**What is XML:**

XML vocabulary documents are like HTML, documents in that they are text-based and consist of markup(encoded descriptions of a document’s logical structure) and content(document text not interpreted as markup). Markup is evidenced via tags(angle bracket-delimited syntactic constructs) and each tag has a name. Furthermore, some tags have attributes (name-value pairs).

The key difference between XML and HTML is that XML invites you to create your own vocabularies with its own tags and rules, whereas HTML gives you a single precreated vocabulary with its own fixed set of tags and rules.

**Language Features Tour:**

**XML Declaration:**

*<?xml version="1.0"?>*

The initial version of this specification (1.0) was introduced in 1998 and is widely implemented.

*<?xml version="1.0" encoding="ISO-8859-1"?>*

When your native language isn’t English or when you’re called to create XML documents that include non-ASCII characters, you need to properly specify encoding.

**Element and Attributes:**

An *element* is a portion of the document delimited by a *start tag* (such as <name>) and an *end tag* (such as </name>), or is an *empty-element tag* (a standalone tag whose name ends with a forward slash (/), such as <break/>).

Elements can contain child elements, content, or *mixed content* (a combination of child elements and content).

Attributes provide additional details about elements.

**Character References and CDATA Sections:**

Cannot place a literal < character between a start tag and an end tag. One solution to this problem is to replace the literal character with a *character reference* For example & #0931 represent Σ the Greek capital letter sigma.

<expression>6 < 4</expression>. You could replace the < with numeric reference & #60.

Suppose you want to embed an HTML or XML document within an element. XML provides an alternative in the form of a CDATA (character data) section. 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. Without namespaces, an XML parser couldn’t distinguish between same-named elements or other language features that mean different things, such as two same-named title elements from two different languages.

**Comments and Processing Instructions:**

XML documents can contain *comments*, which are character sequences beginning with <!-- and ending with -->. They can appear anywhere after the XML declaration except within tags, cannot be nested.

*instruction* is an instruction that’s made available to the application parsing the document. The instruction begins with <? and ends with ?>.

**Well-Formed Documents:**

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

Tags must be nested correctly*. <b><i>XML</b></i>*, XML parser would report an error.

*All attribute values must be quoted.* Either single quotes (') or double quotes (") are permissible.

*Empty elements must be properly formatted.*

*Be careful with case.* XML is a **case-sensitive** language in which tags differing in case

Namespace 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.

**Document Type Definition:**

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.