

<<疯狂XML讲义>>

图书基本信息

书名：<<疯狂XML讲义>>

13位ISBN编号：9787121140495

10位ISBN编号：7121140497

出版时间：2011-8

出版时间：电子工业出版社

作者：李刚

页数：500

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## &lt;&lt;疯狂XML讲义&gt;&gt;

## 内容概要

由李刚编著的本书主要以XML为核心，深入地介绍了XML的各种相关知识。

本书作为疯狂Java体系图书之一，依然保持该体系图书系统、全面的特点：不仅详细介绍了XML文档的各种知识，还通过案例示范了实际开发中如何应用XML知识。

本书主要分为五个部分。

第一部分介绍了XML、DTD、XML

Schema等基础知识，这些知识主要教读者如何定义有效的XML文档，这部分内容是深入学习后面知识的基础，也是每个使用XML的开发者都应该掌握的基础。

第二部分介绍了CSS、XSLT和XPath等知识，通过使用CSS或XSLT，可直接将XML文档当成视图技术。

第三部分介绍了DOM、SAX、JAXP、dom4j、

JDOM、XQuery和XQJ等，其中的DOM、SAX、JAXP、dom4j和JDOM都以结构化的方式来创建、解析XML文档，从而可以将XML文档作为数据传输工具，而

XQuery则是一种新规范，通过XQuery可以查询XML文档中的数据，就像使用SQL查询关系数据库的数据一样。

第四部分介绍了Web

ServicesService和CXF，其中CXF是Java领域的主流Web

ServicesService框架，简单易用。

第五部分则提供了两个综合性案例：基于XML的电子公告系统和电子拍卖系统，让读者将前面所学应用到实际项目中。

，尤其是电子拍卖系统，。

它是一个包含5个表，表之间具有复杂关联映射的系统，该案例采用目前最流行、最规范的轻量级Java EE架构，将整个应用分为领域对象层、DAO层和业务逻辑层，然后用CXF将业务逻辑组件包装成Web ServicesService，从而可以将现有的Java

EE项目与其他平台的、其他语言的异构项目进行整合，具有极好的指导价值和借鉴意义。

本书大部分章节后都提供了相应的编程习题，供开发者巩固所学，将理论融入实际开发之用。

本书是疯狂Java体系丛书之一，学习本书无须任何XML基础，但DOM、SAX、dom4j、JDOM解析部分包含了大量Java编程，因此建议先认真阅读疯狂Java体系的《疯狂Java讲义》一书。

## <<疯狂XML讲义>>

### 作者简介

李刚，从事10多年的Java EE应用开发现任传智播客广州中心教学总监。

疯狂Java实训营创始人，疯狂Java体系图书作者。

曾任LTTEON公司的J2EE技术主管，负责该公司的企业信息化平台的架构设计。

曾任广州电信、广东龙泉科技等公司的技术培训导师。

曾兼任广东技术师范学院计算机科学系的兼职副教授。

2007年3月26日的《电脑报》专访人物。

培训的学生已在华为、立信、普信、网易、电信盈科等公司就职。

国内著名的高端IT技术作家，已出版《疯狂Java讲义》、《疯狂Android讲义》、《轻量级Java EE企业应用实战》、《疯狂Ajax讲义》、《疯狂XML讲义》、《经典Java EE企业应用实战》、《struts 2.1权威指南》、《Ruby On Rails敏捷开发最佳实践》等著作。

