup> about items available from online shopping facilities.

RDF is intended for situations in which this info needs to be processed by application rather than being only displayed to people

RDF provides a common framework in expressing these info so it can be exchanged b/w applications means that the inf' may be made as table th available other than those for which it was originally created.

### Features of RDF →

1) RDF is designed to be read by computers.

RDF was designed to provide a common way to describe inf' so it can be read & understood by computer app's.

RDF description are not designed to be display on the web.

~~2) RDF is written in XML →~~

RDF documents are written in XML. The XML lang. used by RDF is called RDF/XML.

By using XML, RDF info can easily be exchanged b/w different types of computers using different type of operating system + application lang.

~~3) RDF & the Semantic Web →~~

The RDF lang. is a part of the W3C's Semantic web activity. Semantic web is where (a) web info has exact meaning

~~b) Web info can be understood & processed by computer~~

## LIMITATIONS

- 1) RDF syntax is too verbose
- 2) The triplet notation is not expressive enough

## Application :

① RDF Site Summary (RSS).

RSS

Language for publishing info about updates made to a web-page.  
It is often used for the ~~missing~~ diminishing / distributing news articles, summaries & sharing web-block content.

② FOAF (friend of a friend)

design to describe people, their interest & interconnection.

③ DOAC (Description of a career)

Supplements FOAF to allow sharing of resume info.

④ DOAP (Description of a project)

It is designed to describe software projects

\* **[HyperText]** A system of writing & displaying text that enables the text to be linked in a multiple ways, to be available at several level of detail & to contain links to related document is called HyperText. It is the organization of info. units into connected association that user can choose to make. It is the main concept that led to the invention of WWW. It is non sequential method for reading a document displayed on comp. Screen. It can be designed to perform various tasks; for instance when a user clicks on it, bubble with a word defn may load or video clip may run.

→ **[Hypermedia]** It is a term used as a logical extension of term HyperText in which audio, video, plain-text & non linear hyperlinks intertwine to create a generally non-linear medium of info. It is relatively new term that describe two new technology

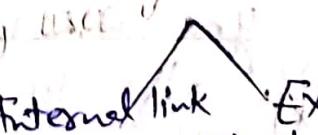
### (i) Multimedia

↓  
It refers to the capabilities of modern comp. to provide info. to user in a no. of diff. form

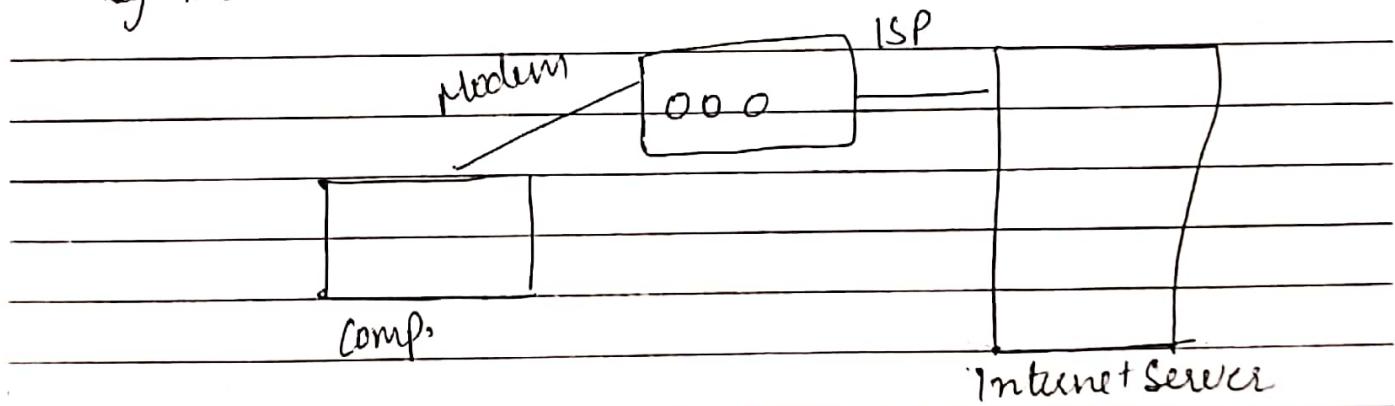
### (ii) Hyperlink

↓ It is a reference or navigation element in a document to another section of same document that automatically brings the referred info. to user when navigation element is selected.

Used to link a section within some webpage or website. Only relative URL is required.

  
Internal link      External link  
Used to link a section within other webpage. Absolute URL is required.

\* How web servers work → A web server is a program running on server comp. It consists of website containing no. of web page. A webpage simply special type of comp. file written in special design lang. HTML. The web server constantly & passively waits for a request for a webpage from browser prog. running on the client & when any such request is received it locates that corresponding page & sends it to requesting client computer. Every website has a server process that listens to TCP connection request coming from client. After connection is established the client send one req. & server send one response. This request-response model is governed by protocol called HTTP.



Let's say you want to visit any website first enter address or URL of website in browser. Then your browser req. the web page from webserver that host that website. The server sends the data over the internet to your comp. Your web browser interpret the data, displaying it on your comp. screen. That website also has links to other site. When you click on that link you access the webserver for the glue that holds the web together is called hyper text or links.

→ HTTP → Hyper text transfer Protocol is a communication protocol used to connect to server on www. It is to establish connection with server & transmit HTML Pages to user browser. It defines how messages are formatted & transmitted and what actions Web Server & browsers should take in response to various commands.

Three versions of HTTP

- ① HTTP 1.0. 9
- ② HTTP 1.0
- ③ HTTP 1.1

It is the protocol for distributed, collaborative, hypermedia info. System. It is foundation of data communication for www.

HTTP Session → It is sequence of the request-response transaction.

