University Of Babylon
College of Computer Technology
Department Of Information Networks

Web Page Design

Introduction to XML

HAIDER M. HABEEB
MSC. INFORMATION TECHNOLOGY

XML

- What is XML?
  - XML stands for eXtensible Markup Language
  - XML is a markup language much like HTML
  - XML was designed to carry data
  - XML tags are not predefined.
  - XML is designed to be self-descriptive
  - XML is a W3C Recommendation

XML

- What You Should Already Know?
  - HTML
  - JavaScript

Ready OR NOT?

J If NOT, you can go home J

The Difference Between XML and HTML

- XML is not a replacement for HTML.
- Different goals:
  - XML was designed to transport and store data, with focus on what data is
  - HTML was designed to display data, with focus on how data looks

- HTML is about displaying information.
- XML is about carrying information.

Is XML a Database?

- XML document is a collection of data.
- XML document doesn’t make much difference between the other files that store data.
- XML in a database format is a self describing, portable, and can describe data in tree or graph structure.
- XML is a sort of Database Management System (DBMS).

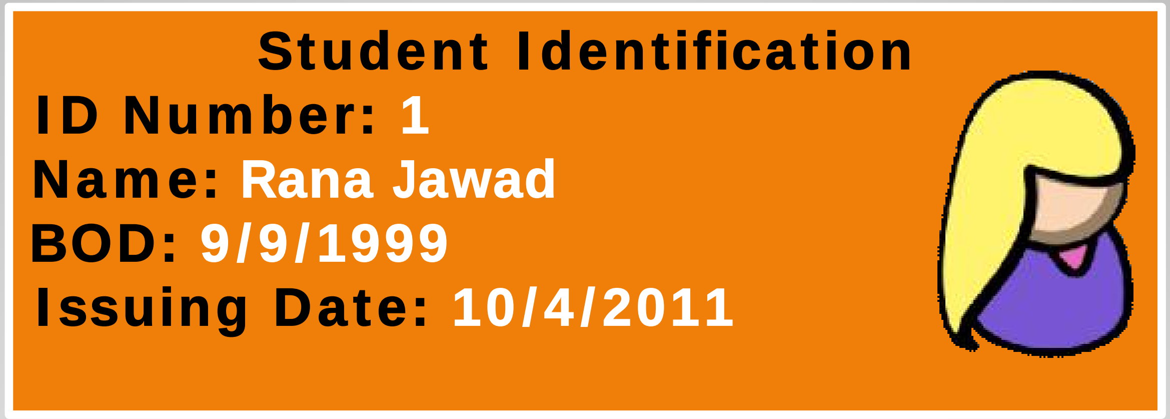
XML File

- How to write and store XML file?
- As you did before with CSS and JavaScript files.
- By using text file in different extension:
  - .css for CSS file
  - .js for JavaScript
- Then, .xml for XML file.

It's too easy, right?

XML structure

- Look at the following student ID?



Student Identification

ID Number: 1

Name: Rana Jawad

BOD: 9/9/1999

Issuing Date: 10/4/2011

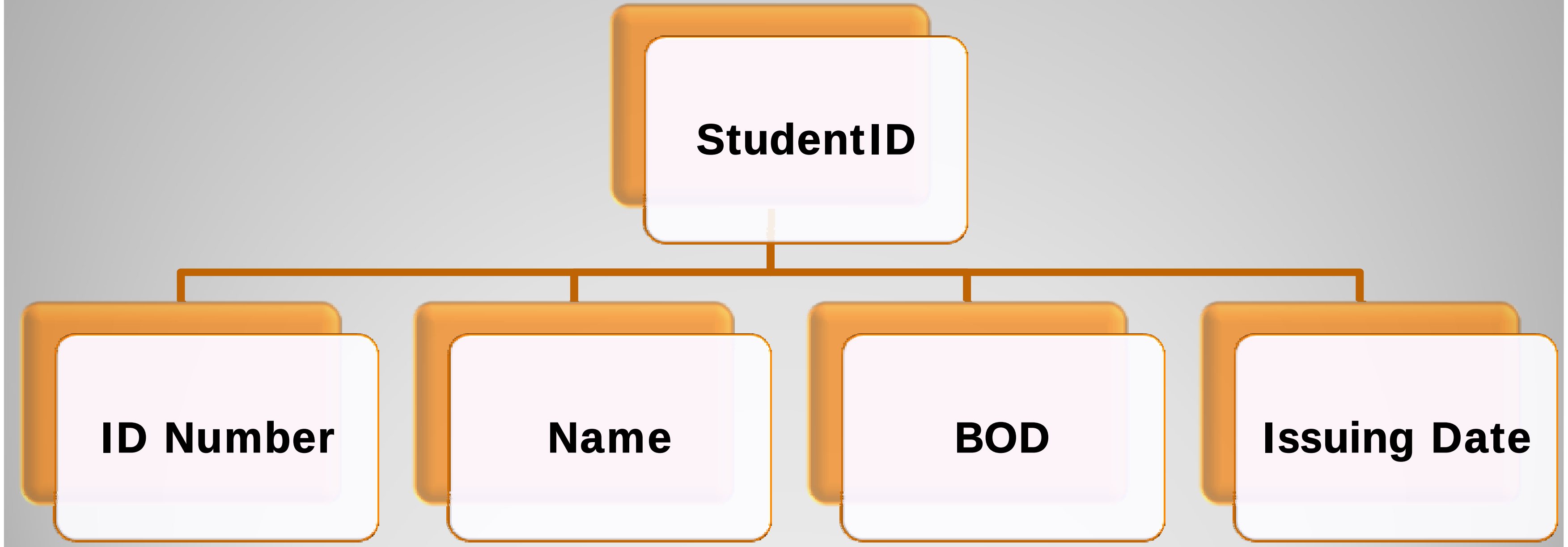
- Think about the main items of this ID!!!!
- Which items are constant and which variables?