其中疯狂Java体系图书都是已经得到广泛的市场认同，并多次重印的畅销图书，并被多家高校选作教材，部分图书已被翻译成繁体中文版、输出到台湾地区。

## &lt;&lt;疯狂XML讲义&gt;&gt;

## 书籍目录

## 第1章 XML概述

## 1.1 XML的起源

## 1.1.1 标记语言

## 1.1.2 XML的基本概念

## 1.1.3 XML和HTML

## 1.2 XML的优势

## 1.2.1 简单易用的标记语言

## 1.2.2 严格的格式

## 1.2.3 数据逻辑和显示逻辑分离

## 1.3 XML和Java EE

## 1.3.1 配置描述

指定Web组件的配置信息是不是只能采用web.xml文件呢？

## 1.3.2 简化的数据交换

## 1.3.3 Web Service

## 1.4 XML的编辑工具

## 1.4.1 普通文本编辑工具

## 1.4.2 XMLSpy简介

## 1.4.3 使用XMLSpy编辑

## 1.4.3 XML文档

## 1.4.4 Stylus Studio简介

## 1.4.5 使用Stylus Studio编辑

## 1.4.5 XML文档

## 1.5 XML的竞争对手

## 1.5.1 Java的Annotation

## 1.5.2 轻量级的数据交换格式

## 1.5.2 ——JSON

## 1.6 本章小结

## 第2章 XML文档规则

## 2.1 XML文档的分类

## 2.1.1 格式不良的XML文档

## 2.1.2 格式良好但无效的

## 2.1.2 XML文档

## 2.1.3 有效的XML文档

## 2.2 XML文档的整体结构

## 2.2.1 有且仅有一个根元素

## 2.2.2 元素必须合理结束

## 2.2.3 元素之间必须合理嵌套

## 2.2.4 元素的属性必须有值

## 2.3 XML声明

UTF-8不是兼容Unicode吗？

那UTF-8也应该支持中文啊，  
为什么上面的XML文档会  
出现字符集错误的情况呢？

## &lt;&lt;疯狂XML讲义&gt;&gt;

## 2.4 XML元素的基本规则

## 2.4.1 合法的标签名

## 2.4.2 嵌套子元素

## 2.4.3 空元素

## 2.5 字符数据

## 2.5.1 使用实体引用

## 2.5.2 使用CDATA标记

## 2.6 注释

## 2.7 处理指令

## 2.8 W3C对于属性的使用建议

## 2.9 换行处理

## 2.10 本章小结

## 本章练习

## 第3章 DTD详解

## 3.1 XML语义约束

我一直有一个疑问：XML太“随意”了，我们想怎样定义元素都可以，想怎样嵌套子元素也行，想怎样定义属性也行，真的是这样吗？

## 3.2 引入DTD

## 3.2.1 内部DTD

## 3.2.2 外部DTD

## 3.2.3 公用DTD

## 3.3 DTD文档的结构

## 3.3.1 验证XML文档的有效性

## 3.4 定义元素

## 3.4.1 定义任意类型的元素

## 3.4.2 定义空元素

## 3.4.3 定义字符串内容的元素

## 3.4.4 定义混合内容

## 3.5 定义子元素

## 3.5.1 有序的子元素

## 3.5.2 互斥的子元素

## 3.5.3 子元素出现的频率

## 3.5.4 组合子元素

## 3.5.5 无序的子元素

上面的DTD中明明定义了<书名.../>、<价格.../>、<作者.../>和<简要介绍.../>4个元素之间具有互斥关系，它们怎么可以同时出现呢？

