

Agenda



- XML
- AJAX

XML Resources



- **w3schools:**

<http://www.w3schools.com/xml/default.asp>

- **Lynda.com:**

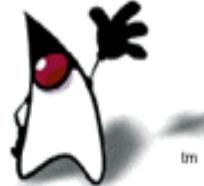
<http://www.lynda.com/XML-tutorials/What-XML/145930/164600-4.html>



XML – what is it?

- XML is e**X**tensible M**a**rku**p** L**a**ngu**age**
- A Markup Language is used to annotate (that is, structure) content in a way that is syntactically distinguishable from the content
- XML is a subset (profile) of SGML, Standard G**e**n**a**ralized M**a**rku**p** L**a**ngu**age**
- SGML itself is for defining markup languages

XML – what's it used for?



- XML is W3C recommendation
- It's for structuring and describing information, emphasizing simplicity, generality and usability across the Internet
- XML is widely used for
 - representing data (structures)
 - facilitating data interchange
 - specifying configuration
 - specifying layout



RSS feed examples

- Ottawa weather

http://weather.gc.ca/rss/city/on-l18_e.xml

- CBC news stories

<http://www.cbc.ca/cmlink/rss-topstories>

More Examples of XML



- SOAP, XHTML, AJAX, MS Office
- http://en.wikipedia.org/wiki/List_of_XML_markup_languages
- XML is not a single language, but is a family of many many many languages

What's an XML language?



- Imagine a printer that accepts content in the form of XML documents, and it produces the appropriate hard copy
- Suppose that printer understands a certain language, it knows attribute **font="courier"** but it does not know **groove="slimey"** or **akh="trap"**
- The set of all possible documents that the printer could understand defines a single XML language



Printers

- We could create a DTD or a Schema (slides below) to concisely define what is a valid document for our printer
- That DTD or Schema would define a single member language of the XML language family
- Oh... speaking of printers:

<http://www.zdnet.com/article/samsung-hops-back-on-android-bandwagon-with-printers/>

SGML vs HTML vs XML



- SGML: Standard Generalized Markup Language
- XML is an application profile of SGML
- Tim Berners-Lee intended HTML to be an application profile of SGML (HTML4)
- HTML and XML look similar, but general HTML violates XML restrictions
 - For example, in HTML opening tags with no corresponding closing tag are allowed, but that's not well-formed XML
- HTML5 and XHTML2 are parallel streams

XML languages are declarative



- **Declarative** solutions concentrate on describing the solution itself
- **Procedural/Imperative** solutions describe steps that should be carried out in order to achieve the solution
- Declarative technology is thought to be less likely to conflict with unforeseen future processing needs and techniques.
- Declarative examples: XML, SQL, regular expressions, logic programming, functional programming, some new Java 8 features

Representing Information with XML



- Single root element
- Elements can have attributes
- Elements can contain other elements

```
<MyElement my_attr="my_attr">  
    <SubElement>  
        Contents of a SubElement  
    </SubElement>  
    <SubElement>  
        Contents of another SubElement  
    </SubElement>  
</MyElement>
```

Business Card example from Lynda.com



- Consider Contact information
 - Name
 - Phone numbers (several?)
 - Email address (several?)

Joe Marini

1-613-555-1212

1-416-555-1234

1-819-555-4321

joe@example.com

With XML markup



```
<Contact>
```

```
    <Name>Joe Marini</Name>
```

```
    <Phone type="home">1-613-555-  
1212</Phone>
```

```
    <Phone type="work">1-416-555-  
1234</Phone>
```

```
    <Phone type="cell">1-819-555-  
4321</Phone>
```

```
    <Email>joe@example.com</Email>
```

```
</Contact>
```

Pieces of an XML document



- XML document declaration
- Elements and Attributes
- Comments
- Entity References (might need)
- Character Data (probably won't need)
- Processing Instructions (probably won't need)

XML document declaration



```
<?xml version="1.0" encoding="UTF-8"  
standalone="yes"?>
```

- optional (but recommended)
- If it's included, it must be the very first thing in the xml file (no characters, not even white space, can come before it)
- Character encoding attribute
- standalone attribute (ignore DTD or not)

Elements (aka tags)



- Names: begin with underscore or a letter, followed by letters, digits, periods, hyphens and underscores
- "XML" cannot begin a tag (no matter what capitalization)
- Case sensitive



Attributes

- Specified on opening tags only
- Same naming constraints as Elements
- An attribute appears one or zero times on any one Element
- Case sensitive

Attributes vs Elements



- These two pieces of XML say basically the same thing:

```
<person sex="female">  
  <firstname>Anna</firstname>  
  <lastname>Smith</lastname>  
</person>
```

versus

```
<person>  
  <sex>female</sex>  
  <firstname>Anna</firstname>  
  <lastname>Smith</lastname>  
</person>
```

Attributes vs Elements (cont'd)



- The first says something like "there is a person of type female with a firstname that is "Anna" and a lastname that is "Smith"
- The second says something like "there is a person with a sex that is "female", a firstname that is "Anna", and a lastname that is "Smith"



