158.258 Web Development

Introduction to XML

Computer Science and Information Technology
School of Mathematical and Computational Sciences
Massey University
(AKLI, DISD and MTUI)

Revised: 2022-08-03

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.

Software for XML Database Development

- Its possible to use Native XML Database:
  - *eXist DB*
  - *BaseX*
- VS Code has XML Tools Extension
- Eclipse has XML development pludins

Section 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*

```xml
<?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>
```

stored?

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

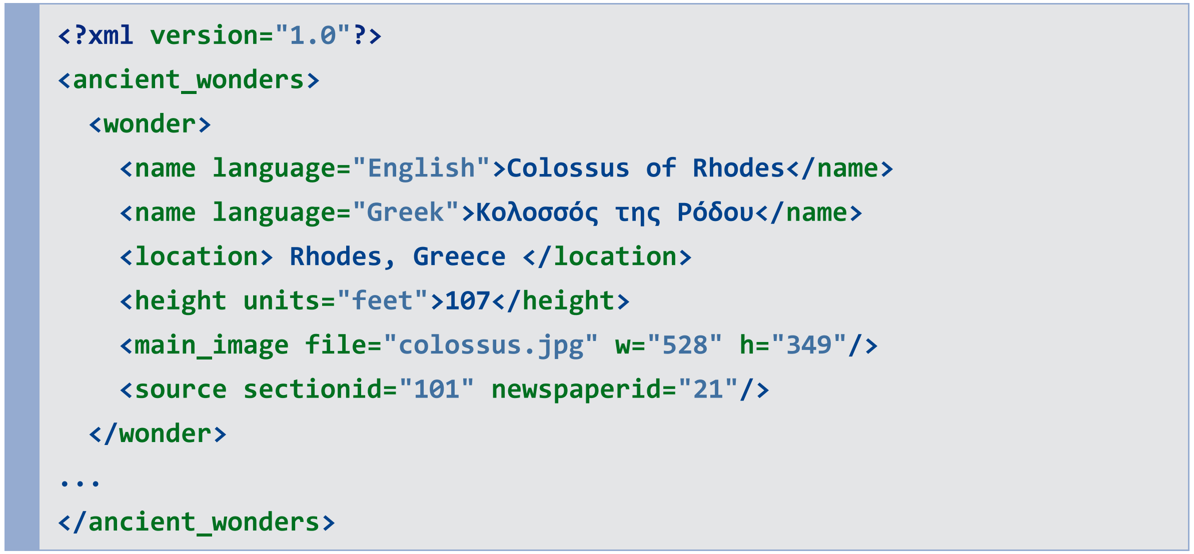
The Power of XML

- XML tags are different from HTML tags

  – XML tags describe the contents that they enclose;

- XML is easily extended and adapted

  – you can define



```
<?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>
```

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*

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)*

```
<?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>
```

