

LANZHOU UNIVERSITY OF TECHNOLOGY

结课作业

学生姓名 王肖肖

学 号 1817700343

专业班级 三班

学 院 软件学院

目 录

[（一） XML简介 1](#_Toc23974)

[1． 什么是XML 1](#_Toc31353)

[2． XML 与 HTML 的主要差异 2](#_Toc30842)

[（二） XML用途 2](#_Toc24245)

[（三） XML树结构 2](#_Toc2073)

[（四） XML语法规范 3](#_Toc13612)

[（五） XML元素 5](#_Toc26487)

[1. 什么是XML元素 5](#_Toc17529)

[2. XML命名规则 6](#_Toc21347)

[（六） XML属性 6](#_Toc21891)

[1. XML属性必须加双引号 6](#_Toc8311)

[2.避免 XML 属性？ 7](#_Toc29135)

[（七） XML验证 7](#_Toc13467)

[1. 验证 XML 文档 7](#_Toc8048)

[2. 形式良好的 XML 文档 7](#_Toc32308)

[（八） XML验证器 8](#_Toc6476)

[（九） XML浏览器 8](#_Toc8773)

[（十） DTD技术： 9](#_Toc14156)

## XML简介

### 什么是XML

1. XML 指可扩展标记语言（EXtensible Markup Language）
2. XML 是一种标记语言，很类似 HTML
3. XML 的设计宗旨是传输数据，而非显示数据
4. XML 标签没有被预定义。您需要自行定义标签。
5. XML 被设计为具有自我描述性。
6. XML 是 W3C 的推荐标准

### XML 与 HTML 的主要差异

1. XML 不是 HTML 的替代。
2. XML 和 HTML 为不同的目的而设计：
3. XML 被设计为传输和存储数据，其焦点是数据的内容。
4. HTML 被设计用来显示数据，其焦点是数据的外观。
5. HTML 旨在显示信息，而 XML 旨在传输信息

## XML用途

XML 应用于 web 开发的许多方面，常用于简化数据的存储和共享。

1. XML 把数据从 HTML 分离
2. XML 简化数据共享

在真实的世界中，计算机系统和数据使用不兼容的格式来存储数据。

XML 数据以纯文本格式进行存储，因此提供了一种独立于软件和硬件的数据存储方法。

这让创建不同应用程序可以共享的数据变得更加容易。

1. XML 简化数据传输

通过 XML，可以在不兼容的系统之间轻松地交换数据。

对开发人员来说，其中一项最费时的挑战一直是在因特网上的不兼容系统之间交换数据。

由于可以通过各种不兼容的应用程序来读取数据，以 XML 交换数据降低了这种复杂性。

1. XML 简化平台的变更

升级到新的系统（硬件或软件平台），总是非常费时的。必须转换大量的数据，不兼容的数据经常会丢失。

XML 数据以文本格式存储。这使得 XML 在不损失数据的情况下，更容易扩展或升级到新的操作系统、新应用程序或新的浏览器

1. XML 使您的数据更有用

由于 XML 独立于硬件、软件以及应用程序，XML 使您的数据更可用，也更有用。

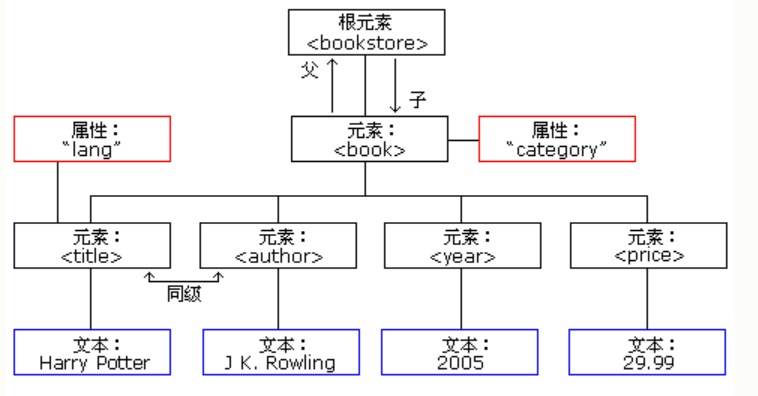
不同的应用程序都能够访问您的数据，不仅仅在 HTML 页中，也可以从 XML 数据源中进行访问。

