第一章

XML 基础知识

课程目标

- XML 简介
- XML 文档的组成
- XML 的语法规则
- 元素的语法
- XML 文档的有效性

XML简介

XML 的起源和目的

XML 是 Extensible Markup Language 的缩写，即可扩展标记语言。它是一种用来创建的标记的标记语言。1996 年，万维网协会（或者叫 W3C，http://www.w3c.org）开始设计一种可扩展的标记语言，1998 年 2 月，XML1.0 成为了 W3C 的推荐标准。

使用 XML 标记语言可以做到数据或数据结构在任何编程语言环境下的共享。例如我们在某个计算机平台上用某种编程语言编写了一些数据或数据结构，然后用 XML 标记语言进行处理，那样的话，其他人就可以在其他的计算机平台上来访问这些数据或数据结构，甚至可以用其他的编程语言来操作这些数据或数据结构了。这就是 XML 标记语言作为一种数据交换语言存在的价值。

XML和HTML的区别

XML 和 HTML 都是用于操作数据或数据结构，在结构上大致是相同的，但它们在本质上却存在着明显的区别，它们的区别主要有以下几点：

语法要求不同

在 HTML 中不区分大小写，在 XML 中对大小写要求非常严格。

标记不同

HTML 使用固有的标记，而 XML 没有固有标记。

作用不同

HTML 用于显示页面，而 XML 用于描述页面内容的数据或数据的结构。 HTML 把数据和显示合在一起，在页面中把这些数据显示出来，而 XML 则将数据和显示分开。

XML的优势

每种语言的产生都能完成某些特定的功能， XML 作为一种标记语言也不例外。 XML 最大的优势在于它能对各种编程语言编写的数据进行管理，使得在任何平台下都能通过解析器来读取 XML 数据。它的优势可归纳为以下几点：

- **数据的搜索**：在 XML 中可以提取文档中任何位置的数据，
- **数据的显示**：XML 将数据的结构和数据的显示形式分开，根据需要使数据呈现出多种显示方式。如 HTML 、 PDF 等格式。
- **数据的交换**：XML 标记语言的语法非常简单，可以通过解析器在任何机器上解读。并可以在各种计算机平台上使用。逐渐成为一种数据交换的语言。

XML文档的组成

XML 文档也属于纯文本文件，该文档一般如下四部分组成：

- XML 文档的声明
- XML 文档类型定义
- XML 文档注释
- XML 标识及其内容

前三部分都是可选的

按照这种文档格式来编写的一个 XML 文件，如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
        <!--XML 文档注释 -->
        <?xml:stylesheet type="text/xsl"
            href="stu.xsl"?>
        <!-- 班级中学生的信息 -->
        <class>
                <student>
                        <name>Jone</name>
                        <age>20</age>
                </student>
        </class>
```

XML文档有效性

结构良好的 XML 文档

如果某个文档符合 XML 语法规范，那么我们就说这个文档是“结构良好”的文档。

有效的 XML 文档

所谓有效的 XML 文档是指通过了 DTD 的验证的，具有良好结构的 XML 文档， XML 文档可分为结构良好的 XML 文档和有效的 XML 文档，以及他们之间的关系。即具有结构良好的 XML 文档并不一定就是有效的 XML 文档，反之一个有效的 XML 文档必定是一个结构良好的 XML 文档。

XML的基本语法

XML 的语法规则

XML 的语法规则既简单又严格，非常容易学习，在使用过程中只需认真仔细，没有多大困难。一般 XML 的语法规则大致可归纳为以下几点：

- **结束标记不可忽略**

  在 HTML 中某个标记有起始标记，却可以没有结束标记，但在 XML 文档中却不可以。

- **区分大小写**

  在 XML 中严格区分大小写，主要表现在开始标记和结束标记的大小写必须相同。还包括文档的声明部分和文档类型定义部分的大小写区分。

- **正确的嵌套包含**

元素

元素是 XML 文档的重要组成部分，在 XML 文档中必须存在元素。 XML 文档的元素一般是由标记头、标记末和标记间的字符串数据构成，如下代码所示：

```
<root>
        <a>this is test</a>
</root>
```

元素 a 的值

元素 a 的元素名或标签名

XML 文档中的第一个元素被称为根元素，在任何一个 XML 文档中有且只有一个元素被称为根元素。其余所有的元素都是子元素，子元素必须正确的嵌套在根元素中。

标记间的字符串数据就是该元素的值，在 XML 中，如果元素的值中存在空格，那么这些空格将按原样解析出来

实体

预定义实体表如下所示：

实体名	引用格式	表示的符号
lt	<	<
gt	>	>
amp	&	&
apos	'	'
quot	"	"

实体在 **XML** 文档中的一般引用格式如下：

& 实体名 ;

属性

属性是用来修饰某个元素的，如：

