UNIVERSITY OF SRI JAYEWARDENEPURA

Faculty of Technology

Department of Information and Communication Technology

ITC 3013 - Service Oriented Architecture & Web Services

Lab Sheet 1 – XML Basics

1. What is XML?

XML (eXtensible Markup Language) is a text-based format used to organise and describe data in a structured way, making it easy to store and share information between different systems.

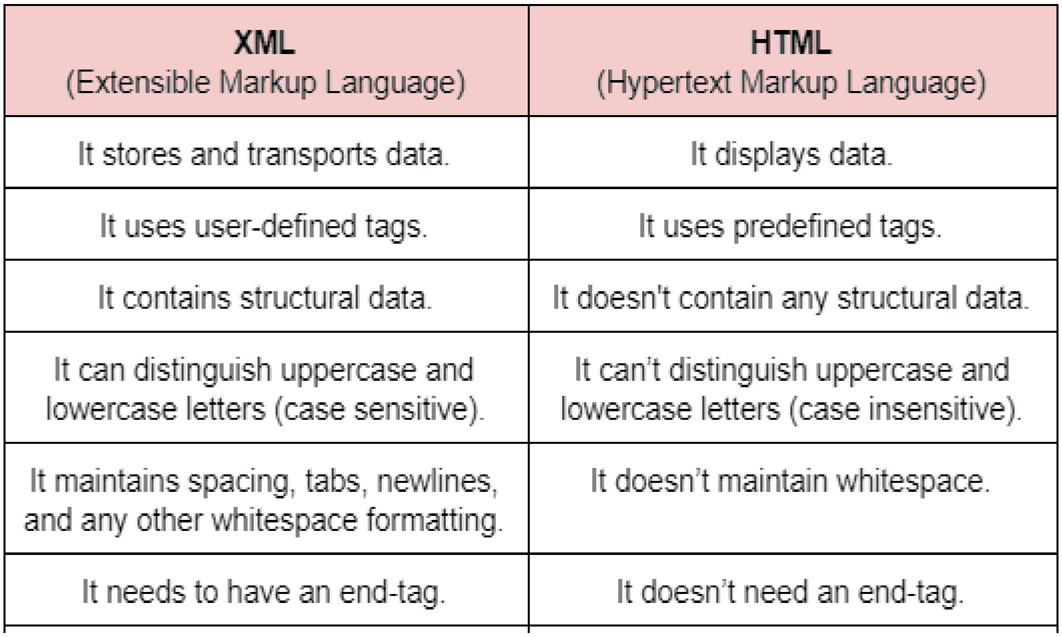
1.1 Comparison between XML and HTML:

Similarities:

- Both are markup languages.
- Both use tags to define elements and their

structure. Differences:

- XML is designed to store and transport data, while HTML is used for displaying data and creating web pages.
- XML focuses on describing the content, while HTML focuses on presenting the content.

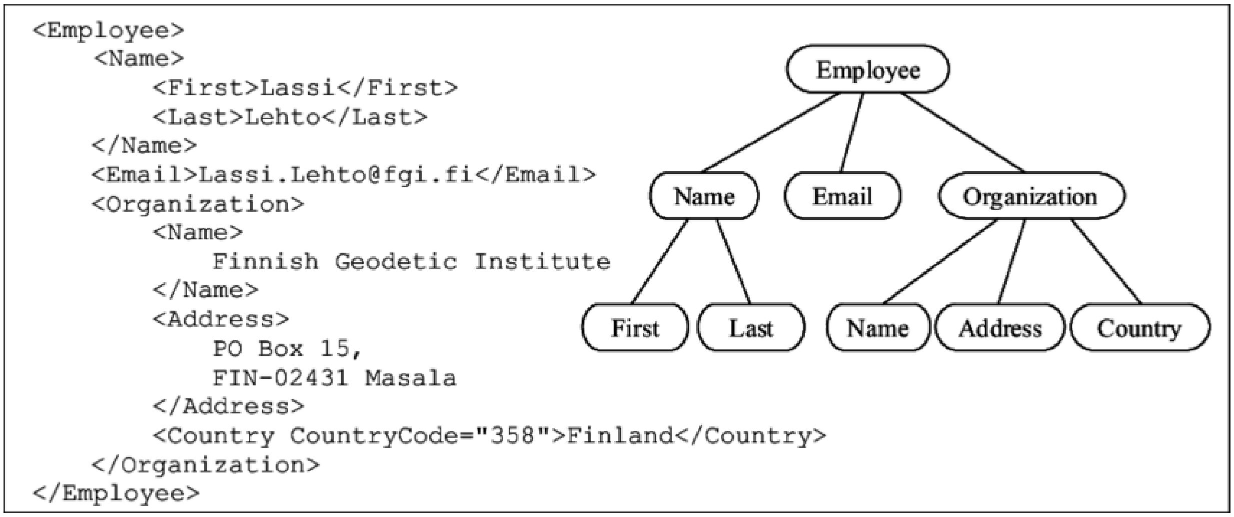


XML (Extensible Markup Language)	**HTML** (Hypertext Markup Language)
It stores and transports data.	It displays data.
It uses user-defined tags.	It uses predefined tags.
It contains structural data.	It doesn't contain any structural data.
It can distinguish uppercase and lowercase letters (case sensitive).	It can't distinguish uppercase and lowercase letters (case insensitive).
It maintains spacing, tabs, newlines, and any other whitespace formatting.	It doesn't maintain whitespace.
It needs to have an end-tag.	It doesn't need an end-tag.

