eXtensible Markup Language (XML): The Basics

KUDA DUBE

Lecture Outline

1. Lecture Goals
2. Learning Outcomes
3. Materials and Resources
4. Introduction to XML
5. The Syntax of XML
6. Comparison between XML and HTML
7. Exercise: Designing an XML document

Lecture Goals

- Introduce XML
- Present the rules of writing XML documents
- Describe the syntax of XML
- Compare XML to HTML

Learning Objectives

At the end of this lecture, you should be able to:

1. Explain what XML is;
2. Describe the syntax of XML;
3. Apply the correct syntax of XML when writing XML documents;
4. Compare and contrast XML with HTML;
5. Design an XML document for a given domain.

Materials and Resources

- **Extensible Markup Language (XML) 1.0 (Third Edition)**
  - W3C Recommendation 04 February 2004
  - http://www.w3.org/TR/2004/REC-xml-20040204/
- **Extensible Markup Language (XML) 1.1**
  - http://www.w3.org/TR/2004/REC-xml11-20040204/
- **XML Tutorial**
  - http://www.w3schools.com/xml/default.asp
- **E.R. Harold & W. Scott Means: XML in a Nutshell. O'Reilly 2001.**

Software for XML Development

- Eclipse or any text editor;
- Browser;
- See STREAM section for this week!

Introduction to XML

1.1 Introduction: What is XML?

- Specification for:
  - *Storing information*
  - *Describing the structure of information*
- No tags of its own. Tags are user-created.
- Tags must adhere to rules of the XML specification.

In the opposite example:

- *What information is being stored?*
- *What is the structure of the information?*
- *What tags were created to describe the information and its structure?*

```
<?xml version="1.0"?>
<my_children>
    <child>
            <name>Bart</name>
            <gender>M</gender>
            <age>7</age>
    </child>
    <child>
            <name>Lisa</name>
            <gender>F</gender>
            <age>4</age>
    </child>
    <child>
            <name>Molly</name>
            <gender>F</gender>
            <age>2</age>
    </child>
</my_children>
```

1.2 Introduction: The Power of XML

- XML tags are different from HTML tags – ***XML tags describe the contents that they enclose;***
- XML is easily extended and adapted – *you can define your own custom mark-up language;*
- XML allows data sharing among systems & organisations – *advantage of text files;*
- XML is a free and non-proprietary specification - *created by W3C*

```
<?xml version="1.0"?>
<ancient_wonders>
 <wonder>
  <name language="English">Colossus
   of Rhodes</name>
  <name language="Greek">Κολοσσός
   της Ρόδου</name>
  <location> Rhodes, Greece </location>
  <height units="feet">107</height>
  <main_image file="colossus.jpg"
   w="528" h="349"/>
  <source sectionid="101"
   newspaperid="21"/>
 </wonder>
...
</ancient_wonders>
```