Request Message → It consist of :

- ① Request line such as GET /images/logo.png  
HTTP/1.1 which req. a resource called /images/logo.png from Server.
- ② Headers
- ③ empty line
- ④ optional msg- body

Request Method → HTTP request is a collection of lines sent to the server by browser

It includes

→ Request line → type of doc. requested, method which must be specified & version of Protocol. The line is made up of 3 elements.

- (1) Method
- (2) URI
- (3) Version

HTTP Response is it is a collection of lines sent to the server by browser. It includes:

- (1) Status line
- (2) Response Header fields
- (3) Body of the Response

invented by Tim Berners-Lee

→ HTML → Hyper text markup lang.

- (1) Hypertext is ordinary text that has been dressed up with extra features such as formatting, images, multimedia & links to other documents.
- (2) Markup is the process of taking ordinary text & adding extra symbol.
- (3) language refers to the way of communication b/w webpage which has its own syntax & rules.

#### Characteristics:-

- (1) It is easy to understand & easily modified.
- (2) It provides a flexible way to design the web page.
- (3) Graphic, video, sound can also be used & imported to give attractive look to web page.
- (4) Effective presentation can be made.
- (5) HTML documents can be displayed on any platform.

attr(SIZE → height of input  
WIDTH, NO SHADE (create a flat look)  
ALIGN (left, right, center))

Page No.	
Date	

(9)

HTML tags → It is used to mark up HTML element  
These are surrounded by two characters.

< and >

angle brackets called

→ It normally comes in pair like <b>&</b>  
↓            ↓  
Start tag    end tag

→ It is not case sensitive

<b> means the same as <B>

List of tags +

<a>, <abbr> <applet> <b>, <big>  
<frame>, <frameset>, <h1>, <h2> - <h6>  
<head>, <hr>, <html> <style> etc

HTML attributes → It defines a property for an element,  
consist of an attribute / value pair  
& appears within element's start tag.

⇒ URL → Uniform Resource Locator. It is an address for web page or any other files on internet. Every URL on the web is unique. Particular syntax.

Several types of URL

- ① Simple webpage URL
- ② URL with account name
- ③ IP address in URL
- ④ URL with path name & file name
- ⑤ URL with port numbers.

Ex:- http://www.yahoo.com

transfer  
Protocol

Domain name.

The format of URL follows:

(1) How can the page be accessed?

(2) Where can the page be found?

(3) What is the file name correspond to page?

How / where / what

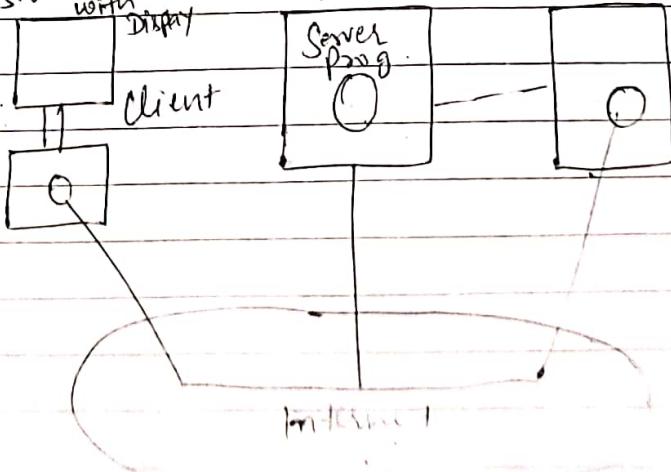
for ex: HTTP://PubPages.umn.edu/index.html

### \* How the Browser Work :-

Web browser consist of

SW that runs on Comp. & display home page on web. A web browser display info. on your comp by interpreting the HTML that is used to build home pages on web.

User's Comp. with display      Remote Comp. 1      Remote Comp. N



working of web-browsers.

A web browser acts as the client in the user's interaction. Using this prog., a user requests for web page that is stored on web server. The web server locates this web page & sends it back to client comp. The web browser then interprets the web page written in the HTML language & then displays it on the client comp. Screen. Interaction b/w web browser & web server is:

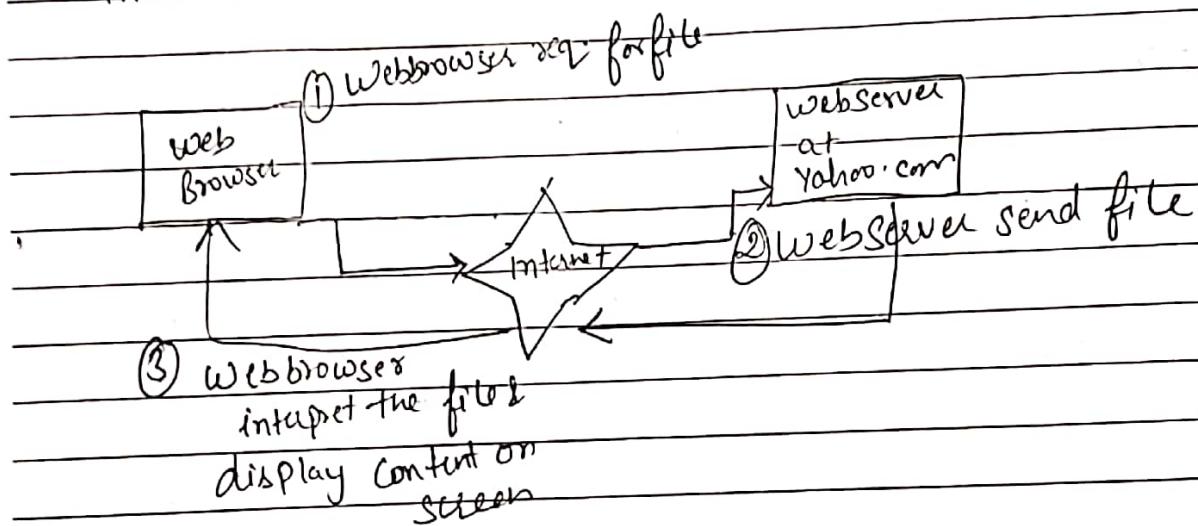


fig. Interaction b/w web browser & web server

→ **MIME Types** → MIME stands for Multipurpose Internet Mail Extensions

Type	SubType	Description	Extension
① Application	Postscript	Printable Postscript	.esp
	Text	Tex document	.text
② Audio	Aiff	Apple Sound	.aif, .aiff
	Au	Sun Microsystem Sound	.au, .snd
	Midi	Musical instrument	.midi
③ Image	Gif	digital interface Graphics Interchange format	.gif
	Jpeg	Joint Photographic Expert Group	.jpg
	Png	Portable Network Graphics	.png

