XML Review

Extensible Markup Language

What is XML

- XML stands for eXtensible Markup Language.
- A markup language is used to provide information about a document.
- Tags are added to the document to provide the information.
- HTML tags tell a browser how to display the document
  - Explain how the data should look
- XML tags give a reader some idea what the data means or how it is to be interpreted
  - Ie, XML tags are generally mnemonic
  - Explain how the data should be interpreted

What is XML Used For?

- XML documents are used to transfer data from one place to another often over the Internet.
- XML subsets are designed for particular applications.
- One is RSS (Rich Site Summary or Really Simple Syndication). It is used to send breaking news bulletins from one web site to another.
- A number of fields have their own subsets. These include chemistry, mathematics, and book publishing.
- Many of these subsets are registered with the W3Consortium and are available for anyone's use.
  - Ex: EBXML – a United Nations XML subset for Electronic Business transactions

Advantages of XML

- XML is text (Unicode) based.
  - Often takes up less space when compared to binary
  - Can be transmitted efficiently
  - Easy to parse
  - machine and human readable
- One XML document can be displayed differently in different media.
  - Phone, tablets, print, browsers, etc.
- XML documents can be modularized. Parts can be reused.

Example of an HTML Document

```
<?xml version="1.0"/>
<html>
  <head><title>Example</title></head.
<body>
  <h1>This is an example of a page.</h1>
  <h2>Some information goes here.</h2>
</body>
</html>
```

Example of an XML Document

```
<?xml version="1.0"/>
<address>
  <name>Alice Lee</name>
  <email>alee@aol.com</email>
  <phone>212-346-1234</phone>
  <birthday>1985-03-22</birthday>
</address>
```

Difference Between HTML and XML

- HTML tags have a fixed meaning and **browsers know** what it is.
- XML tags are different for different applications, and **users know** what they mean.
- HTML tags are used for display.
- XML tags are used to describe documents and data.

XML is Rules-based vs Grammar-based

- Tags are enclosed in angle brackets.
- Tags come in pairs with start-tags and end-tags.
- Tags must be properly nested.
  - **<name><email>…</name></email> is not allowed.**
  - **<name><email>…</email><name> is.**
- Tags that do not have end-tags must be terminated by a '/'.
  -
 is an xhtml example
  - Called self-closing tags

More XML Rules

- Tags are case sensitive.
  - **<address> is not the same as <Address>**
- Tags may not contain '<' or '&'.
- Tags follow Java naming conventions
  - They must begin with a letter and may not contain white space.
- Documents must have a single *root* tag that begins the document.

Well-Formed Documents

- An XML document is said to be **well-formed if it follows all the rules**
  - That's really it…
- An XML parser is used to check that all the rules have been obeyed.
- Recent browsers such as Edge, Chrome and Firefox come with XML parsers.
- Parsers are also available for all modern programming languages and environments (ie, Python)

XML Example Revisited

```
<?xml version="1.0"/>
<address>
   <name>Alice Lee</name>
   <email>alee@aol.com</email>
   <phone>212-346-1234</phone>
   <birthday>1985-03-22</birthday>
</address>
```

- Markup for the data aids understanding of its purpose.
- A flat text file is not nearly so clear. For example:

Alice Lee

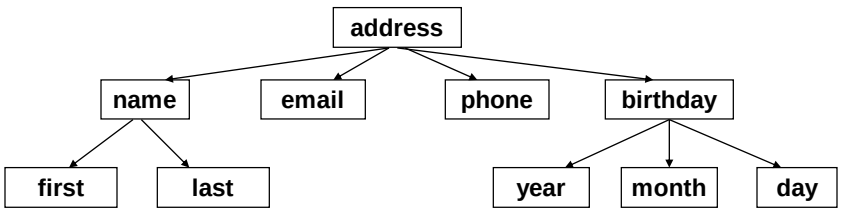
alee@aol.com

212-346-1234

1985-03-22

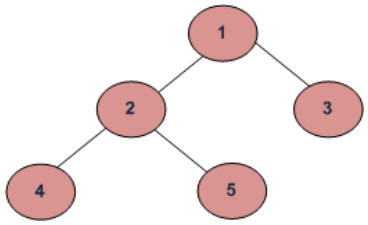
- **The last line looks like a date, but what is it for?**

XML Files are "Trees"

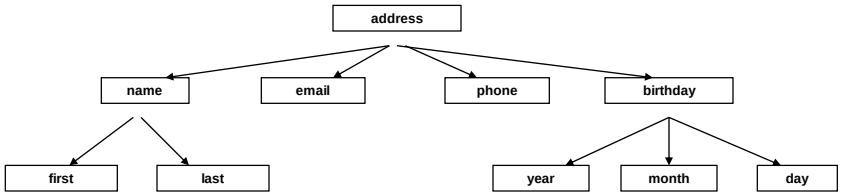



```
<?xml version = "1.0" ?>
<address>
    <name>
        <first>Alice</first>
        <last>Lee</last>
    </name>
    <email>alee@aol.com</email>
    <phone>123-45-6789</phone>
    <birthday>
        <year>1983</year>
        <month>07</month>
        <day>15</day>
    </birthday>
</address>
```

XML Files are "Trees"



- An XML document has a single root node.
- The tree is a general ordered tree.
  - A parent node may have any number of children.
  - Child nodes are ordered, and may have siblings.
- Preorder traversals are usually used for getting information out of the tree.
  - Example: Preorder traversal for the above figure is 1 2 4 5 3.



Validity

- A well-formed document has a tree structure and **obeys all** the XML rules.
- A particular application may add more data rules via an XML schema
- Many specialized schemas have been created to describe particular knowledge areas and domains
  - Business, the sciences, etc.
  - These range from disseminating news bulletins (RSS) to chemical formulas

XML Schemas

- Schemas are themselves XML documents.
- They were standardized after DTDs and provide more information about the document.
- They have a number of data types including string, decimal, integer, Boolean, date, and time.
- They divide elements into simple and complex types.
- They also determine the tree structure and how many children a node may have.

Schema for Address Example

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="address">
  <xs:complexType>
        <xs:sequence>
              <xs:element name="name" type="xs:string"/>
              <xs:element name="email" type="xs:string"/>
              <xs:element name="phone" type="xs:string"/>
              <xs:element name="birthday" type="xs:date"/>
        </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

XML Parsers

- There are two principal models for parsers.
- DOM – Document Object Model
  - Creates an in memory parse tree
  - Requires a **complete** tree traversal
- SAX – Simple API for XML
  - Uses a call-back method
  - May or may not completely traverse a tree – controlled by the user
  - While efficient, SAX parsers are more difficult for the average programmer
- In practice, SAX parsers are used for extremely large documents as they are more memory efficient