Introduction to XML

XML was designed to describe data and focus on what data is.

HTML was designed to display data and focus on how data looks.

What is XML?
XML stands for EXtensible Markup Language
XML is a markup language much like HTML

XML does not DO anything
XML was not designed to DO anything.

XML is used to structure, store and to send information.

```
<note>
   <to>Tom</to>
   <from>Jan</from>
   <heading>Reminder</heading>
   <notebody>Don't forget me this weekend!</notebody>
</note>
```

The tags used to mark up HTML documents are
predefined. The author of HTML documents can only use tags
that are defined. (like <p>, <h1>, etc.).

XML tags are not predefined. You must define your own tags.
<to> and <from> are not defined in any XML standard!

What's the big deal?
Example of data transmission.
You are trying to transmit the following table:

```
------------------------------
ID   Age  HourlyPay RoomNumber
------------------------------
46   25   12        51
48   33   19        36
45   85   14        51
.....
```

We send this table from one computer to another computer.
A transmission error happens. One number gets dropped.
(Number 51 in first line).
The receiving computer sees this:

```
ID   Age  HourlyPay RoomNumber
46   25   12        48
33   19   36        45
85   14   51
```

After the end (hundreds of lines) the computer sees
that there is one number missing.
It has no idea which one. Now one XML way to send this data:

```
<data>
  <row>
     <ID>46</ID>
     <Age>25</Age>
     <HourlyPay>12</HourlyPay>
     <RoomNumber>51</RoomNumber>
  </row>
  <row>
     <ID>48</ID>
     <Age>33</Age>
   ETC. ETC.
</data>
```

Note that this is an extremely wasteful representation!
(We waste disk space and band-width). And yet it is OK.

Now a more complex example of XML:

```
<states drawingBy="Jim">

        <state>
                <name>New Jersey</name>
                <number>1</number>
        </state>




        <state>
                <name>New York</name>


                <number>2</number>
        </state>

        <gov>
                <location>Washington</location>
        </gov>
        <valid_data/>
</states>
```

Tree interpretation of XML:

The very first (and very last) tag become the root of a tree.
There MUST BE a root. ONE ROOT.

Every other matching pair of tags becomes one node.
If a pair of tags is contained in another pair (nested),
the contained pair becomes a child of the containing pair.
Children have a defined order.

Text becomes a child of the node corresponding to the tag that
encloses the text. Text is always a leaf node.
Text has no surrounding box.

XML allows single tags (they are begin and end tag at the
same time). Single tags always become leaves with a BOX around it.
Example: <capital/>

For the above and for the stuff below, see
http://web.njit.edu/~geller/632/Protected/lecture7figures.doc

Box interpretation of XML

All XML elements must be properly nested.
(That means boxes may not overlap!)

In HTML some elements can be improperly nested within each other like
this:

This text is bold and italic

In XML all elements must be properly nested within each other like this:

This text is bold and italic

Even Worse:

In HTML some elements do not have to have a closing tag. The
following code is legal in HTML:

This is a paragraph
This is another paragraph

In XML all elements must have a closing tag:

This is a paragraph
This is another paragraph

XML tags are case sensitive
HTML tags are not.

With XML, the tag <Letter> is different from the tag <letter>.

Opening and closing tags must therefore be written with the same
case.

XML elements can have attribute/value pairs
just like in HTML. In XML the value must
always be quoted with DOUBLE quotes.

```
    attribute  value
        |        |
        |        |
        |        |
        V        V
<note date="12/11/2002">
</note>
```

There are no rules about when to use attributes,
and when to use child elements. You should try to
avoid attributes.

Example:
<note date="12/11/2002"> ..... </note>
is almost the same as:

```
<note>
   <date>12/11/2002</date>
</note>
```

which is considered better.

Rule about attributes has one exception:
You may assign ID references to elements.
These ID references can be used to access XML elements.
Example:

```
  <note ID="p501">
```

One case when you use attributes is if this is information
for the programmer. Example: Author information.

```
<addressbook author="James Geller">
```

This information (author) is not of interest to the USER
of the address book.