XML structure

- The items are:
- Student Identification (StudentID)
  - ID Number
  - Name
  - BOD
  - Issuing Date

- Note: every student should has these information for issuing ID.

XML structure - Tree



XML structure - Tags

- Lets put them inside suited tags:

```
<studentID>
        <IdNumber>1</IdNumber>
        <Name>Rana Jawad</Name>
        <BOD>9/9/1999</BOD>
        <IssueDate>10/4/2011</IssueDate>
</studentID>
```

- What about the next student?

```
<studentID>
        <IdNumber>2</IdNumber>
        <Name>Ahmed Sameer</Name>
        <BOD>3/3/1998</BOD>
        <IssueDate>10/4/2011</IssueDate>
</studentID>
```

XML Does Not DO Anything

- Maybe it is a little hard to understand, but XML does not DO anything.
- XML was created to structure, store, and transport information.
- The previous example is a student ID, stored as XML:
- It is quite self descriptive.
- But still, this XML document does not DO anything.
- It is just information wrapped in tags.
- Someone must write a piece of software to send, receive or display it.

With XML You Invent Your Own Tags

- The tags in the example above (like <Name> and <BOD>) are not defined in any XML standard.
- These tags are "invented" by the author of the XML document.
- That is because the XML language has no predefined tags.
- The tags used in HTML are predefined.
- HTML documents can only use tags defined in the HTML standard (like <p>, <h1>, etc.).
- XML allows the author to define his/her own tags and his/her own document structure.

How Can XML be Used?

- XML Separates Data from HTML
- XML Simplifies Data Sharing
- XML Simplifies Data Transport
- XML Simplifies Platform Changes
- XML Makes Your Data More Available
- XML is Used to Create New Internet Languages

More about XML Structure

– **Can you modify the student ID?**

```
<Identification>
     <student id=“1”>
          <Name>Rana Jawad</Name>
          <BOD>9/9/1999</BOD>
          <IssueDate>10/4/2011</IssueDate>
     </student>

     <student id=“2”>
          <<Name>Ahmed Sameer</Name>
          <BOD>3/3/1998</BOD>
          <IssueDate>10/4/2011</IssueDate>
     </student>
</Identification>
```