2. How to write XML syntax and what are the rules?

2.1 Basic syntax rules of XML

- XML documents must have a single root element that contains all other elements.
- XML elements must have a start tag, content, and an end tag, except for self-closing elements.
- XML is case-sensitive.
- Attribute values must be enclosed in single or double quotes.
- XML comments start with <!-- and end with -->.



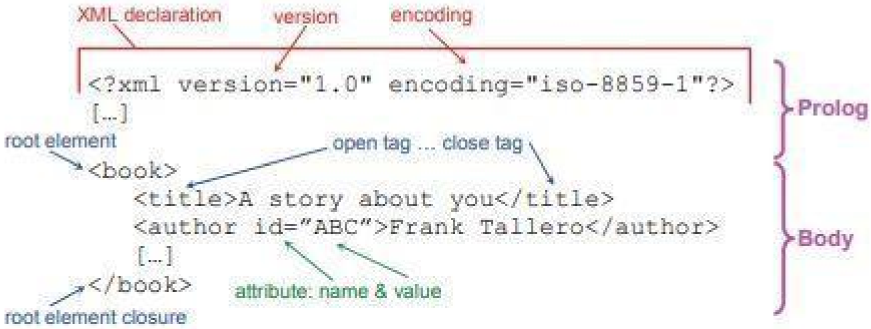
```
<Employee>
    <Name>
        <First>Lassi</First>
        <Last>Lehto</Last>
    </Name>
    <Email>Lassi.Lehto@fgi.fi</Email>
    <Organization>
        <Name>
            Finnish Geodetic Institute
        </Name>
        <Address>
            PO Box 15,
            FIN-02431 Masala
        </Address>
        <Country CountryCode="358">Finland</Country>
    </Organization>
</Employee>
```

2.2 XML elements

- XML elements are defined as opening and closing tags, like <element>content</element>.
- The content can contain other elements or text data.

2.3 XML attributes

- XML attributes are additional pieces of information that can be added to elements.
- They are defined within the start tag, like <element attribute="value">content</element>.

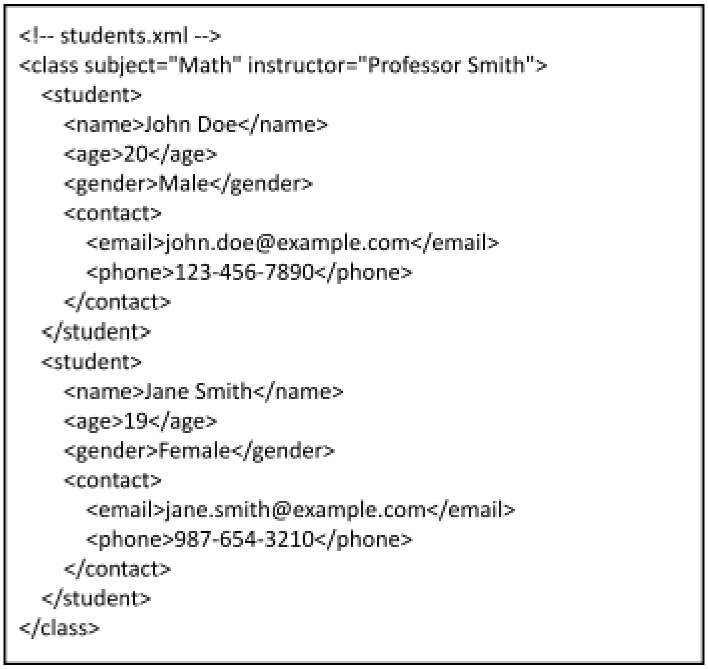


```
<?xml version="1.0" encoding="iso-8859-1"?>
[...]
<book>
   <title>A story about you</title>
   <author id=”ABC”>Frank Tallero</author>
   [...]
</book>
```

Papaleo, Laura. (2013). Introduction to XML and its applications. 10.1142/9789812836304_0006.

3. Create an XML document

In the text editor, create below XML document.



