**Chapter 1 Introduction to XML**

# What Is XML?

XML (eXtensible Markup Language) is a metalanguage for defining vocabularies (like HTML).

XML is user-defined tags and rules.

# **XML Declaration**

The XML declaration minimally looks like *<?xml version="1.0"?>*

At the begin XML declaration’s encoding attribute isn’t present, an XML parser looks for the byte order-mark (BOM), which is created by an editor program, when it saves the document according to UTF-8 or some other encoding.

Use *<?xml version="1.0" encoding="ISO-8859-1"?>* for a non-English Western European language.

# **Elements and Attributes**

- The structure of an XML document is anchored in a root element (the topmost element).

- Elements can contain child elements, content, or mixed content, also contains whitespace.

- The XML specification permits whitespace to be added to a document.

- Element and attribute names may contain any alphanumeric characters including “\_”, “-“, “.”, “:”. The colon should only be used with namespaces and *names cannot contain whitespace.*

# **Character References and CDATA Sections**

- Certain characters cannot appear literally in the content that appears between a start tag and an end tag or within an attribute value. (<)

=> Character references are classified as numeric character references or character entity references:

+ A numeric character reference

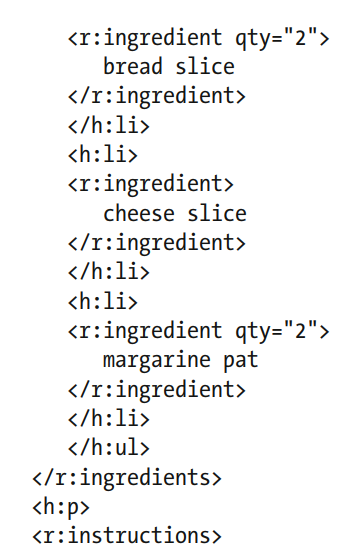
+ A character entity reference

- XML provides an alternative in the form of a CDATA (character data) section, which is a section of literal HTML or XML markup and content surrounded by the <![CDATA[ prefix and the ]]> suffix.

# **Namespaces**

- Namespaces are used to prevent name conflicts when elements and other XML language features appear.

- The namespace URI is associated with a namespace prefix by specifying, typically in an XML document’s root element, either the xmlns attribute by itself or the xmlns:prefix and assigning the URI to this attribute.



- A tag’s attributes don’t need to be prefixed when those attributes belong to the element.

# **Comments and Processing Instructions**

- XML documents can contain comments, which are character sequences beginning with <!—and ending with -->.

- The XML declaration isn’t a processing instruction.

# **Well-Formed Documents**

XML documents has rules:

+ All elements must either have start and end tags or consist of empty-element tags.

+ Tags must be nested correctly.

+ All attribute values must be quoted

+ Empty elements must be properly formatted.

+ XML is a case-sensitive

XML parsers that are aware of namespaces enforce two additional rules:

+ Each element and attribute name must not include more than one colon character.

+ No entity names, processing instruction targets, or notation names (discussed later) can contain colons.

# **Valid Documents**

Grammar documents are written in a special language. Two commonly used grammar languages are Document Type Definition and XML Schema.

## **Document Type Definition**

- Element declarations support three other content specifiers:

*<!ELEMENT name ANY>* to allow any type of element content

*<!ELEMENT name EMPTY>* to disallow any element content.

An element contains mixed content, you would specify *#PCDATA* and a list of element names, separated by vertical bars (|).

**For example,** *<!ELEMENT ingredient (#PCDATA | measure | note)\*>:* ingredient element can contain a mix of parsed character data, zero or more measure elements, and zero or more note elements.

- A document can have internal and external DTDs:

*<!DOCTYPE recipe SYSTEM "http://www.javajeff.ca/dtds/recipe.dtd" [ <!ELEMENT ...>]>.*

- The internal DTD is referred to as the internal DTD subset and the external DTD is referred to as the external DTD subset. Neither subset can override the element declarations of the other subset.

- XML doesn’t allow references to external general entities to appear in attribute values.

## **XML Schema**

* XML Schema is a grammar language for declaring the structure, content, and semantics (meaning) of an XML document.
* XML Schema predefines 19 primitive types: *anyURI, base64Binary, boolean, date, dateTime, decimal, double, duration, float, hexBinary, gDay, gMonth, gMonthDay, gYear, gYearMonth, NOTATION, QName, string, and time.*
* The order of element and attribute declarations isn’t significant within a schema.