**XML:**

**Components in XML:** The following are the different components which together form an XML**.**

* Tags
* Elements
* Attributes
* References
* Comments
* Namespaces

**Tags:**

* Start tag:  Any tag in an XML file starts with a start tag. The following text colored in red represents a start tag:
  + <name> QAFox </name>
* End tag: Any tag in an XML file which is used for ending a start tag. The following text colored in red represents an end tag in XML:
  + <name> QAFox </name>

And, / in the end tag is known as slash or solidus

**Elements:**

Elements in XML must start with a letter or underscore and can have letters, digits, period, underscore etc .. but not white space. The following is an example of an element:

<name>sample</name>

Text**‘**name’ called in the above example is an element name, whereas text ‘sample’ is **element value**

Following are the rules to keep in mind while writing the XML elements:

1. XML element names are case sensitive
2. XML document should have one root element
3. Elements should not be overlapped

**Following is the structure of XML elements:**

*<parent>*

*<child>*

*<sub-child>…………..</sub-child>*

*</child>*

*</parent>*

**Understanding XML declaration or prolog:**

*<? xml  version=”1.0” encoding=”UTF-8” ?>*is the XML declaration or prolog of XML. Which is the very first statement in XML. It is explained indetailed below:

**version=”1.0”**– Version of XML

There are two versions of XML i.e. 1.0 and 1.1. Mostly used one is 1.0 and 1.1 only used when one needs any special features like using non-ASCII characters. We can find non-ascii characters here:

<http://rbutterworth.nfshost.com/Tables/common>

**UTF**–  It is an encoding type and is nothing but Universal Character Set Transformation Format.

* UTF itself has two versions i.e. UTF-8 and UTF-16.
* 8 and 16 represent the number of bits used to represent a character.

We have other encoding types along with UTF. The following is the list:

* UTF-8 and UTF-16
* US-ASCII
* ISO-8859-1
* ISO-8859-2
* Big5
* And others. All the other types are mentioned

here: <https://www.iana.org/assignments/character-sets/character-sets.xml>

XML declaration statement *<? xml  version=”1.0” encoding=”UTF-8” ?>*doesn’t have any end tag, hence it is not an element and can be called as a declaration.

**Following rules to be followed while writing the above declaration:**

* It is optional. If it appears in a document then it should be on top. Not even whitespace or comment come before it
* All XML parsers are required to support UTF-8 and UTF-16
* It is case sensitive

**XML Attributes:**

Property of XML element called as XML attributes. It consists of name and value separated by an equal sign. Following example elaborates about XML attribute:

*<book name=”mine”>*

The text name=”mine” is nothing but an XML attribute. And, text ‘name’ is attribute name and text ‘mine’ is attribute value and it is should always be in single or double quotes.

Few rules of XML attributes to keep in mind are:

1. The attribute is only placed inside the start tag with white space associated
2. Only one value for the same attribute in the same start-tag

**XML References:**

A reference is used to add additional text or markup in an XML document, where an error may occur if the character is typed directly. Such as ***< , &, >,*** etc.. can be replaced by ***&amp;, &lt;, &gt;,*** etc. References start with **& (ampersand)** and end with **;(semicolon)**

**References are of two types:**

* Entity References
* Character References

**Entity references:**Starts with **& (ampersand)** and ends with **;(semicolon).**Below example illustrates entity reference:

If you write <salary> salary < 100 </salary>, throws XML error. Hence it can be fixed using entity reference as <salary>salary **&lt;** 1000</salary>.

* Here, ‘**&lt;**’ is the equivalent of ‘**<**’ symbol
* List of entity references are shown here: <http://xml.silmaril.ie/specials.html>

**Character references:**starts with **&#**and ends with **;** (semicolon).

* Examples: &#97→a, &#65→A, &#90→ z etc.
* List of character references are shown here: <http://myhandbook.info/codes_htmlchr.html>

**XML Comments:**Declaration of XML comments is shown below:

<!-comment goes here->

XML comments are same like any other language comments. Which are added as notes or lines for understanding the purpose of the code. Also, comments can be used to include terms and conditions links or any other related links. They may appear anywhere in the code.