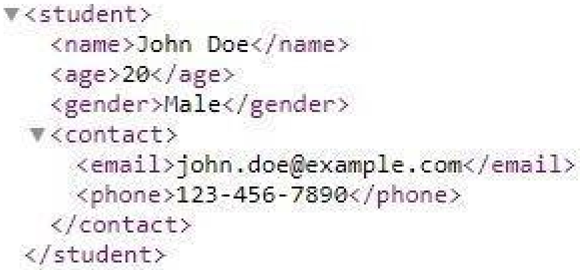
```
<!-- students.xml -->
<class subject="Math" instructor="Professor Smith">
  <student>
    <name>John Doe</name>
    <age>20</age>
    <gender>Male</gender>
    <contact>
      <email>john.doe@example.com</email>
      <phone>123-456-7890</phone>
    </contact>
  </student>
  <student>
    <name>Jane Smith</name>
    <age>19</age>
    <gender>Female</gender>
    <contact>
      <email>jane.smith@example.com</email>
      <phone>987-654-3210</phone>
    </contact>
  </student>
</class>
```

3.1 Nested Elements

In XML, a nested element refers to an element that is placed inside another element. The above students.xml file already includes the contact element nested within the student element.

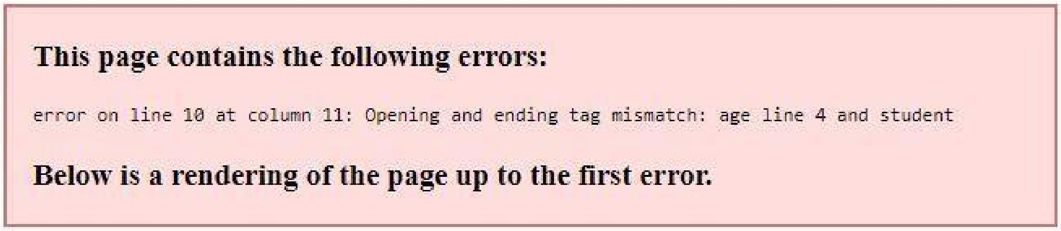
4. How to validate and test an XML document?

1. Save the students.xml file and open it with a web browser. The browser should display the contents of the XML file without any errors.



```
▼<student>
   <name>John Doe</name>
   <age>20</age>
   <gender>Male</gender>
  ▼<contact>
     <email>john.doe@example.com</email>
     <phone>123-456-7890</phone>
   </contact>
 </student>
```

2. To introduce an intentional error, modify the students.xml file by removing the closing tag for the age element in one of the student entries.
3. When attempting to open the modified XML file in the browser, an error message will indicate that the document is not well-formed, and the specific error location will be highlighted.



This page contains the following errors:

```
error on line 10 at column 11: Opening and ending tag mismatch: age line 4 and student
```

Below is a rendering of the page up to the first error.

5. Well-Formed vs. Valid XML

- ★ **Well-formed XML** means that the XML document follows the basic syntax rules of XML, such as having a single root element, properly nested elements, and correctly closed tags.
- ★ **Valid XML** means that the XML document not only follows the basic syntax rules but also adheres to a defined structure and rules specified in a Document Type Definition (DTD) or an XML Schema (XSD).

Exercise 1
Create an XML structure to represent a person's profile with attributes like name, age, and email.

Exercise 2
Build an XML structure for a bookstore that includes details about books such as title, author, genre, and price. (Ensure to include details for at least two books).

Exercise 3
Build an XML structure to store information about employees in a company, including their ID, name, department, and salary. (Ensure to include details for at least two employees)

Exercise 04

Create an XML structure to represent a menu of a restaurant, including categories like appetizers, main courses, and desserts. (Ensure to include details for at least three menus)

Exercise 05

You are tasked with creating an XML document to represent a library catalog. The catalog should include information about various books available in the library, such as the title, authors, publication year, ISBN, publisher, and availability details. Each book may have multiple authors. Additionally, the availability section should specify the number of copies currently available and the total number of copies in the library.

Your XML document should follow a hierarchical structure with the root element named <library>. Under this element, there should be multiple <book> elements, each representing a specific book in the library. Each <book> element must contain the following sub-elements:

- ➢ <title>: The title of the book.
- ➢ <authors>: A list of authors who contributed to the book. This element should contain one or more <author> sub-elements to represent each author.
- ➢ <publication_year>: The year the book was published.
- ➢ <isbn>: The International Standard Book Number (ISBN) of the book.
- ➢ <publisher>: The name of the publisher responsible for releasing the book.
- ➢ <availability>: This element should contain two sub-elements: a. <available_copies>: The number of copies currently available for borrowing. b. <total_copies>: The total number of copies available in the library (including both borrowed and available copies).

Please create an XML document with at least three books in the library catalog. Feel free to use any relevant book titles, author names, publication years, ISBNs, and availability details. Ensure that your XML structure is well-formed and adheres to the guidelines provided.

You need to submit to the LMS.

Deadline: 1st of July
File Naming Convention: SOAWS_Lab1_<IndexNumber>.pdf