

21世纪高等职业教育信息技术类规划教材

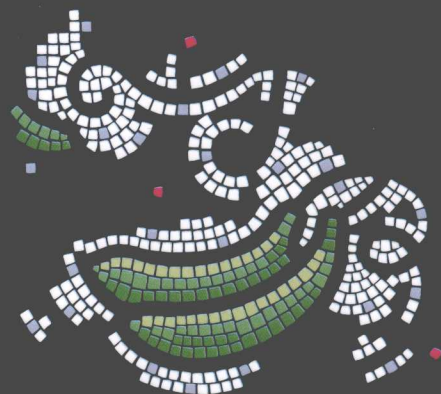
21 Shiji Gaodeng Zhiye Jiaoyu Xinxu Jishulei Guihua Jiaocai

XML 实用教程

XML SHIYONG JIAOCHENG

耿祥义 张跃平 编著

- 强调XML的基础知识
- 突出XML的核心内容
- 结合实例讲解难点关键点



人民邮电出版社
POSTS & TELECOM PRESS

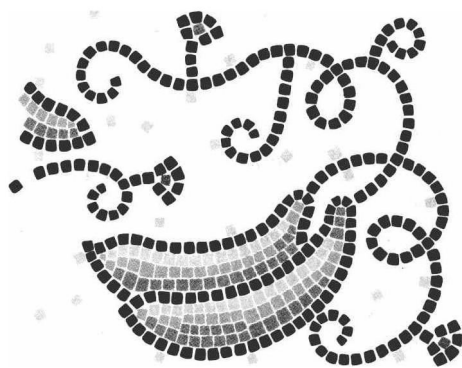
21世纪高等职业教育信息技术类规划教材

21 Shiji Gaodeng Zhiye Jiaoyu Xinxi Jishulei Guihua Jiaocai

XML 实用教程

XML SHIYONG JIAOCHENG

耿祥义 张跃平 编著



人民邮电出版社

北京

图书在版编目(CIP)数据

XML实用教程 / 耿祥义, 张跃平编著. —北京: 人民邮电出版社, 2009.10
21世纪高等职业教育信息技术类规划教材
ISBN 978-7-115-20789-0

I. X… II. ①耿…②张… III. 可扩展语言, XML—程序设计—高等学校: 技术学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2009)第093717号

内 容 提 要

本书结合实例详细地讲解了XML的基础知识。全书共分9章, 内容包括XML简介、规范的XML文件、有效的XML文件、XML与CSS、XML与XSL、基于DOM的解析器、XML Schema模式、XML与数据库以及综合实训等内容。

本书本着“知识够用, 应用为主”的原则, 精选示例, 贴近实际, 力求拓展学生的应用能力。

本书可作为高职高专院校相关专业的教材, 对于XML的初学者, 也不失为一本好的参考书。

21世纪高等职业教育信息技术类规划教材

XML实用教程

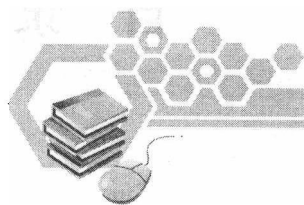
-
- ◆ 编 著 耿祥义 张跃平
责任编辑 潘春燕
执行编辑 王 威
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京楠萍印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 11.75
字数: 300千字
印数: 1-3 000册
- 2009年10月第1版
2009年10月北京第1次印刷

ISBN 978-7-115-20789-0/TP

定价: 20.00元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

前 言



XML 是由万维网联盟定义的一种语言，是表示结构化数据的行业标准。它使得 Internet 上的数据相互交流更加方便，让文件的内容更加显而易懂。XML 不仅提供了直接在数据上工作的通用方法，还可以将用户界面和结构化数据相分离，允许不同来源数据的无缝集成和对同一数据的多种处理。其应用越来越广。

目前，各高职高专院校开设了 XML 技术基础课程，受到广大师生和业界的普遍欢迎。由于 XML 技术还在不断发展的过程中，因此需要不断推出新的、实用性强的 XML 教材。作者从事多年的 XML 课程教学，收集、整理了一大批经典例题，在此通过本书将多年来对 XML 技术的探索心得与大家分享。

本书分为 9 章。前 8 章主要介绍 XML 基础知识及基本应用，第 9 章是综合实训，通过 3 个实训项目，帮助学生将前面所学内容融会贯通，以期达到举一反三的效果。