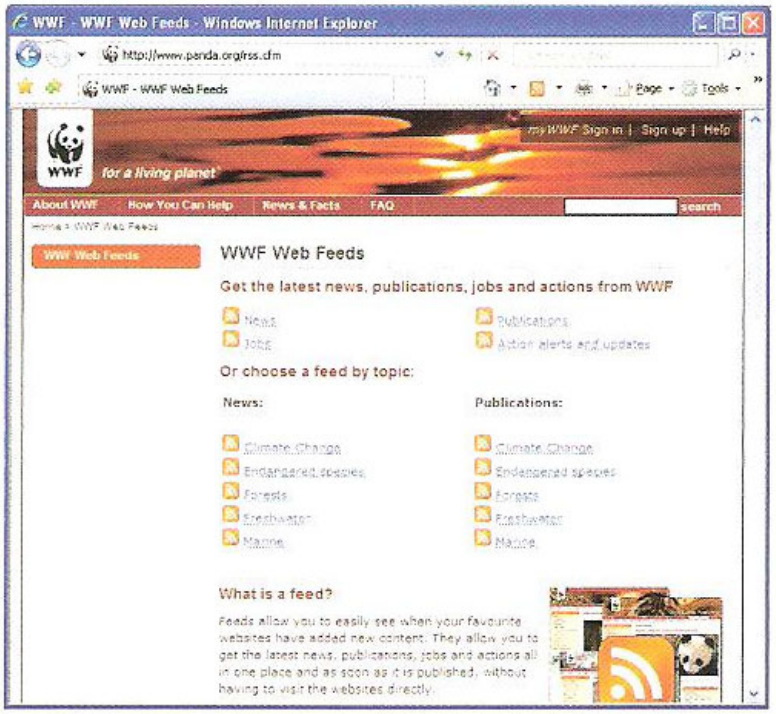
1.3 Introduction: Extending XML

- XML tags created from scratch:
  - *browsers have no idea how to display them*;
- XSL – eXtensible Style Sheet Language:
  - *allow specifying how an XML document should be displayed*
- XSL = XSLT + XPath + XSL-FO *(we look at XSL later)*
- Structure of XML documents specify tags used:
  - *DTD (Document Type Definition)*, and
  - *XML Schema* language

```
<?xml version="1.0"?>
<ancient_wonders>
...
 <wonder>
   <name language="English">Statue of Zeus at Olympia</name>
   <name language="Greek">Δίας μυθολογία</name>
   <location>Olympia, Greece</location>
   <height units="feet">39</height>
   <main_image file="zeus.jpg" w="528" h="349"/>
 </wonder>
...
</ancient_wonder>
```


```
<html>
   <head>   <title>Wonders of the World</title>  </head>  <body>
  <hr/>
  <p align="center">
   <strong> STATUE OF ZEUS AT OLYMPIA </strong><br/>
   <img src="zeus.jpg" width="528" height="349"/></p>
    <center>The Statue of Zeus at Olympia (<em>Δίας μυθολογία</em>) was
located in Olympia, Greece and stood 39 feet tall.
</center>   <br/>
 </body>
</html>
```

1.4 Introduction: *XML in the Real World*



RSS (Really Simple Syndication) *is an easy way for you to "subscribe" to news, podcasts and other content from Web sites that offer RSS feeds. Once you've subscribed to your favorite feeds, instead of needing to browse to the sites you like, information from these sites is delivered to you.*

Some believe that Google Suggest was instrumental in bringing Ajax to the forefront of Web development circles. The idea is simple: as you type, Google Suggest displays matching search terms which you can choose instead of continuing to type. Try it!
www.google.com/webhp?complete=1&hl=en

AJAX = HTML + JavaScript + XML + more!

Syntax of XML

2.1 XML Syntax: *Structure of XML Documents*

- XML documents are created by the author;
- XML document is self-describing:
  - Tags should describe the data that they contain!
- XML Declaration:

<?xml version="1.0"?>

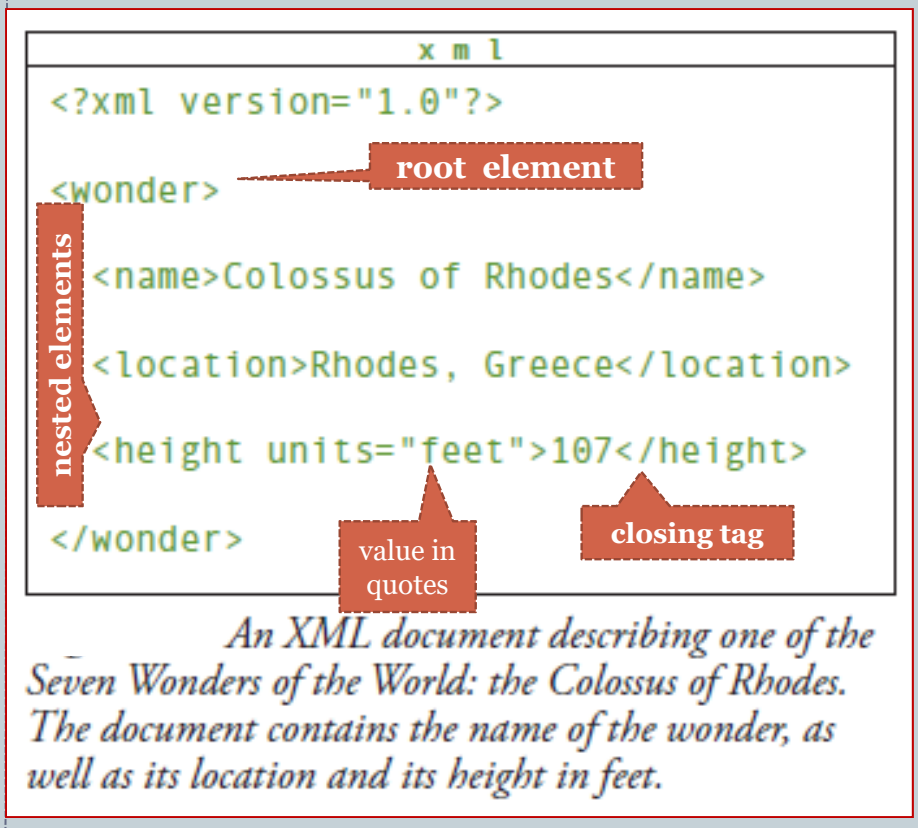
- Root element follows declaration: <wonder>;
- Child elements: <name>, <location> and <height>;
- Attribute: *units*

