

Title: Basics of XML.

Aims:

- What is XML?
- Rules and Syntax of XML
- Implementation of XML

What is XML?

XML stands for eXtensible Markup Language, which is markup language more like HTML. It is designed to store and transport data. XML is recommended by W3C (World Wide Web Consortium) in early February of 1998.

While HTML was designed to display data focusing on how data looks, XML was designed to carry data focusing on what data is. XML does not have any predefined tags like HTML consists. XML stores data in plain text format, which provides a software and hardware independent way of storing, transporting, and sharing data. And it makes it easier to expand or upgrade to new operating systems, new applications, new browsers without losing data.

XML Tree Structure:

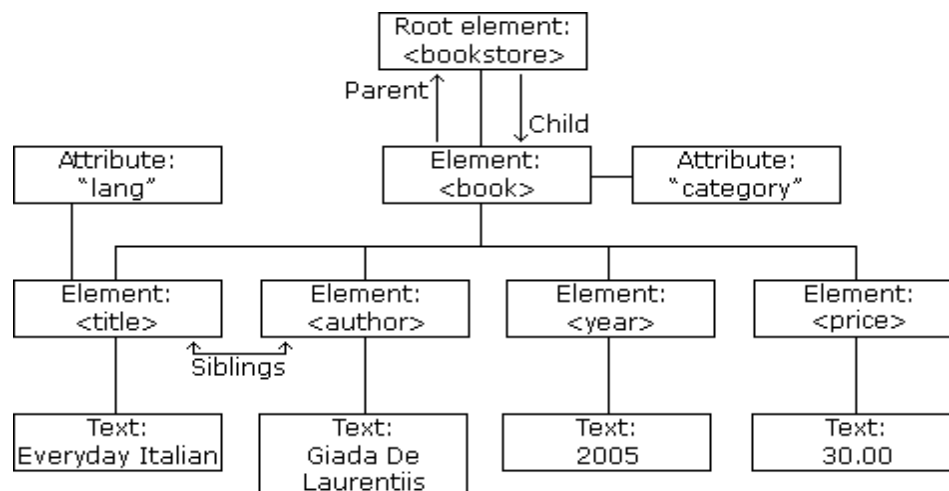


Fig 01: XML Tree Structure.

XML documents form a tree structure that starts at "the root" and branches to "the leaves". XML documents are formed as element trees. The tree starts at a root element and branches from the root to child elements. All elements can have sub elements i.e. child elements.

Syntax of XML (Rules)

- XML documents must have a root element.
- The XML Prolog.
 - `<?xml version="1.0" encoding="UTF-8"?>`
- All elements must have a closing tag.
 - `<title>Harry Potter</title>`
- XML tags are case sensitive.
 - `<title>Harry Potter</title>` ✓
 - `<title>Harry Potter</Title>` ✗
- All elements must be properly nested.
 - `<i>This text is bold and italic</i>` ✗
 - `<i>This text is bold and italic</i>` ✓
- Attributes values must be quoted.
 - `<note date="12/11/2007">`
- Entity references.

<	<	less than
>	>	greater than
&	&	ampersand
'	'	apostrophe
"	"	quotation mark

XML Elements:

An element can contain:

- Text
- Attributes
- Other elements
- Mix of the above

Example:

```
<eligibility>age > 18</eligibility>
<eligibility>age &gt; 18</eligibility>
```

```
<book category="children">
  <title>Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
```

In the above example:

- `<title>`, `<author>`, `<year>`, and `<price>` have **text content** because they contain text
- `<bookstore>` and `<book>` have **element contents**, because they contain elements.
- `<book>` has an **attribute**.

An element can be empty without any content. `<element></element>` or `<element />`

XML Naming Rules:

- Element names are case-sensitive
- Element names must start with a letter or underscore
- Element names cannot start with letters xml (or XML, or Xml, etc)
- Element names can contain letters, digits, hyphens, underscores, and periods
- Element names cannot contain spaces.

XML Attributes:

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

In here, gender is an attribute. And it has value. Always the attribute value must be quoted with either a single or double quotation.

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

In here, gender is an element.

In both the way the same information is going to be resulted.

Implementation of XML

The XML code for the above Fig 01 is given below.

```
<?xml version="1.0" encoding="UTF-8"?>  
<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>
```

Questions

1. Write an XML document containing root / parent element, attributes, child elements, sub-child elements

- 0. University Lecturers
- 1. Bank Customers
- 2. Hospital Patients

Find your last digit of your registration number.
Get the last digit and divide by 3,

If the remainder is 0 = University Lecturers
1 = Bank Customers
2 = Hospital Patients

Example: SEU/IS/17/ICT/145,
Last Digit is 5
 $5/3$
Reminder = 2
Therefore, Topic is Hospital Patients