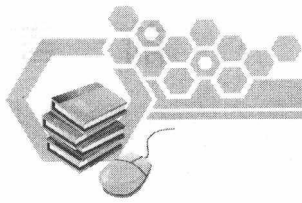
书中示例都是经过精心挑选的，既能帮助理解知识，同时又具有启发性，所有代码都测试通过。在讲解过程中以“实用”为原则，力求通过示例程序的分析与讲解，使学生快速熟悉 XML 规范、掌握 XML 编程技能，通过课堂练习和课后习题，拓展学生的实际应用能力。在内容编写方面，我们力求细致全面、重点突出；在文字叙述方面，注意言简意赅、通俗易懂；在示例程序选取方面，我们强调例子的针对性和实用性。

本书实例开发中用到的程序源代码以及电子教案可以在“人民邮电出版社教学服务与资源网 (www.ptpedu.com.cn)”上免费下载，以供读者学习和使用。

由于编写时间仓促，加之水平有限，书中难免存在错误和不妥之处，敬请广大读者批评指正。

编 者

2009 年 6 月



目 录

第 1 章 XML 简介 1	2.10 标记的子孙关系..... 24
1.1 什么是 XML..... 1	习题..... 24
1.2 XML 文件的编辑与保存..... 3	第 3 章 有效的 XML 文件 28
1.3 XML 和 HTML 有何不同..... 5	3.1 有效的 XML 文件概述..... 29
1.4 XML 解析器..... 7	3.1.1 初识 DTD..... 30
1.5 XML 的优势..... 7	3.1.2 DTD 文件的保存..... 30
习题..... 8	3.1.3 XML 文件与 DTD 文件的关联..... 31
第 2 章 规范的 XML 文件 9	3.2 怎样检查有效性..... 32
2.1 XML 声明..... 10	3.3 DTD 中的元素..... 34
2.1.1 XML 声明中的版本属性..... 10	3.3.1 元素的定义..... 34
2.1.2 XML 声明中的编码属性..... 10	3.3.2 约束标记的子标记..... 34
2.1.3 XML 声明中的独立属性..... 12	3.3.3 约束标记只包含文本数据..... 38
2.2 标记..... 12	3.3.4 约束标记的混合内容..... 38
2.2.1 空标记..... 12	3.3.5 EMPTY 和 ANY..... 40
2.2.2 非空标记..... 12	3.4 DTD 的完整性..... 40
2.2.3 标记的名称..... 14	3.5 DTD 中的属性约束列表..... 40
2.2.4 根标记..... 14	3.5.1 ATTLIST 属性约束列表..... 41
2.3 特殊字符..... 15	3.5.2 属性的默认值..... 42
2.4 CDATA 段..... 15	3.5.3 属性类型..... 46
2.5 标记的文本数据..... 16	3.6 内部 DTD..... 52
2.6 属性..... 16	3.7 IGNORE 和 INCLUDE..... 55
2.6.1 属性的构成..... 16	3.8 DTD 与名称空间..... 56
2.6.2 使用属性的原则..... 17	习题..... 59
2.7 注释..... 18	第 4 章 XML 与 CSS 61
2.8 名称空间..... 18	4.1 XML 关联 CSS..... 62
2.8.1 有前缀和无前缀的名称空间..... 19	4.1.1 初识 CSS..... 62
2.8.2 标记中声明名称空间..... 19	4.1.2 关联 CSS..... 62
2.8.3 名称空间的作用域..... 19	4.2 标记与样式表..... 63
2.8.4 名称空间的名字..... 22	4.2.1 标记的名字与样式表的名称..... 63
2.9 区分大小写..... 23	4.2.2 CSS 的显示规则..... 64
	4.3 设置文本的显示方式..... 66