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

<wonder>

  <name>Colossus of Rhodes</name>

  <location>Rhodes, Greece</location>

  <height units="feet">107</height>

</wonder>
```

An XML document describing one of the Seven Wonders of the World: the Colossus of Rhodes. The document contains the name of the wonder, as well as its location and its height in feet.

2.2 XML Syntax: **Rules for Writing XML**

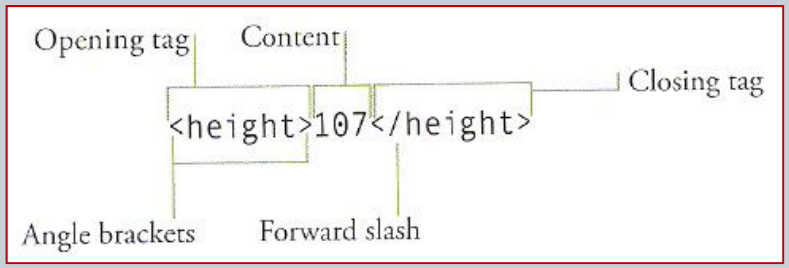
1. A root element is required;
2. Closing tags are required;
3. Elements must be properly nested;
4. Case matters: <name>, <Name>, <NAME> are different tags;
5. Values must be enclosed in quotation marks.



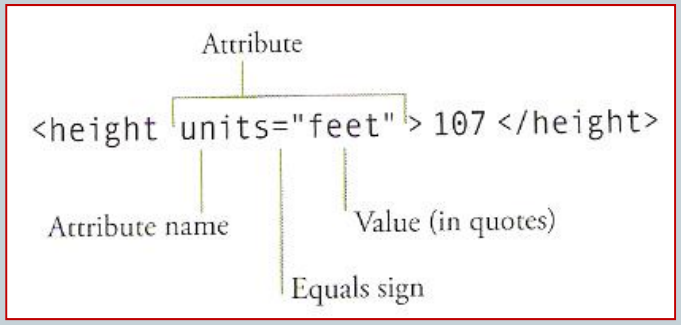
An XML document describing one of the Seven Wonders of the World: the Colossus of Rhodes. The document contains the name of the wonder, as well as its location and its height in feet.

2.3 XML Syntax: *XML Elements, Attributes and values*

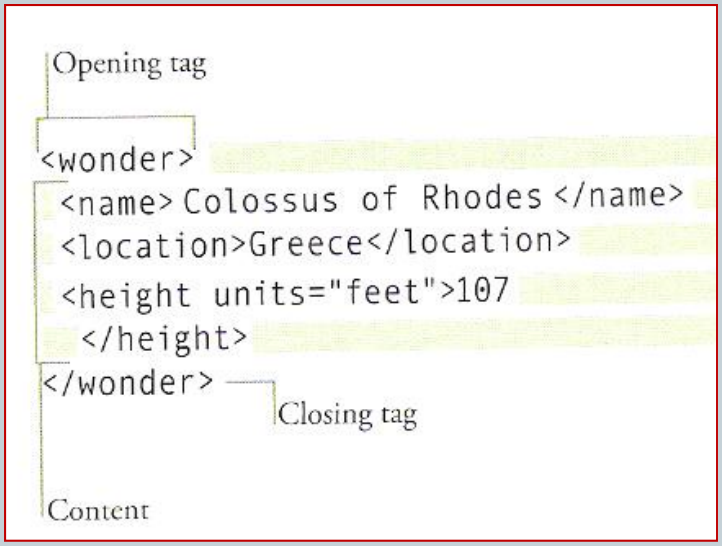
XML Elements



XML Attributes and Values



Nested Elements



2.4.1 XML Syntax: *Attributes vs Child Elements I*

Why:

