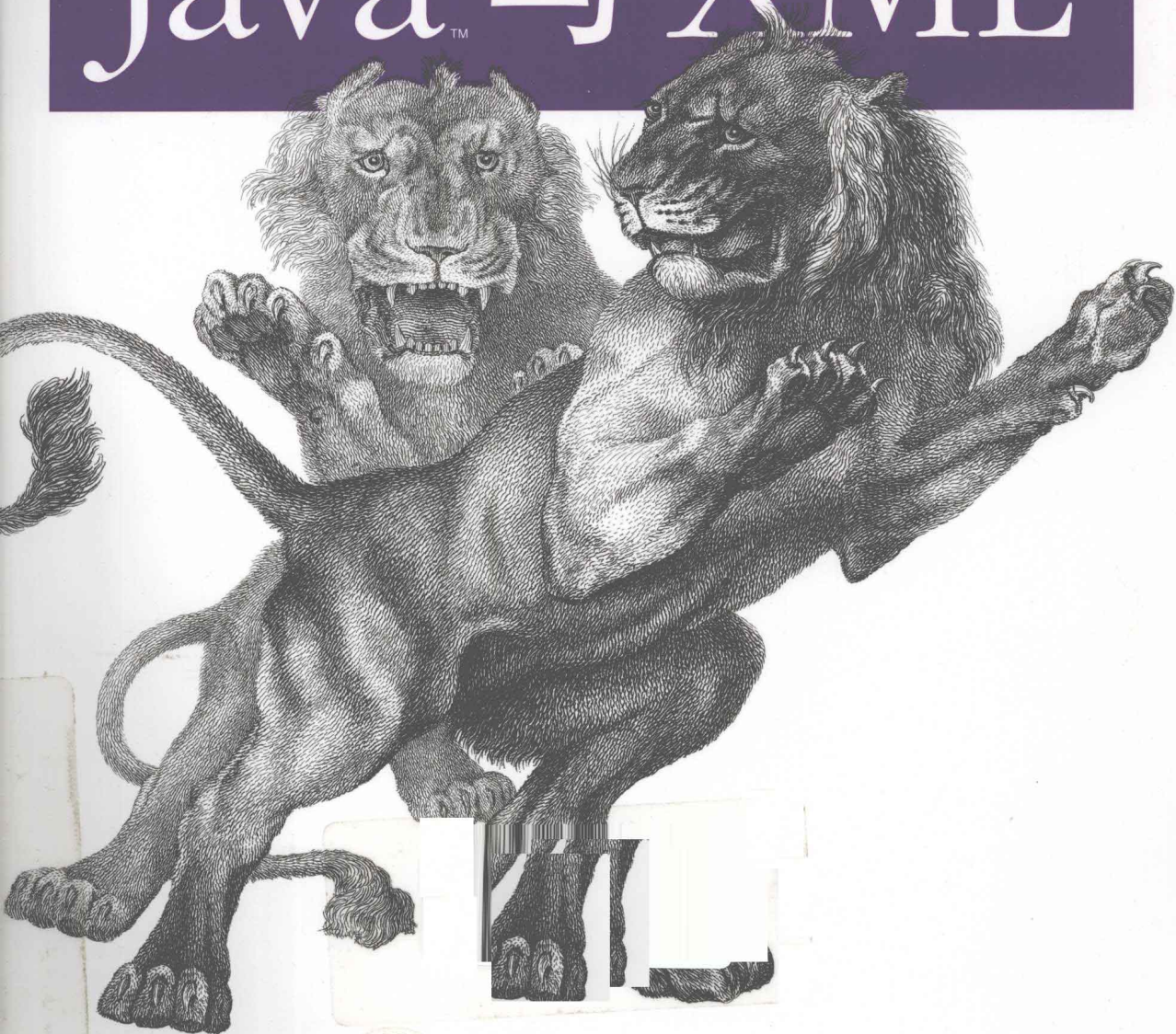


Java & XML

第三版

Java 与 XML



O'REILLY®

中国电力出版社



Brett D. McLaughlin & Justin Edelson 著

江文 唐俊 刘柯 译

第三版

Java™与XML

Brett D. McLaughlin & Justin Edelson 著
江文 唐俊 刘柯 译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权 中国电力出版社 出版

