Web Programming

Module Code- K72T001M07

ICT in NVQ Level-05

Ministry of Youth Affairs & Skills Development

Vocational Training Authority of Sri lanka

NITC NATIONAL IT CENTER

Topic Web contents development - XML

- Understand & Explain the Outline of XML
- Write XML code

© 2013 National IT Center. All rights reserved

What is XML?

- eXtensible Markup Language
- A language for describing data
- A tool for building new data description languages
- Open to define in any way you want
- Not a program or a process or a function
  - It does not do anything
- A markup language. That means that “marks” or tags are used to identify the data elements. Data and tags are generally stored as plain text.

© 2014 VTA-National IT Centre. All rights reserved.

Expected use for XML

- “XML is going to be the main language for exchanging financial information between businesses over the Internet. A lot of interesting B2B applications are under development .”

© 2014 VTA-National IT Centre. All rights reserved.

XML uses

A Document Type Definition or an XML Schema to specify what tags are allowed or required.

A validation service to confirm that a given XML document is syntactically correct.

© 2014 VTA-National IT Centre. All rights reserved.

An XML file

- Easy rules, but very strict
- First line is the version and character set used:
  - <?xml version="1.0" encoding="ISO-8859-1"?>
- The rest is user defined tags
- Every tag has an opening and a closing

© 2014 VTA-National IT Centre. All rights reserved.

An example

Book Title: My First XML

Chapter 1: Introduction to XML

- What is HTML
- What is XML

Chapter 2: XML Syntax

- Elements must have a closing tag
- Elements must be properly nested

```
<book> <title>My First
    XML</title>
<prod id="33-657"
    media="paper"></prod>
<chapter>Introduction to XML
    <para>What is
    HTML</para>
<para>What is XML</para>
    </chapter>
<chapter>XML Syntax
    <para>Elements must have
    a closing tag</para>
<para>Elements must be
    properly nested</para>
</chapter>
</book>
```

© 2014 VTA-National IT Centre. All rights reserved.

XML Elements

Elements have Content

Elements can have different content types.

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

An element can have **element** content, **mixed** content, **simple** content, or **empty** content. An element can also have **attributes**.

In the example above, book has **element content**, because it contains other elements. Chapter has **mixed content** because it contains both text and other elements. Para has **simple content** (or **text content**) because it contains only text. Prod has **empty content**, because it carries no information.

In the example above only the prod element has **attributes**. The **attribute** named id has the **value** "33-657". The **attribute** named media has the **value** "paper".

© 2014 VTA-National IT Centre. All rights reserved.

Element naming

- **XML elements must follow these naming rules:**

- Names can contain letters, numbers, and other characters

- Names must not start with a number or punctuation character

- Names must not start with the letters xml (or XML or Xml ..)

- Names cannot contain spaces

© 2014 VTA-National IT Centre. All rights reserved.

Elements and Attributes

Example

<note date="12/11/99"> <to>Tove</to> <from>Jani</from>

<heading>Reminder</heading> <body>Don't forget me this

weekend!</body> </note>

© 2014 VTA-National IT Centre. All rights reserved.

Example

```
<note>
    <date>12/11/99</date>
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder
</heading> <body>Don't
    forget me this
    weekend!</body>

 </note>
```

```
<note>
<date> <day>12</day>
    <month>11</month>
    <year>99</year> </date>
 <to>Tove</to>
 <from>Jani</from>
    <heading>Reminder
    </heading>
<body>Don't forget me this
    weekend!</body>
 </note>
```

© 2014 VTA-National IT Centre. All rights reserved.

Attributes & Elements rule

- Use elements to describe data
- Use attributes to present information that is not part of the data
  - For example, the file type or some other information that would be useful in processing the data, but is not part of the data.

© 2014 VTA-National IT Centre. All rights reserved.

Defining the structure of the data

- A Document Type Definition, or an XML Schema defines the tags and their organization.
- XML files can be validated against the definition before a program tries to process the data.
  - Then you don't have to worry about accounting for all kinds of error conditions.

© 2014 VTA-National IT Centre. All rights reserved.