通过 XML，您的数据可供各种阅读设备使用（手持的计算机、语音设备、新闻阅读器等），还可以供盲人或其他残障人士使用。

## 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”。

1. XML文档实例



<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>

<book category="WEB">

<title lang="en">Learning XML</title>

<author>Erik T. Ray</author>

<year>2003</year>

<price>39.95</price>

</book>

</bookstore>

## XML语法规范

XML 的语法规则很简单，且很有逻辑。这些规则很容易学习，也很容易使用。

1. 所有 XML 元素都须有关闭标签

在 HTML，经常会看到没有关闭标签的元素：

<p>This is a paragraph

<p>This is another paragraph

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：

<p>This is a paragraph</p>

<p>This is another paragraph</p>

注释：您也许已经注意到 XML 声明没有关闭标签。这不是错误。声明不属于XML本身的组成部分。它不是 XML 元素，也不需要关闭标签。

1. 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>

注释：打开标签和关闭标签通常被称为开始标签和结束标签。不论您喜欢哪种术语，它们的概念都是相同的。

1. 在 HTML 中，常会看到没有正确嵌套的元素：

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

在 XML 中，所有元素都必须彼此正确地嵌套：

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

在上例中，正确嵌套的意思是：由于 <i> 元素是在 <b> 元素内打开的，那么它必须在 <b> 元素内关闭。

1. 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>

1. 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。

1. 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，请用实体引用来代替 "<" 字符：

<message>if salary &lt; 1000 then</message>

在 XML 中，有 5 个预定义的实体引用：

|  |  |  |
| --- | --- | --- |
| &lt; | < | 小于 |
| &gt; | > | 大于 |
| &amp; | & | 和号 |
| &apos; | ' | 单引号 |
| &quot; | " | 引号 |

注释：在 XML 中，只有字符 "<" 和 "&" 确实是非法的。大于号是合法的，但是用实体引用来代替它是一个好习惯。

1. 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 -->

1. 在 XML 中，空格会被保留

HTML 会把多个连续的空格字符裁减（合并）为一个：

HTML: Hello my name is David.

输出: Hello my name is David.

在 XML 中，文档中的空格不会被删节。

1. XML 以 LF 存储换行

在 Windows 应用程序中，换行通常以一对字符来存储：回车符 (CR) 和换行符 (LF)。这对字符与打字机设置新行的动作有相似之处。在 Unix 应用程序中，新行以 LF 字符存储。而 Macintosh 应用程序使用 CR 来存储新行。

## 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。

<bookstore>

<book category="CHILDREN">

<title>Harry Potter</title>

<author>J K. Rowling</author>

<year>2005</year>

<price>29.99</price>

</book>

<book category="WEB">

<title>Learning XML</title>

<author>Erik T. Ray</author>

<year>2003</year>

<price>39.95</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。

1. XML命名规则

XML 元素必须遵循以下命名规则：

1. 名称可以含字母、数字以及其他的字符
2. 名称不能以数字或者标点符号开始
3. 名称不能以字符 “xml”（或者 XML、Xml）开始
4. 名称不能包含空格

## XML属性

从 HTML，你会回忆起这个：<img src="computer.gif">。"src" 属性提供有关 <img> 元素的额外信息。

在 HTML 中（以及在 XML 中），属性提供有关元素的额外信息：

<img src="computer.gif">

<a href="demo.asp">

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>

1. XML属性必须加双引号

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 &quot;Shotgun&quot; Ziegler">

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。

不要做这样的蠢事（这不是 XML 应该被使用的方式）：

<note day="08" month="08" year="2008"

to="George" from="John" heading="Reminder"

