Chapter 1:

Introducing XML

**What is XML?**

XML (eXtensible Markup Language ) is a metalanguage (a language used to describe other languages) for defining vocabularies (custom markup languages), which is the key to XML’s importance and popularity.

**Language Features Tour**

XML provides several language features for use in defining custom markup languages: XML declaration, elements and attributes, character references and CDATA sections, namespaces, and comments and processing instructions

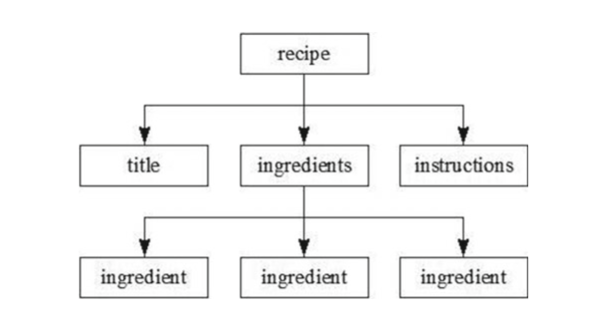
**XML Declaration**The XML declaration minimally looks like <?xml version="1.0"?> in which the nonoptional version attribute identifies the version of the XML specification to which the document conforms. The initial version of this specification (1.0) was introduced in 1998 and is widely implemented.

An XML document usually begins with the XML declaration, which is special markup telling an XML parser that the document is XML

XML supports Unicode, which means that XML documents consist entirely of characters taken from the Unicode character set

The final attribute that can appear in the XML declaration is standalone

**Elements and Attributes**



The XML declaration is a hierarchical (tree) structure of elements

Unlike in HTML, you can choose the root element for your XML documents

Elements can contain child elements, content, or mixed content

An XML element’s start tag can contain one or more attributes

**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

Solution:

-A numeric character reference refers to a character via its Unicode code point and adheres to the format &#nnnn

-A character entity reference refers to a character via the name of an entity (aliased data) that specifies the desired character as its replacement text

A CDATA section 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.

A namespace is a Uniform Resource Identifier (URI) -based container that helps differentiate XML vocabularies by providing a unique context for its contained identifiers

**Comments and Processing Instructions**

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

Comments are used to clarify portions of a document.

Comments are not content.

XML also permits processing instructions to be present. A processing instruction is an instruction that’s made available to the application parsing the document.

**Well-Formed Documents**

XML mandates that XML documents follow certain 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

-Be careful with case

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.

An XML document that conforms to these rules is well formed

**Valid Documents**

A valid document adheres to constraints.

A parser that performs validation compares an XML document to a grammar document

**Document Type Definition**

Document Type Definition (DTD) is the oldest grammar language for

specifying an XML document’s grammar. DTD grammar documents (known as DTDs) are written in accordance to a strict syntax that states what elements may be present and in what parts of a document, and also what is contained within elements (child elements, content, or mixed content) and what attributes may be specified

A DTD-based validating XML parser requires that a document include a document type declaration identifying the DTD that specifies the document’s grammar before it will validate the document.

A document type declaration appears immediately after the XML declaration and is specified in one of the following ways:

-<!DOCTYPE root-element-name SYSTEM uri> references an external but private DTD via uri.

-<!DOCTYPE root-element-name PUBLIC fpi uri> references an external but public DTD via fpi, a formal public identifier

-<!DOCTYPE root-element [ dtd ]> references an internal DTD, one that is embedded within the XML document.

A notation is an arbitrary piece of data that typically describes the format of unparsed binary data

It’s also common to use notations to specify binary data types via media types

General entities are entities referenced from inside an XML document via general entity references, syntactic constructs of the form &name

General entities are classified as internal or external

Parameter entities are classified as internal or external.

**XML Schema**

XML Schema is a grammar language for declaring the structure ,content, and semantics (meaning) of an XML document

XML Schema provides restriction (reducing the set of permitted values through constraints), list (allowing a sequence of values), and union (allowing a choice of values from several types) derivation methods for creating new simple types from these primitive types.

XML Schema requires that each element have a name and (unlike DTD) be associated with a type, which identifies the kind of data stored in the element.