4.3.1 块方式	66	5.9.2 属性值条件	105
4.3.2 行方式	67	5.9.3 子标记条件	106
4.3.3 列表方式	69	5.9.4 子标记及属性条件	107
4.3.4 不显示	71	5.9.5 子标记及属性、属性值条件	107
4.4 字体	71	5.10 xsl:choose 的用法	108
4.5 文本样式	73	5.11 xsl:element 的用法	110
4.6 边框	75	5.12 xsl:comment 的用法	112
4.7 边缘	77	习题	113
4.8 颜色和背景	78	第 6 章 基于 DOM 的解析器	115
4.9 显示图像	78	6.1 初识 DOM 解析器	115
4.10 设置鼠标的形状	80	6.1.1 创建 DOM 解析器的步骤	116
4.11 处理层叠	80	6.1.2 Document 对象的结构	116
习题	81	6.2 节点的类型	118
第 5 章 XML 与 XSL	84	6.2.1 Node 接口	118
5.1 XSL 样式表文件与 XSL 变换	84	6.2.2 Node 接口的常用方法	119
5.1.1 初识 XSL 样式表文件	84	6.2.3 节点的子孙关系	119
5.1.2 XML 关联 XSL 文件	85	6.2.4 使用递归方法输出节点中 的数据	119
5.1.3 XSL 变换与 HTML	86	6.3 Document 节点	121
5.2 XSL 样式表文件中的模板	87	6.4 Element 节点	121
5.2.1 模板	87	6.5 Text 节点	123
5.2.2 主模板与 XSL 处理器	88	6.6 Attr 节点	126
5.3 标记与模板匹配	90	6.7 DocumentType 节点	128
5.3.1 XML 子标记匹配的模板	90	6.8 处理空白	129
5.3.2 任意级别 XML 子标记匹配 的模板	91	6.9 验证规范性和有效性	131
5.3.3 具有指定属性的 XML 子标记 匹配的模板	93	6.10 使用 DOM 生成 XML 文件	131
5.3.4 使用“[]”和“ ”给出 XML 标记 匹配的模板	94	6.10.1 Transformer 对象	132
5.4 模板调用	95	6.10.2 用于修改 Document 的常用方法	132
5.4.1 模板调用标记	95	6.10.3 用 DOM 建立 XML 文件	133
5.4.2 模板调用标记的执行过程	95	习题	136
5.5 常用的 XSL 标记	97	第 7 章 XML Schema 模式	137
5.6 xsl:for-each 的用法	99	7.1 什么是 XML Schema	137
5.7 xsl:value-of 的用法	100	7.2 初识 XML Schema	138
5.8 xsl:copy 的用法	103	7.2.1 XML Schema 模式的保存	138
5.9 xsl:if 的用法	105	7.2.2 XML Schema 模式的特点	138
5.9.1 属性条件	105	7.2.3 根标记	139
		7.2.4 元素标记	139



7.2.5 属性标记	140	和 Web 服务目录	165
7.3 XML Schema 模式的验证	140	9.3.1 配置端口	165
7.4 简单类型元素	143	9.3.2 配置 Web 服务目录	166
7.5 复杂类型元素	145	9.4 JSP 页面简介	168
7.6 属性	147	9.4.1 JSP 页面的基本结构	168
7.7 正则表达式	149	9.4.2 page 指令	169
7.7.1 正则表达式概述	149	9.4.3 数据和方法的声明	169
7.7.2 使用正则表达式	150	9.4.4 Java 程序片与表达式	170
习题	151	9.4.5 JSP 内置对象 request	170
第 8 章 XML 与数据库	152	9.5 在 JSP 中使用 XML 文件	172
8.1 JDBC	153	9.5.1 XML 文件	172
8.2 Microsoft Access 数据库	153	9.5.2 XSL 文件	172
8.2.1 建立数据库	153	9.5.3 JSP 页面	173
8.2.2 创建表	154	9.6 实训题目 2: 在 JSP 中使用 XML 查询商品	175
8.3 连接数据库	154	9.7 Java Servlet 简介	175
8.3.1 JDBC-ODBC 桥接器	154	9.7.1 编写 Servlet 类	175
8.3.2 ODBC 数据源	155	9.7.2 字节码文件的保存	177
8.3.3 建立连接	156	9.7.3 配制 web.xml 文件	177
8.4 XML 至数据库	157	9.7.4 请求 servlet 对象	178
8.5 数据库至 XML	160	9.8 在 Java Servlet 中使用 XML 文件	178
习题	162	9.8.1 XML 文件	179
第 9 章 综合实训	163	9.8.2 CSS 文件	179
9.1 JSP 简介	163	9.8.3 JSP 页面与 Servlet	179
9.2 Tomcat 服务器	164	9.9 实训题目 3: 在 Java Servlet 中使用 XML 查询车次	182
9.3 实训题目 1: 配置端口号			

第 1 章

XML 简介

本章难点

- XML 和 HTML 有何不同

随着网络的迅速发展以及规模的扩大,对信息的规范性提出了更加严格的要求,XML 就是在这一背景下诞生的一种数据格式标准。

在学习这门课程之前,读者应当初步了解 HTML,并有一定的 Java 语言基础。另外,需要强调的是,由于许多 Web 技术都和 XML 有关,因此本课程的学习,对于进一步学习、理解 Web 技术是非常必要的,XML 的先导知识与后继技术的关系如图 1.1 所示。本书将在第 9 章结合 JSP 和 Servlet 具体讲解 web 技术。