DTD

- Internal
  - Put the dtd right into the XML file

```
<?xml version="1.0"?>
<!DOCTYPE note [
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)> ]>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend</body> </note>
```

Example

- External

```
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM
   "note.dtd">
 <note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder
   </heading>
<body>Don't forget me this
   weekend!</body>
 </note>
```

Where the note.dtd file=

```
<!ELEMENT note
   (to,from,heading,body)> <!ELEMENT
   to (#PCDATA)> <!ELEMENT from
   (#PCDATA)> <!ELEMENT heading
   (#PCDATA)> <!ELEMENT body
   (#PCDATA)>
```

© 2014 VTA-National IT Centre. All rights reserved.

Why use a DTD?

- With DTD, each of your XML files can carry a description of its own format with it.
- With a DTD, independent groups of people can agree to use a common DTD for interchanging data.
- Your application can use a standard DTD to verify that the data you receive from the outside world is valid.
- You can also use a DTD to verify your own data

© 2014 VTA-National IT Centre. All rights reserved.

The building blocks of XML documents

- Elements
- Tags
- Attributes
- Entities
- PCDATA
- CDATA

© 2014 VTA-National IT Centre. All rights reserved.

XML document components

- Elements can contain text, other elements, or be empty. Examples of empty HTML elements are "hr", "br" and "img".
- Tags are used **to markup elements**.
- Attributes provide **extra information about elements**.
- Entities are variables used to **define common text**. Entity references are references to entities.
- PCDATA - Parsed Character Data
- CDATA - Text that will not be parsed (Think comment)

Syntax

Element with only character data:

<!ELEMENT element-name (#PCDATA)>

example:<!ELEMENT from (#PCDATA)>

Element with any data:

<!ELEMENT element-name ANY>

example:<!ELEMENT note ANY>

Elements with children

<!ELEMENT element-name (child-element-name)> or

<!ELEMENT element-name (child-element-name,child-element-name, ...)>

example: <!ELEMENT note (to,from,heading,body)>

Full declaration of note

<!ELEMENT note (to,from,heading,body)>

<!ELEMENT to (#PCDATA)>

<!ELEMENT from (#PCDATA)>

<!ELEMENT heading (#PCDATA)>

<!ELEMENT body (#PCDATA)>

© 2014 VTA-National IT Centre. All rights reserved.

Repeating elements

- Naming an element means it appears exactly once.
- Name+ means it appears one or more times
- Name* means it appears 0 or more times.
- Name? Means it appears 0 or one time.

© 2014 VTA-National IT Centre. All rights reserved.

Mixed content

Example

<!ELEMENT note (#PCDATA|to|from|header|message)*>

- The example above declares that the "note" element can contain parsed character data and any number of "to", "from", "header", and/or "message" elements.

© 2014 VTA-National IT Centre. All rights reserved.

XML organization

- Organization in structured logical way
  - XML- XML Schema
    - Standard syntax
    - Guarantee quality, correctness
    - Rich set of basic types help standardization
    - Abundance of XML parsers helps construction of analysis tools

© 2014 VTA-National IT Centre. All rights reserved.

- Make up an XML definition for some data you want to work with.
  - An address book, perhaps?
  - What would you put into an address book?
  - How would you define an address?
  - What would you do with an address book definition once you had it?

© 2014 VTA-National IT Centre. All rights reserved.

The purpose of XML

- Is to describe data
- Is to facilitate the cross platform sharing of information
- Is independent of any hardware, software, operating system, programming language…

© 2014 VTA-National IT Centre. All rights reserved.

Namespaces

- Conflicting names make consistent interpretation impossible
- XML identifies namespaces to make the usage clear.

© 2014 VTA-National IT Centre. All rights reserved.

Example

```
<table>
  <tr>
  <td>Apples</td>
  <td>Bananas</td>
  </tr>
</table>
```

```
<table>
  <name>African Coffee
   Table</name>
  <width>80</width>
  <length>120</length>
</table>
```

The two different uses of the tag “table” lead to conflicts.

© 2014 VTA-National IT Centre. All rights reserved.