## 3.6 定义元素属性

## 3.6.1 对属性的约束规则

## 3.6.2 定义属性类型

## 3.7 定义实体

## 3.7.1 定义实体

## 3.7.2 定义参数实体

## 3.7.3 外部实体

## 3.7.4 外部参数实体

## &lt;&lt;疯狂XML讲义&gt;&gt;

## 3.8 定义符号

## 3.8.1 未解析实体

XML文档里的内容不是应该由XML解析器负责处理吗？

如果XML解析器不负责处理，那这些数据岂不就失去作用了？

## 3.8.2 ENTITY和ENTITIES

## 3.8.2 类型的属性

## 3.8.3 NOTATION类型的属性

## 3.9 使用XMLSpy创建DTD

## 3.10 本章小结

本章练习

## 第4章 XML Schema基本语法

## 4.1 DTD和Schema

## 4.1.1 Schema概述及其优势

## 4.1.2 Schema在Java EE中的应用

## 4.2 XML Schema入门

## 4.2.1 XML Schema根元素

## 4.2.2 在XML中引用无命名

## 4.2.2 空间的Schema

## 4.2.3 在XML中引用有命名

## 4.2.3 空间的Schema

## 4.3 Schema中的注释

## 4.4 理解Schema的数据类型

## 4.5 Schema内置类型

## 4.5.1 字符串及相关类型

## 4.5.2 数值类型

## 4.5.3 日期、时间类型

## 4.5.4 boolean类型

## 4.5.5 anyURI类型

## 4.5.6 二进制数据

## 4.6 使用限制派生新类型

## 4.6.1 指定基类型的两种方式

## 4.6.2 指定类型的两种方式

## 4.6.3 范围约束

## 4.6.4 长度约束

## 4.6.5 精度约束

## 4.6.6 枚举约束

## 4.6.7 正则表达式约束

## 4.6.8 空白处理

## 4.7 使用&lt;list.../&gt;派生列表类型

## 4.7.1 限制列表类型

## 4.8 使用&lt;union.../&gt;派生联合类型

## 4.8.1 限制联合类型

## 4.9 列表和联合结合使用

## 4.10 阻止派生新的简单类型

## 4.10.1 使用final属性

## 4.10.2 为约束指定fixed属性

## &lt;&lt;疯狂XML讲义&gt;&gt;

## 4.11 合并多个Schema

- 4.11.1 使用include元素
- 4.11.2 使用redefine元素
- 4.11.3 使用import元素

## 4.12 XMLSpy中关于Schema的

## 4.12 操作

- 4.12.1 创建新的Schema
- 4.12.2 为XML创建Schema
- 4.12.3 为XML分配Schema

## 4.13 本章小结

## 第5章 XML Schema高级知识

## 5.1 使用anyType定义任意类型

## 5.2 定义复杂类型

- 5.2.1 定义复杂类型的方式
- 5.2.2 扩展简单类型

既然派生复杂类型的方式有两种，那接下来是不是应该介绍“限制简单类型来派生复杂类型”了？

## 5.2.3 包含属性的两种方式

## 5.2.4 扩展包含简单内容的复杂

## 5.2.4 类型

## 5.2.5 使用派生类型的另一种方式

在上面的Schema中先定义了一个book\_Type类型，然后扩展该类型增加了一个price属性，为何不在定义book\_Type类型时一次性添加isbn、name和price三个属性，而非要一次又一次地添加呢？这样做有实际意义吗？

