TUTORIAL 1
INTRODUCING XML

• XML stands for Extensible Markup Language. A markup language specifies the structure and content of a document.

• Because it is extensible, XML can be used to create a wide variety of document types.
INTRODUCING XML

• XML is a subset of the Standard Generalized Markup Language (SGML) - introduced in the 1980s.
• SGML is very complex and can be costly.
• Creation of Hypertext Markup Language (HTML) - more easily used markup language.
• XML can be seen as sitting between SGML and HTML – easier to learn than SGML, more robust than HTML.
THE LIMITS OF HTML

- HTML was designed for formatting text on a Web page. It was not designed for dealing with the description of the content of a Web page. Additional features have been added to HTML, but they do not solve data description or cataloging issues in an HTML document.

- Because HTML is not extensible, it cannot be modified to meet specific needs. Browser developers have added features making HTML more robust, but this has resulted in a confusing mix of different HTML standards.
THE LIMITS OF HTML

- HTML cannot be applied consistently. Different browsers require different standards making the final document appear differently on one browser compared with another.
## XML VOCABULARIES

<table>
<thead>
<tr>
<th>XML Vocabulary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel Definition Format (CDF)</td>
<td>Automatic delivery of information from Web publishers to PCs, PDAs, cell phones, and other information devices</td>
</tr>
<tr>
<td>Chemical Markup Language (CML)</td>
<td>Coding of molecular and chemical information</td>
</tr>
<tr>
<td>Extensible Hypertext Markup Language (XHTML)</td>
<td>HTML written as an XML application</td>
</tr>
<tr>
<td>Mathematical Markup Language (MathML)</td>
<td>Presentation and evaluation of mathematical equations and operations</td>
</tr>
<tr>
<td>Musical Markup Language (MML)</td>
<td>Display and organization of music notation and lyrics</td>
</tr>
<tr>
<td>Open Financial Exchange (OFX)</td>
<td>Exchange of financial data between financial institutions, businesses, and consumers via the Internet</td>
</tr>
<tr>
<td>Real Simple Syndication (RSS)</td>
<td>Distribution of news headlines and syndicated columns</td>
</tr>
<tr>
<td>Synchronized Multimedia Integration Language (SMIL)</td>
<td>Editing of interactive audiovisual presentations involving streaming audio, video, text, and any other media type</td>
</tr>
<tr>
<td>Voice Markup Language (VoiceXML)</td>
<td>Creation of audio dialogues that feature synthesized speech, digitized audio, and speech recognition</td>
</tr>
</tbody>
</table>
XML VOCABULARIES - Examples

http://www.w3.org/Math/

MML being referred to Medical Markup Language
WELL-FORMED AND VALID XML DOCUMENTS

- An XML document is well-formed if it contains no syntax errors and fulfills all of the specifications for XML code as defined by the W3C.

- An XML document is valid if it is well-formed and also satisfies the rules laid out in the DTD or schema attached to the document.

- DTD or schema not required but helpful to ensure XML documents follow specific vocabulary.
THE STRUCTURE OF AN XML DOCUMENT

- XML documents consist of three parts
  - The prolog
  - The document body
  - The epilog
THE STRUCTURE OF AN XML DOCUMENT

• The prolog is optional and provides information about the document itself. It’s the XML declaration

```xml
<?xml version="v #" encoding="encoding type"
standalone="yes|no" ?>
```

*Attributes explained on page 11*

• **XML is case sensitive**
THE STRUCTURE OF AN XML DOCUMENT

• The document body contains the document’s content in a hierarchical tree structure.

• The epilog is also optional and contains any final comments or processing instructions.
THE STRUCTURE OF AN XML DOCUMENT: CREATING THE PROLOG

- The prolog consists of four parts in the following order:
  - XML declaration
  - Miscellaneous statements or comments (optional)
  - Processing instructions (optional)
  - Document type declaration (optional)
THE STRUCTURE OF AN XML DOCUMENT:
THE XML DECLARATION

• The XML declaration is always the first line of code in an XML document. It tells the processor what follows is written using XML. It can also provide any information about how the parser should interpret the code.