中国电力出版社

图书在版编目 (CIP) 数据

Java与XML: 第3版/ (美) 麦克劳夫林 (McLaughlin, B. D.) 等著; 江文, 唐俊, 刘柯译. —北京: 中国电力出版社, 2010.9

书名原文: Java & XML, Third Edition

ISBN 978-7-5123-0748-3

I. J… II. ①麦… ②江… ③唐… ④刘… III. ①JAVA语言—程序设计②可扩展语言, XML—程序设计 IV. TP312

中国版本图书馆CIP数据核字 (2010) 第158371号

北京市版权局著作权合同登记

图字: 01-2007-1764号

©2006 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Electric Power Press, 2010.
Authorized translation of the English edition, 2006 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由O'Reilly Media, Inc. 出版2006。

简体中文版由中国电力出版社出版2010。英文原版的翻译得到O'Reilly Media, Inc.的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc.的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式复制。

书 名/ Java与XML (第三版)

书 号/ ISBN 978-7-5123-0748-3

责任编辑/ 刘炽

封面设计/ Ellie Volckhausen, Karen Montgomery, 张健

出版发行/ 中国电力出版社 (<http://www.cepp.sgcc.com.cn>)

地 址/ 北京市东城区北京站西街19号 (邮政编码100005)

经 销/ 全国新华书店

印 刷/ 航远印刷有限公司印刷

开 本/ 787毫米×980毫米 16开本 28.25印张 529千字

版 次/ 2011年3月第一版 2011年3月第一次印刷

印 数/ 0001—3000册

定 价/ 69.00元 (册)

O'Reilly Media, Inc.介绍

为了满足读者对网络和软件技术知识的迫切需求，世界著名计算机图书出版机构 O'Reilly Media, Inc.授权中国电力出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc.是世界上在 Unix、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时也是联机出版的先锋。

从最畅销的*The Whole Internet User's Guide & Catalog*（被纽约公共图书馆评为20世纪最重要的50本书之一）到GNN（最早的Internet门户和商业网站），再到WebSite（第一个桌面PC的Web服务器软件），O'Reilly Media, Inc.一直处于Internet发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc.是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc.具有深厚的计算机专业背景，这使得O'Reilly Media, Inc.形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc.所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc.还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc.依靠他们及时地推出图书。因为O'Reilly Media, Inc.紧密地与计算机业界联系着，所以O'Reilly Media, Inc.知道市场上真正需要什么图书。

目录

前言	1
第1章 介绍	7
XML 1.0	7
XML 1.1	15
XML文档转换	16
更多内容	24
第2章 约束	26
DTDs	27
XML Schema	33
RELAX NG	40
第3章 SAX	47
安装SAX	47
使用SAX解析文档	51
内容处理类	55
错误处理类	69
第4章 SAX高级技术	75
属性和特征	75
解析实体	78
符号与非解析实体	82

DefaultHandler类	84
扩展接口	84
Filters与Writers	88
第5章 DOM	95
文档对象模型 (Document Object Model)	95
序列化	100
修改和创建XML.....	113
名称空间	123
第6章 DOM模块	127
检验支持的模块	127
DOM Level 2 模块.....	131
DOM Level 3模块.....	144
第7章 JAXP	156
不仅仅是API	156
解析XML.....	157
处理XSL.....	167
XPath	178
XML 验证	190
第8章 使用StAX进行拉式解析	198
StAX 基础	198
StAX 工厂	201
使用StAX解析	201
使用StAX输出文档	228
工厂属性	236
StAX的基本问题	239
XmlPull	239
第9章 JDOM	244
基础知识	244