XML Documents Form a Tree Structure

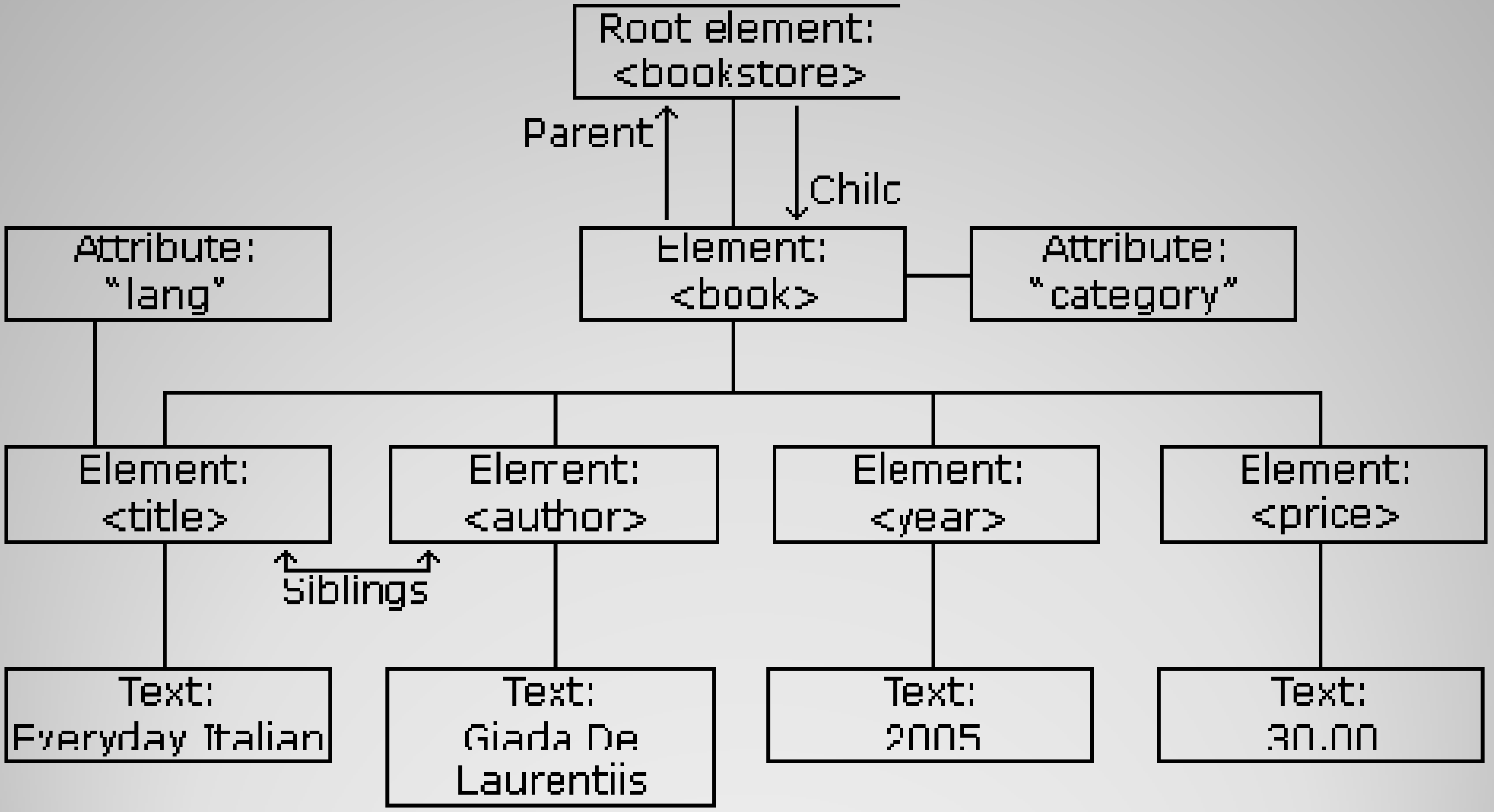
- XML documents must contain a root element.
- The tree starts at the root and branches to the lowest level of the tree.
- All elements can have sub elements (child elements):

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

- The terms parent, child, and sibling are used to describe the relationships between elements.
- Children on the same level are called siblings.
- All elements can have text content and attributes (just like in HTML).

XML Documents Form a Tree Structure

– **Example:**



XML Documents Form a Tree Structure

- The previous tree 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>
```

XML Syntax Rules

- All XML Elements Must Have a Closing Tag
- XML Tags are Case Sensitive
- XML Elements Must be Properly Nested
- XML Documents Must Have a Root Element
- XML Attribute Values Must be Quoted
- Entity References:
  - &lt; < :less than
  - &gt; > greater than
  - &amp; & ampersand
  - &apos; ' apostrophe
  - &quot; " quotation mark
- Comments in XML: <!-- This is a comment -->
- White-space is Preserved in XML

XML Naming Rules

- 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

Best Naming Practices

- Make names descriptive: <first_name>,
- <book_title> not <the_title_of_the_book>.
- Avoid "-" characters.
- Avoid "." characters.
- Avoid ":" characters. Colons are reserved to be used for something called namespaces
- A good practice is to use the naming rules of your database for the elements in the XML documents.
- Non-English letters like éòá are perfectly legal in XML, but watch out for problems if your software vendor doesn't support them.