```
<root>
        <a attribute="aa">th</a>
</ro
```

属性值

属性名

关于元素的属性需注意如下几个问题：

- **属性的值必须用引号括起来，如：　attribute1=“aa”　或 attribute3=‘aa'；**
- **元素的属性以名和值成对出现；**
- **用来修饰同一个元素的属性的属性名不能相同　；**
- **属性值不能包含“ &” 、“ ’”、“ <” 等字符。**

XML 树结构

- **XML 文档形成了一种树结构，它从“根部”开始，然后扩展到“枝叶”。**
- **一个 XML 文档实例**
- XML 使用了简单的具有自我描述性的语法：
  - <?xml version="1.0" encoding="ISO-8859-1"?>
  - <note>
    - <to>George</to>
    - <from>John</from>
    - <heading>Reminder</heading>
    - <body>Don't forget the meeting!</body>
  - </note>

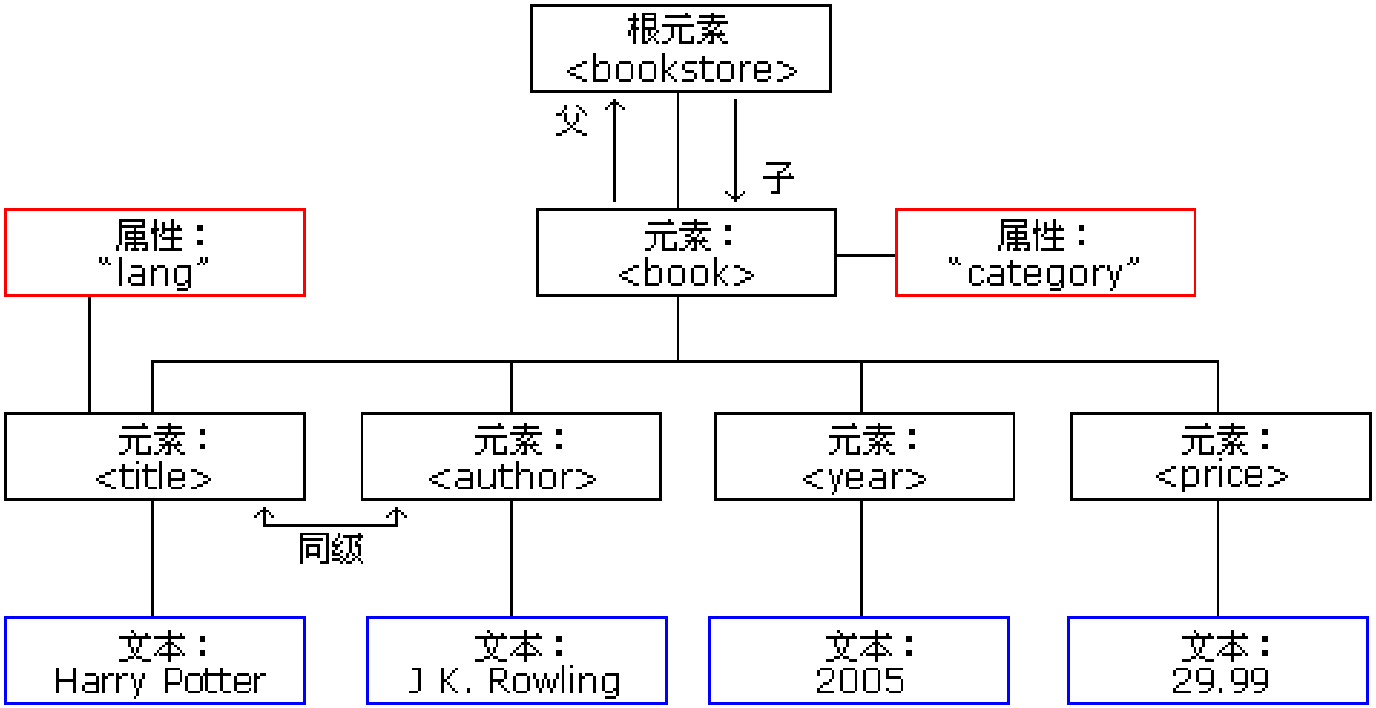
XML 树结构

- 第一行是 XML 声明。它定义 XML 的版本 (1.0) 和所使用的编码 (ISO-8859-1 = Latin-1/ 西欧字符集)。
- 下一行描述文档的*根元素* <note> 接下来 4 行描述根的 4 个*子元素*（ to, from, heading 以及 body ）：
- 最后一行定义根元素的结尾： </note>

XML 文档形成一种树结构

- XML 文档必须包含*根元素*。该元素是所有其他元素的父元素。
- XML 文档中的元素形成了一棵文档树。这棵树从根部开始，并扩展到树的最底端。
- 所有元素均可拥有子元素：
- <root>
  - <child>
    - <subchild>.....</subchild>
  - </child>
- </root>
- 父、子以及同胞等术语用于描述元素之间的关系。父元素拥有子元素。相同层级上的子元素成为同胞（兄弟或姐妹）。
- 所有元素均可拥有文本内容和属性。

实例


根元素
<bookstore>
父
子
属性：
"lang"
元素：
<book>
属性：
"category"
元素：
<title>
元素：
<author>
元素：
<year>
元素：
<price>
同级
文本：
Harry Potter
文本：
J K. Rowling
文本：
2005
文本：
29.99

实例

- <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>
- </bookstore>

XML 语法规则

- **所有 XML 元素都须有关闭标签**
- 在 XML 中，省略关闭标签是非法的。所有元素都*必须*有关闭标签：
- This is a paragraph
- This is another paragraph

XML 语法规则

- **XML 标签对大小写敏感**
- XML 元素使用 XML 标签进行定义。
- XML 标签对大小写敏感。在 XML 中，标签 <Letter> 与标签 <letter> 是不同的。
- 必须使用相同的大小写来编写打开标签和关闭标签：
- <Message> 这是错误的。 </message>
  <message> 这是正确的。 </message>

XML 语法规则

- **XML 必须正确地嵌套**
- 在 XML 中，所有元素都*必须*彼此正确地嵌套：
- This text is bold and italic
- 在上例中，正确嵌套的意思是：由于 元素是在 元素内打开的，那么它必须在 元素内关闭。

XML 语法规则

- **XML 文档必须有根元素**
- XML 文档必须有一个元素是所有其他元素的*父元素*。该元素称为*根元素*。
- <root>
  - <child>
    - <subchild>.....</subchild>
  - </child>
- </root>

XML 语法规则

- **XML 的属性值须加引号**