PropsToXML	249
XMLProperties	259
更多的JDOM类	270
JDOM 和工厂类	278
JDOM的常见问题	283
第10章 dom4j.....	288
概述	288
使用dom4j读和写文档.....	291
遍历文档	297
Transformations	305
专用工厂	307
第11章 使用JAXB实现数据绑定	310
数据绑定基础	310
介绍JAXB	314
使用JAXB	322
其他的绑定框架	346
第12章 使用RSS的内容聚合	348
RSS是什么?	349
创建一个RSS文档	356
读取RSS订阅源	365
ROME模块	369
第13章 XML呈现	384
XML和模型-视图-控制器模式	384
使用JSP转换HTML	390
XSLT的使用	393
Ajax	401
Flash	414

第14章 展望	429
XML应用机	429
XQuery	430
快速信息集	430
更多内容	430
附录 SAX特征和属性	433

前言

欢迎阅读《Java & XML》第三版。无论你是否曾经读过本书的前作，我想你都会在本书中找到感兴趣的内容。自从本书的第二版出版后，有很多内容都发生了变化，曾经的草案已经最终发布成为规范了，其中不但增加了一些新的API，还增加了一些关于XML的新用法。其中变化最大的二个API，也许要算是本书第二版中曾经讨论过的JDOM和JAXB了，最终发布版与在第二版编写时所发布的预览版之间出现了重大变化。虽然我也希望本书的内容永远不会过时，但是Java和XML世界中的变化速度，使得要做到这一点显然不太可能。不过，即使一些技术细节发生变化，我也相信本书的内容可以满足未来几年大部分的需求。

本书的结构

本书的结构分成三个部分：第一部分由第1章和第2章组成，它们介绍了XML的基本知识和相关标准。第二部分由第3章～第11章构成，解释了各种各样与XML有关的API，它们可以完成创建、处理、输出，以及与XML文档有关的几乎任何操作。最后一部分中，第12章和第13章描述了XML的两个重要应用，然后就是最后一章和附录了。

第1章 介绍

本章会介绍XML的基本知识，如果你之前从来没有使用过XML，本章会让你学到学习后续章节所需的全部知识。在介绍XLST和XPath之前，还会简要回顾一下XML1.0和1.1之间的区别。

第2章 约束

本章会介绍三种定义XML文档结构的方法：DTD、W3C XML Schema和RELAX NG Schema，以及介绍如何使用这些标准来定义一个XML文档的结构和如何确保

一个XML文档符合该结构（也就是验证）。同时你还可以学会如何在不同的约束类型之间进行转换。

第3章 SAX

本书中第一种要介绍的用于处理XML的Java API是：用于XML的简单API（SAX），其中会详细介绍解析的全过程和SAX所捕获的各种事件，并演示其应用。

第4章 SAX高级技术

本章将深入讨论SAX，其中会涉及到那些在API中不常使用但却功能强大的部分方法。从中读者可以了解到如何使用XML Filter进行链式回调，如何使用XML Writer输出XML，同时还可以学会使用诸如LexicalHandler和DeclHandler等一些不常用的处理方法的应用。

第5章 DOM

本章将沿着XML知识体系继续前进，讲述另一种Java和XML API：文档对象模型（DOM）。在本章，读者可以了解到DOM的基础知识和最新规范（DOM Level 3），以及如何读写DOM树等内容。

第6章 DOM模块

本章进一步讲述DOM，会介绍各种Level2和Level3中的模块，如：Traversal、Range、Events、Style、HTML、Load and Save和Validation。

第7章 JAXP

本章将研究用于XML处理的Java API（JAXP），包括JAXP1.3中介绍的XPath和Validation API。

第8章 使用StAX进行拉式解析

用于XML的流式API（StAX）是最新的Java XML API标准。本章将会学习如何使用StAX，以及它与SAX和DOM之间的区别。同时还会简要介绍一下目前仍在使用的StAX的前身——XMLPull。