```
<?xml version="1.0"?>
<email priority="urgent">
  <to>Conrad</to>
  <cc>Duncan</cc>
  ...
</email>
```

And not:

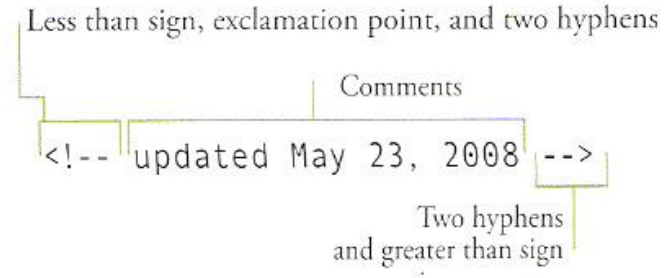
```
<?xml version="1.0"?>
<email priority="urgent“ to=“Conrad” cc=“Duncan” ... >
...
</email>
```

2.4.1 XML Syntax: *Attributes vs Child Elements* II

Attribute or child element – which should I use?

- There is a lot of debate about this question.
- Often both approaches can be used to *express the same thing*.
- Attributes are often more *convenient* to use, and the documents are more *compact*.
- Attributes should be used for *meta information* attached to the respective element.
- Attributes are not as *flexible* and *expressive* as elements (e.g., *a child element can also contain children*).

2.5 XML Syntax: *Writing Comments in XML*



XML comments have the same syntax as HTML comments.

```
<?xml version="1.0"?>
<ancient_wonders>
<wonder>
<name language="English">Colossus of Rhodes</name>
 <name language="Greek">Κολοσσός της Ρόδου</name>
<location>Rhodes, Greece</location>
    <height units="feet">107</height>
    <main_image filename="colossus.jpg"
    w="528" h="349"/>
<!-- the research on this wonder of the world
    came in part from the sectionid of the
    newspaper (and the newspaper's id)
    identified in the source tag below -->
<source sectionid="101"
    newspaperid="21"/>
</wonder>
</ancient_wonders>
```

Note the way ***empty elements*** are written in XML

2.6 XML Syntax: *Pre-defined Entities - 5 Special Symbols*

To write the five predefined entities:

- Type **&** to create an ampersand character (&).
- Type **<** to create a less than sign (<).
- Type **>** to create a greater than sign (>).
- Type **"** to create a double quotation mark (").
- Type **'** to create a single quotation mark or apostrophe (').

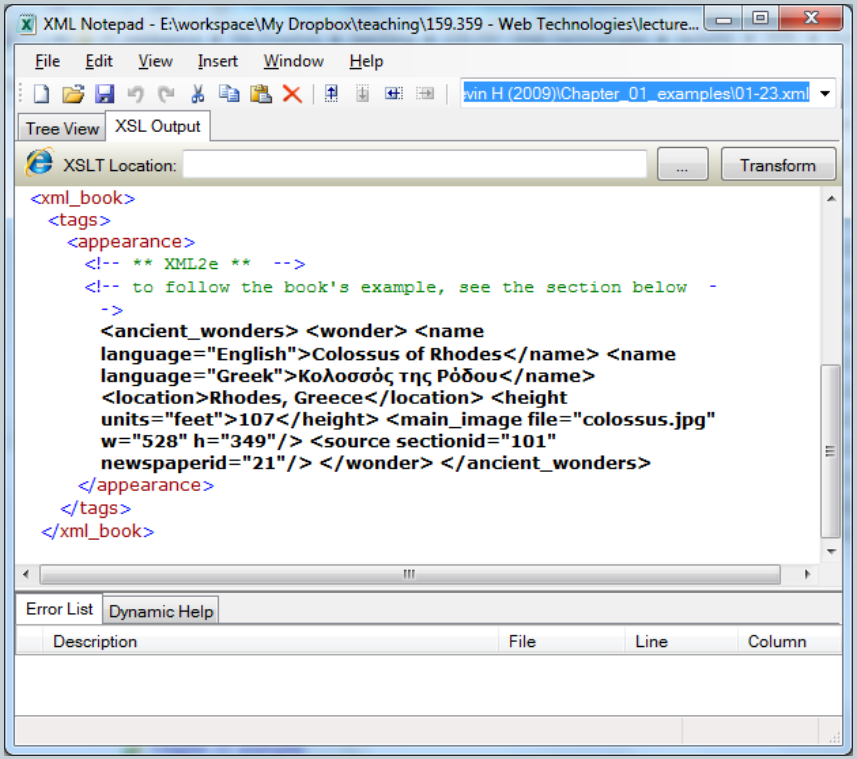
```
<?xml version="1.0"?>
<ancient_wonders>
<wonder>
 <name language="English">Colossus
    of Rhodes</name>
<name language="Greek">Κολοσσός
    της Ρόδου</name>
<location>Rhodes, Greece</location>
<height units="feet">&lt; 107</height>
<main_image filename="colossus.jpg"
    w="528" h="349"/>
<source sectionid="101"
    newspaperid="21"/>
</wonder>
</ancient_wonders>
```

Output of parsing: **<107**

2.7 XML Syntax: *Displaying Elements as Text*

CDATA section prevents XML processor from interpreting enclosed XML elements and attributes.

