e-PG Pathshala

Subject: Computer Science

Paper: Web Technology

Module: XHTML, XML

Module No: CS/WT/7

Quadrant 1 – e-text

Learning Objectives

The last module explains about Hyperlinks in HTML, Frames, how to define Frames and the purpose of the Frameset document. Moreover, the module also explains about HTML Images . In this module we will learn about XHTML, its syntax, tags, and document type definitions. We will learn how to use XHTML to create Web pages. In this module we will also understand the basics of creating an XML document

XHTML – eXtensible Hyper Text Markup Language

XHTML is HTML defined as an XML application. It is identical to HTML but it is a stricter and cleaner HTML. It is compatible to HTML 4.0.1 and supported by all browsers. XHTML is used to define and organize the page content but not to format or style it.

XHTML uses the elements and attributes of HTML. Its uses the syntax of XML (e**X**tensible **M**arkup **L**anguage).

Need for XHTML

XHTML combines the strength of HTML and XML. XHTML provides the web page with a more consistent and well-structured format so that the web pages can be easily parsed and processed by present and future web browsers. Also, XHTML pages can be rendered by all XML enabled browsers.

XHTML integrates the concepts of XML with HTML and hence it is relatively easy to introduce new elements or additional element attributes. It also conforms to the rules and standards of XML.

The following, “bad” HTML document will work fine in most browser even if it does not follow HTML rules:

```
<html>
<head>
<body>
<p>a paragraph...<br>
<a href=“#”>test
</html>
```

But browsers running on hand-held devices (e.g. mobile phones) have small computing power and cannot interpret “bad” markup language.

The difference between HTML and XML is that HTML is designed to structure (and display) data and XML is designed to describe and structure data. XHTML specifies that everything must be marked up correctly

XHTML – Base syntactic rules

The basic syntactic rules to be followed when creating XHTML files are given as below,

- XHTML elements must be properly nested

  ```
  <b><i> Italic and bold text </b></i>
  <b><i> Italic and bold text </i></b>
  ```

- XHTML elements must always be closed. For every element there should be a opening tag and a closed tag. The <p> tag,
 tag, <img> tag doesn't have a closing tag in HTML whereas in XHTML there is a closing tag for every element. For example

  ```
  <p> A paragraph...
  <br>
  <img src=“foo.jpg”>

  <p> A paragraph...</p>
  <br />
  <img src=“foo.jpg” />
  ```

- XHTML elements must be in lowercase
- XHTML elements must have one <html> root element (which contains a <head> and a <body>)

XHTML – other syntactic rules

The other syntactic rules to be followed is that,

- The attribute names must be in lower case

- The attribute values must be enclosed in double quotes

```
<table width=300px>
<table width="300px">
```

- The "id" attribute replaces the "name" attribute
- XHTML DTD defines mandatory elements
- Attribute minimization is forbidden in XHTML. The values has to be specified with attributes.

```
<input checked>
<input disabled>
<input checked="checked" />
<input disabled="disabled" />
```

General format of an XHTML document

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

The <!Doctype>,<html>,<head>,<title>,<body> elements are mandatory in an XHTML file.

DTD – Document Type Definition

A DTD specifies the syntax of a document written in a Standard Generalized Markup Language (SGML) such as HTML, XHTML and XML. It specifies the hierarchical structure of the document, element names and types, element content type and attribute names and values.

XML 1.0 defines three DTDs such as Strict, Transitional and Frameset DTD.

DOCTYPE must be specified on the first line when a DTD is defined.

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

The DOCTYPE tells the web browser which language is being used for the set of instructions that follow. In this module, we will be using XHTML Transitional, which allows us more flexibility than XHTML Strict.

W3C has recommended the use of a Document Type Definition to identify the type of markup language used in a web page.

XHTML 1.0 Transitional . This is the least strict specification for XHTML 1.0. It allows the use of both Cascading Style Sheets and traditional formatting instructions such as fonts.

XHTML 1.0 Strict . This requires exclusive use of Cascading Style Sheets.

XHTML 1.0 Frameset. This standard is required for pages using XHTML frames.

DTD example (internal to XHTML file)

```
<!DOCTYPE course [
<!ELEMENT course (lecture+)>
<!ELEMENT lecture (title,bibliography,notes,examples)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT bibliography (#PCDATA)>
<!ELEMENT notes (#PCDATA)>
<!ELEMENT examples (#PCDATA)>

<!ATTLIST course professor CDATA #REQUIRED>
<!ATTLIST course title CDATA #REQUIRED>
<!ATTLIST course yearofstudy CDATA #REQUIRED>
<!ATTLIST course date CDATA #IMPLIED>
]>
```

We will study more about DTD in the next module.