Attributes vs Elements

- There are no concrete rules
- The first form considers "sex" to be about the **<person>** tag
- Many prefer the second form, reasoning that information about a person should be represented as an element of a person
- It's not simple, for example, see:

<http://www.ibm.com/developerworks/library/x-eleatt/>

Comments



- <!-- this is a comment -->
- To be read by humans (or machines, I suppose) but not considered part of the document
- Cannot appear inside an Element tag, between the < and >



Entities

- General Entities (defined in a DTD) are replaced by a full string: **©right;**
&author;
- Used to encode otherwise illegal characters
- Character Entities are replaced by a troublesome character: **<** **<** **>**
' **&**
- http://en.wikipedia.org/wiki/List_of_XML_and_HTML_character_entity_references



CDATA

- Character Data Sections
- Part of the document, but not parsed by XML parser
- Begin with **<![CDATA[**
- End with **]]>**
- Used for program code, or any string that could be interpreted as XML-meaningful characters (example: an html document)

Processing Instructions



- Special instructions for the XML parser
- <?targetName instruction ?>
- Example

```
<?SpellCheckMode mode="en-GB" ?>
```

- In our Android XML documents, we shouldn't need these

XML Syntax



- Must have single root element
- Must be well formed (HTML is not, but browsers can deal with it because the tags are well-known)
- Empty tags must be closed with />
- Attributes cannot be minimized
 - cannot say **<option selected>**
 - must say **<option selected="selected">**
- Tags must be properly nested
 - cannot say ** <i> hi </i>**
 - must say **<i> hi </i>**

XML well-formed tester



- An example of a tool that checks whether xml is well-formed
- It does not validate, but does check that XML is well-formed (we'll learn what the difference is later!)

http://www.w3schools.com/xml/xml_validator.asp



XML Namespaces

- We can see a namespace at the top of our Android Manifest file, in the **<Manifest>** tag:

```
<manifest  
    xmlns:android="http://schemas.android.com/apk/res/android"
```

... >

- The value is often in the form of a URL, but just needs to be unique

XML Namespaces (cont'd)



- Namespaces prevent tags from different tag sets from conflicting
- Namespace without the colon is the default
- Namespaces with a colon need to be used with a colon

Namespaces example



```
<table>  
    <tr><td>a</td><td>b</td></tr>  
    <tr><td>c</td><td>d</td></tr>  
</table>
```

versus

```
<table>  
    <type>Coffee</type>  
    <price>199.00</price>  
    <material>wood</material>  
</table>
```

Namespace example (cont'd)



```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:furn="http://www.furniture.org/items">

  <table>
    <tr><td>
      <furn:table>
        <furn:type>Coffee</furn:type>
        <furn:material>maple</furn:material>
      </furn:table>
    </td></tr>
  </table>
</html>
```

CSS and XML



- Cascading Style Sheets can be used with XML to display it

http://www.w3schools.com/xml/xml_display.asp

- There is also XSLT
- eXtensible Stylesheet Language Transformations
- XSLT beyond the scope of this lesson

Well-formed and Valid XML



- **Well-formed:** the document contains XML that parses correctly (none of the above forbidden HTML-isms, for example)
- **Valid:** the document is well-formed and satisfies a set of rules (which define a single XML language)
 - example rules:
 - tag **B** occurs only inside tag **A**
 - attribute **X** can have only values "a", "b", or "c"



Valid XML

- There are two ways to specify these rules for validation
 - I. Document Type Definition document (DTD)
 - old way, not XML syntax
 - more limited
 - 2. XML Schema
 - new way, XML syntax
 - more powerful

XML DTDs



- A DTD constrains XML content (to a specific XML language)
- Can be included inline in an XML document, or can be separate

DTD Syntax and Content



- DTDs contain markup declarations that define document content
- Can specify the valid
 - Elements
 - Attributes
 - Entities
 - Notations
 - Processing Instructions
 - Comments
 - Parameter Entity References



DTD Syntax

- inline

```
<!DOCTYPE Name [  
    Document Type Definition  
]>
```

- in a separate file

```
<!DOCTYPE Name SYSTEM "filename.dtd">
```

(that filename can be a URL)

XML Schemas



- Alternative to DTDs for defining XML languages
- Allow more control over the language
- XML Schemas are written in XML (XML Schemas are XML documents themselves)
- extensible to additions
- support data types
- support namespaces

An XML "note" document with Schema



```
<?xml version="1.0"?>  
  
<note  
    xmlns="http://www.w3schools.com"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
    instance"  
    xsi:schemaLocation="http://www.w3schools.com  
    note.xsd">  
  
    <to>Tove</to>  
  
    <from>Jani</from>  
  
    <heading>Reminder</heading>  
  
    <body>Don't forget me this weekend!</body>  
  
</note>
```

XML Schemas



- We won't go into details on XML Schemas, but take a look at this:

http://www.w3schools.com/xml/xml_schema.asp



Coming soon...AJAX

- Asynchronous Javascript and XMLHttpRequest
- Make web applications more responsive