```
<?xml version="1.0"?>
<xml_book>
      <tags>
      <appearance>

<![CDATA[
<ancient_wonders>
 <wonder>
  <name language="English">Colossus of Rhodes</name>
  <name language="Greek">Κολοσσός της Ρόδου</name>
  <location>Rhodes, Greece</location>
  <height units="feet">107</height>
  <main_image file="colossus.jpg" w="528" h="349"/>
  <source sectionid="101" newspaperid="21"/>
 </wonder>
</ancient_wonders>
]]>

      </appearance>
      </tags>
</xml_book>
```



*Using **Internet Explorer** from **XML Notepad** shows that the elements within the **CDATA section** are treated as text.*

2.8 XML Syntax: Processing Instructions


```
<?xml version="1.0"?>
<?inlook driver="com.ediabolo.mail.DefaultDriver"?>
<email priority="urgent">
    <to>Max</to>
    ...
</email>
```

- Processing instructions can be used to pass additional information to applications.
- The start with "<?" and end with "?>"
- The content is not part of the data.

NB: A similar syntax is used in order to embed PHP script in HTML.

Comparison of XML with HTML

3.1 XML vs HTML

This is a summary of the differences between XML and HTML (syntax):

Tags must be closed

```
<p>This is a paragraph
<p>This is another paragraph
```

HTML

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

XML

Empty tags

```
<br>
```

HTML

```
<br/>
```

XML

3.2 XML vs HTML (ctd)

XML is case-sensitive

<p>This is ok in HTML </P>	HTML
<p>This is not ok in XML</P>	XML

Attribute values are mandatory and must always be quoted

- (in html, they must be quoted as well acc. To HTML 4.0, but this was not the case in older versions and all major browsers accept unquoted values as well.)

<hr noshade>	HTML
<hr shade="noshade"/>	XML

3.3 XML vs HTML (ctd)

XML preserves white spaces while HTML strips off whitespaces.

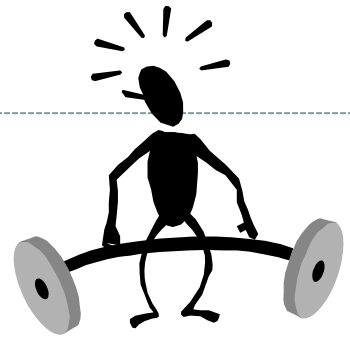
XML uses LF (line feed) as new line character:

- *Most windows applications use CR LF (carriage return, line feed),*
- *Unix applications use LF and*
- *Mac applications use CR.*

XML Basics Exercise

DESIGNING AND WRITING AN XML DOCUMENT

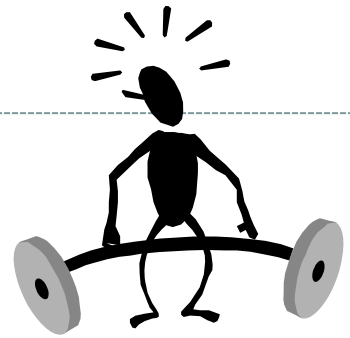
4.1 Exercise 1



Design a sample XML document representing an email. Take into account that an email has:

1. A content type (plain text, html or rtf)
2. The name and email address of the sender
3. A timestamp when the email has been send
4. The pop server used to send the email
5. The reply-to address (to necessarily the same as the sender address)
6. One or many receivers (to) (name/email)
7. One or many cc's
8. A subject
9. A body
10. Attachments

4.2 Exercise 1 (ctd)



Try to use tags to group related information.
Use comments to improve readability of the XML file.
Verify that the file is well-formed using at least two tools (e.g., Mozilla, Cooktop).

Next Topics …

- Constraint Models for Web Data:
  1. JSON schema
  2. Document Type Definitions
  3. XML Schema