From HTML to XML (eXtensible Markup Language)

- HTML describes the presentation of the content

```xml
&lt;h1&gt;Bibliography&lt;/h1&gt;
&lt;p&gt;&lt;i&gt;Foundations of Databases&lt;/i&gt;&lt;br&gt;Abiteboul, Hull, and Vianu&lt;br&gt;Addison Wesley, 1995…&lt;/p&gt;
```

- XML describes only the content

```xml
&lt;bibliography&gt;
  &lt;book&gt;
    &lt;title&gt;Foundations of Databases&lt;/title&gt;
    &lt;author&gt;Abiteboul&lt;/author&gt;
    &lt;author&gt;Hull&lt;/author&gt;
    &lt;author&gt;Vianu&lt;/author&gt;
    &lt;publisher&gt;Addison Wesley&lt;/publisher&gt;
    &lt;year&gt;1995&lt;/year&gt;
  &lt;/book&gt;
&lt;/bibliography&gt;
```

- Separation of content from presentation simplifies content extraction and allows the same content to be presented easily in different looks

Other nice features of XML

- Portability: Just like HTML, you can ship XML data across any platforms
  - Relational data requires heavy-weight protocols, e.g., JDBC
- Flexibility: You can represent any information (structured, semi-structured, documents, …)
  - Relational data is best suited for structured data
- Extensibility: Since data describes itself, you can change the schema easily
  - Relational schema is rigid and difficult to change
XML terminology

- Tag names: book, title, ...
- Start tags: <book>, <title>, ...
- End tags: </book>, </title>, ...
- An element is enclosed by a pair of start and end tags: <book>...</book>
  - Elements can be nested:
    <book>, <title>...</title>...
  - Empty elements: <is_textbook/>
  - Can be abbreviated: <is_textbook/>
- Elements can also have attributes: <book ISBN="..." price="80.00">

Well-formed XML documents

A well-formed XML document
- Follows XML lexical conventions
  - Wrong: <section>We show that x < 0.</section>
  - Right: <section>We show that x &lt; 0.</section>
  - Other special entities: > becomes &gt; and & becomes &amp;
- Contains a single root element
- Has tags that are properly matched and elements that are properly nested
  - Right: <section>, <subsection>...</subsection>...</section>
  - Wrong: <section>, <subsection>...</subsection>...</subsection>

More XML features

- Comments: <!-- Comments here -->
- CDATA: <![CDATA[Tags: <book>,...]]>
- ID’s and references

  - <person id="012">name="Homer"/name></person>
  - <person id="023">name="Marge"/name></person>
  - <person id="034">father="012" mother="034">name="Bart"/name></person>
- Namespaces allow external schemas and qualified names

  - <book xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.example.com/xsd" xmlns:myCitationStyle="http://example.com/my-citation-style.xsd">...
- Processing instructions for apps: ? ...java applet... ?
- And more...
Valid XML documents

- A valid XML document conforms to a Document Type Definition (DTD)
- A DTD is optional
- A DTD specifies:
  - A grammar for the document
  - Constraints on structures and values of elements, attributes, etc.

Example:
```
<!DOCTYPE bibliography [ ...
  <!ELEMENT bibliography (book*)>
  <!ELEMENT book (title, author*, publisher?, year?, section*)>
  <!ATTLIST book ISBN CDATA #REQUIRED>
  <!ATTLIST book price CDATA #IMPLIED>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT author (#PCDATA)>
  <!ELEMENT publisher (#PCDATA)>
  <!ELEMENT year (#PCDATA)>
  <!ELEMENT section (title, (#PCDATA), section*)>
]
```

DTD explained
```
<!DOCTYPE bibliography [ ...
  bibliography is the root element of the document
  bibliography consists of a sequence of one or more book elements
  book consists of a title, zero or more authors, an optional publisher, and zero or more sections, in sequence
  book has a required ISBN attribute which is a unique identifier
  book has an optional (IMPLIED) price attribute which contains character data
  Other attribute types include IDREF (reference to an ID), IDREFS (space-separated list of references), enumerated list, etc.
]
```

DTD explained (cont’d)
```
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
<!ELEMENT year (#PCDATA)>
  title, author, publisher, and year all contains parsed character data (#PCDATA)
<!ELEMENT section (title, (#PCDATA), section*)>
  Each section starts with a title, followed by some optional text and then zero or more subsections
```

PCDATA is text that will be parsed (<> will be treated as markup tags and &lt; etc. will be treated as entities); CDATA is unparsed character data.
Using DTD

- DTD can be included in the XML source file
  ```xml
  <xml version="1.0"/>
  <!DOCTYPE bibliography [ ]>
  <bibliography>
  </bibliography>
  ```

- DTD can be external
  ```xml
  <xml version="1.0"/>
  <!DOCTYPE bibliography SYSTEM "../dtds/bib.dtd">
  <bibliography>
  </bibliography>
  ```

Why use DTD's?

- Benefits of using DTD

- Benefits of not using DTD

XML versus relational data

<table>
<thead>
<tr>
<th>Relational data</th>
<th>XML data</th>
</tr>
</thead>
<tbody>
<tr>
<td>schema is always fixed in advance and difficult to change</td>
<td></td>
</tr>
<tr>
<td>simple, flat table structures</td>
<td></td>
</tr>
<tr>
<td>ordering of rows and columns is unimportant</td>
<td></td>
</tr>
<tr>
<td>data exchange is problematic</td>
<td></td>
</tr>
<tr>
<td>&quot;native&quot; support in all serious commercial DBMS</td>
<td></td>
</tr>
</tbody>
</table>