## 5.2.6 限制包含简单内容的复杂类型

## 5.2.7 限制anyType派生新类型

为什么没有通过扩展anyType来派生新类型呢？

## 5.2.8 包含子元素的两种方式

## 5.2.9 空元素类型

## 5.2.10 混合内容类型

## 5.3 复杂类型的进一步派生

## 5.3.1 限制空元素类型

## 5.3.2 扩展空元素类型

## 5.3.3 限制包含子元素的类型

## 5.3.4 扩展包含子元素的类型

## 5.3.5 限制混合内容类型

## 5.3.6 扩展混合内容类型

## 5.4 阻止派生新的类型

## 5.5 通配符

## 5.5.1 元素通配符

## 5.5.2 属性通配符

## 5.6 元素替换

## 5.6.1 阻止自己被替换

## 5.6.2 阻止指定派生类型的替换

## &lt;&lt;疯狂XML讲义&gt;&gt;

## 5.7 抽象元素和抽象类型

## 5.7.1 抽象元素

## 5.7.2 抽象类型

## 5.8 一致性约束

## 5.8.1 key约束

## 5.8.2 unique约束

## 5.8.3 keyref约束

## 5.9 元素组与属性组

## 5.10 定义符号

## 5.11 本章小结

## 本章练习

如何确定一个XML元素所在的命名空间呢？

## 第6章 命名空间详解

## 6.1 使用命名空间

## 6.2 Schema的命名空间支持

## 6.2.1 在Schema中使用命名空间

## 6.2.2 命名空间对XML文档的作用

## 6.2.3 为属性使用命名空间限定

## 6.2.4 命名空间对一致性约束的影响

## 6.2.5 局部元素和局部属性的强制限定

## 6.3 命名空间和DTD

## 6.4 本章小结

## 第7章 使用CSS显示XML文件

## 7.1 样式单简介

## 7.1.1 显示XML的两种常用样式单

## 7.1.2 样式单的优势

## 7.2 CSS的基本用法

## 7.2.1 CSS基本语法

## 7.2.2 引入外部样式文件

## 7.2.3 使用内部CSS样式

## 7.2.4 使用内联样式

## 7.3 对XML文档有效的CSS

## 7.3 选择器

## 7.4 使用CSS显示XML

## 7.5 本章小结

## 第8章 使用XSLT显示XML

## 8.1 XSL概述

## 8.2 XSLT入门

## 8.2.1 XSLT转换入门

XSLT为什么不直接将XML根元素当成根元素呢？

## 8.2.2 使用template元素定义模板

## 8.2.3 使用apply-templates处理子节点

## 8.2.4 使用value-of输出节点内容

## 8.2.5 匹配节点的模式

## 8.2.6 mode属性



## &lt;&lt;疯狂XML讲义&gt;&gt;

## 8.3 XSLT转换分类

## 8.3.1 客户端转换和服务器端转换

## 8.3.2 Xalan处理器

## 8.3.3 Saxon处理器

我发现用Saxon进行实时转换和用Xalan进行实时转换的JSP页面代码完全相同，这是为什么呢？

## 8.4 XSLT的内置模板规则

## 8.5 流程控制元素

## 8.5.1 分支处理

## 8.5.2 循环控制

## 8.5.3 排序控制

## 8.6 控制空白的处理方式

## 8.7 创建结果树

## 8.7.1 创建元素和属性

## 8.7.2 创建文本

## 8.7.3 创建处理指令

## 8.7.4 创建注释

## 8.7.5 复制

## 8.7.6 输出格式化数值

## 8.8 变量和参数

## 8.8.1 为变量和参数指定值

## 8.8.2 全局和局部的变量和参数

## 8.8.3 改变参数值

## 8.9 使用命名模板

## 8.10 包含和导入

## 8.10.1 使用import导入

## 8.10.2 使用include包含

## 8.10.3 解决模板定义冲突

## 8.11 XSLT 1.1的fallback支持

## 8.12 指定输出格式

## 8.12.1 转换XML文档的相关格式

## 8.12.2 转换HTML文档的相关格式

## 8.13 XSLT的内置函数

## 8.13.1 使用system-property函数

## 8.13.2 使用current函数返回当前节点集

## 8.13.3 使用element-available和

## 8.13.3 function-available 函数

## 8.13.4 使用unparsed-entity-uri函数

## 8.13.5 使用document 函数处理多个源

## 8.13.5 XML文档

## 8.13.6 使用format-number 函数

## 8.13.7 使用key函数

## 8.13.8 使用generate-id函数

## 8.14 XSLT 2.0的常用新功能

## 8.14.1 分组

## 8.14.2 多文档输出

## 8.14.3 字符映射

## &lt;&lt;疯狂XML讲义&gt;&gt;

- 8.14.4 < xsl:value-of > 元素的改进
- 8.14.5 数据类型绑定
- 8.14.6 正则表达式支持
- 8.14.7 用户自定义函数
- 8.15 使用XMLSpy管理XSLT
- 8.15 操作
- 8.16 本章小结
- 本章练习
- 第9章 XPath语言详解
- 9.1 XPath语言简介
- 9.1.1 XPath节点
- 9.1.2 XPath基本概念
- 9.1.3 节点关系
- 9.2 绝对路径和相对路径
- 9.2.1 XPath基路径
- 9.3 XPath基础语法
- 9.3.1 轴 ( axis )
- 9.3.2 节点测试 ( node-test )
- 9.3.3 限定谓词 ( predicate )
- 9.3.4 简化写法
- 9.4 运算符和表达式
- 9.4.1 算术运算符
- 9.4.2 比较运算符
- 9.4.3 逻辑运算符
- 9.4.4 组合多个路径的运算符
- 9.5 XPath 2.0新增的表达式
- 9.5.1 for表达式
- 9.5.2 if表达式
- 9.5.3 some/every判断表达式
- 9.6 XPath 2.0的类型支持
- 9.7 内置函数库
- 9.7.1 字符串相关函数
- 9.7.2 数值相关函数
- 9.7.3 日期和时间相关函数
- 9.7.4 boolean值相关函数
- 9.7.5 节点相关函数
- 9.7.6 序列相关函数
- 9.8 本章小结
- 第10章 DOM、SAX和JAXP解析
- 10.1 DOM、SAX和JAXP概述
- 接口的实现不是类吗，怎么接口的实现还是接口啊？
- 10.2 JAXP的DOM支持
- 10.2.1 XML文档和DOM模型
- 10.2.2 DOM树中的对象类型
- 10.2.3 DOM解析器
- 10.2.4 使用DTD验证XML文档

