# XML - Documents

An XML *document* is a basic unit of XML information composed of elements and other markup in an orderly package. An XML *document* can contains wide variety of data. For example, database of numbers, numbers representing molecular structure or a mathematical equation.

A XML DOCUMENT CONSITS OF

1. Elements

2. Attributes

3. Entity References

XML Document Example

A simple document is shown in the following example −

<?xml version = "1.0"?>

<contact-info>

<name>Tanmay Patil</name>

<company>TutorialsPoint</company>

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

</contact-info>

The following image depicts the parts of XML document.



Document Prolog Section

**Document Prolog** comes at the top of the document, before the root element. This section contains −

* XML declaration
* Document type declaration

Document Elements Section

**Document Elements** are the building blocks of XML. These divide the document into a hierarchy of sections, each serving a specific purpose. You can separate a document into multiple sections so that they can be rendered differently, or used by a search engine. The elements can be containers, with a combination of text and other elements.

XML **Character Entities**

This means, entities are the placeholders in XML. These can be declared in the document prolog or in a DTD. There are different types of entities and in this chapter we will discuss Character Entity.

Both, HTML and XML, have some symbols reserved for their use, which cannot be used as content in XML code. For example, **<** and **>** signs are used for opening and closing XML tags. To display these special characters, the character entities are used.

There are few special characters or symbols which are not available to be typed directly from the keyboard. Character Entities can also be used to display those

symbols/special characters.

## Types of Character Entities

There are three types of character entities −

* Predefined Character Entities
* Numbered Character Entities
* Named Character Entities

### Predefined Character Entities

They are introduced to avoid the ambiguity while using some symbols. For example, an ambiguity is observed when less than ( **<** ) or greater than ( **>** ) symbol is used with the angle tag (**<>**). Character entities are basically used to delimit tags in XML. Following is a list of pre-defined character entities from XML specification. These can be used to express characters without ambiguity.

* Ampersand − **&amp;**
* Single quote − **&apos;**
* Greater than − **&gt;**
* Less than − **&lt;**
* Double quote − **&quot;**

### Numeric Character Entities

The numeric reference is used to refer to a character entity. Numeric reference can either be in decimal or hexadecimal format. As there are thousands of numeric references available, these are a bit hard to remember. Numeric reference refers to the character by its number in the Unicode character set.

General syntax for decimal numeric reference is −

&# decimal number ;

General syntax for hexadecimal numeric reference is −

&#x Hexadecimal number ;

The following table lists some predefined character entities with their numeric values −

| **Entity name** | **Character** | **Decimal reference** | **Hexadecimal reference** |
| --- | --- | --- | --- |
| quot | " | &#34; | &#x22; |
| amp | & | &#38; | &#x26; |
| apos | ' | &#39; | &#x27; |
| lt | < | &#60; | &#x3C; |
| gt | > | &#62; | &#x3E; |

### Named Character Entity

As it is hard to remember the numeric characters, the most preferred type of character entity is the named character entity. Here, each entity is identified with a name.

For example −

* 'Aacute' represents capital https://www.tutorialspoint.com/xml/images/namedentity_1.png character with acute accent.
* 'ugrave' represents the small https://www.tutorialspoint.com/xml/images/namedentity_2.png with grave accent.

Another Example of XML DOCUMENT: Books

*File: books.xml*

**<bookstore>**

**<book** category="COOKING"**>**

**<title** lang="en"**>**Everyday Italian**</title>**

**<author>**Giada De Laurentiis**</author>**

**<year>**2005**</year>**

**<price>**30.00**</price>**

**</book>**

**<book** category="CHILDREN"**>**

**<title** lang="en"**>**Harry Potter**</title>**

**<author>**J K. Rowling**</author>**

**<year>**2005**</year>**

**<price>**29.99**</price>**

**</book>**

**<book** category="WEB"**>**

**<title** lang="en"**>**Learning XML**</title>**

**<author>**Erik T. Ray**</author>**

**<year>**2003**</year>**

**<price>**39.95**</price>**

**</book>**

**</bookstore>**

The root element in the example is <bookstore>. All elements in the document are contained within <bookstore>.

The <book> element has 4 children: <title>,< author>, <year> and <price>.