		Virtual Reality Modeling	World
④ Model	VRML	lang-	.wrl
⑤ Text	HTML sgml	Hyper-text markup lang- Std. Generalized Markup lang.	.txt .sgml
⑥ Video	Avi Mpeg	Microsoft Audio Interlaced Moving Picture Experts Group	.avi .mpeg, .mpg

## → PLUG-INS & HELPER APP.

Plugin & helper app

do the same thing they extend the power of the browser.

Plugins are more tightly integrated with browser

so there is less work for you to do.

plugins are capable of playing audio, showing movies, running automation & working with calendars & other things.

The adobe acrobat is an ex. of Plugins.

Helper App. or helper is stand alone prog.

that are used to process or display data

that is not directly integrated into web page

Helpers do not display their info. inside

the browser window instead they launched

in their own window.

### Plug-ins

- Closely tied to browser
- Display inside browser window
- Broad selection available.
- Launches quickly.
- Installation involves down. loading Plug-in & running procedure.

### Helper App

- Prog. stands alone.
- Display in Separate window
- Broad Selection available.
- Many launches more slowly than plug ins.
- Configuring involves download the helper app & editing your preferences.

\* Standard HTML → frames  
forms

Frame → with frames, you can display more than one HTML document in same browser window.  
Each HTML document is called a frame & each frame is independent of others.

Disadvantage - (1) web developer must keep track of more HTML documents.

(2) It is difficult to print the entire page.

Frameset tag + define how to divide window into frame.

→ Each frameset defines a set of rows or columns.

frame tag → It defines what HTML document to put into each frame.

```
for ex → <frameset cols = "25%, 75%">  
          <frame src = "frame-a.html">  
          <frame src = "frame-a.html">  
      </frameset>
```

frame Tag →

<frameset> Define a set of frames.  
<frame> Define subwindow

Usage → Common Content

form verification

Table of Content.

forms → It allows you to type info. into fields on a browser. Screen & submit that info. to a web server.

which sends the data without displaying any info. → specifies URL to send the data.

Text field = `<form method="post" action=" " >`  
Name: `<input type="text" size="10" name="name" >`  
`</form>`

form

Radio Button → `<input type="radio" name="size" value="large" >`

In this we have used to quizzes, Questionnaire & other website that give user a MCQ.

→ Value → it will be sent if user choose radio button.

→ name → which set of radio buttons

Check Box → Multiple items to be selected for certain group.

Dropdown list

It is created with `<select>` & `<option>`

tag.

It is the list itself

available  
for  
choice.

Ex:-

```
<html>  
<Body>  
<form>
```

username: `<br>`

`<input type="text" name="username">`  
`<br>`

Email id: <br>

```
<input type = "text" name = "username"> email.id">  
<br> <br>
```

Submit <input type = "submit" value = "Submit" >

```
</form>
```

```
</body>
```

```
</html>
```

O/P

Username :

Email id :

submit



~~XML~~



HTML, XML, XHTML, and the W3C.

~~Reg.~~

CH

## Program: Timetable

<HTML>

<HEAD>

<TITLE>

TIMETABLE

</TITLE>

</HEAD>

<CH1 ALIGN = "CENTER"> TIMETABLE (SESSION : JAN-MAY 2012) </CH1>

<Table border = "2" border color = "Red" bgcolor = "Blue" > <Br>

<th>

<td> <sup>&nbsp;</sup> <td> 1 <td> 2 <td> 3

8

<tr>

<td>

<td> &nbsp <td> 9:45 - 10:35 <td>

3:55

<th> ~~MONDAY~~

<td> MONDAY <td> WEDNESDAY <td rowspan = "2" > LAB </td>

</td>

</table>

<Body>

</HTML>

### Mandatory XHTML Elements

All XHTML documents must have a DOCTYPE declaration. The html, head, title, and body elements must be present.

This is an XHTML document with a minimum of required tags:

```
<!DOCTYPE Doctype goes here>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Title goes here</title>
  </head>
  <body>
    </body>
  </html>
```

*Specimen Copy  
Not For Sale  
Arun Kr. Singh*

#### ~~12.1.~~ Difference between HTML and XML

HTML	XML
HTML stands for Hyper text markup language	XML stands for extensible markup language
HTML was designed to carry data (what you want to display) and layout (how you want to display it) information.	XML was designed to carry data but not the layout (presentation) informed.
HTML is a not a case sensitive language.	XML is a case sensitive language.
In HTML, the elements are pre-defined.	In XML, the elements are defined by the developer.
Browsers try to render every part of HTML file, regardless of errors or incorrect coding.	In XML, browser generates for mis-coded XML files.
Work in web browser (All type)	Work in Netscape 6, Internet Explorer 5, 6

#### 1.12.2. Similarity between HTML and XML

Both HTML and XML are mark up languages used to design web pages. Both use text files and therefore are software and hardware independent. In XML, there is no concise list of XML tags because XML

Example :

```
<HEAD>
<TITLE>Stamp Collecting World </TITLE>
<META name="description" content="Everything you wanted
to know about stamps, from prices to history.">
<META name="keywords" content="stamps, stamp collecting,
stamp history, prices, stamps for sale">
</HEAD>
```

### 1.23 SEMANTIC TAGS FOR AIDING SEARCHES

A Semantic Tag is a collection of filters that transforms "regular tags" into RDF triples.