## &lt;&lt;疯狂XML讲义&gt;&gt;

10.2.5 使用DOM解析XML文档

10.2.6 使用DOM创建XML文档

10.2.7 使用DOM修改XML文档

10.2.8 解析DTD信息

如果我想获取DTD中的元素定义、属性定义等信息该怎么办？

10.2.9 DOM和命名空间

10.3 JAXP的SAX支持

10.3.1 SAX的处理机制

10.3.2 SAX解析器和监听器

为何XMLReader在调用时不需要传入SAX解析事件的监听器呢？

SAX解析不是总是基于事件机制的吗？

XMLReader和SAXParser到底什么关系？

我们到底应该用哪个呢？

10.3.3 使用DTD验证XML的有效性

10.3.4 使用SAX解析XML文档

10.3.5 SAX和命名空间

10.4 DOM和SAX的比较

10.5 使用XML Schema验证XML

10.5 文档

10.5.1 SchemaFactory和验证

上面的SchemaFactory的newInstance()方法中怎么还有一个schemaLanguage参数，它代表什么呢？

10.5.2 获取节点的类型信息

10.6 浏览器对DOM的支持

10.7 本章小结

本章练习

第11章 使用dom4j处理XML文档

11.1 dom4j简介

11.1.1 dom4j的封装和优势

11.1.2 下载和安装dom4j

11.2 dom4j常用API

11.3 使用dom4j访问XML文档

11.3.1 验证XML文档

11.3.2 使用dom4j解析XML文档

11.3.3 使用访问者模式遍历XML文档

11.4 使用dom4j创建XML文档

11.5 修改XML文档

11.6 使用dom4j获取命名空间信息

11.7 本章小结

本章练习

第12章 使用JDOM处理XML文档

12.1 JDOM简介

12.1.1 JDOM常用API

12.1.2 下载和安装JDOM

## &lt;&lt;疯狂XML讲义&gt;&gt;

- 12.2 使用JDOM访问XML文档的
- 12.3 实例
  - 12.2.1 验证XML文档的有效性
  - 12.2.2 使用JDOM解析XML文档
- 12.3 使用JDOM创建XML文档
- 12.4 修改XML文档
- 12.5 使用JDOM获取命名空间信息
- 12.6 选择dom4j还是JDOM
- 12.7 本章小结
  - 本章练习
- 第13章 XQuery详解
  - 13.1 XQuery简介
    - 13.1.1 XQuery和XPath的关系
    - 13.1.2 XQuery和XSLT的关系
  - 13.2 使用XQuery
    - 13.2.1 使用XMLSpy查看XQuery结果
    - 13.2.2 使用Saxon执行XQuery查询
  - 13.3 基本表达式
    - 13.3 (Primary Expressions)
      - 13.3.1 直接量 (Literal)
      - 13.3.2 变量引用
      - 13.3.3 圆括号表达式
      - 13.3.4 上下文项表达式
      - 13.3.5 函数调用
  - 13.4 序列表达式
    - 13.4.1 构造序列
    - 13.4.2 过滤表达式
    - 13.4.3 组合节点序列
  - 13.5 算术表达式
  - 13.6 比较表达式
    - 13.6.1 值比较
    - 13.6.2 通用比较
    - 13.6.3 节点比较
  - 13.7 逻辑表达式
  - 13.8 构造器
    - 13.8.1 直接构造
    - 13.8.2 计算构造
  - 13.9 FLWOR表达式
    - 13.9.1 for、let和return子句
    - 13.9.2 where子句
    - 13.9.3 order by子句
  - 13.10 if表达式
  - 13.11 some/every判断表达式
  - 13.12 与序列类型有关的表达式
    - 13.12.1 instance of
    - 13.12.2 typeswitch
    - 13.12.3 cast