- 与 HTML 类似，XML 也可拥有属性（名称 / 值的对）。
- 在 XML 中，XML 的属性值须加引号。请研究下面的两个 XML 文档。第一个是错误的，第二个是正确的：
- <note date=08/08/2008>
  - <to>George</to>
  - <from>John</from>
- </note>
- <note date="08/08/2008">
  - <to>George</to>
  - <from>John</from>
- </note>
- 在第一个文档中的错误是，note 元素中的 date 属性没有加引号。

XML 语法规则

- **实体引用**
- 在 XML 中，一些字符拥有特殊的意义。
- 如果你把字符 "<" 放在 XML 元素中，会发生错误，这是因为解析器会把它当作新元素的开始。
- 这样会产生 XML 错误：
- <message>if salary < 1000 then</message>
- 为了避免这个错误，请用一个*实体引用*来代替 "<" 字符：
- <message>if salary < 1000 then</message>

XML 语法规则

- 在 XML 中，有 5 个预定义的实体引用:
- < < 小于
- > > 大于
- &amp ; & 和号
- ' ' 单引号
- " " 引号

XML 语法规则

- **XML 中的注释**
- 在 XML 中编写注释的语法与 HTML 的语法很相似：
- <!-- This is a comment -->

XML 元素

- **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。
- <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* (category="CHILDREN")。

XML 元素

- **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。

XML 元素

- **最佳命名习惯**
- 使名称具有描述性。使用下划线的名称也很不错。
- 名称应当比较简短，比如： <book_title> ，而不是： <the_title_of_the_book> 。
- 避免 "-" 字符。如果您按照这样的方式进行命名： "first-name" ，一些软件会认为你需要提取第一个单词。
- 避免 "." 字符。如果您按照这样的方式进行命名： "first.name" ，一些软件会认为 "name" 是对象 "first" 的属性。
- 避免 ":" 字符。冒号会被转换为命名空间来使用（稍后介绍）。
- XML 文档经常有一个对应的数据库，其中的字段会对应 XML 文档中的元素。有一个实用的经验，即使用数据库的名称规则来命名 XML 文档中的元素。

XML 属性

- **XML 元素可以在开始标签中包含属性，类似 HTML。**
- **属性（Attribute）提供关于元素的额外信息。**
- **XML 属性**
- 属性通常提供不属于数据组成部分的信息。在下面的例子中，文件类型与数据无关，但是对需要处理这个元素的软件来说却很重要：
- <file type="gif">computer.gif</file>

XML 属性

- **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"> 或者这样也可以：
- <person sex='female'>
- 注释：如果属性值本身包含双引号，那么有必要使用单引号包围它，就像这个例子：
- <gangster name='George "Shotgun" Ziegler'> 或者可以使用实体引用：
- <gangster name="George "Shotgun" Ziegler">

XML 属性

- **XML 元素 vs. 属性**
- 请看这些例子：
- <person sex="female">
  - <firstname>Anna</firstname>
  - <lastname>Smith</lastname>
- </person>
- <person>
- <sex>female</sex>
  - <firstname>Anna</firstname>
