**XML Schemas**

* “Schemas” is a general term--DTDs are a form of XML schemas
  + According to the dictionary, a schema is “a structured framework or plan”
* When we say “XML Schemas,” we usually mean the W3C XML Schema Language
  + This is also known as “XML Schema Definition” language, or XSD
  + I’ll use “XSD” frequently, because it’s short
* DTDs, XML Schemas, and RELAX NG are all XML schema languages

**An XML Schema:**

* **defines elements that can appear in a document**
* **defines attributes that can appear within elements**
* **defines which elements are child elements**
* **defines the sequence in which the child elements can appear**
* **defines the number of child elements**
* **defines whether an element is empty or can include text**
* **defines default values for attributes**

**The purpose of a Schema is to define the legal building blocks of an XML document, just like a DTD.**

**Why XML Schemas?**

* **DTDs provide a very weak specification language**
  + **You can’t put any restrictions on text content**
  + **You have very little control over mixed content (text plus elements)**
  + **You have little control over ordering of elements**
* **DTDs are written in a strange (non-XML) format**
  + **You need separate parsers for DTDs and XML**
* **The XML Schema Definition language solves these problems**
  + **XSD gives you much more control over structure and content**
  + **XSD is written in XML**

**XML schemas are better than DTD**

**XML Schemas**

* **are easier to learn than DTD**
* **are extensible to future additions**
* **are richer and more useful than DTDs**
* **are written in XML**
* **support data types**

**Referencing a schema**

* **To refer to a DTD in an XML document, the reference goes *before* the root element:**
  + **<?xml version="1.0"?>  
    <!DOCTYPE rootElement SYSTEM "*url*">  
    <rootElement> ... </rootElement>**
* **To refer to an XML Schema in an XML document, the reference goes *in* the root element:**
  + **<?xml version="1.0"?>  
    <rootElement  
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
     (The XML Schema Instance reference is required)  
     xsi:noNamespaceSchemaLocation="*url*.xsd">  
     (This is where *your* XML Schema definition can be found)  
     ...  
    </rootElement>**
  + **The file extension is .xsd**
  + **The root element is <schema>**
  + **The XSD starts like this:**
  + **<?xml version="1.0"?>  
    <xs:schema xmlns:xs="http://www.w3.rg/2001/XMLSchema">**
* A “simple” element is one that contains text and nothing else
  + A simple element cannot have attributes
  + A simple element cannot contain other elements
  + A simple element cannot be empty
  + However, the text can be of many different types, and may have various restrictions applied to it
* If an element isn’t simple, it’s “complex”
  + A complex element may have attributes
  + A complex element may be empty, or it may contain text, other elements, or both text and other elements

**Defining a simple element**

* A simple element is defined as  
   <xsd:element name="***name***" type="***type***" />  
  where:
  + ***name*** is the name of the element
  + the most common values for ***type*** are  
     xsd:boolean xsd:integer  
     xsd:date xsd:string  
     xsd:decimal xsd:time
* Other attributes a simple element may have:
  + default="***default value***" *if no other value is specified*
  + fixed="***value***" *no other value may be specified*
* Attributes themselves are always declared as simple types
* An attribute is defined as  
   <xsd:attribute name="***name***" type="***type***" />  
  where:
  + *name* and *type* are the same as for xsd:element

**Defining a Complex type**

* **A complex element is defined as  
   <xs:element name="*name*">  
   <xs:complexType>  
   *... information about the complex type...*  
   </xsd:complexType>  
   </xsd:element>**
* **Example:  
   <xsd:element name="person">  
   <xsd:complexType>  
   <xsd:sequence>  
   <xsd:element name="firstName" type="xsd:string" />  
   <xsd:element name="lastName" type="xsd:string" />  
   </xsd:sequence>  
   </xsd:complexType>  
   </xsd:element>**
* **<xsd:sequence> says that elements must occur in this order**
* **Remember that attributes are always simple types**

**Xs:sequence**

* **an example of a complex type whose elements must occur in a specific order:**

**<xsd:element name="person">  
 <xsd:complexType>  
 <xsd:sequence>  
 <xsd:element name="firstName" type="xsd:string" />  
 <xsd:element name="lastName" type="xsd:string" />  
 </xsd:sequence>  
 </xsd:complexType>  
 </xsd:element>**