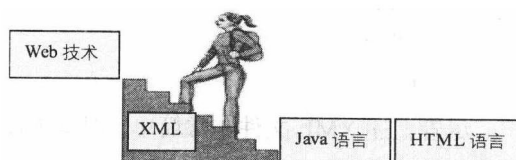


图 1.1 XML 的先导知识与后继技术

1.1 什么是 XML

随着网络的迅速发展,万维网联盟(World Wide Web Consortium, W3C)认识到信息规范化的重要性,即让人们用一种规范化的格式来处理信息,从而可以使人们更加方便地交换信息,如图 1.2 所示。

XML 是 eXtensible Markup Language 的缩写,是由万维网联盟定义的一种语言,称之为可扩展标记语言。所谓可扩展性,是指 XML 允许用户按照 XML 规则自定义标记。

XML 文件是由标记及其所标记的内容构成的文本文件,与 HTML 文件不同的是,这些标记可自由定义,其目的是使 XML 文件能够很好地体现数据的结构和含义。W3C 推出 XML 的主要目的是使 Internet 上的数据相互交流更方便,让文件的内容更加显而易见。

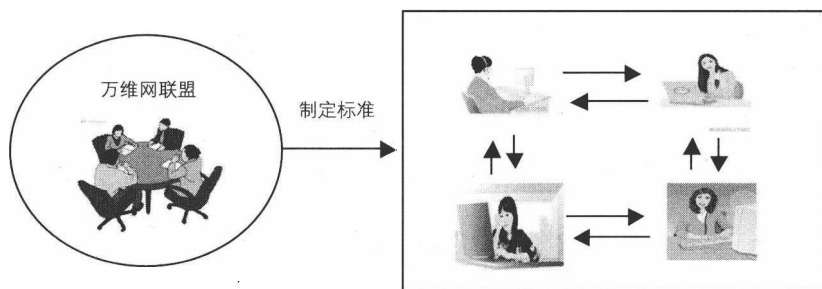


图 1.2 W3C 负责制定和信息有关的标准

以下是一个简单的 XML 文件。

first.xml

```
<?xml version="1.0" ?>
<学生>
  <姓名> 薛校好
    <性别> 男 </性别>
    <出生日期> 1991/8/15 </出生日期>
  </姓名>
  <姓名> 姚笔叶
    <性别> 女 </性别>
    <出生日期> 1992/10/31 </出生日期>
  </姓名>
</学生>
```

尽管上述 XML 文件非常简单，但基本体现了 XML 文件的基本结构，总结如下（有关细节将在后续章节讲解）。

- XML 文件包含一个 XML 声明，其位置必须是 XML 文件的首行（有关 XML 声明将在 2.1 节详细介绍）：

```
<?xml version="1.0" ?>
```

- XML 文件中包含有若干标记，每个标记都是由开始标记和结束标记构成的。
- XML 文件有且仅有一个根标记，其他标记都必须封装在根标记中，文件的标记必须形成树形结构。

- 标记的开始标记和结束标记之间的内容称为该标记所标记的内容，简称标记的内容。一个标记的内容中可以包含文本或其他标记，其中包含的标记称为该标记的子标记。

上面的 XML 文件的根标记的开始标记是“<学生>”、结束标记是“</学生>”，该根标记有两个子标记：“<姓名>...</姓名>”。“<姓名>...</姓名>”的内容既有文本也有子标记，比如其中一个标记：“<姓名>...</姓名>”中的文本是“薛校好”，子标记是“<性别>...</性别>”和“<出生日期>...</出生日期>”。

XML 文件必须符合一定的语法规则（后续章节会详细地讲解 XML 的语法规则），只有符合这些规则，XML 文件才可以被 XML 解析器解析，以便利用其中的数据，如图 1.3 所示。



图 1.3 XML 解析器

下面的 XML 文件都是错误的，其中的“errorOne.xml”没有根标记，“errorTwo.xml”虽然有根标记，但根标记的两个“节目”子标记有交叉，导致标记没有形成树形结构。

errorOne.xml

```

<节目> 西部探险
  <播出时间>20 点 22 分</播出时间>
</节目>
<节目> 山中橘红
  <播出时间>22 点 38 分</播出时间>
</节目>
  
