XML

eXtensible Markup Language.

**.**XML was designed to store and transport data.

**.**XML was designed to be both human- and machine-readable.

For example;

<?xml version="1.0" encoding="UTF-8"?>  
<note>  
  <to>Gurpreet</to>  
  <from>Atul</from>  
  <heading>Reminder</heading>  
  <body>you are my buddy</body>  
</note>

**.**XML plays an important role in many different IT systems.

**.**XML is often used for distributing data over the Internet.

**.**It is important (for all types of software developers!) to have a good understanding of XML.

**The Difference Between XML and HTML**

XML and HTML were designed with different goals:

* XML was designed to carry data - with focus on what data is
* HTML was designed to display data - with focus on how data looks
* XML tags are not predefined like HTML tags are

**XML Does Not Use Predefined Tags**

**.**The XML language has no predefined tags.The tags in the example above (like <to> and <from>) are not defined in any XML standard. These tags are "invented" by the author of the XML document.

**.**HTML works with predefined tags like <p>, <h1>, <table>, etc.

**.**With XML, the author must define both the tags and the document structure.

**XML is Extensible**

**.**Most XML applications will work as expected even if new data is added (or removed).

**.**Imagine an application designed to display the original version of note.xml (<to> <from> <heading> <data>).

**.**Then imagine a newer version of note.xml with added <date> and <hour> elements, and a removed <heading>.

**.**The way XML is constructed, older version of the application can still work:

<note>  
  <date>2015-09-01</date>  
  <hour>08:30</hour>  
  <to>Tove</to>  
  <from>Jani</from>  
  <body>Don't forget me this weekend!</body>  
</note>

# XML Syntax Rules

* **All XML elements must have a closing tag.**

For example;

<p>This is a paragraph.  
<br>

* **XML tags are case sensitive.**

For example;

<Message>This is incorrect</message>  
<message>This is correct</message>

* **All XML elements must be properly nested.**

For example;

<b><i>This text is bold and italic</b></i>

* **All XML documents must have a root element.**

For example;

<b><i>This text is bold and italic</b></i>

* **Attribute values must always be quoted.**

for example;

<note date=12/11/2007>  
  <to>Tove</to>  
  <from>Jani</from>  
</note>

**There are 5 pre-defined entity references in XML:**

|  |  |  |
| --- | --- | --- |
| &lt; | < | less than |
| &gt; | > | greater than |
| &amp; | & | ampersand |
| &apos; | ' | apostrophe |
| &quot; | " | quotation mark |

Only < and & are strictly illegal in XML, but it is a good habit to replace > with &gt; as well.

# XML - Elements

XML elements can be defined as building blocks of an XML. Elements can behave as containers to hold text, elements, attributes, media objects or all of these.

Each XML document contains one or more elements, the scope of which are either delimited by start and end tags, or for empty elements, by an empty-element tag.

## Syntax

Following is the syntax to write an XML element:

<element-name attribute1 attribute2>

....content

</element-name>

where

* **element-name** is the name of the element. The *name* its case in the start and end tags must match.
* **attribute1, attribute2** are attributes of the element separated by white spaces. An attribute defines a property of the element. It associates a name with a value, which is a string of characters. An attribute is written as:

name = "value"

*name* is followed by an *=* sign and a string *value* inside double(" ") or single(' ') quotes.

## Empty Element

An empty element (element with no content) has following syntax:

<name attribute1 attribute2.../>

Example of an XML document using various XML element:

<?xml version="1.0"?>

<contact-info>

<address category="residence">

<name>Tanmay Patil</name>

<company>TutorialsPoint</company>

<phone>(011) 123-4567</phone>

<address/>

</contact-info>

## XML Elements Rules

Following rules are required to be followed for XML elements:

* An element *name* can contain any alphanumeric characters. The only punctuation mark allowed in names are the hyphen (-), under-score (\_) and period (.).
* Names are case sensitive. For example, Address, address, and ADDRESS are different names.
* Start and end tags of an element must be identical.
* An element, which is a container, can contain text or elements as seen in the above example.

**XML Attributes**

XML attributes are normally used to describe XML elements, or to provide additional information about elements. From HTML you can remember this construct: <IMG SRC="computer.gif">. In this HTML example SRC is an attribute to the IMG element. The SRC attribute provides additional information about the element.

Attributes are always contained within the start tag of an element. Here are some examples:

|  |
| --- |
| HTML examples:  <img src="computer.gif">  <a href="demo.asp">  XML examples:  <file type="gif">  <person id="3344"> |

**XML Elements vs. Attributes**

Take a look at these examples:

<person gender="female">  
  <firstname>Anna</firstname>  
  <lastname>Smith</lastname>  
</person>

<person>  
  <gender>female</gender>  
  <firstname>Anna</firstname>  
  <lastname>Smith</lastname>  
</person>

In the first example gender is an attribute. In the last, gender is an element. Both examples provide the same information.

There are no rules about when to use attributes or when to use elements in XML.

**XML DTD**

An XML document with correct syntax is called "Well Formed".

An XML document validated against a DTD is both "Well Formed" and "Valid".

**Valid XML Documents**

A "Valid" XML document is a "Well Formed" XML document, which also conforms to the rules of a DTD:

<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE note SYSTEM "Note.dtd">  
<note>  
<to>Tove</to>  
<from>Jani</from>  
<heading>Reminder</heading>  
<body>Don't forget me this weekend!</body>  
</note>

The DOCTYPE declaration, in the example above, is a reference to an external DTD file.

**.**The purpose of a DTD is to define the structure of an XML document. It defines the structure with a list of legal elements:

<!DOCTYPE note  
[  
<!ELEMENT note (to,from,heading,body)>  
<!ELEMENT to (#PCDATA)>  
<!ELEMENT from (#PCDATA)>  
<!ELEMENT heading (#PCDATA)>  
<!ELEMENT body (#PCDATA)>  
]>

The DTD above is interpreted like this:

* !DOCTYPE note defines that the root element of the document is note
* !ELEMENT note defines that the note element must contain the elements: "to, from, heading, body"
* !ELEMENT to defines the to element to be of type "#PCDATA"
* !ELEMENT from defines the from element to be of type "#PCDATA"
* !ELEMENT heading defines the heading element to be of type "#PCDATA"
* !ELEMENT body defines the body element to be of type "#PCDATA"

PCDATA means parse-able text data.

## Using DTD for Entity Declaration

A doctype declaration can also be used to define special characters and character strings, used in the document:

### Example

<?xml version="1.0" encoding="UTF-8"?>  
  
<!DOCTYPE note [  
<!ENTITY nbsp "&#xA0;">   
<!ENTITY writer "Writer: Donald Duck.">  
<!ENTITY copyright "Copyright: W3Schools.">  
]>  
  
<note>  
<to>Tove</to>  
<from>Jani</from>  
<heading>Reminder</heading>  
<body>Don't forget me this weekend!</body>  
<footer>&writer;&nbsp;&copyright;</footer>  
</note>.