第9章 JDOM

本章将介绍JDOM，一种Java特定的对象模型API，会对JDOM 1.0的方方面面进行详细介绍。

第10章 dom4j

本章讨论另一个Java特定的对象模型API——dom4j，将介绍包括dom4j所独有的、面向对象的转换API在内的等与JDOM和DOM不同的功能。

第11章 使用JAXB实现数据绑定

使用数据绑定，应用程序就不再需要考虑与XML文档的处理细节了，只要使用Java

对象模型就够了。Sun公司开发了一套用于XML数据绑定的标准JAXB(针对XML绑定的Java体系结构)。在本章你将学习到JAXB1.0和2.0的有关内容，它们都是使用数据绑定的基础，同时还会对数据绑定的使用时机等有关问题进行讨论。

第12章 使用RSS的内容聚合

本章介绍如何使用XML进行内容聚合，其中，还对RSS标准和Atom两者进行了简要的讨论。另外，你还将学习到如何使用前述章节讨论的API和特定于RSS的API ROME来创建和读取RSS订阅源。甚至还可以学习到如何针对苹果公司的iTunes Podcast目录订阅创建自己的播客订阅源。

第13章 XML呈现

在本章中，我们将学习多种使用XML在Web应用程序中进行呈现或可视化的技术，还可以了解到XML是如何快速地从新鲜事物变成主流的动态Web应用技术中的关键组件的。

第14章 展望

最后一章对一些在本书中没有详细介绍的技术进行了概述。对于那些有更高要求的读者来说，可能会在将来继续学习这些内容。

附录 SAX特征和属性

本附录详细描述了SAX2.0解析实现中可用的所有特征和属性。

排版约定

本书使用以下排版约定：

等宽字体 (Constant width)

用于描述命令行中的输入，输出和示例代码。

等宽斜体 (*Constant width Italic*)

表示示例和注册表键值中的占位符（用来代替一个实际的名字）。

等宽黑体 (**Constant width bold**)

表示用户输入以及代码中的高亮部分。

斜体 (*Italic*)

用来表示目录名、程序名和命令名，以及命令行选项和网址。

注意： 用来表示秘诀、建议或通用注解。

警告： 用来表示警告或注意事项。

使用示例代码

本书的目的在于帮助读者完成工作。通常，可能会在程序和文档中使用本书代码。除非大量的复制本书中代码，否则读者无需联系我们，获取许可。例如，程序中使用了本书中的一些代码片断，这种情况就不必获取许可。出售或发布O'Reilly图书中的示例CD-ROM必须获得许可。回答问题时，引用本书的内容及示例代码无需获得许可，而将本书中大量的示例代码合并到自己的产品文档中，这种行为是需要获得许可的。

我们对以下内容重视但无需责任归属，包括标题、作者、出版社以及ISBN。例如：
"Java & XML, Third Edition, by Brett McLaughlin and Justin Edelson. Copyright 2007 O'Reilly Media, Inc., 978-0-596-10149-7."

如果读者感觉对示例代码的使用已经超出了上面提到的使用许可，请通过电子邮件与我们联系，邮箱是：permissions@oreilly.com。

联系我们

本书的内容都经过测试，尽管我们做了最大的努力，但错误和疏忽仍然是在所难免的。如果你发现有什么错误，或者是对将来的版本有什么建议，请通过下面的地址告诉我们：

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街2号成铭大厦C座807室（100035）
奥莱利技术咨询（北京）有限公司

本书的网页上列出了勘误，示例和其他相关信息，可以从以下页面进行访问：

原文书

<http://www.oreilly.com/catalog/9780596101497>

中文书

<http://www.oreilly.com.cn/book.php?bn=9787512307483>

如果想要发表关于本书的评论或询问技术问题，请发送邮件到：

bookquestions@oreilly.com

关于图书，会议，资源中心和O'Reilly网络的其它信息，请查看我们的站点：

<http://www.oreilly.com>

<http://www.oreilly.com.cn>

致谢

没有O'Reilly公司员工的努力，本书就不可能面世，这里要特别感谢我的编辑Mike Loukides和他的助手Caitrin McCullough，以及产品专家Marlowe Shaeffer和Linley Dolby。此外，还要感谢参与本书评审的全部人员，Colin Chow, Greg Graf, Noah Jones, Michael Rosencrantz, Victor Toni, 和Deepak Vohra他们都对本书的改进做出了贡献。