```

errorTwo.xml

```

<?xml version="1.0" ?>
<CCTV6>
  <节目> 西部探险
    <播出时间>20 点 22 分</播出时间>
  <节目> 山中橘红
    <播出时间>22 点 38 分</播出时间>
  </节目>
</CCTV6>
  
```

1.2 XML 文件的编辑与保存

1. 编辑与保存

XML 文件是具有特殊扩展名（.xml）的文本文件，因此需使用纯文本编辑器来编辑 XML 文件。本书以 Windows 系统自带的“记事本”做编辑器。XML 文件保存的扩展名必须为“.xml”，如“Example.xml”、“hello.xml”等。需要特别注意的是，名字区分大小写，hello.xml 和 Hello.xml 是完全不同的 XML 文件的名字。

对于初学者，不建议使用集成开发工具来编写 XML 文件，尽管各种集成开发工具也提供了用于编辑 XML 文件的编辑器，但同时也屏蔽了 XML 的许多知识点，这非常不利于学习 XML。但是，在掌握了 XML 知识后，在具体开发项目时，应当选择一个流行的、成熟的集成开发工具来编写 XML 文件，这有利于更好、更快地编写 XML 文件，例如，XMLSpy 就是非常优秀的 XML 集成开发工具。对于已掌握 XML 知识的人，学习使用各种集成开发工具是没有任何困难的。

一个 XML 文件应当以 XML 声明作为文件的第一行，在其前面不能有空白或其他处理指令或注释。XML 声明以“<?xml”标识开始，以“?”标识结束。注意“<?”和“xml”之间，以及



“?” 和 “>” 之间不要有空格。

以下是一个最基本的 XML 声明：

```
<?xml version="1.0" ?>
```

一个简单的 XML 声明中可以只包含属性 version，目前该属性的值可以取 1.0（1.1 还没有正式公布，1.1 增加了一些极少被使用的功能），指出该 XML 文件使用的 XML 版本。XML 声明中还可以指定 encoding 属性的值，该属性规定 XML 文件采用哪种字符集进行编码。例如：

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

如果在 XML 声明中没有显示地指定 encoding 属性的值，那么该属性的默认值为 UTF-8 编码。如果 encoding 属性的值为 UTF-8，XML 文件就必须按照 UTF-8 编码来保存，这样 XML 解析器就会识别 XML 中的标记并正确解析标记中的内容。

假如使用文本编辑器“记事本”编辑 XML 文件，在保存文件时，必须将“保存类型”选择为“所有文件”，将“编码”选择为“UTF-8”，见图 1.4。如果在保存文件时，系统总是给文件名尾加上“.txt”，那么在保存文件时可以将文件名用双引号括起来。



图 1.4 保存 XML 文件

2. 怎样检查 XML

XML 的语法规则非常严格，这一点和 HTML 有很大的不同，HTML 本身语法十分不严格，严重影响网络信息的传送和共享。W3C 吸取了 HTML 发展的教训，对 XML 指定了严格的语法标准，例如，标记要有一个开始标记和结束标记，所有的标记都必须合理嵌套，即形成树形结构。也就是说 XML 文件必须符合一定的语法规则，只有符合这些规则，XML 文件才可以被 XML 解析器解析，以便利用其中的数据。

XML 文件分为规范的 XML 文件和有效的 XML 文件，符合 W3C 制定的基本规则的 XML 文件称为规范的 XML 文件，规范的 XML 文件如果再符合额外的一些约束就称为有效的 XML 文件。有关 XML 的详细语法将从第 2 章开始讲述。为了检查 XML 文件是否规范，一个简单的办法就是用浏览器（比如 IE 6.0）打开 XML 文件，如果 XML 是规范的，浏览器将显示 XML 源文件，否则将显示出错信息。图 1.5 显示了 1.1 节中规范的 XML 文件 first.xml；图 1.6 显示了 1.1 节中不规范的 XML 文件 errorOne.xml。

