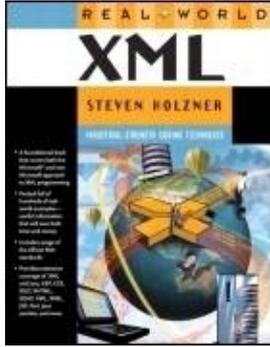


[Team LiB]



[Table of Contents](#)

[Index](#)

Real World XML

By [Steven Holzner](#)

Publisher : New Riders Publishing

Pub Date : January 16, 2003

ISBN : 0-7357-1286-7

Pages : 1200

Steven Holzner's friendly, easy-to-read style has turned this book (formerly known as Inside XML) into the leading reference on XML. Unlike other XML books, this one is packed with hundreds of real-world examples, fully tested and ready to use!

Holzner teaches you XML like no other author can, covering every major XML topic today and detailing the ways XML is used now--connecting XML to databases (both locally and on web servers), styling XML for viewing in today's web browsers, reading and parsing XML documents in browsers, writing and using XML schemas, creating graphical XML browsers, working with the Simple Object Access Protocol (SOAP), and a great deal more. Real World XML is designed to be the standard in XML coverage--more complete, and more accessible, than any other.

"The author's approach is definitely bottom up, written in a highly personable tone. He makes efficient use of example code, which sets this book apart from many I have read in the past. His examples bring to life the code without overwhelming the reader, and he does not present any examples for which the reader has not been prepared. In addition, no prior knowledge of XML is assumed. As such, this is an excellent book for both beginners and intermediate level web designers and programmers. Experts, too, will find this book of value, due to its emphasis on real world applicability. Overall, this book will benefit all web developers and programmers, with a special emphasis on beginner and intermediate developers."-Donna A. Dulo, MS, MA, Senior Systems Engineer, U.S. Department of Defense

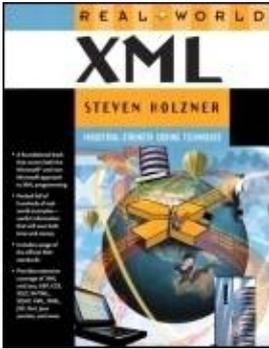
"This book will provide a brilliant basis for anyone wishing to keep up to speed with the new XML developments."-Mr. Andrew Madden, Department of Computer Science, University of Wales

"I found this book's strengths to be: its exhaustive specification reference for the conscientious developer; access to the official specs, which is key; the wide variety of choices provided for all aspects of XML; several alternatives provided for each editor, browser, parser, stylesheet transform engine, and programming language; and working examples that show the power of the tools used."-Jaime Ryan, Software

Developer/Documentation Manager, Blue Titan Software

[[Team LiB](#)]

[[Team LiB](#)]



[Table of Contents](#)

[Index](#)

Real World XML

By [Steven Holzner](#)

Publisher : New Riders Publishing

Pub Date : January 16, 2003

ISBN : 0-7357-1286-7

Pages : 1200

[Copyright](#)

[Reviewers Praise Real World XML](#)

[Comments from the Previous Edition](#)

[About the Author](#)

[About the Technical Reviewers](#)

[Acknowledgments](#)

[Preface](#)

[What Is Inside This Book](#)

[Who Is This Book For?](#)

[At What Level This Book Is Written](#)

[Conventions Used](#)

[Chapter 1. Essential XML](#)

[Markup Languages](#)

[What Does XML Look Like?](#)

[What Does XML Look Like in a Browser?](#)

[What's So Great About XML?](#)

[Well-Formed XML Documents](#)

[Valid XML Documents](#)

[Parsing XML Yourself](#)

[XML Resources](#)

[XML Editors](#)

[XML Browsers](#)

[XML Parsers](#)

[XML Validators](#)

- CSS and XSL
- XLinks and XPointers
- URLs Versus URIs
- ASCII, Unicode, and the Universal Character System
- XML Applications

Chapter 2. Creating Well-Formed XML Documents

- The World Wide Web Consortium
- What Is a Well-Formed XML Document?
- Markup and Character Data
- The Prolog
- The XML Declaration
- Comments
- Processing Instructions
- Tags and Elements
- The Root Element
- Attributes
- Building Well-Formed Document Structure
- CDATA Sections
- XML Namespaces
- Infosets
- Canonical XML

Chapter 3. Valid Documents: Creating Document Type Definitions

- Valid XML Documents
- Creating Document Type Declarations
- Creating Document Type Definitions
- Validating Against a DTD
- Element Declarations
- ANY
- Child Element Lists
- "v2" width="21">
- Choices
- Mixed Content
- Empty Elements
- DTD Comments
- A DTD Example
- External DTDs
- Using Document Type Definitions with URLs
- Public Document Type Definitions
- Using Both Internal and External DTDs
- Namespaces and DTDs

Creating Subsequences with Parentheses

Chapter 4. DTDs: Entities and Attributes

- Entities
- Attributes
- Creating Internal General Entities
- Creating External General Entities
- Building a Document from Pieces
- Predefined General Entity References