XHTML validation

A valid XHTML document is an XHTML document which obeys the rules of the DTD specified by the <!Doctype> tag. The official W3C XHTML validator is,

http://validator.w3.org/check/referer

XML

XML stands for eXtensible Markup Language. It is a markup language is used to provide information about a document. XML is a meta markup language for text documents / textual data. Tags are added to the document to provide the extra information.

The basic difference between HTML and XML is, HTML tags tell a browser how to display the document whereas XML tags give a reader some idea what some of the data means.

What is XML Used For?

XML documents are used to transfer data from one place to another often over the Internet. XML is text (Unicode) based. It takes up less space and can be transmitted efficiently.

One XML document can be displayed differently in different media such as HTML, video, CD or DVD. We only have to change the XML document in order to change all the rest. Moreover, XML documents can be modularized so that the parts can be reused.

An example of an XML Document is given here,

```
<?xml version="1.0"/>
<address>
    <name>ABCD</name>
    <email>abcd@annauniv.edu</email>
    <phone>044-2235-1234</phone>
    <birthday>1985-03-22</birthday>
</address>
```

All information in an XML file has markup for the data which aids in understanding its purpose. The XML language is very expressive that means semantics comes along with the data. It is well structured, easy to read and write from programs.

Difference Between HTML and XML

- HTML tags have a fixed meaning and browsers know what it is.
- XML tags are different for different applications, and users know what they mean.
- HTML tags are used for display.
- XML tags are used to describe documents and data.

XML Rules

Following are the rules to be followed when creating an XML file,

- Tags are enclosed in angle brackets.
- Tags come in pairs with start-tags and end-tags.
- Tags must be properly nested.
  - <name><email>...</name></email> is not allowed.
  - <name><email>...</email><name> is.
- Tags that do not have end-tags must be terminated by a '/'.
  -
 is an example.

More XML Rules

- Tags are case sensitive.
  - <address> is not the same as <Address>
- XML in any combination of cases is not allowed as part of a tag.
- Tags may not contain '<' or '&'.
- Tags follow JAVA naming conventions, except that a single colon and other characters are allowed. They must begin with a letter and may not contain white space.
- Documents must have a single *root* tag that begins the document.

Encoding

XML (like Java) uses Unicode to encode characters. Unicode comes in many flavors. The most common one used in the West is UTF-8. UTF-8 is a variable length code where the characters are encoded in 1 byte, 2 bytes, or 4 bytes.

Well-Formed Documents

An XML document is said to be well-formed if it follows all the rules. An XML parser is used to check that all the rules have been obeyed. Recent browsers such as Internet Explorer 5 and Netscape 7 come with XML parsers. Parsers are also available for free download over the Internet. Java 1.4 also supports an open-source parser.

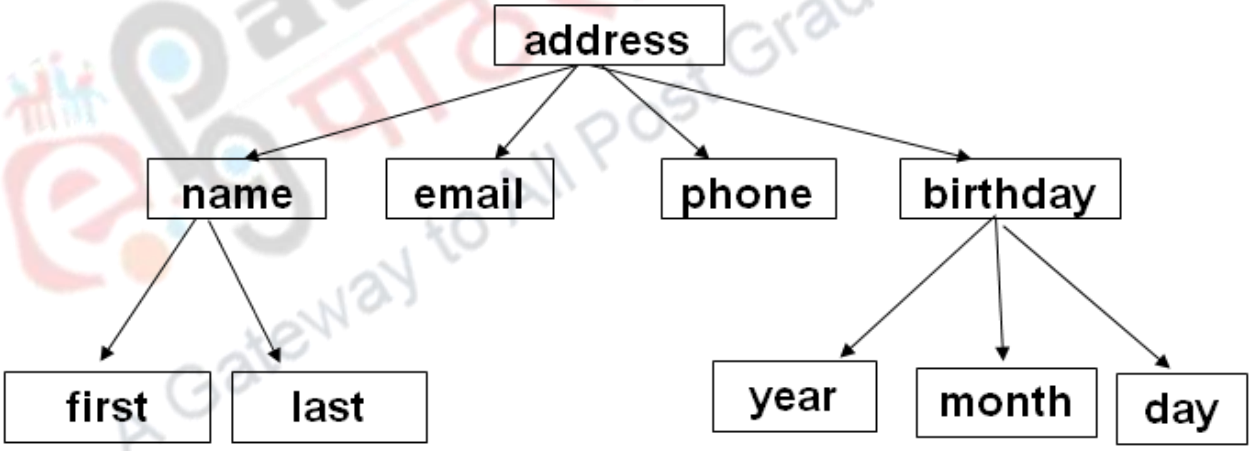
Example

This is an example of an XML file,

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

```
<address>
        <name>
            <first>ABC</first>
            <last>XYZ</last>
    </name>
    <email>abc@annauniv.edu</email>
    <phone>044-2235-6789</phone>
    <birthday>
            <year>1976</year>
            <month>09</month>
    <day>26</day>
        </birthday>
</address>
```