Namespace use

```
<h:table xmlns: h="http://
    www.w3.org/TR/html4/">
   <h:tr>
   <h:td>Apples</h:td>
   <h:td>Bananas</h:td>
   </h:tr>
</h:table>
```

```
<f:table xmlns: f="http://
    www.w3schools.com/furniture">
   <f:name>African Coffee
    Table</f:name>
   <f:width>80</f:width>
   <f:length>120</f:length>
</f:table>
```

The use of a prefix (h or f here) clearly separates the two meanings of the tag "table."

Tag attribute xmlns (xml name space) identifies each prefix with a unique name. The link is not used by the parser to check the definition. The link may contain information about the namespace.

© 2014 VTA-National IT Centre. All rights reserved.

Default name space

- Using a default namespace in an element avoids the need to repeat the prefix on each child element.

```
<table
   xmlns="http://www.w3.org/TR/html4/">
  <tr>
  <td>Apples</td>
  <td>Bananas</td>
  </tr>
</table>
```

Here, all subelements of “table” have the name related to the namespace indicated in the xmlns attribute.

© 2014 VTA-National IT Centre. All rights reserved.

XML Schema

- Alternative to DTD
  - More data types
  - Syntax based on XML
  - Ability to reuse and redefine existing schema
- Defines the elements, attributes that can appear in a document -- and their data types
- Defines relationships (parent-child) and order and number of child elements
- Defines fixed and default values for elements and attributes

© 2014 VTA-National IT Centre. All rights reserved.

Schema use

- Use of a schema assures that the communicating entities have the same understanding of how data is represented and how it should be interpreted.
- Classic case: dates. Is 02-01-2006 February 1, 2006 or January 2, 2006?
  - Schema specifies how the data values are to be interpreted.

© 2014 VTA-National IT Centre. All rights reserved.

Syntax and semantics

- Syntax determines the rules for a well-formed statement.
- Semantics determines how a well-formed statement will be used.
- Use of a schema can help detect errors in use within a well-formed statement.
  - Ex: May include a range of legal values.

© 2014 VTA-National IT Centre. All rights reserved.

Schema example for “note”

```
<?xml version="1.0"?>
<xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">


<xs:element name="note">
   <xs:complexType>
    <xs:sequence>
    <xs:element name="to" type="xs:string"/>
    <xs:element name="from" type="xs:string"/>
    <xs:element name="heading" type="xs:string"/>
    <xs:element name="body" type="xs:string"/>
    </xs:sequence>
   </xs:complexType>
</xs:element>
</xs:schema>
```

© 2014 VTA-National IT Centre. All rights reserved.

Reference to a schema

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

<note
xmlns="http://www.w3schools.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3schools.com note.xsd">
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

© 2014 VTA-National IT Centre. All rights reserved.

Common simple data types

- xs:string
- xs:decimal
- xs:integer
- xs:boolean
- xs:date
- xs:time

Default or fixed values:

- <xs:element name="color" type="xs:string" default="red"/>
- <xs:element name="color" type="xs:string" fixed="red"/>

© 2014 VTA-National IT Centre. All rights reserved.

Attributes

All attributes are declared as simple types. Elements with attributes are considered complex type.

Syntax for defining an attribute:

```
<xs:attribute name="xxx"
  type="yyy"/>
```

Defining an attribute for an element:

```
<lastname
  lang="EN">Smith</lastname>
```

Corresponding attribute definition:

```
<xs:attribute name="lang"
  type="xs:string"/>
```

© 2014 VTA-National IT Centre. All rights reserved.

Keywords

- XML – **Extensible Markup Language**
- DTD - **Document Type Definition**

© 2014 VTA-National IT Centre. All rights reserved.

Summary

- How to work with XML tags

© 2014 VTA-National IT Centre. All rights reserved.

Questions

1. What are the advantage of using XML?

2. What are basic CSS types

© 2014 VTA-National IT Centre. All rights reserved.

Reference

- http://www.w3schools.com
- http://www.tizag.com

© 2014 VTA-National IT Centre. All rights reserved.