• The complete syntax is:

```xml
<?xml version="version number" encoding="encoding type" standalone="yes | no" ?>
```

• A sample declaration might look like this:

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
```
THE STRUCTURE OF AN XML DOCUMENT: INSERTING COMMENTS

- Comments or miscellaneous statements go after the declaration. Comments may appear anywhere after the declaration.
- The syntax for comments is:

  ```xml
  <!-- comment text -->
  ```

- This is the same syntax for HTML comments
ELEMENTS

- Elements are the basic building blocks of XML files.

- Elements contain an opening tag and a closing tag
  – Content is stored between tags
ELEMENETS

• A closed element, has the following syntax:
  
  <element_name>Content</element_name>

• Example:
  
  <artist>Miles Davis</artist>
• Element names are case sensitive

• Elements can be nested, as follows:
  <tracks>Kind of Blue
    <track>So What (:22)</track>
    <track>Blue in Green (5:37)</track>
  </tracks>
ELEMENTS

- Nested elements are called child elements.
- Elements must be nested correctly. Child elements must be enclosed within their parent elements.
- Elements names must begin with a letter or the underscore character (\_) and may not contain blank spaces
ELEMENTS

• All elements must be nested within a single **document** or **root element**. There can be only **one** root element. If an XML document does not have the **root**, it’s not considered well-formed and is rejected.

• An **open** or **empty** element is an element that contains no content. They can be used to mark sections of the document for the XML parser.

Ex.: `<element />`

New Perspectives on XML, 2nd Edition
Tutorial 1
ELEMENTS
Charting the Element Hierarchy

- a parent can only contain a single child

* - an element can contain zero or more sub elements

+ - an element must contain at least one sub element

? – an element may or may not have sub elements

Write code (pages 18, 19)
WORKING WITH ATTRIBUTES

• An **attribute** is a feature or characteristic of an element. Attributes are text strings and must be placed in single or double quotes. The syntax is:

  `<element_name attribute="value"> …
  </element_name>`

• In the case of an empty element:

  `<element_name attribute="value" />`
ATTRIBUTES

- They are considered text strings => attributes
  - VALUES can contain spaces
    - Must begin with letter or _ (underscore)
    - Spaces not allowed in attributes NAMES
    - Never begin with word “xml”
    - Can appear only once within the element

Add attributes to jazz.xml – pages 20, 21
ELEMENTS AND ATTRIBUTES: ADDING ELEMENTS TO THE JAZZ.XML FILE

This figure shows the revised document

```xml
<item>
  <title>Kind of Blue</title>
  <artist>Miles Davis</artist>
  <tracks>
    <track length="9:22">So What</track>
    <track length="9:46">Freddie Freeloader</track>
    <track length="5:37">Blue in Green</track>
    <track length="11:33">All Blues</track>
    <track length="9:26">Flamenco Sketches</track>
  </tracks>
</item>

<item>
  <title>Cookin'</title>
  <artist>Miles Davis</artist>
  <tracks>
    <track length="5:57">My Funny Valentine</track>
    <track length="9:53">Blues by Five</track>
    <track length="4:22">Airegin</track>
    <track length="13:03">Tune-Up</track>
  </tracks>
</item>

<item>
  <title>Blue Train</title>
  <artist>John Coltrane</artist>
  <tracks>
    <track length="10:39">Blue Train</track>
    <track length="9:06">Moment's Notice</track>
    <track length="7:44">Locomotion</track>
    <track length="7:55">I'm Old Fashioned</track>
    <track length="7:03">Lazy Bird</track>
  </tracks>
</item>
```
WORKING WITH ATTRIBUTES

- When to use or not use attributes?

If you take out ALL XML tags (even attributes) – the remaining text would have the total content

If not, then the attribute should become an element (if the attribute should be displayed)
CHARACTER REFERENCES

- Special characters, such as the symbol for the British pound, can be inserted into your XML document by using a **character reference**. The syntax is:

```
&#nnn;
```

**nnn** = character number from ISO/IEC (international numbering system referencing chars from practically any language)
CHARACTER REFERENCES

- Character references in XML are the same as in HTML.
- You can also use the entity reference
  
  \texttt{\&entity}

  \textit{Ex.}: \texttt{\&amp;} = \texttt{\&38} = Ampersand

- To insert HTML code within XML

  \texttt{<htmcode>&#60;img src="banner.jpg" width="90" height="15" /&gt;</htmcode>
This figure shows commonly used character reference numbers

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Character Reference</th>
<th>Entity Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>©</td>
<td>&amp;#169;</td>
<td></td>
<td>Copyright symbol</td>
</tr>
<tr>
<td>®</td>
<td>&amp;#174;</td>
<td></td>
<td>Registered trademark symbol</td>
</tr>
<tr>
<td>™</td>
<td>&amp;#153;</td>
<td></td>
<td>Trademark symbol</td>
</tr>
<tr>
<td>&lt;</td>
<td>&amp;#60;</td>
<td>&lt;</td>
<td>Less than symbol</td>
</tr>
<tr>
<td>&gt;</td>
<td>&amp;#62;</td>
<td>&gt;</td>
<td>Greater than symbol</td>
</tr>
<tr>
<td>&amp;</td>
<td>&amp;#38;</td>
<td>&amp;</td>
<td>Ampersand</td>
</tr>
<tr>
<td>&quot;</td>
<td>&amp;#22;</td>
<td>&quot;</td>
<td>Double quote</td>
</tr>
<tr>
<td>'</td>
<td>&amp;#27;</td>
<td>'</td>
<td>Apostrophe (single quote)</td>
</tr>
<tr>
<td>£</td>
<td>&amp;#163;</td>
<td></td>
<td>Pound sign</td>
</tr>
<tr>
<td>€</td>
<td>&amp;#128;</td>
<td></td>
<td>Euro sign</td>
</tr>
<tr>
<td>¥</td>
<td>&amp;#165;</td>
<td></td>
<td>Yen sign</td>
</tr>
</tbody>
</table>
CHARACTER REFERENCES

This figure shows the revised Jazz.XML file

```xml
<Item>
  <title>Kind of Blue</title>
  <price>US: $11.99</price>
  <price>UK: &pound;8.39</price>
  <artist>Miles Davis</artist>
  <tracks>
    <track length="9:22">So What</track>
    <track length="9:46">Freddie Freeloader</track>
    <track length="5:37">Blue in Green</track>
    <track length="11:33">All Blues</track>
    <track length="9:26">Flamenco Sketches</track>
  </tracks>
</Item>

<Item>
  <title>Cookin'</title>
  <price>US: $7.99</price>
  <price>UK: &pound;5.59</price>
  <artist>Miles Davis</artist>
  <tracks>
    <track length="5:57">My Funny Valentine</track>
    <track length="7:53">Blues by Five</track>
    <track length="4:22">Airegine</track>
    <track length="13:03">Tune-Up</track>
  </tracks>
</Item>

<Item>
  <title>Blue Train</title>
  <price>US: $8.99</price>
  <price>UK: &pound;6.29</price>
  <artist>John Coltrane</artist>
  <tracks>
    <track length="10:39">Blue Train</track>
    <track length="9:06">Moment's Notice</track>
    <track length="7:11">Locomotion</track>
    <track length="7:35">I'm Old Fashioned</track>
    <track length="7:08">Lazy Bird</track>
  </tracks>
</Item>
```
PARSED CHARACTER DATA

- **Text Characters:** parsed character data, character data, white space

- **Parsed character data,** or **pcdata** consists of all those characters that XML treats as parts of the code of XML document
  - The XML declaration
  - The opening and closing tags of an element
  - Empty element tags
  - Character or entity references
  - Comments
PARSED CHARACTER DATA

<temp> <32 degrees</temp>

Why this would generate an error?
What should you do to avoid this error?

<temp> &lt;32 degrees</temp>
CHARACTER DATA

- **Character data** the symbols remaining after you remove parsed character data = data content

- **White space** refers to any space in a document. HTML applies **white space stripping** (consecutive occurrences of white space = single white space). XML treats white space slightly different (no stripping). White space – from spacebar, from ENTER, from Tab

**Note:** IE browser transforms XML into HTML => white space stripping is applied
WHITE SPACES

• Ignored when it is the only character between element tags
• Ignored within prolog, epilog and within element tags
• Not ignored when within an attribute VALUE
CDATA SECTIONS

- A CDATA section is a large block of text the XML processor will interpret only as text.

- The syntax to create a CDATA section is:

  ```xml
  <! [CDATA [ Text Block ] ]>
  ```
CDATA SECTIONS

• In this example, a CDATA section stores several HTML tags within an element named HTMLCODE:

```xml
<htmlcode>
  <![CDATA[
    <h1>The Jazz Warehouse</h1>
    <h2>Your Online Store for Jazz Music</h2>
  ]]>]
</htmlcode>
```
CDATA SECTIONS

- May be placed anywhere in the document
- Cannot be nested within other CDATA
- Cannot be empty
- Cannot have “]]>” – this is the ending mark!

Continue coding jazz.xml – page 27
CDATA SECTIONS

This figure shows the revised Jazz.XML file

```
<items>
  <message>
    <![CDATA[Here are some of the latest specials from the Jazz Warehouse. Please note that all Miles Davis & John Coltrane CDs will be on sale for the month of March.]]>
  </message>
</item>
```

CDATA section
PARSING AN XML DOCUMENT

Figure 1-15  Parsing an XML document

The author writes an XML document in an XML editor.

The document is submitted to an XML processor, which evaluates and parses the document (checking for errors in syntax or document structure).

The processed document is then displayed to the user in whatever format is used by the XML processor.
PARSING AN XML DOCUMENT

- XML parsers are very strict => The XML code accepted by one parser should be accepted by other parsers.

- XML parsers interpret pcdata and resolve any character or entity references.
DISPLAYING AN XML DOCUMENT IN A WEB BROWSER

- XML documents can be opened in Internet Explorer or in Netscape Navigator.

- If there are no syntax errors, IE will display the document’s contents in an expandable/collapsible outline format including all markup tags.

- Netscape will display the contents but neither the tags nor the nested elements.

- Older browsers displays only the data content
DISPLAYING AN XML DOCUMENT IN A WEB BROWSER

• To display the Jazz.xml file in a Web browser:

1. Start the browser and open the Jazz.xml file located in the tutorial.01x/tutorial folder of your Data Disk.
2. Click the minus (-) symbols.
3. Click the resulting plus (+) symbols.
DISPLAYING AN XML DOCUMENT IN A WEB BROWSER

Here are some of the latest specials from the Jazz Warehouse. Please note that all Miles Davis & John Coltrane CDs will be on sale for the month of March.