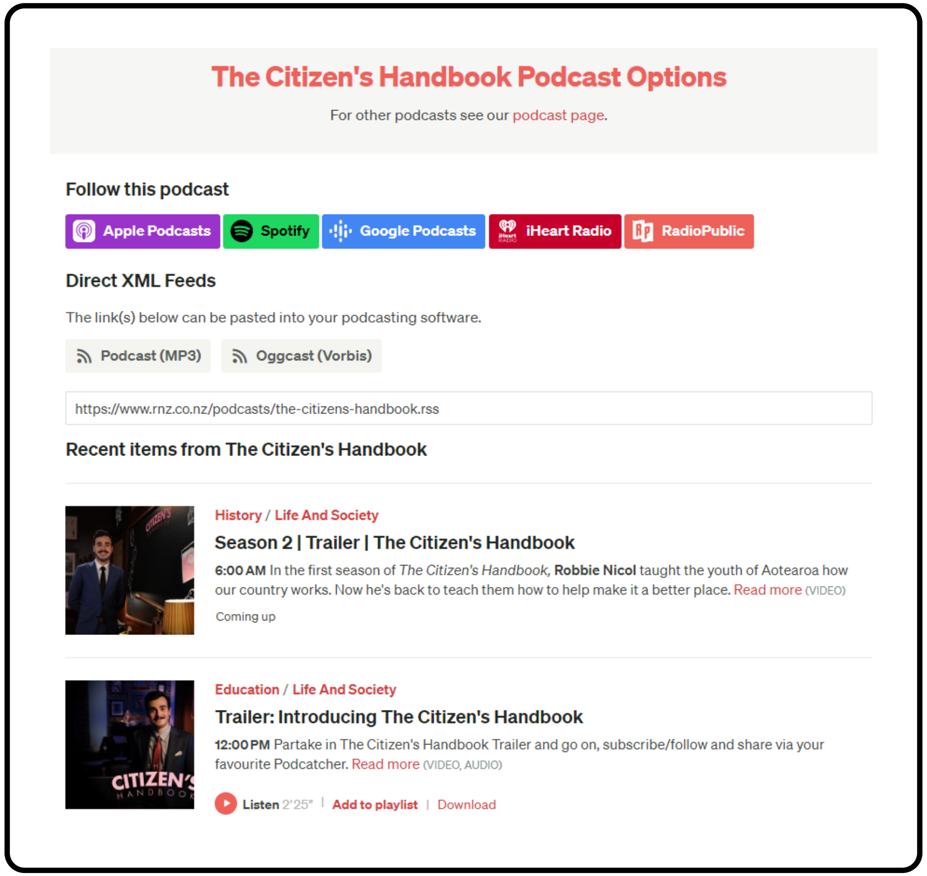
After applying XSL Transform, gives the following:

```
<html>
  <head>
    <title>Wonders of the World</title>
  </head>
  <body>
    <hr />
    <p align="center">
```

- Structure of XML documents specify tags used:
  - *DTD (Document Type Definition), and*
  - *XML Schema language*

```
    <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>
```

XML in the Real World



For other podcasts see our podcast page.

Follow this podcast

Direct XML Feeds

The link(s) below can be pasted into your podcasting software.

https://www.rnz.co.nz/podcasts/the-citizens-handbook.rss

Recent items from The Citizen's Handbook

History / Life And Society

Season 2 | Trailer | The Citizen's Handbook

6:00 AM In the first season of *The Citizen's Handbook*, **Robbie Nicol** taught the youth of Aotearoa how our country works. Now he's back to teach them how to help make it a better place. Read more (VIDEO)

