XML - lang for defining other tagged based languages. Derives from SGML.

Using XML DTDs to specify a lang.

To specify a tagged based language:

1. Specify its elements (i.e., its tags)

   Ex) `<!ELEMENT BOOKLIST (BOOK)>
       ↑
       b/w `<BOOKLIST>`   `</BOOKLIST>`

   tags can have 0 or more
   `<BOOK>` `</BOOK>` items

2. Specify the attributes each tag has

   `<!ATTLIST BOOK GENRE
     (Fiction|NonFiction)
     REQUIRED>`

   say book tag has a genre attribute
   `<BOOK GENRE = "NonFiction">

3. Specify entity references — these are abbreviations for common parts of lang.

   Ex) `<!ENTITY nbsp " " >

   `&nbsp;` is an abbreviation for
   (the unicode character `&#160`
<!DOCTYPE Matrix PUBLIC "-//W3C//DTD MathML 1.0//EN" "http://www.w3.org/MATH/DTD/mathml1.dtd">

<!-- "matrices" contain zero or more "matrix" elements. -->
<!-- ELEMENT matrices (matrix) -->

<!-- "matrix" contains one or more "row" elements. -->
<!-- ELEMENT matrix (row+) -->
<!-- "matrix" has 2 attributes which are "numberOfRows" and "numberOfColumns" -->
<!-- ATTLIST matrix
    numberOfRows  CDATA #REQUIRED
    numberOfColumns  CDATA #REQUIRED
 -->

<!-- "row" contains one ore more "column" elements. -->
<!-- ELEMENT row (column+) -->

<!-- "column" contains zoro or one "matrixInColumn" element (in the future it will be able to: -->
<!-- ELEMENT column (matrix?) -->
<!-- "column" has 1 attribute which is "content" (it is a matrix element in the matrix it can -->
<!-- ATTLIST column
    content  CDATA #REQUIRED
 -->

]]>
See XML overhead

Languages that have already been defined: MathML, X3D, SOAP, WML, VoiceML, etc.

**XML Schema**

Notice an XML DTD is very much like specifying a schema for a database. DTDs are actually more specific; for instance, a DTD almost only has only one element in it of #PCDATA (character text).

But... with a database schema can specify int, float, etc.

XML Schema is a new specification on DTDs to handle this. Talk about that later.

---

**Three Tiers Architecture**

What are the ways one connects application w/ databases?
Single Tiers

Main Frame
App
DBMS

Client
Server

Problem: tend to be characterized because graphical displays don't tend to scale well to thousands and up.

Two Tiers

DBMS

Client app

Client app

Net

Now can have GUI's but no central place to update business logic.

Three Tiers

DBMS

Web server
App/Business logic

Client

Presentation Tiers - web pages w/ javascript

Middle tiers - app/business logic
C++ or Java or perl or php, etc

Data Management Tiers - DBMS
More on presentation layer

Learned HTML. Now learn more: forms

```html
<form action="myformhandler.php" method="GET" name="form1">
  <input type="text" name="first-name"/>
  <input type="submit" value="Turn in Form" name="submit"/>
</form>
```

How does this get sent to web server?