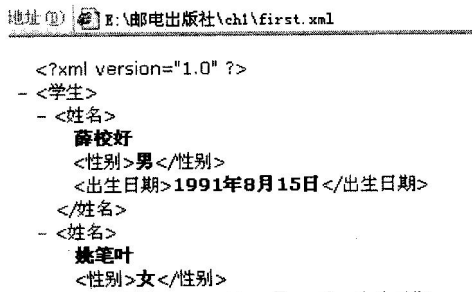


图 1.5 用浏览器打开规范的 XML 文件

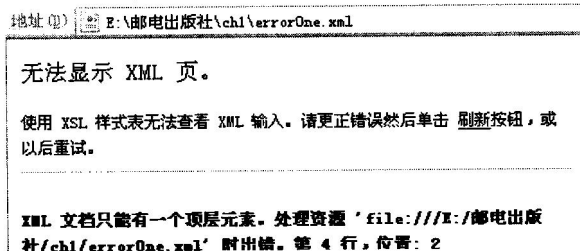


图 1.6 用浏览器打开不规范的 XML 文件



1.3 XML 和 HTML 有何不同

尽管 XML 和 HTML 都是 W3C 定义的语言，但两者有很大的不同。

1. HTML 的核心

HTML 是用来编写 Web 页的语言（超文本标记语言），HTML 的核心是把数据和数据的显示外观捆绑在一起，即 HTML 中标记的出发点不是为了体现数据的含义，而是体现数据的显示格式，整个 HTML 文件的目的是为了组织数据，而不是为了组织某些数据的显示外观。如果某个应用系统只想使用 HTML 中所包含的数据而不需要它的显示外观，那么应用系统将 HTML 中的数据和外观分离是非常困难的，如图 1.7 所示。

HTML 不允许用户自定义标记，目前的 HTML 大约有 100 多个标记。HTML 不能体现数据的组织结构，只能描述数据的显示格式。下列 HTML 将数据分别用黑体 1、黑体 2 和黑体 3 来显示。

show.html

```
<<html>
  <H1> 张小三
    <H2> 男 </H2>
  </H1>
  <H2> 李翠花
    <H3> 女 </H3>
  </H2>
</html>
```

对于上述 HTML 文件，一个不认识汉字的英国人并不知道“张小山”和“李翠花”是人的名字。因为 HTML 中标记的出发点不是为了体现数据的含义，而是为了体现数据的显示格式。

浏览器能将 HTML 中“<H1>...</H1>”、“<H2>...</H2>”和“<H3>...</H3>”所标记的文本内容分别用黑体 1、黑体 2 和黑体 3 显示在浏览器中（如图 1.8 所示）。

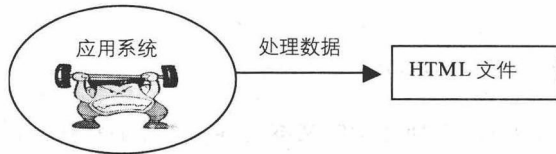


图 1.7 HTML 文件不关心数据的组织结构

地址 (D) E:\邮电出版社\ch1\show.html

张小三

男

李翠花

图 1.8 用浏览器打开 HTML 文件

2. XML 的核心

和 HTML 不同的是，XML 的核心是描述数据的组织结构，让 XML 可以作为数据交换的标准格式。XML 可自定义标记，其标记名称是对所标记的数据内容含义的抽象，而不是数据的显示格式，而且 XML 文件通过其中的标记来表示数据的组织结构。例如，对于下述 XML 文件 second.xml:

**second.xml**

```
<student>
  <name> 张大山
    <sex> 男 </sex>
  </name>
  <name> 李翠花
    <sex> 女 </sex>
  </name>
</student>
```

即使是不认识汉字的英国人也很清楚地知道“张大山”和“李翠花”是学生的名字，并清楚地知道 second.xml 中数据的组织结构。

XML 非常关心数据的组织结构，以便 XML 解析器按照其组织结构分解出数据（有关解析器的详细内容将在第 6 章讲述），XML 本身不提供数据的显示格式。

XML 有效地分离数据的组织结构和显示外观，即不将显示外观和其中的标记直接进行关联，因此浏览器不能直接显示 XML 文件中的标记的内容。如果需要浏览器显示 XML 文件中标记的内容，就必须以某种方式告诉浏览器如何显示，一种方式是使用层叠样式表（CSS）；另一种方式是使用 XSL 变换，如图 1.9 所示（这部分内容和显示 XML 文件中标记的内容的有关知识将在第 4 章和第 5 章讲述）。



图 1.9 让 XML 文件和显示数据外观的 CSS 或 XSL 相关联

以下的 XML 文件将自己关联到一个层叠样式表，以便浏览器能显示 XML 文件中标记的文本内容。层叠样式表中最重要的组成部分就是样式表，其作用是说明 XML 文件中的标记内容用何种方式来显示。例如，如果想用黑体来显示 XML 文件中“<name>...</name>”所标记的文本内容，就可以在层叠样式表文件中包含如下样式表：

```
name
{ display:block;font-size:36pt; font-style:normal;font-weight:bold
}
```