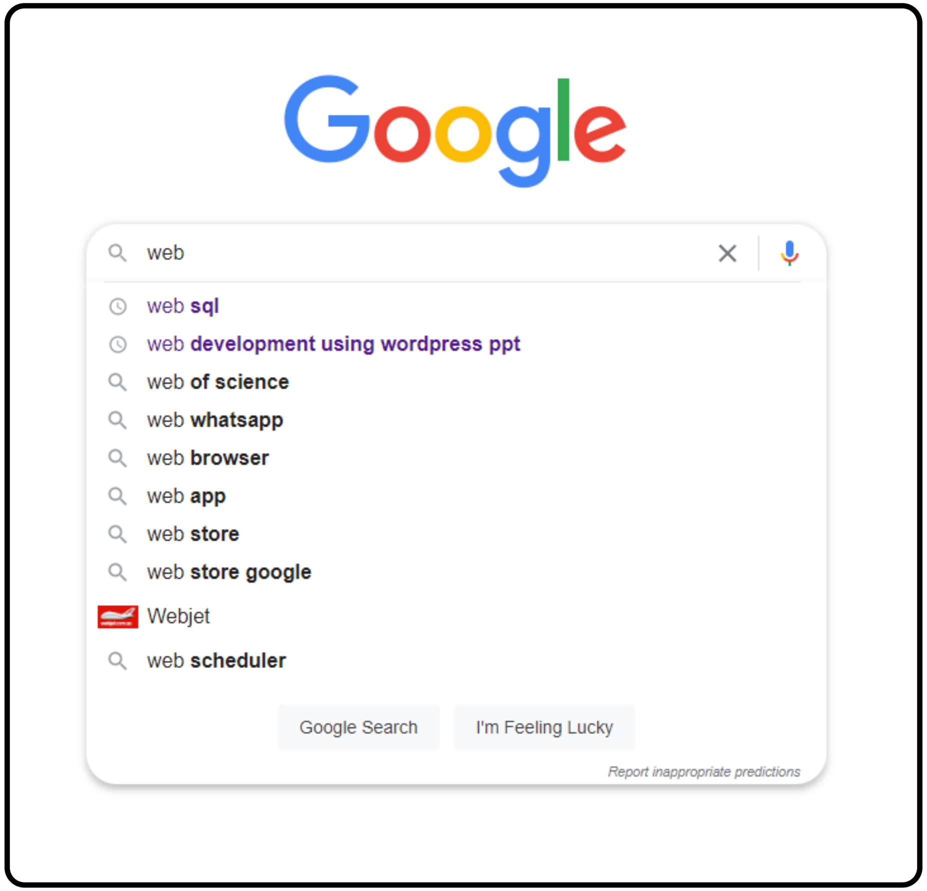
Coming up

Education / Life And Society

Trailer: Introducing The Citizen's Handbook

12:00 PM Partake in The Citizen's Handbook Trailer and go on, subscribe/follow and share via your favourite Podcatcher. Read more (VIDEO, AUDIO)

RSS *(Really Simple Syndication)* is an easy way for you to "subscribe" to news, podcasts and other content



web

- web **sql**
- web **development using wordpress ppt**
- web **of science**
- web **whatsapp**
- web **browser**
- web **app**
- web **store**
- web **store google**
- Webjet
- web **scheduler**

Report inappropriate predictions

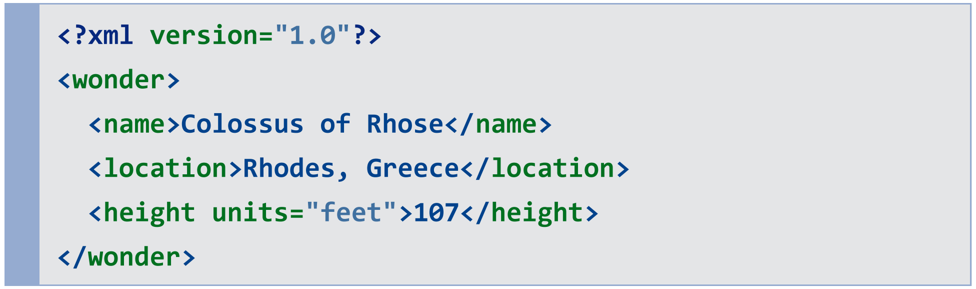
Some argue that Google Suggest was instrumental in bringing AJAX to the forefront of Web from Websites that offer RSS feeds. Once you have subscribed to your favourite feeds, instead of needing to browse to the sites you like, information from these sites is delivered to you.

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!

AJAX = HTML + JavaScript + XML + more

Section 2: Syntax of XML

- 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>`;



```xml
<?xml version="1.0"?>
<wonder>
  <name>Colossus of Rhose</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 it's location and it's height in feet.

- Child elements: `<name>`, `<location>`, `<height>`;
- Attribute: `units`

Rules for Writing XML

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

```xml
<?xml version="1.0"?>
<wonder> <!-- root element -->
  <!-- The following are nested elements -->
  <name>Colossus of Rhose</name>
  <location>Rhodes, Greece</location>
  <height units="feet">107</height> <!-- Closing tag -->
  <!-- Note the value of units is in quotes above -->
</wonder>
```

An XML document describing one of the *Seven Wonders of the World*, the *Colossus of Rhodes*.

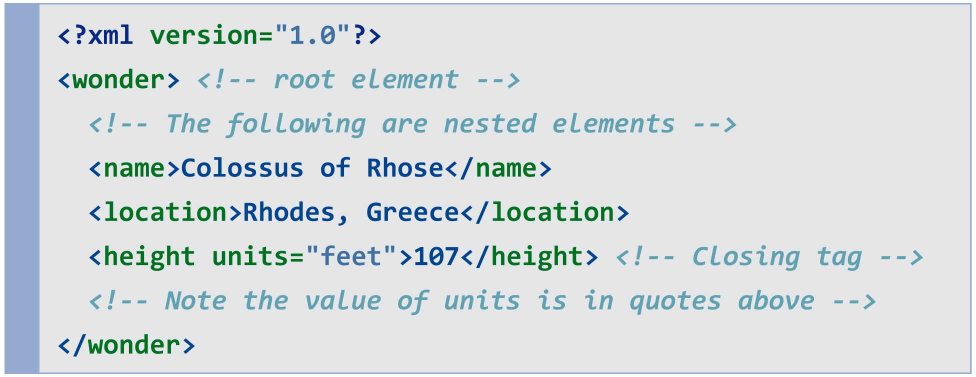
XML Elements, Attributes and Values

XML Elements

- Opening tag with angle brackets: `<height>`
- Content: `102`
- Closing tag with angle brackets and forward slash: `</height>`

XML Attributes and Values

- Attribute: `units="feets"`
- Attribute name: `units`
- Value in quotes: `"feet"`