```
<xml version="1.0" encoding="UTF-8" standalone="yes"/>
<!-- This document contains data on Jazz Warehouse special offers -->
<items>
  <message>
    <![CDATA[
      Here are some of the latest specials from the Jazz Warehouse. Please note that all Miles Davis & John Coltrane CDs will be on sale for the month of March.
    ]]>
  </message>
  <item>
    <title>Kind of Blue</title>
    <price>US: $11.99</price>
    <price>UK: £8.99</price>
    <artist>Miles Davis</artist>
    <tracks>
      <track length="9:22">So What</track>
      <track length="9:46">Freddie Freeloader</track>
      <track length="5:37">Blue in Green</track>
      <track length="11:30">All Blues</track>
      <track length="9:26">Flamenco Sketches</track>
    </tracks>
  </item>
  <item>
    <title>Cookin'</title>
    <price>US: $7.99</price>
    <price>UK: £5.59</price>
    <artist>Miles Davis</artist>
    <tracks>
      <track length="5:57">My Funny Valentine</track>
      <track length="5:33">Blues by Five</track>
      <track length="4:22">Airegin</track>
      <track length="13:03">Tune-Up</track>
    </tracks>
  </item>
  <item>
    <title>Blue Train</title>
    <price>US: $6.99</price>
    <price>UK: £6.29</price>
    <artist>John Coltrane</artist>
    <tracks>
      <track length="18:39">Blue Train</track>
      <track length="9:06">Moment's Notice</track>
      <track length="7:11">Locomotion</track>
      <track length="7:55">I'm Old Fashioned</track>
      <track length="7:03">Lazy Bird</track>
    </tracks>
  </item>
</items>
```
LINKING TO A STYLE SHEET

• Link the XML document to a style sheet to format the document. The XML processor will combine the style sheet with the XML document and apply any formatting codes defined in the style sheet to display a formatted document.

• There are two main style sheet languages used with XML:
  – Cascading Style Sheets (CSS) and Extensible Style Sheets (XSL)
CSS x XSL

- XSL more powerful
- CSS easier to learn
- CSS with more browser support for now
LINKING TO A STYLE SHEET

• There are some important benefits to using style sheets:
  – By separating content from format, you can concentrate on the appearance of the document
  – Different style sheets can be applied to the same XML document
  – Any style sheet changes will be automatically reflected in any Web page based upon the style sheet
APPLYING A STYLE TO AN ELEMENT

• To apply a style sheet to a document, use the following syntax:

   \[ \text{selector \{attribute1:value1; attribute2:value2; …\}} \]

• selector is an element (or set of elements) from the XML document.

• \textit{attribute} and \textit{value} are the style attributes and attribute values to be applied to the document.
APPLYING A STYLE TO AN ELEMENT

• For example:

    artist {color:red; font-weight:bold}

• will display the text of the artist element in a red boldface type.
CREATING PROCESSING INSTRUCTIONS

• The link from the XML document to a style sheet is created using a processing statement.

• A processing instruction is a command that gives instructions to the XML parser.
CREATING PROCESSING INSTRUCTIONS

• For example:

```xml
<?xml-stylesheet type="style" href="sheet" ?>
```

• *Style* is the type of style sheet to access and *sheet* is the name and location of the style sheet.
This figure shows the cascading style sheet stored in the jw.css file

```css
message {
  display: block; width: 400px; color: blue; text-align: center;
  font-size: 10pt; font-family: Arial, Helvetica, sans-serif;
  border: 3px solid blue; background-color: ivory;
  margin: 10px; padding: 15px;
}

item {
  display: block; font-size: 14pt; color: red;
  font-family: Arial, Helvetica, Sans-serif;
  margin: 20px;
}

title {
  display: block; font-size: 16pt; color: blue;
  font-weight: bold;
  font-family: Arial, Helvetica, sans-serif
}

priceus, priceuk {
  color: black; font-size: 12pt; font-weight: bold;
  font-family: Times New Roman, Times, Serif;
  margin-left: 20px;
}

artist {
  display: block; font-size: 12pt; color: black;
  font-style: italic; font-weight: bold;
  font-family: Times New Roman, Times, Serif;
  margin-left: 20px
}

track {
  display: list-item; font-size: 9pt; color: black;
  list-style-type: circle;
  font-family: Arial, Helvetica, sans-serif;
  margin-left: 35px
}
```
This figure shows how to link the JW.css style sheet to the Jazz.xml file.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!-- This document contains data on Jazz Warehouse special offers -->
<?xml-stylesheet type="text/css" href="jw.css" ?>

<items>
  <message>
    <![CDATA[
    Here are some of the latest specials from the Jazz Warehouse. Please note that all Miles Davis & John Coltrane CDs will be on sale for the month of March.
    ]]> 
  </message>
</items>
```

processing instruction to access the jw.css style sheet
THE JAZZ.XML DOCUMENT
FORMATTED WITH THE JW.CSS
STYLE SHEET

This figure shows the formatted jazz.xml file.
CASE PROBLEMS

Lab Work

4 Cases – pages 39-45