When a "regular tag" is filtered by a Semantic Tag, it is substituted into a template by the Semantic Tag Processor Service (Fig. 14).

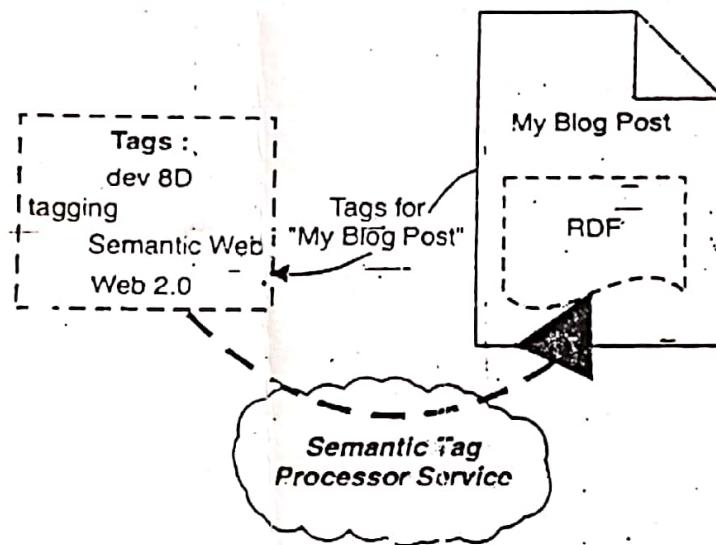


Fig 14. "My Blog Post" with embedded RDF metadata

**Use Case 1: Automatic generation of RDF metadata using a REST interface.**

A participant of a recent happiness-enhancing event has authored an XHTML article on the Internet ("My Blog Post"). The article is tagged with the following keywords: "dev8D", "tagging", "Semantic Web", "Web 2.0". Before the article is published, the Content Management System (CMS) that hosts the article issues a request to the Semantic Tag Processor Service. The results of all matching Semantic Tags (a bag of RDF sub-graphs) are combined to form a single RDF document. The CMS embeds the RDF document into the XHTML content of the article (Figure).

## Lists of semantic tags are :

Tag	What it is	When to use it
<a>	Anchor (most commonly a link)	Vital. Use to create links in content. Use the title attribute whenever the contents of the <a>...</a> pair do not accurately describe what you'll get from selecting the link. Title attribute often displays as a tooltip in visual browsers, which may be a helpful usability aid.
<abbr>	Defines an abbreviation	Works in a similar way to <dfn> and <acronym>, using a title attribute (displays a tooltip in standard visual browsers). e.g. <abbr title="Hypertext markup language">HTML</abbr>
<ACRONYM>	Defines an acronym	Works in a similar way to <abbr> and <dfn>, using a title attribute (displays a tooltip in standard visual browsers).
<ADDRESS>	Used for marking up a physical (e.g. mailing) address	Not commonly used. Recommend looking into microformats, which allow for more details and interoperability.
<APPLET>	Inserts a Java applet	The old way to insert a Java applet. Use <object> instead today.
<AREA>	Hotspot in image map	Avoid using image maps where possible. Occasionally needed.
<BASE>	Specifies the base location of the document	Used only when necessary. Adjusts any relative links or paths within the document.

<b>&lt;BASEFONT&gt;</b>	Gets default font size	Display info - never use it
<b>&lt;BIG&gt;</b>	Larger text	Display info - never use it
<b>&lt;BLINK&gt;</b>	Makes text blink	You go to hell if you use this
<b>&lt;BLOCKQUOTE&gt;</b>	Large quoted block of text	Use for any quoted text that constitutes one or more paragraphs (note: should contain <b>&lt;p&gt;</b> tags as well). Use <b>&lt;q&gt;</b> for quotations within a paragraph. Often used in conjunction with <b>&lt;cite&gt;</b> to cite the quotation's source.
<b>&lt;BODY&gt;</b>	Document body	Essential (unless you're using frames)
<b>&lt;BR&gt;</b>	Line break	This is arguably display information. Still in common use, but use with restraint
<b>&lt;B&gt;</b>	Bold text	Display info - never use it
<b>&lt;BUTTON&gt;</b>	Used for a standard clickable button within a form	Often better than <b>&lt;input type="button" /&gt;</b> or <b>&lt;input type="submit" /&gt;</b> , as it allows you to assign different styles based on the HTML element alone, whereas differentiating style based on the type of input is less well supported.
<b>&lt;CAPTION&gt;</b>	Caption for a table: describes the table's contents	The correct way to assign a title to a table
<b>&lt;CENTER&gt;</b>	Centred block	Display info - never use it. Use <b>&lt;div&gt;</b> or some other block-level tag with the style <b>text-align:center</b> instead

<b>&lt;CITE&gt;</b>	Defines a citation	Defines the source of a quotation in conjunction with content in <q> or <blockquote> pairs.
<b>&lt;CODE&gt;</b>	Defines an extract of code	Not commonly used. Similar to <pre> tag, but collapses consecutive white spaces and line breaks in the source.
<b>&lt;COL&gt;</b>	Identifies a particular column in a table	Can be very useful, e.g. <col class="namecol"> can be applied to each first column in a series of tables, then the width of each column may be set to be equal in the stylesheet, overriding the table's natural tendency to adjust its own column widths to fit its contents.
<b>&lt;DFN&gt;</b>	Definition of a term	Works in a similar way to <abbr> and <acronym>, using a titleattribute (displays a tooltip in standard visual browsers).
<b>&lt;DIR&gt;</b>	Directory list	Now deprecated. Use a standard <ul> or other list instead.
<b>&lt;DIV&gt;</b>	Division	Specifies a logical division within a document. Use it to separate or identify chunks of content that are not otherwise distinguished naturally using other tags.  One of the most common HTML tags.
<b>&lt;DL&gt;</b>	Definition list	Contains one or more definition-term / definition-description pairs.

