XML知识点总结

# XML基础

XML被设计用来传输和存储数据

## 1.什么是xml

XML 指可扩展标记语言（EXtensible Markup Language）

XML 是一种标记语言，很类似 HTML

XML 的设计宗旨是传输数据，而非显示数据

XML 标签没有被预定义。您需要自行定义标签。

XML 被设计为具有自我描述性。

XML 是 W3C 的推荐标准

## 2. XML 与 HTML 的主要差异

XML 不是 HTML 的替代。

XML 和 HTML 为不同的目的而设计：

XML 被设计为传输和存储数据，其焦点是数据的内容。

HTML 被设计用来显示数据，其焦点是数据的外观。

HTML 旨在显示信息，而 XML 旨在传输信息。

## 3. XML的用途

XML 把数据从 HTML 分离

XML 简化数据共享

XML 简化数据传输

XML 简化平台的变更

XML 使您的数据更有用

XML 用于创建新的 Internet 语言

## 4. XML树结构

**例如：**

<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>

**其中**：

根元素是 <bookstore>。文档中的所有 <book> 元素都被包含在 <bookstore> 中。

<book> 元素有 4 个子元素：<title>、< author>、<year>、<price>

## 5. XML语法规则

**1.所有 XML 元素都须有关闭标签**

<p>This is a paragraph</p>

**2.XML 标签对大小写敏感**

<message>这是正确的。</message>

**3.XML 必须正确地嵌套**

<b><i>This text is bold and italic</i></b>

**4.XML 文档必须有根元素**

<root>

<child>

<subchild>.....</subchild>

</child>

</root>

**6.XML 的属性值须加引号**

<note date="08/08/2008">

<to>George</to>

<from>John</from>

</note>

**7.XML 中的注释**

<!-- This is a comment -->

**8.在 XML 中，空格会被保留**

## 9. XML元素

**1.什么是 XML 元素？**

XML 元素指的是从（且包括）开始标签直到（且包括）结束标签的部分。

元素可包含其他元素、文本或者两者的混合物。元素也可以拥有属性。

**例如：**

<bookstore>

<book category="CHILDREN">

<title>Harry Potter</title>

<author>J K. Rowling</author>

<year>2005</year>

<price>29.99</price>

</book>

</bookstore>

在上例中，<bookstore> 和 <book> 都拥有元素内容，因为它们包含了其他元素。<author> 只有文本内容，因为它仅包含文本。

在上例中，只有 <book> 元素拥有属性

**2.XML 命名规则**

XML 元素必须遵循以下命名规则：

名称可以含字母、数字以及其他的字符

名称不能以数字或者标点符号开始

名称不能以字符 “xml”（或者 XML、Xml）开始

名称不能包含空格

可使用任何名称，没有保留的字词。

**3.XML 元素是可扩展的**

## 10. XML属性

**1.XML 属性必须加引号**

属性值必须被引号包围，不过单引号和双引号均可使用。比如一个人的性别，person 标签可以这样写：

<person sex="female">

**2.避免 XML 属性**

因使用属性而引起的一些问题：

属性无法包含多重的值（元素可以）

属性无法描述树结构（元素可以）

属性不易扩展（为未来的变化）

属性难以阅读和维护

请尽量使用元素来描述数据。而仅仅使用属性来提供与数据无关的信息

**3.针对元数据的 XML 属性**

有时候会向元素分配 ID 引用。这些 ID 索引可用于标识 XML 元素，它起作用的方式与 HTML 中 ID 属性是一样的。

## 11. XML验证

拥有正确语法的 XML 被称为“形式良好”的 XML。

通过 DTD 验证的 XML 是“合法”的 XML。

**形式良好的 XML 文档**

“形式良好”或“结构良好”的XML 文档拥有正确的语法**。**

“形式良好”（Well Formed）的 XML 文档会遵守前几章介绍过的 XML 语法规则：

**XML 文档必须有根元素**

**XML 文档必须有关闭标签**

**XML 标签对大小写敏感**

**XML 元素必须被正确的嵌套**

**XML 属性必须加引号**

**12.DOM和SAX**

**DOM**的全称是Document Object Model，也即文档对象模型。基于DOM的XML分析器将一个XML文档转换成一个对象模型的集合，应用程序挣是通过对这个对象模型的操作，来实现对XML文档数据的操作。

**DOM**接口提供了一种通过分层对象模型来访问XML文档信息的方式，这些分层对象模型依据XML的文档结构形成一颗节点树。也就是说，DOM强制使用树模型来访问XML文档中的信息。

**SAX**的全称是Simple APIs for XML,也即XML简单应用程序接口。SAX提供的顺序呢模式是一种顺序模式，这是一种快速读写XML数据的方式。应用程序通过这些事件处理函数实现对XML文档的访问，因而SAX接口也被称作事件驱动接口。

**13.DOM的使用**：

　　DOM的基本对象有5个：**Document、Node、NodeList、Element和Attr**。

　　我们总是先通过解析XML源文件而得到一个Document对象，然后再来执行后续的操作：

　　1、createAttribute(String):用给定的属性名创建一个Attr对象，并可在其后使用setAttributeNode方法来设置在某一个Element对象上面。

　　2、createElement(String):用给定的标签名创建一个element对象，代表XML文档中的一个标签，然后就可以在这个Element对象上添加属性或进行其他操作。

　　3、createTextNode(String):用给定的字符串创建一个Text对象，Text对象代表了标签或属性中所包含的纯文本字符串。

　　4、getElementsByTagName(String)：返回一个NodeList对象，它包含了所有给定标签名字的标签。

　　5、getDocumentElement()：返回一个代表这个DOM树的根节点的Element对象，也就是代表了XML文档根元素的那个对象。

**14.Node对象**是DOM结构中最为基本的对象，它代表了文档树中的一个抽象的节点。Node对象所包含的主要方法有：

　　1、appendChild(org.w3c.dom.Node)：为这个节点增肌一个子节点，并放在所有子节点的最后。如果这个子节点已经存在，则先把它删掉再添加进去。

　　2、getFirstChild()：如果节点存在子节点，则返回第一个子节点。（getLastChild()返回最后一个子节点）。

　　3、getNodeName():根据节点的类型返回节点的名称。

　　4、getNodeValue():返回节点的值。

**15.Element对象**代表的是XML文档中的标签元素，它继承于Node,也是Node的最主要的子对象。在标签中可以包含属性，因而Element对象中有存取其属性的方法。而任何Node中定义的方法，也可以用在Element对象上面。

　　1、getElementsByTagName(String):返回一个NodeList对象，它包含在这个标签中其下的子孙节点中具有给定标签名的标签。

　　2、getAttribute(String):返回标签中给定属性名称的属性值。因为XML文档中允许有实体属性出现，而这个方法对这些实体属性并不适用。这时需要用getAttributeNodes()方法得到一个Attr对象来进行进一步的操作。