```
<?xml version="1.0"?>
<wonder> <!-- root element -->
  <!-- The following are nested elements -->
  <name>Colossus of Rhose</name>
  <location>Rhodes, Greece</location>
  <height units="feet">107</height> <!-- Closing tag -->
  <!-- Note the value of units is in quotes above -->
</wonder>
```

An XML document describing one of the *Seven Wonders of the World*, the *Colossus of Rhodes*.

Attributes vs Child Elements

- Attribute or child element – which should I use?
- There has been a 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.

- **Why:**

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

- **And not:**

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

- Attributes are not as flexible and expressive as elements (e.g., a child element can also contain children).

Comments in XML

Comments in XML are the same as in HTML: `<!-- this is a comment -->`

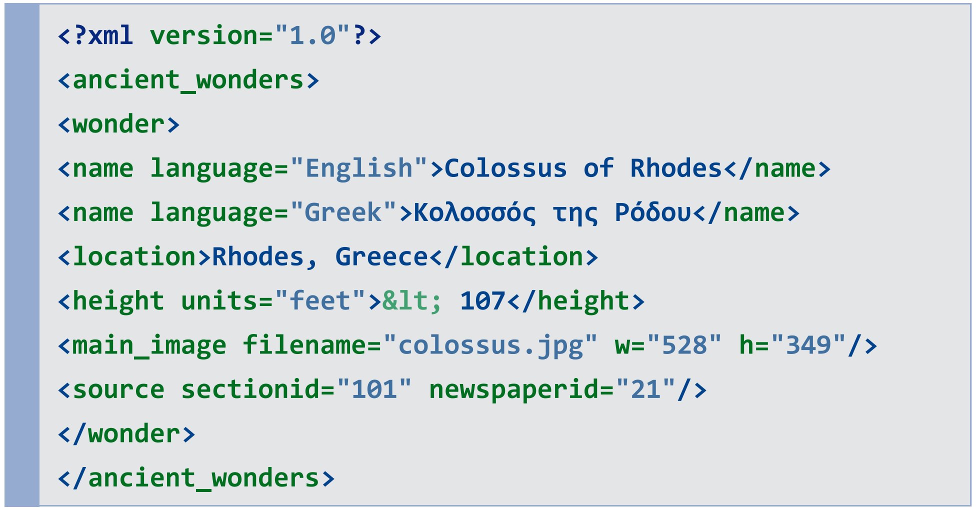
```
<?xml version="1.0"?>
<ancient_wonders>
<wonder>
<name language="English">Colossus of Rhodes</name>
<name language="Greek">Κολοσσός της Ρόδου</name>
<location>Rhodes, Greece</location>
<height units="feet">107</height>
<main_image filename="colossus.jpg" w="528" h="349" />
<!--
The research on this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

Pre-defined Entities — 5 Special Symbols

- XML pre-defined entities are the same as in HTML.

- To write the five pre-defined entities:

  - *Type* `&amp;` *to create an ampersand character (&).*
  - *Type* `&alt;` *to create the less than sign (<).*
  - *Type* `&gt;` *to create the greater than sign (>).*
  - *Type* `&quot;` *to create a*



```
<?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>
```

double quotation mark (").