<DT>	Definition term	Used as part of a <dt></dt><dd></dd> pair within a definition list (<dl></dl>)
<DD>	Definition description	
<EM>	Emphasis	Commonly used in place of the old <i> (italics) tag to indicate emphasis (but less than <strong>)
<FONT>	Font settings	Display info - never use it
<FORM>	Input form	Essential for data input
<H1>	Level 1 header	Aim to have one H1 on each page, containing a description of what the page is about.
<H2>	Level 2 header	Defines a section of the page
<H3>	Level 3 header	Defines a sub-section of the page (should always follow an H2 in the logical hierarchy)
<H4>	Level 4 header	Etc. Less commonly used
<H5>	Level 5 header	Less commonly used. Only complex academic documents will break down to this level of detail.
<H6>	Level 6 header	Less commonly used
<HEAD>	Document head	Essential. Contains information about a page that does not constitute content to be communicated as part of the page.
<HR>	Horizontal rule	Display info with no semantic value - never use it. "Horizontal", by definition, is a visual attribute.

<b>&lt;HTML&gt;</b>		Core element of every web page.
<b>&lt;IMG &gt;</b>	Show an image	Vital. Always use the alt or longdesc attributes when the image has content value
<b>&lt;INPUT&gt;</b>	Input fields within forms	Vital. (I prefer to use <button> for buttons and submit buttons though)
<b>&lt;ISINDEX&gt;</b>	Old type of search input	Not really used any more. Use <form> instead.
<b>&lt;I&gt;</b>	Italicised text	Display info - never use it
<b>&lt;KBD&gt;</b>	Keyboard input	Display info - never use it
<b>&lt;LINK&gt;</b>	Defines a relationship to another document	Commonly used to reference external stylesheets, but has other minor uses.
<b>&lt;LI&gt;</b>	List item	Specifies an item in an unordered or ordered list (<ul> or <ol>)
<b>&lt;MAP&gt;</b>	Client-side image map	May have occasional value, but only use when absolutely necessary
<b>&lt;MARQUEE&gt;</b>	Makes text scroll across the screen	See <blink>
<b>&lt;MENU&gt;</b>	Menu item list	Deprecated. Do not use. Use other standard list types instead.
<b>&lt;META&gt;</b>	Meta-information	Useful way to insert relevant information into the <head> section of the page that does not need to be displayed.
<b>&lt;OL&gt;</b>	Ordered list	Type of list where the order of elements has some meaning. Generally rendered with item numbers (best managed with CSS).

<b>&lt;OPTION&gt;</b>	Selection list option	Vital for options within a drop-down control.
<b>&lt;PARAM&gt;</b>	Parameter for Java applet	Used in conjunction with an <b>&lt;object&gt;</b> or <b>&lt;applet&gt;</b> tag to pass additional setting information at runtime.
<b>&lt;PRE&gt;</b>	Preformatted text	Renders text in a pre-formatted style, preserving line breaks and all spaces present in the source. May be useful. ( <i>This one's a paradox, as it is strictly display info that applies only to visual browsing, but it's still so commonly used and useful that I'm hesitant to advise against using it.</i> )
<b>&lt;P&gt;</b>	Paragraph	Only use to denote a paragraph of text. Never use for spacing alone.
<b>&lt;Q&gt;</b>	Short quotation	Use for inline quotations (whereas <b>&lt;blockquote&gt;</b> should be used for quotations of a paragraph or more). Often used in conjunction with <b>&lt;cite&gt;</b> to cite the quotation's source.
<b>&lt;SAMP&gt;</b>	Denotes sample output text	Similar to the <b>&lt;code&gt;</b> tag. Rarely used & void.
<b>&lt;SCRIPT&gt;</b>	Inline script (e.g. JavaScript)	It's better to have all scripts as separate files than to write inline or in the <b>&lt;head&gt;</b> section, however still has its uses.
<b>&lt;SELECT&gt;</b>	Selection list	A drop-down selector for a form.

<b>SMALL</b>	<b>Smaller text</b>	<b>Display info - never use it</b>
<b>&lt;SPAN&gt;</b>	An inline span within text	Use to apply meaning (and style) to a span of text that goes with the flow of content (whereas a<div> tag is block-level and breaks the flow)
<b>&lt;Strikeout&gt;</b>		Display info - never use it
<b>&lt;STRONG&gt;</b>	Strong emphasis	Use this instead of the old <b>tag.
<b>&lt;STYLE&gt;</b>	CSS style settings	Normally used in <head> section of a page. Try to use external stylesheets, to enable you to apply different styles for different output media.
<b>&lt;SUB&gt;</b>	Subscript text	Arguably display info - recommend using alternative tags (e.g. <cite>). May be required in some academic uses e.g. Chemical formulas.
<b>&lt;SUP&gt;</b>	Superscript text	
<b>&lt;TABLE&gt;</b>	Table	Use for repeated data that has a naturally tabular form. Never use for layout purposes.
<b>&lt;TD&gt;</b>	Table data cell	A cell containing actual data. If a cell actually contains a descriptor or identifier for a row or column, use a <th> (table header) tag, not a <td>. This usually applies to column headers (within a<thead>), column footers (within a <tfoot>), as well as row headers (usually the first cell in a row in the <tbody>).
<b>&lt;TEXTAREA&gt;</b>	Multi-line text input area in a form	Essential