In XML files, the data are represented with tags in such a way that they form a tree with a root. The above XML file can be viewed as a tree which is shown below.



The root element of the XML file is <address>. The <address> element has four children <name>, <email>, <phone> and <birthday>. The <name> element has two children as <first> and <last>. The <birthday> element has three children as <year>, <month> and <day>. Hence any XML file can be represented as a tree.

XML Trees

An XML document has a single root node. The tree is a general ordered tree. A parent node may have any number of children. Child nodes are ordered, and may have siblings. Preorder traversals are usually used for getting information out of the tree.

XML Documents

An XML document consists of Elements, Attributes plus some other details such as textual information, namespaces, processing instruction and soon.

A Simple XML Document

```
<article>
  <author>ABC</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

Elements in XML Documents

In XML all tags are user definable/ freely definable. In the above code we have defined tags: article, title, author. All tags start with a start tag: **<article>** etc. and end with a end tag: **</article>** etc.

Elements: `<article> ... </article>`

Elements can have a name (article) and a content (...). Elements may be nested.

Elements may be empty:

```
<this_is_empty/>
```

Element content is typically a parsed character data (PCDATA), i.e., strings with special characters, and/or nested elements (*mixed content* if both).

Each XML document has exactly one root element and forms a tree. Elements with a common parent are ordered.

Elements vs. Attributes

Elements may have attributes (in the start tag) that have a **name** and a **value**,

For example the element section has a attribute 'number' whose value is '1'.

<section number="1">.

The difference between elements and attributes are that only one attribute with a given name per element can be defined but an arbitrary number of subelements can be defined for an element. Moreover, attributes have no structure, they are simply strings while elements can have subelements.

An example is given below,

<person born="1912-06-23" died="1954-06-07"> Alan Turing</person> proved that...

Namespaces

Namespaces are generally provided to avoid element name conflicts. Name conflicts in XML can easily be avoided using a name prefix.

Namespaces in XML specification defines syntax for qualifying element or attribute names with a namespace identifier. Element/attribute names can be qualified with a namespace prefix as shown below,

(QName = prefix:local_name)

A namespace prefix is an abbreviation for a namespace identifier (URI). Namespace prefixes are mapped to namespace identifiers through namespace declarations.

(xmlns:prefix='namespace identifier')

Namespace declarations are placed within element start tags just like attributes.

Namespace Example

```
<d:student
  xmlns:d = 'http://www.develop.com/student' xmlns:i='urn:schemas/develop.
  com:identifiers'   xmlns:p = 'urn:schemas/develop.com:programming/languages'>
  <i:id>3235329</i:id>
  <name>XXX</name>
  <p:language>C#</p:language>
  <d:rating>9.5</d:rating>
</d:student>
```

When a namespace is defined for an element, all child elements with the same prefix are associated with the same namespace.

Default Namespace

Default namespace may be set for an element and its content but *not* for its attributes. For example,

```
<book xmlns = "http://www-dbs/dbs">
  <description>...</description>
<book>
```

The default namespace can be overridden in the elements by specifying the namespace there using prefix or default namespace.

```
<d:student xmlns:d='http://www.develop.com/student' xmlns='urn:foo' id='3235329'>
  <name>XYZ</name>
  <language xmlns=''>C#</language>
  <rating>35</rating>
</d:student>
```

Defining XML Data Formats

A well-formed document has a tree structure and obeys all the XML rules. A particular application may add more rules in either a DTD (document type definition) or in an XML schema. Well-formed document is defined as the document that adheres to the XML syntax rules.

Valid document is defined as the document that adheres to the rules defined in the corresponding DTD document. Only the valid documents are valuable in terms of sharing and retrieving information. Hence every XML file adheres to Document Type Definitions or XML Schema.

Summary

This module provides an insight into XHTML. The module provides an introduction to XML, XML elements, attributes and namespaces. The module also provides an introduction to Document Type Definitions (DTD) which would be discussed in detail in the next module.