该样式表的作用是告知浏览器将 XML 文件中“<name>...</name>”所标记的文本内容用黑体显示在一个“块区域”中；而

```
sex
{ display:line;font-size:12pt; font-style:italic;
}
```

告知浏览器将 XML 文件中标记“<sex>...</sex>”所标记的文本内容用斜体显示在一行中。



当使用层叠样式表和 XML 关联时，XML 文档中的标记的名字不要含有非 ASCII 码字符。

以下的“showXML.css”是一个简单的层叠样式表，该文件可以保存为编码为“ANSI”的文件或编码为“UTF-8”的文件，并与其关联的 XML 文件存放在同一目录中。

showXML.css

```
name
```



```
{ display:block;font-size:18pt;font-weight:bold
}
sex
{ display:line;font-size:16pt;font-style:italic
}
birthday
{ display:line;font-size:9pt;font-weight:bold
}
```

XML 文件使用操作指令:

```
<?xml-stylesheet ?>
```

关联到某个层叠样式表, 例如:

```
<?xml-stylesheet href="showXML.css" type="text/css" ?>
```

以下是一个与 showXML.css 层叠样式表相关联的 XML 文件 three.xml, 用浏览器打开 three.xml 文件, 浏览器就会按照 showXML.css 中的样式表来显示 XML 文件中标记的文本内容, 效果见图 1.10。

地址  E:\邮电出版社\ch1\three.xml

张大山 男 1991年9月10日
李翠花 女 1992年11月23日

图 1.10 使用 CSS 显示 XML 中的数据

three.xml

```
<?xml version="1.0" ?>
<?xml-stylesheet href="showXML.css" type="text/css" ?>
<student>
  <name> 张大山
    <sex> 男 </sex>
    <birthday> 1991年9月10日 </birthday>
  </name>
  <name> 李翠花
    <sex> 女 </sex>
    <birthday> 1992年11月23日 </birthday>
  </name>
</student>
```

1.4 XML 解析器

XML 解析器是 XML 和应用程序之间的一个软件组织, 其目的是为应用程序从 XML 文件中解析出所需要的数据。比如, 应用程序可能需要 XML 文件所标记的商品价格, 并准备对价格进行数据分析处理。现在普遍使用的 XML 解析器都是 Java 语言编写的, 本书将采用这样的解析器, 有关解析器的详细内容将在第 6 章讲述。

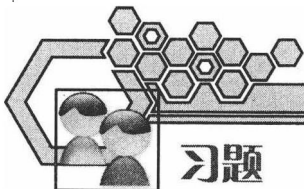
1.5 XML 的优势

XML 作为表示结构化数据的行业标准, 已获得广泛的行业支持。XML 在采用简单的标准化格式组织数据方面是一个革命性的进步, 不仅提供了直接在数据上工作的通用方法, 还将数据的结构和显示相分离, 允许不同来源数据的无缝集成和对同一数据的多种处理。从数据描述语言的角度看, XML 是灵活的、可扩展的、有良好的结构和约束; 从数据处理的角度看, 它足够简单且易于阅读, 几乎和 HTML 一样易于学习, 同时又易于被应用程序处理。许多网络应用都在大量使



用 XML，XML 已经成为网络应用技术的基础。

通过本书的进一步学习，我们会逐步体会到 XML 的优点，例如，通过 1.3 节学习已经初步体会到了 XML 是怎样将数据的组织结构和显示相分离的，如果想修改 three.xml 的显示外观，只需修改它所关联的 CSS 即可。



