



Data processing technologies (TTD)

Class 1

XML

Standing for *eXtensible Markup Language*, XML is a markup language used to store and transport data designed to be both human and machine readable. Although XML became less popular in the last couple of years (being replaced with JSON), it still plays an important role in many different IT systems and it is used in many aspects of web development.

XML doesn't depend on platform nor software nor programming language. This means it is possible to write a program in any language on any platform to send, receive or store data using XML. Mainly, just like PHP includes, for instance, XML is a complement to HTML used to separate data from presentation. You will be surprised to know that XML doesn't actually do anything. The XML tags, wrapped around different contents, are simply used to structure data. The XML file can then be used in a program to display the data.

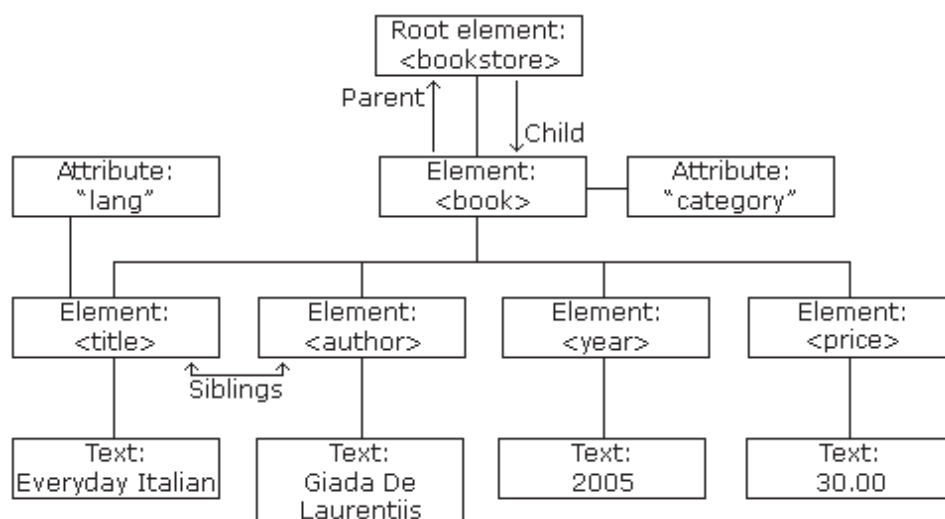
XML structure

XML resembles HTML in this sense that it uses opening and ending tags. But, XML using a self-describing syntax, there are no predefined tags in XML and they must be created accordingly to the database needs. Fields are grouped within entries, and entries are grouped within a table in a child-sibling relationship.

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

Explanation:

The root can be viewed as the database itself or a table. The root contains all entries, the children which contain the same subchildren consisting into fields.



Example of bookstore's database nodetree.

```

<?xml version="1.0" encoding="UTF-8"?>           // XML prolog
<booklist>                                         // Table
  <book>                                           // Entry
    <title>Harry Potter</title>                   // Field
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book>
    <title>The flowers of evil</title>
    <author>Charles Baudelaire</author>
    <year>1857</year>
    <price>39,95</price>
  </book>
</booklist>

```

Explanation:

The first line is called the XML prolog and tells the browsers what language is being used, its version number and characters encoding. In XML, there are no pre-defined tags. In the above example, the database's table (the root) is defined by the tag `<booklist>`, and the entries are defined by `<book>` tags containing different fields.

Viewing XML files

Once an XML document has been created and saved using XML extension, opening it in a browser would look like the example below.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

▼ <bookstore>
  ▼ <book>
    <title>Poèmes Français</title>
    <author>Réjean Thomas</author>
    <year>2006</year>
    <price>14.99</price>
  </book>
  ▼ <book>
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  ▼ <book>
    <title>Le petit prince</title>
    <author>Antoine de Saint-Exupéry</author>
    <year>1943</year>
    <price>19.99</price>
  </book>
  ▼ <book>
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
  ▼ <book>
    <title>L'avalée des avalés</title>
    <author>Réjean Ducharme</author>
    <year>1966</year>

```

Important notes regarding XML coding

XML prolog

The XML prolog is optional but, if it exists, it must come first in the document and doesn't have a closing tag. It is a good idea to use it though since XML document may use international characters. To avoid errors, encoding used should be specified (or simply save your XML the file as UTF-8, the default character encoding for XML).

XML tags

It is mandatory that all XML tags have a closing tag (except the prolog which isn't a XML tag). XML tags are case sensitive, they must start with a letter or an underscore and they can use letters, digits, hyphens, underscores, and periods. They cannot start with xml nor contain a space, and that they must be properly nested. It is recommended to choose short descriptive names and to prefer underscores to dashes.

Empty element

An opening and ending tags with no content in between is called an empty element in XML and the two tags can then be replaced with a self closing tag. Although, empty elements can still have attributes.

```
<element> </element>
```

OR

```
<element />
```

XML attributes

XML can use attributes, just like HTML. If attributes are used, they must always be quoted. The following example shows two ways of coming to the same result, with and without an attribute. The basic rule should be to store data as elements and metadata as attributes.

| | |
|--|--|
| <pre><note date="2020-01-22"> <to>Students</to> <from>Teacher</from> </note></pre> | <pre><note> <date>2020-01-22</date> <to>Students</to> <from>Teacher</from> </note></pre> |
|--|--|

An even better way would be like the following :

| | |
|--|--|
| <pre><note id="117"> <date> <year>2020</year> <month>01</month> <day>22</day> </date> <to>Students</to> <from>Teacher</from> </note></pre> | <pre>// Here, an ID is used for JavaScript reference // Using elements instead of attributes makes // it expandable // Attributes cannot contain multiple values</pre> |
|--|--|

Use of symbols

Be careful as some symbols may cause errors. The symbol smaller than ("<") and ampersand ("&") are strictly illegal in XML, so it must be coded using predefined entities. It is actually a good idea to systematically code all symbols. Note that the white spaces are not truncated in XML like they are in HTML.

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

CDATA

Standing for *Character Data*, CDATA is defined as blocks of plain text which are not parsed and that would otherwise be recognized as markup. When using CDATA, you are telling the parser a specific section of the document contains no markup and has to be treated as plain text. This is especially useful for text containing symbols and special characters.

```
<![CDATA[  
    <message>Example...</message>           // Tags would be treated as plain text  
]]>
```

Online resources

<https://onlinexmltools.com/>
<https://www.freeformatter.com/>
<https://codebeautify.org/xmlviewer>

Styling XML files using CSS

Although this isn't the most proper way to display XML data, it is possible to use a CSS file to style an XML document's content. To do so, it is simply needed to link the XML document to a stylesheet.

XML file :

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="mystyle.css"?>
<booklist>
  <book>
    ...
```

CSS file :

```
bookstore {
  display: block;
  width: 600px;
  margin: auto;
  column-count: 2;
}

book {
  display: block;
  margin-bottom: 20px;
  padding: 10px;
  background-color : rgba(0,0,0,.1);
  break-inside: avoid-column;
  border: solid 1px black;
  box-shadow: 3px 3px 6px rgba(0,0,0,.25);
}

title,author,year,price {display : block;}

title {
  font-size : 25px;
  font-weight : bold;
}
```

Poèmes Français

Réjean Thomas
2006
14.99

Learning XML

Erik T. Ray
2003
39.95

Harry Potter

J K. Rowling
2005
29.99

L'avalée des avalés

Réjean Ducharme
1966
39.95