站在巨人的肩膀上，对于一本书的作者而言尤其应该这样。显然，没有全世界开发者的努力和贡献，就不会有本书所讨论的技术。需要感谢的人还有还多，这里无法一一列举，只能一并感谢。

Brett D. McLaughlin

看到作为第三版的本书即将出版，是我今年所遇到的最酷的事情之一，主要是因为它代表了团队努力的成果。当我被拉入技术书籍的“阴暗面”Head First的世界时，我还曾一度怀疑是否还会有Java & XML的第三版。不过要感谢Justin Edelson的美妙天分，为此我们有了比第一版和第二版更好的第三版。Justin 来到项目团队后，对已经完成的文字和技术细节进行润色和加工，并在短短几个月的时间内研制了数量惊人的大量材料。Justin，能与你共事并在本书的封面上一同署名，这令我感到无比自豪，你为本书的出版做出了巨大的贡献。

对于本书能够最终完成，我一直想对O'Reilly的团队说声“谢谢”，尤其是Mike Loukides。正是本书使我与O'Reilly开始有了合作，看到大家齐心协力为第三版的面世而努力工作，真是一件令人愉快的事情。谢谢大家了，能够编写一本由O'Reilly出版的书，这真是一件乐事。

Justin Edelson

首先要感谢Brett和 Mike，非常感谢你们给了我这样一个宝贵的机会。其次是我的妻子Elizabeth，没有你的帮助和支持，我根本无法完成这样一份工作。然后是Owen，尽管你要读到本书还需要几年的时间，但对我来说你就是我每天创作灵感的源泉。接下来是我

和Beth的父母以及我的兄弟姐妹，谢谢你们为我所做的一切。最后是我们的狗，Burt和Lola，谢谢你们在沙发上偶尔为我腾出的空地。

谢谢在MTV共事的每一个人，特别是 Lily Chang, Dmitry Grinberg, Steven Hsu, Glenn Goldstein, Greg Clayman和 Nick Rock，感谢你们的支持和帮助。谢谢同一办公室的伙伴 Dan Brustein，过去的10年中，如果没有你将完全不一样。谢谢Michael Joyce，感谢你曾给出的指点。

最后，我想将本书献给我的祖父Max Edelson，没有他，我生命中大多数美好的事情（包括合著本书）都不会发生。

在接下来的两章中，我将提供一个关于XML及其校验的速成课程。虽然已经有许多关于XML及其相关规范的资料可以参考，但我还是建议通过快速阅读本章并将之运用于Java，来掌握这部分知识，对于有些完全不熟悉XML的读者，你们可能想阅读如下参考书：

- 《XML in a Nutshell》，作者：Elliote Rusty Harold 和W. Scott Means
- 《Learning XML》，作者：Erik Ray
- 《Learning XSLT》，作者：Michael Fitzgerald
- 《XSLT》，作者：Doug Tidwell

这些都是O'Reilly公司出版的书籍，我已经将它们放置在我自己的案头上了。说到这儿，让我们一起来学习吧。

XML 1.0

所有的XML文档都源自XML 1.0推荐标准（Recommendation），可以到<http://www.w3.org/TR/REC-xml>查看完整的标准。示例1-1是遵循此标准的一个XML文档。我将用它来说明几个重要的概念。

示例1-1：典型的XML文档都是非常冗长的

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns="http://purl.org/rss/1.0/" xmlns:admin="http://webns.net/mvcb/"
  xmlns:l="http://purl.org/rss/1.0/modules/link/"
```