习题

1. XML 文件和 HTML 文件有何不同？

2. 如果 XML 文件中的 XML 声明为

```
<?xml version="1.0" ?>
```

XML 文件应使用怎样的编码保存？

3. 登录 <http://www.w3c.org> 网站，了解 XML 的有关话题。

4. 在自己喜欢的搜索引擎中查询“XML Java”，看看能查到哪些相关的话题。

5. 参考 1.3 节的 showXML.css 为下列 XML 文件：“exercise.xml”编写一个显示标记中的文本数据的层叠样式表：show.css。

exercise.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet href="show.css" type="text/css" ?>
<trainSchedule>
  <trainNumber> T211
    <startStation> 北京 </startStation >
    <endStation > 重庆 </endStation >
    <startTime> 20 点 38 分 </startTime >
    <endTime> 07 点 16 分 </endTime>
  </trainNumber>
  <trainNumber> T168
    <startStation> 上海 </startStation >
    <endStation > 北京 </endStation >
    <startTime> 22 点 18 分 </startTime >
    <endTime> 09 点 26 分 </endTime>
  </trainNumber>
</trainSchedule>
```

第2章

规范的 XML 文件

本章难点

- 名称空间

W3C 制定 XML 的目的是让 Internet 上的信息交互更加方便,为此 W3C 吸取了当初 HTML 标准不严格的教训,为 XML 制定了严格的语法规则,需要 XML 文件的系统很容易开发出解析 XML 文件中相关数据的“XML 解析器”,为信息的相互交流奠定了语法保证。

按照 W3C 的有关标准,XML 文件分为规范的 XML (Well-Formed XML) 文件和有效的 XML (Validated XML) 文件,符合 W3C 制定的基本语法规则的 XML 文件称为规范的 XML 文件,规范的 XML 文件如果再符合额外的一些约束就称为有效的 XML 文件。本章介绍规范的 XML 文件,下一章讲解有效的 XML 文件。

一个规范的 XML 文件应当满足如下语法规则 (见图 2.1)。

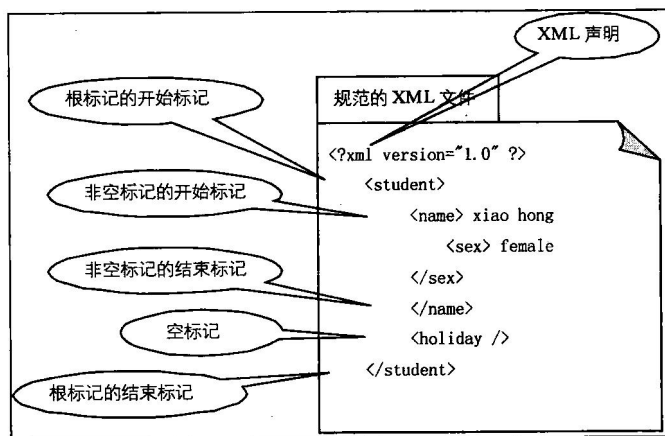


图 2.1 规范的 XML 文件



- XML 文件用“XML 声明”开始。
- XML 文件有且仅有一个根标记。
- XML 的非根标记都必须封装在根标记中。
- 非空标记必须由“开始标记”与“结束标记”构成。
- XML 文件的标记必须形成树形结构，即标记不允许出现交叉。
- 空标记没有“开始标记”和“结束标记”。

人们认为不规范的 XML 文件是没有利用价值的文件，甚至不能称为一个 XML 文件。本章所讲述的内容都是 W3C 所指定的规范标准。

2.1 XML 声明

一个规范的 XML 文件应当以 XML 声明作为文件的第 1 行，在其前面不能有空白、其他处理指令或注释。XML 声明以“<?xml”标识开始、以“>”标识结束。以下是一个最基本的 XML 声明：

```
<?xml version="1.0" ?>
```

W3C 允许在编写 XML 文件时省略 XML 声明，但是如果一个 XML 文件省略 XML 声明，各种 XML 解析器将默认该 XML 文件是有 XML 声明的，而且 XML 声明是：

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

W3C 在 XML 规范中建议每个 XML 文件都包含有 XML 声明，不提倡省略 XML 声明。

2.1.1 XML 声明中的版本属性

一个简单的 XML 声明中可以只包含属性 version，目前该属性的值可以取 1.0，指出该 XML 文件使用的 XML 版本。1.1 版本还没有正式公布，而且仅仅增加了一些极少可能被使用的功能。如果将 version 属性设置为 1.1，用浏览器（IE 6.0）打开 XML 文件时，将得到 XML 版本号设置错误的提示。

2.1.2 XML 声明中的编码属性

XML 声明中也可以指定 encoding 属性的值，该属性规定 XML 文件采用哪种字符集进行编码。如果在 XML 声明中没有指定 encoding 属性的值，那么该属性的默认值是“UTF-8”。例如：

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

声明指定 encoding 属性的值是 UTF-8 编码。如果 XML 使用 UTF-8 编码，那么标记的名称以及标记的内容就可以使用汉字、日文、英文等，XML 解析器就会识别这些标记并正确解析标记中的内容。如果 encoding 属性的值为“UTF-8”，XML 文件应选择“UTF-8”编码来保存（见图 2.2）。



图 2.2 encoding 值是 UTF-8 时 XML 文件的保存