<b>&lt;TH&gt;</b>	Table column or row header cell	May appear in a <b>&lt;thead&gt;</b> (to denote a column header cell), <b>&lt;tbody&gt;</b> (to denote a row header), and in <b>&lt;tfoot&gt;</b> (to denote a column foot cell, e.g. a total)
<b>&lt;TBODY&gt;</b>	Indicates the main body of a data table	<p>It is always worth using this tag, as well as using <b>&lt;thead&gt;</b> and <b>&lt;tfoot&gt;</b> where appropriate.</p> <p>Note that it is permissible to have more than one <b>&lt;tbody&gt;</b>, <b>&lt;thead&gt;</b>, and <b>&lt;tfoot&gt;</b> in the same table.</p>
<b>&lt;THEAD&gt;</b>	The head section of a table	The place to put column header cells ( <b>&lt;th&gt;</b> )
<b>&lt;TFOOT&gt;</b>	The foot section of a table	Good place to put e.g. summary data, such as totals. Note that it goes before the <b>&lt;tbody&gt;</b> tag!
<b>&lt;TITLE&gt;</b>	Document title	Essential
<b>&lt;TR&gt;</b>	Table row	Essential with tables
<b>&lt;TT&gt;</b>	"Teletype" - simulates typewriter output	Similar to <b>&lt;pre&gt;</b> , except that it collapses white space like normal HTML (whereas <b>&lt;pre&gt;</b> leaves all consecutive white space intact). Avoid if possible
<b>&lt;UL&gt;</b>	Unordered list	Essential. Use for lists where the order of items has no particular importance.
<b>&lt;U&gt;</b>	Underline text	Display info - never used
<b>&lt;VAR&gt;</b>	Variable in computer code	Obscure tag - may only be useful in academic documents. Avoid

## 24. THE DOUBLING CODE

The Dublin Core metadata Element Set (DCMES) allows much descriptive information about resources to be expressed. However, in some applications it is desirable to refine the meanings of the DCMES metadata. A method for refining DCMES is encompassed in an extended model known as Qualified Dublin Core metadata [qDC], which requires additional labels and data, known generically as qualifiers.

DCMES metadata may be recorded in many ways. These include, but are not restricted to, tables, database systems, and serializations in XML [DCMES-XML] and HTML [DCMES-HTML]. Though popular, HTML is syntactically limited, particularly for recording more complex information models. Nevertheless, by using suitable conventions most of the requirements of the Qualified DC model may be covered. In this recommendation we describe the methods provided directly by HTML, and explain how to record Qualified DCMES metadata in HTML using the `<meta>` element in the document `<head>`.

### HTML notation

Two elements from the `<head>` of a HTML document are used in recording metadata: `<link>` and `<meta>`. All data must be contained within the values of the attributes of these elements.

#### `<link>` element

A HTML `<link>` element allows a relationship with another document to be recorded. The HTML specification [HTML] defines the attributes of `<link>` elements, of which the following are useful to us here:

`href %URI; (CDATA)` identifies the related resource

`rel %linktypes; (CDATA)` type of link forward from this document to the related resource

`rev %linktypes; (CDATA)` type of link back from this document to the related resource

Usually only one of rel and rev will apply to a single link. We use `<link>` to indicate the location of schemas or definitions of the terms used elsewhere in the document, and then apply a shorthand notation similar to XML namespaces [XML names]. The href attribute records the location of the schema, and the value of the rel attribute indicates that the link is to a schema, and establishes a prefix which can be used with terms elsewhere in the document as in the following example :

`<link rel="schema:DC" href="http://purl.org/dc/elements/1.1/">`

which ties the prefix DC to the schema located at <http://purl.org/dc/elements/1.1/>.

The HTML specification [HTML] describes another method for indicating the schema location, using the profile attribute of the `<head>` element. That method effectively generates a default namespace for the terms used, without requiring a prefix. While the profile method may be used in many instances, care must be taken when mixing terms from multiple schemas.

### `<meta>` element

HTML `<meta>` elements allow a simple list of metadata to be recorded. The HTML specification [HTML] defines the attributes of `<meta>` elements, of which the following are useful to us here:

name	name	the metadata element label
content	CDATA	the metadata value
schema	CDATA	indicates an encoding scheme used for the value
lang	%LanguageCode	indicates the natural language
dir	LTR   RTL	indicates the text direction of the value

### Metadata Elements

The DCMES element-name and value are recorded in the name and content attributes respectively [DCMES-HTML], in the following pattern:

`<meta name="DC.Element" content="Value">`

where Element is one of the 15 DCMES Elements and Value is the value of this element for the resource of interest. The prefix to the element name "DC," refers to the schema indicated in a `<link>` element in the same document, as described above.

In order to clearly record the text and labels used for qualification of DC [qDC] additional positions in the notation are required. For qualified DC we use different positions for qualifiers corresponding to the different classes of qualification.

### Element Refinements

Element Refinements are not supported directly in HTML `<meta>` elements.

## Introduction to Internet and World Wide Web

To refine the meaning of an element, an Element Refinement may be appended to the DCMES element name separated by a dot (.), and stored as the name attribute:

```
<meta name="DC.Element.ER" content="Value">
```

where ER is an element refinement.

This follows much existing practice [DCMES-HTML].

### Value Encoding Schemes

Value Encoding Schemes are supported directly in HTML <meta> elements, using the attributes scheme and lang.

lang is used in cases where the value is in plain-text, and scheme otherwise:

```
<meta name="DC.Element" scheme="schemeA" content="Value coded according to schemeA">
```

```
<meta name="DC.Element" scheme="listB" content="Value selected from listB">
```

```
<meta name="DC.Element" lang="langC" content="Value expressed in language langC">
```

where :

- schemeA is an encoding scheme,
- listB is a controlled vocabulary,
- langC is a language-code,

Specimen COPY  
Not For Sale  
Arun Kr. Singh

If a scheme or lang is specified, then the value in the content must be encoded according to that scheme, including the use of any structure and punctuation.