- *Type `&apos;` to create single quotation mark or apostrophe (')*

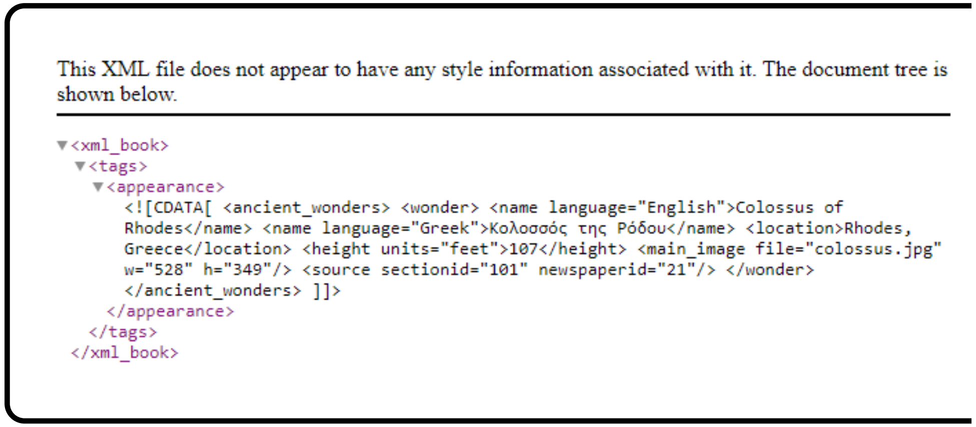
- The output of parsing contents of the `<height>` element is: `<107`

Displaying Elements as Text

```
<?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>
```



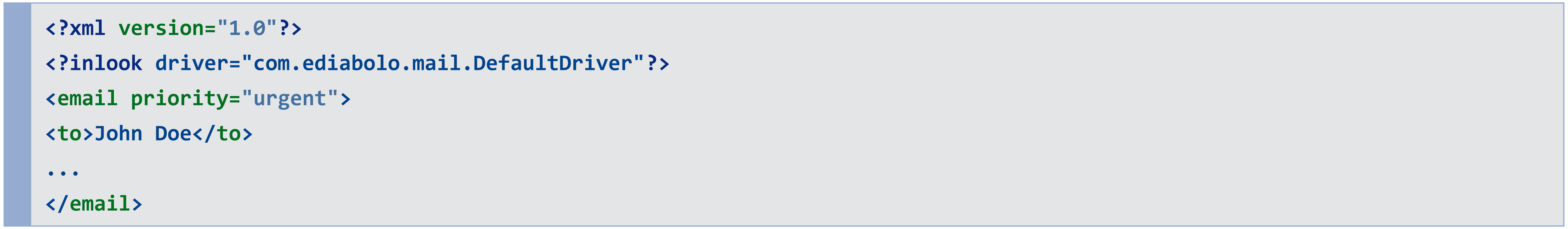
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<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 Google Chrome shows that the elements within the CDATA section are treated as text and not interpreted as XML.

XML Processing Instructions

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



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

Section 3: Comparison of XML with HTML

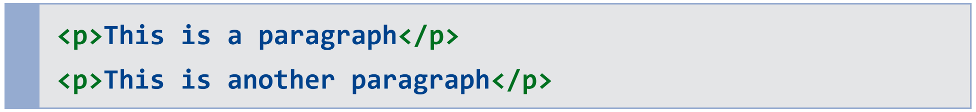
A summary of the differences between XML and HTML (syntax)

- Tags must be closed in XML
  - *Permissive part of HTML5 and earlier versions of HTML5*

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

- XML requires closing all opened tags:

- Attribute values are mandatory and must always be quoted:
  - *(in HTML, they must be quoted as well, but this was not the case in older versions and all major browsers accept unquoted values as well)*
  - *Allowed in HTML: `<hr noshade>`*
  - *In XML write as: `<hr shade="noshade" />`*



