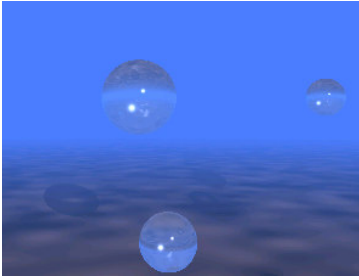


XML Fundamentals



XML, or the extensible markup language, is a W3C recommended standard mark up language designed to facilitate the communication of information between a wide range of systems. XML has emerged as the standard means for communicating data in a non-system specific way. Its historical roots are in Standard General Mark-Up Language (SGML), which means it demonstrates some characteristics with the well-known HTML.

In essence, XML is a language consisting of tags and attributes. A tag is a unit of description. An attribute describes a tag. An example tag may be `<student>`. Attributes of the tag `<student>` may be *age* and *graduation date*. From both a syntactical and conceptual perspective, attributes are contained within the tag. The tag's content is anything that sits between the beginning tag and the end tag. In our example, the start tag may be described as `<student>` and the end tag as `</student>`. These tags describe data. Anything inside the tag is the tag's data. To that extent, HTML and XML are similar.

In implementation, it is the rules of XML that constitute XML. This includes not only the syntax but also the form. Since XML is only a set of rules, it does not prescribe an exhaustive list of core tags, as does HTML. There are a few XML reserved tags that provide standard information about an XML document, but these are not essential to the successful use of an XML data source. XML authors, for all intensive purposes, define their own tags based on a standard set forth by a standardizing body, an architect, or the developer. This full flexibility and extensibility allows for semantically rich markup.

XML source data is typically contained within a text file. It conveys relationships between its content by the relative placement of that content to other content within the document. All that can be known about an XML document is known through its structure and content. This is why XML has sometimes been coined a language of relationships.

If XML has been modeled and authored well, an untrained eye will understand it. Since it is a text-formatted document, it can also be parsed and understood by a wide range of computer platforms and languages. Because XML has this wide range of interoperability, it is excellent for the aggregation of data from multiple sources, and for conversely disseminating information to multiple external clients. As noted in the introduction to the W3C's XML recommendation, "XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere"(W3C).

The core goals of XML as stated by the W3C recommendation are:

- 1.XML shall be straightforwardly usable over the Internet.
- 2.XML shall support a wide variety of applications.
- 3.XML shall be compatible with SGML [Standard Generalize Markup].
- 4.It shall be easy to write programs which process XML documents.
- 5.The number of optional features in XML is to be kept to the absolute minimum, ideally zero.
- 6.XML documents should be human-legible and reasonably clear.
- 7.The XML design should be prepared quickly.
- 8.The design of XML shall be formal and concise.
- 9.XML documents shall be easy to create.
- 10.Terseness in XML markup is of minimal importance. (W3C)

(Table 1 – The Core Goals of XML)

Namespaces are another important aspect of the XML recommendation. The namespace is simply a means of associating an XML vocabulary with a context. Consider the English word *project*. It can be used as a verb, as in, *to project one's voice*, or a noun, as in *my software project*. A namespace allows an author to indicate that the word is being used as a verb or as noun. In other words, it would indicate that this instance of *project* refers to the vocabulary of *valid English nouns* or *valid English verbs*. When using the word *project*, you might declare these namespaces in your document, and then specify to which of these namespaces (verb or noun) this *project* belongs. Beginning XML reminds us that namespaces are a "purely abstract entity; it's nothing more than a group of names that belong with each other conceptually" (Cagle, 68).

XML Syntax Fundamentals

XML is a language of relationships. Each element in the document has a relationship to each other element in the document, and each node within the document can be described as parent, child, or sibling to another element. Each tag should describe its content, and the nesting of tags should describe their relationships and the relationships of the data.

XML's ability to describe these relationships is supported by one core rule. This is the rule of well-formedness. Any functional XML document must be well-formed. This means that every beginning tag is closed in the order it was presented. If tags are nested, then, internal tags must be closed before the parent tag is closed.

Consider the following sample XML code:

```
<product>
  <description>Software</description>
  <productnumber>773412039</productnumber>
</product>
```

(Figure 1 – Sample of Well-Formed XML)

This document is symmetrically labeled. Since XML is case sensitive, the closing and end tags must have the same case. The following document segment is not well-formed:

```
<product>
  <description>Software</description>
  <productname >Microsoft Office
  <productnumber>773412039</productnumber>
</PRODUCT>
```

(Figure 2 – Sample of XML, not Well Formed)

This mark up lacks a closing tag for the <productname> element and the <product> end tag does not have the same case as the <product> start tag. The document segment does not adhere to the syntactical rules of XML, and would therefore not be considered usable XML.

Other fundamental constraints of XML well-formedness include:

- All attribute values must be enclosed in double quotation marks
- Attribute names must not be repeated in the same element.
- XML special characters can not be used as XML data (e.g. <, >, &)

When an XML document has met these criteria, it is considered “well-formed.” If an XML document is not well-formed, it is not usable by XML parsers.

XML Validity

An XML document must also be valid. A valid XML document is one that adheres to the rules of relationships and content specified by the XML schema. A validating parser checks the validity of an XML document using the validation rules specified in a separate document.

XSL

XSL, or the extensible style sheet language, is the standard means of reformatting an XML document. Currently, XSL supports the transformation of XML data into a new XML, HTML, or text document. It is important to note that XSL is a language. A well authored XSL typically contains logic, variables, and other artifacts of traditional programming languages.

XSL syntactical rules are very similar to those of XML. All XSL documents must be well-formed. XSL documents are typically validated against the XSL namespace. The XSL namespace contains all of the valid XSL specific elements and attributes. There are nearly one hundred XSL reserved keywords that define the syntactical language of XSL.

XSL is used on the client-side and the server side. Because parsing an XML document and applying a stylesheet is a memory intensive process, XSL transformations are most often executed on the server side. Server side transformation also means that client machines see no change in the type of data they must consume.