**Follow below instructions while writing comments in XML:**

* Comments are placed anywhere expect inside tag and top of the XML
* XML comments cannot be nested

**Defining Empty XML elements:** An XML element which has no value is known as empty XML element. The following are the examples for empty XML elements:

*<element></element>  or <element/>*

**XML Namespaces:**It provides a way to avoid element name conflicts.

**XML 1:**

<message>

<list>item1</list>

<list>item2</list>

</message>

**XML 2:**

<message>

<list>first</list>

<list>second</list>

</message>

Name conflict will occur when you are trying to mix XML documents from different applications like above XML 1 and XML 2.  To avoid such situations we can add prefix like below:

**XML 1:**

<f:message>

<f:list>item1</f:list>

<f:list>item2</f:list>

</f:message>

**XML 2:**

<s:message>

<s:list>first</s:list>

<s:list>second</s:list>

</s:message>

When using these prefixes in XML, namespace must be defined. Which can be defined by using attribute called ‘xmlns’ in the start tag of an element. And the syntax is shown below:

xmlns:prefix = “URI”

Example: The following is an example for using the above syntax:

<root>

<f:message xmlns:f=”http://www.qafox.com”>

<f:list>item1</f:list>

<f:list>item2</f:list>

</f:message>

<s:message xmlns:s=”http://www.qafox.com”>

<s:list>first</s:list>

<s:list>second</s:list>

</s:message>

</root>

When any prefix is mentioned with xmlns then it will become a qualified namespace. Namespaces can also be declared in the XML root element. Defining a default namespace for an element saves us from using prefixes in all the child elements.  It is shown below:

<message xmlns=”[http://www.qafox.com](http://www.qafox.com/)”>

<list>item1</list>

<list>item2</list>

</message>

**XML constraints:**

1. XML was not designed to display data like HTML.
2. XML doesn’t have predefined tags.

**Uses of XML:**

1. XML can separate data from HTML.
2. XML is used to exchange data
3. XML can be used to share data
4. XML can be used to store data
5. XML can make your data more useful
6. XML can be used to create new \*ML languages

List of \*ML languages are shown here:

<https://en.wikipedia.org/wiki/List_of_XML_markup_languages>

**Elements vs attributes:**

1. Attributes cannot contain multiple values – child elements can
2. Attributes are not easily expandable – for future changes
3. Attributes cannot describe structures—child elements can
4. Attributes are more difficult to manipulate by program code
5. Attribute values are not easy to test against Document Type Definition – DTD: Which is used to define legal elements of an XML document.

**XML Example using all the above expalined components:**

<?xml version=“1.0” encoding=“UTF-8”?>

<!– It is a sample XML document which includes components of XML –>  
<!DOCTYPE note [  
 <!ENTITY writer SYSTEM “https://www.w3schools.com/entities.dtd”>  
 <!ENTITY copyright SYSTEM “https://www.w3schools.com/entities.dtd”>  
]>  
<bookstore>  
  <book category=“children”>  
     <title>Harry Potter</title>  
     <author>J K. Rowling</author>  
     <year>2005</year>  
     <price>29.99</price>  
  </book>  
  <h:book xmlns:h=“http://www.w3.org/TR/html4/”>  
     <h:tr>  
        <h:td>Book1</h:td>  
        <h:td>Book2</h:td>  
     </h:tr>  
  </h:book>  
  <author>&writer;&copyright;</author>  
</bookstore>

Above example illustrates all the above components of XML. Brief of them given below:

* **Tags**shown in blue color. Ex: <bookstore>, <author>, <price>, <h:book>,</h:book>,</bookstore>
* **Element**is a combination of start tag, end tag and data i.e. <h:td>Book1</h:td>’,’<title>Harry Potter</title>’
* **Attributes**mentioned inside start tag i.e.  <book category=“children”> – category is an attribute name and children is attribute value.
* **References** mentioned as ‘&writer;&copyright;’ these set in DOCTYPE note declaration in the top of XML
* **Comments** mentioned as <!– It is a sample XML document which includes components of XML –>
* **Namespaces**mentioned in <h:book xmlns:h=“<http://www.w3.org/TR/html4/>“> and ‘xmlns:h=“<http://www.w3.org/TR/html4/>“’ is a namespace.