  - <lastname>Smith</lastname>
- </person>
- 在第一个例子中，sex 是一个属性。在第二个例子中，sex 则是一个子元素。两个例子均可提供相同的信息。
- 在 HTML 中，属性用起来很便利，但在 XML 中，应尽量避免使用属性。

XML 属性

- **避免 XML 属性?**
- 因使用属性而引起的一些问题:
- 属性无法包含多个值（子元素可以）
- 属性无法描述树结构（子元素可以）
- 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。

XML 属性

- **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。这个例子向我们演示了这种情况：
- <messages>
  - <note id="501">
    - <to>George</to>
    - <from>John</from>
    - <heading>Reminder</heading>
    - <body>Don't forget the meeting!</body>
  - </note>
  - <note id="502">
    - <to>John</to>
    - <from>George</from>
    - <heading>Re: Reminder</heading>
    - <body>I will not</body>
  - </note>
- </messages>
- 上面的 ID 仅仅是一个标识符，用于标识不同的便签。它并不是便签数据的组成部分。

体验项目-编写关于班级学生信息的XML文档

程序的实现要求如下：

（1）用记事本编写某班级的学生信息，要求符合 XML 语言的规范。

（2）编写中每个学生要有姓名、年龄、电子邮箱、身高、电话、单位等信息，单位又包含地址、邮编等信息。每个学生要有电话或手机。每个学生都要有一个“编号”属性作为标识。

（3）该文档是否是结构良好的 XML 文档。

体验项目-编写关于班级学生信息的XML文档

使用记事本编写某班级的学生信息，要求符合XML 语法的规范。学生信息包括姓名、年龄、电子邮箱、身高、电话、单位等；单位又包含地址、邮编等信息，每个学生都要有一个“编号”属性作为标识。例如，姓名为“张三”的学生有两个电子邮箱，每个学生有电话或手机。 XML 代码如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- 以下是某班级的学生信息，每个学生有姓名、年龄、电子邮箱、身高、电话、单位等信息，单位又有地址、邮编等信息，每个学生都要有一个“编号”属性作为标识。名为“张三”的学生有两个电子邮箱，每个学生要有电话或手机。 -->
< 班级 >
< 学生 编号 ="A0001">
< 姓名 > 张三 </ 姓名 >
< 年龄 >23</ 年龄 >
< 电子邮箱 >zhangsan@163.com</ 电子邮箱 >
< 电子邮箱 >zhangsan@yahoo.com</ 电子邮箱 >
< 身高 >179.5</ 身高 >
< 电话 >686868</ 电话 >
< 单位 >
公司
< 地址 > 上海 </ 地址 >
< 邮编 >100002</ 邮编 >
</ 单位 >
</ 学生 >
```

< 学生 编号 ="A0003">
< 姓名 > 李四 </ 姓名 >
< 年龄 >24</ 年龄 >
< 电子邮箱 >lisi@263.com</ 电子邮箱 >
< 身高 >168.0</ 身高 >
< 手机 >135013562554</ 手机 >
< 单位 >
< 地址 > 北京 </ 地址 >
</ 单位 >
</ 学生 >
< 学生 编号 ="A0002">
< 姓名 > 王五 </ 姓名 >
< 年龄 >21</ 年龄 >
< 电子邮箱 >wangwu@163.com</ 电子邮箱 >
< 身高 >179.5</ 身高 >
< 电话 >686868</ 电话 >
< 单位 >XXXX 公司 </ 单位 >
</ 学生 >
</ 班级 >

本章总结

- XML 简介
- XML 文档的组成
- XML 的语法规则
- 元素的语法
- XML 文档的有效性