```

        xmlns:content="http://purl.org/rss/1.0/modules/content/">
<!--Generated by Blogger v5.0-->
<channel rdf:about="http://www.neilgaiman.com/journal/journal.asp">
  <title>Neil Gaiman's Journal</title>
  <link>http://www.neilgaiman.com/journal/journal.asp</link>
  <description>Neil Gaiman's Journal</description>
  <dc:date>2005-04-30T01:57:38Z</dc:date>
  <dc:language>en-US</dc:language>
  <admin:generatorAgent rdf:resource="http://www.blogger.com/" />
  <admin:errorReportsTo rdf:resource="mailto:rss-errors@blogger.com" />
  <items>
    <rdf:Seq>
      <rdf:li rdf:resource="http://www.neilgaiman.com/journal/2005/04/three-photographs.
asp" />
      <rdf:li
rdf:resource="http://www.neilgaiman.com/journal/2005/04/jetlag-morning.asp" />
      <rdf:li
rdf:resource="http://www.neilgaiman.com/journal/2005/04/demon-days.asp" />
      <rdf:li
rdf:resource="http://www.neilgaiman.com/journal/2005/04/more-from-mailbag.asp" />
      <rdf:li
rdf:resource="http://www.neilgaiman.com/journal/2005/04/two-days.asp" />
      <rdf:li
rdf:resource="http://www.neilgaiman.com/journal/2005/04/finishing-things.asp" />
    </rdf:Seq>
  </items>
</channel>

<!-- 未完... -->
</rdf:RDF>

```

注意：有些读者可能会对这个XML文档感兴趣，它是Neil Gaiman博客（<http://www.neilgaiman.com>）的RSS订阅源，其中使用了大量的RSS语法，这些我将会在第12章中进行详细的介绍。

XML1.0标准的许多内容都非常直观，如果你曾手工编写过HTML或SGML代码，那一定不会对元素（element，如例中的items和channel）和属性（attribute，如resource和content）等概念感到陌生。标准中对XML的文档结构和如何使用这些基本要素进行了定义，但其主要精力放在诸如定义空格等棘手的问题上，而不是再引入更多的陌生概念。示例1-1中某些元素的写法可能有点不常见：

[前缀]:[元素名]

如rdf:li，表示XML名称空间中的元素，这我会在后面进行解释。

XML文档可以分为两个基本的组成部分：头部（header），用于为XML解析器和XML程序提供如何处理文档的信息；内容（content），也就是XML数据本身。虽然这种划分

很宽泛，但它有助于将XML文档中给应用程序的指示和文档内容本身区分开来。作为开发人员，理解这种划分是非常重要的。头部就是XML声明，其格式如下：

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

头部中还可以包含XML文档的编码方式和说明文档是独立的还是要引用其他文档才算完整的Standalone属性：

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

头部的其余部分由多个DOCTYPE声明之类的项组成（本例中不存在）。

```
<!DOCTYPE RDF SYSTEM "DTDs/RDF-gaiman.dtd">
```

从上面的声明可以看出，该声明引用了本机DTD目录下的*RDF-gaiman.dtd*文件。只要你引用的是自己系统中的文件，不管使用的是相对路径还是绝对路径，或者是URL，都要加上SYSTEM关键字。与之相对的另一个关键字是PUBLIC，其后接一个公共的标识符。这意味着W3C或其他组织已经定义了该标识符所指定的标准DTD，例如，XHTML 1.0的DTD语句如下所示：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

这里提供了公共标识符（以"-//"开头的字符串），后面是个系统标识符（URL）。如果公共标识符无法解析，那就会使用系统标识符来代替。

在文件头部，可能还会看到处理指令（processing instruction），它们通常被认为是XML文档头部的一部分，而不是内容。这些指令看起来就像下面这个样子：

```
<?xml-stylesheet href="XSL/JavaXML.html.xsl" type="text/xsl"?>  
<?xml-stylesheet href="XSL/JavaXML.wml.xsl" type="text/xsl"  
media="wap"?>  
<?cocoon-process type="xslt"?>
```

每条处理指令都有一个目标（即第一个单词，如xml-stylesheet 或cocoon-process）和数据（其余部分）。通常情况下，数据都是以名-值（name-value）对的形式显示，这样有助于提高可读性。这虽然是种好方法，不过并不作要求，因此不必拘泥于此。

除了第一行之外，XML文档中大段大段的都是内容了，也就是元素、属性，以及放入其中的数据。

根元素

在XML文档中，根元素是最高级别的元素，它是文档中第一个开始且最后一个结束的标