body="Don't forget the meeting!">

</note>

## 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。

### 验证 XML 文档

合法的 XML 文档是“形式良好”的 XML 文档，同样遵守文档类型定义 (DTD) 的语法规则：

<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE note SYSTEM "Note.dtd">

<note>

<to>George</to>

<from>John</from>

<heading>Reminder</heading>

<body>Don't forget the meeting!</body>

</note>

在上例中，DOCTYPE 声明是对外部 DTD 文件的引用。下面的段落展示了这个文件的内容

### 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

<?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验证器

根据 DTD 来验证 XML

只要把 DOCTYPE 声明添加到您的 XML 中，然后点击验证按钮即可：

<?xml version="1.0" ?>

<!DOCTYPE note [

<!ELEMENT note (to,from,heading,body)>

<!ELEMENT to (#PCDATA)>

<!ELEMENT from (#PCDATA)>

<!ELEMENT heading (#PCDATA)>

<!ELEMENT body (#PCDATA)>

]>

<note>

<to>George</to>

<from>John</Ffrom>

<heading>Reminder</heading>

<body>Don't forget the meeting!</body>

</note>

注释：只有在 Internet Explorer 中，可以根据 DTD 来验证 XML。Firefox, Mozilla, Netscape 以及 Opera 做不到这一点。

## XML浏览器

几乎所有的主流浏览器均支持 XML 和 XSLT。

Mozilla Firefox

从 1.0.2 版本开始，Firefox 就已开始支持 XML 和 XSLT（包括 CSS）。

Mozilla

Mozilla 含有用于 XML 解析的 Expat，并支持显示 XML + CSS。Mozilla 同时拥有对 Namespaces 的某些支持。

Mozilla 同样可做到对 XSLT 的执行（XSLT implementation）。

Netscape

自从版本 8 开始，Netscape 开始使用 Mozilla 的引擎，因此它对 XML / XSLT 的支持与 Mozilla 是相同的。

Opera

自从版本 9 开始，Opera 已经拥有对 XML / XSLT（以及 CSS）的支持。版本 8 仅支持 XML + CSS。

Internet Explorer

自从版本 6 开始，Internet Explorer 就开始支持 XML、Namespaces、CSS、XSLT 以及 XPath。

注释：Internet Explorer 5 同样拥有对 XML 的支持，但是 XSL 部分与 W3C 的官方标准不兼容！

## DTD技术：

### DTD概述：

文档类型定义——Document Type Definition

DTD用来描述xml文档的结构，一个DTD文档包含：

元素的定义规则；元素之间的关系规则；属性的定义规则。

### DTD用途：

<1>有了DTD，每个xml文件可以携带一个自身格式的描述,所谓格式描述就是我的xml文档里面可以写哪些东西，比如元素、属性；

<2>有了DTD，不同组织的人可以使用一个通用DTD用来交换数据，xml是随意定义的，有了DTD，我们可以限制多个公司之间按照这种DTD的规则来编写xml文档，由于DTD都是统一格式，所以不同的公司之间，不同的组织之间就可以用这种通用的xml文档格式进行交互数据了；

<3>应用程序可以使用一个标准DTD校检从外部接收的xml数据是否有效。

### 3、DTD编写：

DTD分成三大类，第一是内部DTD,第二是外部DTD,第三是内外结合的DTD。

（1）内部DTD文档(Students.xml)：

（2）就是说这个xml文档里面既包含xml定义，又包含DTD的定义。

这个[]里面是DTD的定义的内容，就是第一个叫内部DTD文档。

（3）外部DTD文档（Students.xml Students.dtd）:

所谓外部就是DTD的定义和我们的xml文档是在不同的文件里面。

那么外部的DTD如何使用呢？那就需要在前面的xml文档的根元素的上面这个位置跟刚才内部的是一样的，要写上

这类年改革文档在同一个文件夹下面，就可以了。

有内部DTD了，为什么还要学习外部DTD呢？？

内部的DTD,它只能验证当前的这个xml文档。那假如说，我们有一个DTD需要验证多个xml文档的话，那就直接使用外部DTD就行。

（4）内外部DTD文档结合

4、DTD的具体定义——元素的定义

元素名称：是指xml文档里面的元素名称

元素类型：

（1）EMPTY；

（2）#PCDATA；

（3）纯元素类型的DTD语法；

（4）ANY；

### 定义属性的语法结构：

xml文档携带数据的方式，不仅仅是xml的元素的内容，还可以是元素的属性。所以说在DTD里面我们有必要对属性进行限制。

6、实体：实体，简单理解就是C#中的常量，我们在DTD里面定义，在xml文档里面去使用。（实际上就是自定义实体，与前面的预定义实体差不多是一回事）。