### 12.3 qDC-HTML

The complete syntax for expressing qualified DC elements in HTML may be summarised :

```
<link  
rel="schema.DC"  
href="http://dublincore.org/qdcmes/1.0/"  
title="DCMES plus DCMI recommended qualifiers">  
<meta name="DC.Element" content="Unqualified value">  
<meta name="DC.Element.ER" scheme="schemeA" content="Value coded according to schemeA">
```

```

<meta name="DC.Element.ER" scheme="listB" content="Value
selected from listB">
<meta name="DC.Element.ER" lang="langC" content="Value
expressed in language langC">

```

where the codewords are :

- Element is one of the 15 DCMES Elements,
- ER represents an Element Refinement,
- schemeA is an encoding scheme,
- listB is a controlled vocabulary,
- langC is a language-code,

In actual instances of DC metadata each of these codewords is replaced by tokens or strings defined in a qDC registry, indicating conformant qualifiers (see examples below).

### qDC-XHTML

XHTML is a reformulation of HTML using XML. The XHTML recommendation [XHTML] describes a number of changes that are necessary to make documents valid XHTML. Two of these concern us here:

1. all attributes must be quoted
2. empty elements, such as <meta> and <link>, must be properly closed with a "/" before the closing >.

Qualified DCMES metadata [qDC] can be recorded using the <meta> element in XHTML, as follows:

```

<link
    rel="schema.DC"
    href="http://dublincore.org/qdcmes/1.0/"
    title="DCMES plus DCMI recommended qualifiers" />
<meta name="DC.Element" content="Unqualified value" />
<meta name="DC.Element.ER" scheme="schemeA"
    content="Value coded according to schemeA" />
<meta name="DC.Element.ER" scheme="listB" content="Value
selected from listB" />
<meta name="DC.Element.ER" lang="langC" content="Value
expressed in language langC" />

```

The space before the "/" is not strictly necessary for XHTML, but is recommended since it allows most HTML clients to treat the XHTML document correctly, thus allowing a single XHTML document to be used for both cases.

### "Dumb-down" - recovering unqualified DC metadata

A client system may be unable to process qDC metadata presented to it for several reasons, particularly:

1. It is only configured to support the basic 15 DCMES.
2. A particular qualifier is encountered that is not supported by client.

In such cases it is necessary to consider how the information degrades to a simpler form.

For qDC metadata recorded in HTML according to the method described here a simple rule may be applied: discard any qualifiers that are not understood.

For encoding schemes the result is straightforward. While the full meaning of an encoded value requires that the client understands the notation, a client system may still process the value found in the content while ignoring the scheme or Lang attribute. Any notation based on character-strings will not inconvenience existing software.

Furthermore, parsing of a value may not even be necessary for resource discovery. The string-matching methods used in most search operations should still find the target text-strings from within extended values. Systems will harvest a text value into an index, regardless of notation, and the resource may still be located by users who will often have a knowledge of specialized notations, independent of the indexing software, and thus will be able to construct sensible and successful queries.

For refined elements the unqualified ("dumb") version is recovered by removing the part of the name following the DCMES element name. This requires more sophistication on the part of client software. Nevertheless, as the hierarchical dot (.) notation includes the DCMES element name earlier within the token, the dumb element should always be clear.

### Which schema?

It is conventional, but not mandatory, for the prefix to use the character string "DC" when recording DC metadata, although any other

XML Sticks for Extensible Markup Language. If anything will support HTML as a part of the Web, the most commonly candidate at the moment seems to be XML. It is another specific app. of SGML (Std. Generalized Markup Lang.) that allows you to create own document type to define new elements, tag, & attributes for own purpose. XML was conceived as means of retaining power & flexibility of SGML without loss of complexity.

- 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.

## 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 éóá are perfectly legal in XML, but watch out for problems if your software vendor doesn't support them.

## XML Elements are Extensible

XML elements can be extended to carry more information.

Look at the following XML example:

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

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

To: Tove  
From: Jani

Don't forget me this weekend!

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

```
<note>
<date>2008-01-10</date>
<to>Tove</to>
<from>Jan</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

Should the application break or crash?

No. 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.

## 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 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

## If Developers Have Sense

If they DO have sense, future applications will exchange their data in XML.

The future might give us word processors, spreadsheet applications and databases that can read each other's data in XML format, without any conversion utilities in between.

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

## An Example XML Document

XML documents use a self-describing and simple syntax:

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

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

The 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 next line describes the **root element** of the document (like saying: "this document is a note"):

```
<note>
```

The next 4 lines describe 4 **child elements** of the root (to, from, heading, and body):

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

And finally the last line defines the end of the root element:

```
</note>
```

You can assume, from this example, that the XML document contains a note to Tove from Jani.

Don't you agree that XML is pretty self-descriptive?

## 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.

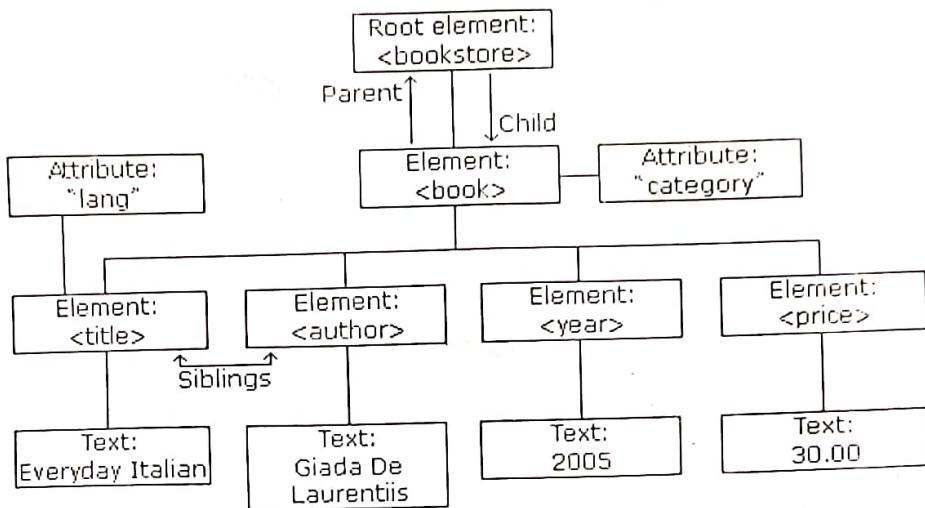
All elements can have sub elements (child elements):

```
<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>.