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

- Empty tags:
  - *Permissive part of HTML5 and earlier versions of HTML allows:* `<br>`
  - *XML requires abbreviated closing of empty tag:* `<br />`
- XML is case-sensitive
  - *HTML allows the following:* `<p>This is ok in HTML </P>`
  - *XML doeos not allow:* `<p>This is not ok in XML</P>`
- 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.*

Topic Summary

- In this topic, we covered the following aspects of XML:
  - *Definition of XML.*
  - *The syntax of XML.*
  - *Creating XML documents.*
  - *Differences between XML and HTML.*

Tutorial Exercise 1: *Designing and Writing an XML Document*

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

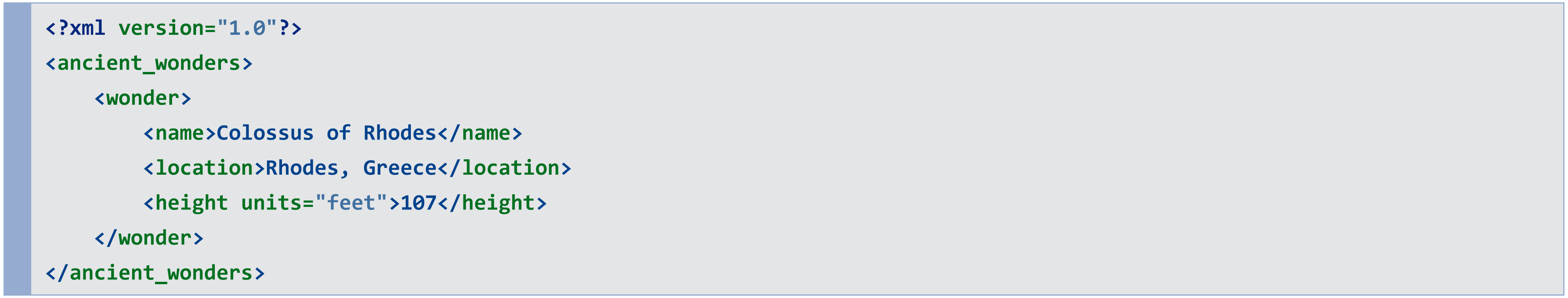
  1. *A content type (plain text, html or rtf)*
  2. *The name and email address of the sender*
  3. *A timestamp when the email has been send*
  4. *The pop server used to send the email*

- 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).

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*

Tutorial Exercise 2: *Ancient Wonders*

- The aim of this tutorial is to help you to practice your understanding of the basics of XML.



```xml
<?xml version="1.0"?>
<ancient_wonders>
    <wonder>
        <name>Colossus of Rhodes</name>
        <location>Rhodes, Greece</location>
        <height units="feet">107</height>
    </wonder>
</ancient_wonders>
```

- Create a new document `wonders.xml` and copy and paste the above code.

Tutorial Exercise 2: *Ancient Wonders*

1. In the above XML document, add another ancient wonder element for the Great Pyramid of Giza located at Giza in Egypt and with the height of 455 feet. Save file wonders.xml

2. For each `<wonder>` in this XML document, just below the `<name>` element, add the `<main_image>`

4. The Colossus of Rhodes also has the Greek name, `Κολοσσός της Ρόδου`. Add a language attribute to the `<name>` element and add a new `<name>` element for the Greek name of the Colossus. Also make the necessary addition to the entry for the Pyramid. Save the XML document `wonders.xml`.

5. Now draw the structure of

elements with attributed filename whose values are `colossus.jpg` and `pyramid.jpg` respectively.

3. The research on the two wonders of the world came in part from the `sectionid` of the newspaper (and the newspaper's id) identified in the sources below:

   - *Colossus of the Rhodes:*

your XML document as a tree with its root at the top. The nodes of the tree should be of various types including *elements, attributes, comments* and *text* but do not include the full text in the diagram to avoid clutter — use the word, `text`, as a placeholder for full text. Save the diagram as `wonders-design.png`.

6. Upload your XML document to your hosted website.

sectionid 56 and newspaperid 21;

- *Pyramid of Giza: sectionid 29 and newspaperid 112*
- *Add the `<source>` element as a child of each `<wonder>` element with attributes `sectionid` and `newspaperid` and as an empty element.*