Kevin Chen

1/4/17

XML Notes

Getting Started

* What is XML?
  + XML = eXtensible Markup Language. W3C Recommendation in 1998. Tag based syntax like HTML, but it is not a replacement for HTML (it is much more general purpose). X means eXtesnible, so you can make your own tags.
  + It is the foundation of several Web technologies. Example: XHTML: HTML formatted as XML syntax. (XML is more strict with syntax.) Also is the basis for RSS and ATOM, which are used for publishing (e.g. blogs). Also the foundation for AJAX (Asynchronous JavaScript and XML). XML is one of the ways of exchanging data over the web using web services.
  + What it is used for: structures and describes information. XML was intended to be used over the internet. Used to exchange data between disparate systems.
* XML-related technologies
  + XPATH – eXtensible Path Language. Similar to how folders and directories on a computer are organized through the use of a path, XPATH finds and extracts information from XML documents.
  + XSLT – XML Stylesheet Language Transformations. Takes XML information and transforms it into something else (pretty much any other file format).
  + XQuery – More advanced querying language than XPath. Like a database language, like SQL except for XML. Used to extract information in a very advanced way from XML data.
  + XPointer/XLink – creates links between and within XML documents.
* Installing the developer tools
  + Any text editor. Development tools: Aptana Studio 3

**How Developers Use XML**

* Describing information with XML
  + “Labels” in the form of tags are used to describe the information.
  + You create your own labels. (X in XML means extensible, so it’s designed to do this)
  + Any XML-aware program can extract data from some XML, even if it doesn’t actually know what the data is.
* Advantages and drawbacks of XML
  + Advantages: XML keeps content separate from presentation. It is an open format that can be read by many applications. Can be used on both the client and server. Has widespread support in multiple languages and runtimes. Makes it possible for disparate systems to exchange data.
  + Drawbacks: not suitable for very large data sets (use databases for very large sets). Some formats, like JSON, may be better in some cases. Some data types, like images, aren’t represented well (these must be encoded, which can get unwieldy). XML can quickly get hard to read when complex.
* Real-world examples of XML
  + RSS is a data format of one of the ways blogs are published. RSS feeds are XML documents that list various items.
  + View page source in order to see the XML of a page.
  + CNN and NY Times use RSS.
  + Used on weather.gov. Doesn’t use RSS, but rather a different dialect of XML.
  + Microsoft Office supports saving files as XML documents.

**Basic Rules of XML**

Types of XML Content

* XML document declaration
  + Example: <?xml version=“1.0” encoding=“UTF-8” standalone=“yes”?>
  + standalone attribute is optional. The declaration is optional, though W3C recommends it.
  + This identifies the file as an XML document. Provides a place for the encoding and standalone attributes. Must be at very beginning – not even whitespace before it.
  + Defaults to UTF-8 if you don’t include it.
  + standalone attribute indicates whether the document is complete by itself.
* Elements and attributes
  + Elements (tags): Must begin with underscore or letter. Can only contain letters, digits, periods, hyphens, and underscores. Can’t contain the word “xml” inside it (with any case combination)
  + Attributes are specified on opening element tags. Same naming rules as elements. Attributes appear only once on a particular element.
* Comments
  + <-!- This is a comment -->
  + They can go everywhere except inside element brackets and before document declaration.
* Character Data Sections (CDATA)
  + They are part of the document, but are not parsed by the XML parser
  + The parser simply skips the content.
  + Defined using <![CDATA[ and ends with ]]>
  + Typically used to contain unescaped textual data.
* Processing Instructions
  + Special instructions to the XML parser
  + Example usage: when the application has options, such as different spellchecking languages, to use a particular set of alphabet.
  + Have the form <?*targetName* instruction ?>
  + The “xml target name is reserved
  + Format: <? *InstructionName InstructionAttributes* ?>
* Entity References
  + Help shorten and modularize XML documents. Provide markup for otherwise illegal characters.
  + General entities are replaced by parser with a full string. Examples: &copyright
  + Character entities &#*characterCode*; Example: &#060;
* Proper XML Syntax