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

In HTML, some elements do not have to have a closing tag:

<p>This is a paragraph.<br/></p>

In XML, it is illegal to omit the closing tag. All elements **must** have a closing tag:

```
<p>This is a paragraph.</p>
<br />
```

**Note:** You might have noticed from the previous example that the XML declaration did not have a closing tag. This is not an error. The declaration is not a part of the XML document itself, and it has no closing tag.

## 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.

## XML Elements Must be Properly Nested

In HTML, you might see improperly nested elements:

```
<b><i>This text is bold and italic</b></i>
```

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

```
<b><i>This text is bold and italic</i></b>
```

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

## 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>
```

## XML Attribute Values Must be Quoted

XML elements can have attributes in name/value pairs just like 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=12/11/2007>
  <to>Tove</to>
  <from>Jani</from>
</note>
```

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

The error in the first document is that the date attribute in the note element is not quoted.  
The parser interprets it as the start of a new element.

## 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>
```

There are 5 predefined entity references in XML:

&lt;	<	less than
&gt;	>	greater than
&amp;	&	ampersand
&apos;	'	apostrophe
&quot;	"	quotation mark

**Note:** Only the characters "<" and "&" are strictly illegal in XML. The greater than character is legal, but it is a good habit to replace it.

## Comments in XML

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

```
<!-- This is a comment -->
```

## White-space is Preserved in XML

HTML truncates multiple white-space characters to one single white-space:

HTML:	Hello	Tove
Output:	Hello Tove	

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

## 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.

## What is an XML Element?

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

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>
```

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.

## XML Naming Rules

XML elements must follow these naming rules:

XHTML document root

An XHTML file must have "html" as the document root. The document root must have the XHTML namespace declared as shown in here:

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

The document root must contain only two elements <head> & <body>

XHTML Example

Attached

XHTML elements

1. XHTML document must be well-formed. The XHTML elements must be completely nested. One inside of another. Overlapping the element tags is forbidden.  
eg. <b><i> text </b> </i> is incorrect  
<b> <i> text </i> </b> is correct.
2. XHTML elements must be coded in lower case.
3. All non-empty XHTML elements must have an opening tag, an ending tag

4. All empty elements must conform to one of the following syntaxes:

<area>  
<base />  
<base> </base>

### XHTML attributes

1) Attribute names must be coded in lower case.

2) The values of the enumerated attributes (having predefined values) must be in lower case.

<input type="text" value="abc" />

3) Attribute values must be quoted using single (' ) or double (" ) quotation marks.

4) Line breaks & multiple white space characters should be avoided within attribute values.

5. Attribute minimization is forbidden. eg  
checked = "checked" for <input> elements.

<input type="checkbox" value="Yes" checked="checked"

6. In HTML 4.01, name attribute is used to add an identifier to <a>, <applet>, <frame>, <iframe> <image> & <map> elements. W3C recommends using the id attribute in favor of the name attribute.

7. If you use the lang attribute for an element you must also add the xml:lang attribute.  
<div lang="en" xml:lang="en"> This is XHTML

XHTML 1.0 Transitional : In this case, you can use the deprecated elements but not the obsolete ones. Syntax is

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

3. XHTML 1.0 Frameset : In this case, you can use <frameset> & <frame> elements. Syntax is

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

\* This doctype declaration must be placed before "html" root element of XHTML file.

### 1) XML declaration

To declare an XHTML file as an XML file, the first line should contain an XML declaration having syntax

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

It is recommended but not required.

### 3) Metadata

If XML declaration in your XHTML file, it is recommended that you include a <meta> tag declaring the charset (character set) used by the document.

D) XHTML : XHTML (Extensible Hyper Text Markup Language).  
XHTML is the new generation of HTML, which conforms to the XML technology. XHTML has facilities inherited from both these markup languages.

1. Like HTML, XHTML can be used to create web pages.
2. Like XML, XHTML can be used with all XML applications. The code is more strict.

To view XHTML document, Browsers must be XHTML compatible.

XHTML has some syntax rules as explained below :

#### 1. DOCTYPE declaration :

XHTML files must contain a DOCTYPE declaration. There are 3 possibilities for this required declaration.

(i) XHTML 1.0 Strict : In this case, you can not use deprecated HTML 4.01 elements (`<applet>`, `<basefont>`, `<center>`, `<dir>`, `<font>`, `<isindex>`, `<menu>`, `<s>`, `<strike>`, & `<u>`) nor obsolete HTML 4.01 elements (`<listing>`, `<plaintext>`, and `<xmp>`). The syntax is :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML  
1.0 Strict//EN" "http://www.w3.org/TR/xhtml  
1/DTD/xhtml1-strict.dtd">
```

## Advantages:

- (1) Sustainability: Web application tend towards XML. Using XHTML now instead of HTML makes any future conversion of the website easier.
- (2) Extensibility: XHTML documents can be supplemented with other forms of markup, MathML, SVG thanks to the use of namespaces.
- (3) Compatibility: XHTML documents are written in compliance with the rules of XML, XML processing programmes can effortlessly convert an XHTML file to another format (PDF, RSS, RTF)
- (4) Efficiency of processing applications: Once Browsers support XHTML documents and the strict rules of XML, they will become quicker thanks to processing routines.

## Disadvantages:

At present, there are hardly any browsers that offer benefits when receiving & processing XHTML documents. Even IE does not support XHTML.

## Limited Practical Use