## &lt;&lt;疯狂XML讲义&gt;&gt;

- 13.12.4 castable
- 13.13 模块和序言
  - 13.13.1 序言
  - 13.13.2 版本声明
  - 13.13.3 模块声明
  - 13.13.4 设置器 ( setter )
  - 13.13.5 导入设置
  - 13.13.6 命名空间声明
  - 13.13.7 默认命名空间声明
  - 13.13.8 变量声明
  - 13.13.9 函数声明
  - 13.13.10 选项声明
- 13.14 使用XQJ执行XQuery
  - 13.14.1 XQJ和JDBC的类比性  
JDBC API和接口我都知道，但JDBC实现好像很少听说啊？
  - 13.14.2 XQJ的编程步骤
- 13.15 本章小结
- 第14章 Web Service详解
  - 14.1 XML和Web Service
    - 14.1.1 Web Service概述
    - 14.1.2 Web Service平台概述
    - 14.1.3 Web Service的广泛应用
  - 14.2 SOA和云计算
    - 14.2.1 SOA和Web Service
    - 14.2.2 云计算和Web Service
  - 14.3 XML和Web Service
    - 14.3.1 SOAP基本语法
    - 14.3.2 Header元素
    - 14.3.3 Body元素
    - 14.3.4 Fault元素
  - 14.4 XML和WSDL
    - 14.4.1 WSDL基本语法
    - 14.4.2 portType元素
    - 14.4.3 binding元素
    - 14.4.4 service元素
    - 14.4.5 WSDL和UDDI
  - 14.5 使用CXF开发Web Service
    - 14.5.1 CXF概述
    - 14.5.2 下载和安装CXF
    - 14.5.3 使用CXF开发Web Service  
是否必须将Web Service服务接口的.class文件复制到客户端呢？
    - 14.5.4 动态客户端
    - 14.5.5 复杂类型的处理
    - 14.5.6 使用拦截器
    - 14.5.7 使用拦截器进行权限控制

## &lt;&lt;疯狂XML讲义&gt;&gt;

- 14.6 整合与Spring开发Web Service
  - 14.6.1 在Spring中使用CXF
  - 14.6.2 通过Spring启用CXF拦截器
  - 14.6.3 在Spring容器中配置Web Service
  - 14.6.3 客户端
- 14.7 本章小结
- 第15章 基于XML的电子公告系统
  - 15.1 系统设计
    - 15.1.1 系统组件的交互关系
  - 15.2 设计XML文档
    - 15.2.1 保存状态的XML文档
    - 15.2.2 定义XML Schema
  - 15.3 定义XSLT样式单
    - 15.3.1 为公告列表定义样式单
    - 15.3.2 为用户评论设计样式单
  - 15.4 实现控制器
    - 15.4.1 添加公告
    - 15.4.2 查看评论列表

为什么不直接将XML文档保存在Web应用根路径下呢？  
这样不是就可以让浏览者直接访问这些XML文档了么？
  - 15.4.3 添加评论
  - 15.5 本章小结
  - 本章练习
- 第16章 Web Service案例
- 第16章 ——电子拍卖系统
  - 16.1 系统功能简介和架构设计
    - 16.1.1 系统功能简介
    - 16.1.2 系统架构设计
  - 16.2 持久层设计
    - 16.2.1 系统实体
    - 16.2.2 系统E-R图和数据表
    - 16.2.3 实现Hibernate PO
    - 16.2.4 管理SessionFactory
  - 16.3 实现系统DAO层
    - 16.3.1 DAO的基础配置
    - 16.3.2 实现系统DAO组件
    - 16.3.3 配置系统DAO组件
  - 16.4 实现业务逻辑层
    - 16.4.1 定义业务逻辑组件接口
    - 16.4.2 依赖注入DAO组件
    - 16.4.3 业务逻辑组件中的异常处理
    - 16.4.4 处理用户竞价
    - 16.4.5 判断拍卖物品状态
    - 16.4.6 事务管理
    - 16.4.7 配置业务层组件
  - 16.5 使用CXF提供Web Service

<<疯狂XML讲义>>

16.5.1 启用CXF支持

16.5.2 实现Web Service

16.5.3 测试Web Service

16.6 本章小结

本章练习

## <<疯狂XML讲义>>

### 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>