What is xml

* **Xml** (eXtensible Markup Language) is a mark up language.
* XML is designed to store and transport data.
* Xml was released in late 90’s. it was created to provide an easy to use and store self describing data.
* XML became a W3C Recommendation on February 10, 1998.
* XML is not a replacement for HTML.
* XML is designed to be self-descriptive.
* XML is designed to carry data, not to display data.
* XML tags are not predefined. You must define your own tags.
* XML is platform independent and language independent.

# Features and Advantages of XML

## 1) XML separates data from HTML

## 2) XML simplifies data sharing

## 3) XML simplifies data transport

#### 4) XML simplifies Platform change

## 5) XML increases data availability

## Example of XML Document

XML documents uses a self-describing and simple syntax:

1. **<?xml** version="1.0" encoding="ISO-8859-1"**?>**
2. **<note>**
3. **<to>**Tove**</to>**
4. **<from>**Jani**</from>**
5. **<heading>**Reminder**</heading>**
6. **<body>**Don't forget me this weekend!**</body>**
7. **</note>**

## Another Example of XML: Books

*File: books.xml*

1. **<bookstore>**
2. **<book** category="COOKING"**>**
3. **<title** lang="en"**>**Everyday Italian**</title>**
4. **<author>**Giada De Laurentiis**</author>**
5. **<year>**2005**</year>**
6. **<price>**30.00**</price>**
7. **</book>**
8. **<book** category="CHILDREN"**>**
9. **<title** lang="en"**>**Harry Potter**</title>**
10. **<author>**J K. Rowling**</author>**
11. **<year>**2005**</year>**
12. **<price>**29.99**</price>**
13. **</book>**
14. **<book** category="WEB"**>**
15. **<title** lang="en"**>**Learning XML**</title>**
16. **<author>**Erik T. Ray**</author>**
17. **<year>**2003**</year>**
18. **<price>**39.95**</price>**
19. **</book>**
20. **</bookstore>**

# XML Attributes

XML elements can have attributes. By the use of attributes we can add the information about the element.

XML attributes enhance the properties of the elements.

**Metadata should be stored as attribute and data should be stored as element.**

1. **<book>**
2. **<book** category="computer"**>**
3. **<author>** A & B **</author>**
4. **</book>**

## Why should we avoid XML attributes

* Attributes cannot contain multiple values but child elements can have multiple values.
* Attributes cannot contain tree structure but child element can.
* Attributes are not easily expandable. If you want to change in attribute's vales in future, it may be complicated.
* Attributes cannot describe structure but child elements can.
* Attributes are more difficult to be manipulated by program code.
* Attributes values are not easy to test against a DTD, which is used to define the legal elements of an XML document.

## Difference between attribute and sub-element

In the context of documents, attributes are part of markup, while sub elements are part of the basic document contents.

In the context of data representation, the difference is unclear and may be confusing.

Same information can be represented in two ways:

**1st way:**

1. **<book** publisher="Tata McGraw Hill"**>** **</book>**

**2nd way:**

1. **<book>**
2. **<publisher>** Tata McGraw Hill **</publisher>**
3. **</book>**

# XML Tree Structure

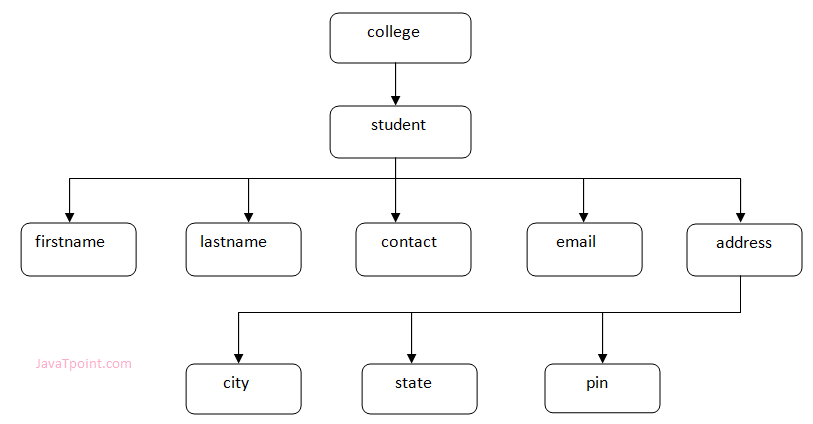
An XML document has a self descriptive structure. It forms a tree structure which is referred as an XML tree. The tree structure makes easy to describe an XML document.

A tree structure contains root element (as parent), child element and so on. It is very easy to traverse all succeeding branches and sub-branches and leaf nodes starting from the root.

## Example of an XML document

1. **<?xml** version="1.0"**?>**
2. **<college>**
3. **<student>**
4. **<firstname>**Tamanna**</firstname>**
5. **<lastname>**Bhatia**</lastname>**
6. **<contact>**09990449935**</contact>**
7. **<email>**tammanabhatia@abc.com**</email>**
8. **<address>**
9. **<city>**Ghaziabad**</city>**
10. **<state>**Uttar Pradesh**</state>**
11. **<pin>**201007**</pin>**
12. **</address>**
13. **</student>**
14. **</college>**

Let's see the tree-structure representation of the above example.



## XML Tree Rules

These rules are used to figure out the relationship of the elements. It shows if an element is a child or a parent of the other element.

**Descendants:** If element A is contained by element B, then A is known as descendant of B. In the above example "College" is the root element and all the other elements are the descendants of "College".

**Ancestors:** The containing element which contains other elements is called "Ancestor" of other element. In the above example Root element (College) is ancestor of all other elements.