With XML, the white space in your document SHOULD NOT be truncated.
(I found this true when programming, but NOT in the browser.)

This is unlike HTML. With HTML, a sentence like this:

Hello   my name is Tom,

will be displayed like this:

Hello my name is Tom,

Use Internet Explorer to check.
Note: Might have to erase the cache when changing a file.

Check public_html/632/test.xml in IE

Comments in XML
The syntax for writing comments in XML is like HTML.

<!-- This is a comment -->

Another example for purposes of naming:

```
<book>
  <title>My First XML</title>
  <prod id="33-657"></prod>
  <chapter>
     Introduction to XML
     <para>What is XML</para>
  </chapter>
</book>
```

<book> has ELEMENT CONTENT, because it contains other elements (tags).

<chapter> has MIXED CONTENT because it contains both:
plain English text and other elements.

<para> has SIMPLE CONTENT because it contains only text.

<prod> has EMPTY CONTENT, because it contains nothing.

XML tags must follow these [simplified] naming rules:

Names can contain letters, numbers, and other characters.
Names must not start with a number or punctuation character.
Names must not start with the letters xml (or XML or Xml ..).
Names cannot contain spaces.
"-" and "." and ":" should not be used in tag names.

Names with an underscore separator are nice.

Examples: <first_name>, <last_name>.
<firstName> <lastName> this is called camel case

A "Well Formed" XML document has correct XML syntax.
("What we learned so far. Almost no constraints."

Single root, correctly nested, correct tags.)

DTDs (Document Type Definitions):

A "Valid" XML document is a "Well Formed" XML document,
which also conforms to the rules of a Document Type Definition (DTD):
A DTD is a little like a database schema for XML files. In brief:

Valid = Well Formed + DTD conformant

```
<?xml version="1.0"?>
<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to      (#PCDATA)>
  <!ELEMENT from    (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body    (#PCDATA)>
]>
<note>
  <to>Tom</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend</body>
</note>
```

The DTD above is interpreted like this:

!ELEMENT note defines the note element as having four elements:
"to,from,heading,body". IN THIS ORDER.

<!ELEMENT to ...> defines the "to" element is of type "#PCDATA".
PCDATA.... PARSED CHARACTER DATA: (A character string)

Empty element. (The BEGIN and END TAG combined into ONE tag.)
<!ELEMENT br EMPTY>   .... line in the DTD   (schema)
XML example:
    .... line in the XML file (data)

Elements with any contents
<!ELEMENT note (any)>

Elements with children (sequence)
<!ELEMENT note (to,from,heading,body)>

Declaring minimum one occurrence of the same element (= one or more)
<!ELEMENT note (message+)>

Declaring zero or more occurrences of the same element
<!ELEMENT note (message*)>

Declaring zero or one occurrences of the same element "optional"
<!ELEMENT note (message?)>

Declaring either/or content (exclusive or)

```
<!ELEMENT note (to,from,header,(message|body))>

Declaring mixed content
<!ELEMENT note (to|from|header|message)*>

----------------
To load a DTD file you use this command:

<!DOCTYPE note SYSTEM "note.dtd">

"note" is the root element.
----------------

NEW EXAMPLES:

First note that there were some differences between IE and
Mozilla.  It's really important to retest all these on IE.

Here is an example that is "three level".
It does not work in IE.
It seemed to work in Mozilla, but I am confused now.

<?xml version="1.0" ?>
<!DOCTYPE note [
  <!ELEMENT note (to,from,message)>
  <!ELEMENT message (heading,body)>
  <!ELEMENT to      (#PCDATA)>
  <!ELEMENT from    (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body    (#PCDATA)>
]>

<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>

So in the above "message" itself does not appear,
because it is itself replaced by heading and body.

The right hand side side is always in (...)
but there can be a *,+,? after the ().




We stop at DTD attributes.  If there is time we do them
in the tutorial.
```