- Creating Internal Parameter Entities
- External Parameter Entities
- Using INCLUDE and IGNORE
- All About Attributes
- Declaring Attributes in DTDs
- Setting Default Values for Attributes
- Attribute Types
- Embedding Non-XML Data in a Document
- Embedding Multiple Unparsed Entities in a Document

Chapter 5. Creating XML Schemas

- Using XML Schemas in Internet Explorer
- Writing XML Schemas
- What Elements Can You Use in Schemas?
- Declaring Types and Elements
- Specifying How Often Elements Can Occur
- Specifying Default Values for Elements
- Specifying Attribute Constraints and Defaults
- Creating Simple Types
- Creating Simple Types Using Facets
- Using Anonymous Type Definitions
- Creating Empty Elements
- Creating Mixed-Content Elements
- Annotating Schemas
- Creating Choices
- Creating Element Groups
- Creating Attribute Groups
- Creating all Groups
- Schemas and Namespaces

Chapter 6. Understanding JavaScript

- What Is JavaScript?
- JavaScript Is Object-Based
- Using Object Properties and Methods in JavaScript
- Using Events in JavaScript
- Programming in JavaScript
- Working with Data in JavaScript
- Commenting Your JavaScript
- Working with JavaScript Operators
- Creating JavaScript if Statements
- Creating JavaScript if&"v2" width="21"> Creating switch Statements
- Creating JavaScript for Loop Statements
- Creating while Loop Statements
- Creating do&"v2" width="21"> Creating Functions in JavaScript
- Passing Values to Functions
- Creating Objects in JavaScript
- Using String Objects in JavaScript
- Using the Array Class to Create Arrays
- Working with Events
- Getting Event Information
- Handling Mouse Events

Chapter 7. Handling XML Documents with JavaScript

- The W3C DOM
- The XML DOM Objects
- The DOMDocument Object
- The XMLDOMNode Object
- The XMLDOMNodeList Object
- The XMLDOMNamedNodeMap Object
- The XMLDOMParseError Object
- The XMLDOMAttribute Object
- The XMLDOMElement Object
- The XMLDOMText Object
- Loading XML Documents
- Using XML Data Islands
- Getting Elements by Name
- Getting Attribute Values from XML Elements
- Parsing XML Documents in Code
- Parsing an XML Document to Display Node Type and Content
- Parsing an XML Document to Display Attribute Values
- Handling Events While Loading XML Documents
- Validating XML Documents with DTDs in Internet Explorer
- Scripting XML Elements
- Editing XML Documents with Internet Explorer

Chapter 8. XML and Data Binding

- Data Binding in Internet Explorer
- Using Data Source Objects
- Binding Data to HTML Elements
- Using Data Binding with XML
- XML Single-Record Binding Using XML Data Islands
- The Properties, Methods, and Events of XML DSOs
- Tabular Data Binding and XML
- Single-Record Data Binding with the XML DSO
- Tabular Data Binding with the XML DSO
- XML and Hierarchical Data
- Handling Variable-Size Hierarchical Data in XML Documents
- Searching XML Data

Chapter 9. Cascading Style Sheets

- Attaching Stylesheets to XML Documents
- Selecting Elements in Stylesheet Rules
- Grouping Elements in Selectors
- Creating Pseudo-Elements
- Classes

- Creating Pseudo-Classes
- Selecting by ID
- Using Contextual Selectors
- Using Inline Styles
- Using Inheritance
- Understanding Cascades
- Creating Style Rules
- Creating Block Elements
- Styling Text
- Setting Colors and Backgrounds
- Margins, Indentations, and Alignments
- Applying Styles to Lists
- Creating Borders
- Displaying Images
- Absolute Positioning
- Relative Positioning
- The Formal Style Property Specifications
- Text Properties
- Font Properties
- Background and Color Properties
- Table Properties
- Positioning and Block Properties
- Box Properties
- Visual Effects Properties
- List Properties

Chapter 10. Understanding Java

- Java Resources
- Writing Java Programs
- Java Is Object-Oriented from the Ground Up
- Getting the Java SDK
- Creating Java Files
- Writing Code: Creating an Application
- Compiling Code
- Running Java Applications
- Commenting Your Code
- Importing Java Packages and Classes
- Creating Variables in Java
- Creating Arrays in Java
- Creating Strings in Java
- Java Operators
- Java Conditional Statements: if, if&"docEmphasis">switch
- Java Loops: for, while, do&"v2" width="21">
- Declaring and Creating Objects
- Creating Methods in Java
- Creating Java Classes

Chapter 11. Java and the XML DOM

- Creating a Parser
- Displaying an Entire Document
- Filtering XML Documents
- Creating a Windowed Browser
- Creating a Graphical Browser

Navigating in XML Documents
 Modifying XML Documents

Chapter 12. Java and SAX

Working with SAX
 Displaying an Entire Document
 Filtering XML Documents
 Creating a Windowed Browser
 Creating a Graphical Browser
 Navigating in XML Documents
 Modifying XML Documents

Chapter 13. XSL Transformations

Using XSLT Stylesheets in XML Documents
 XSL Stylesheets
 Making a Transformation Happen
 Creating XSLT Stylesheets
 The `xsl:apply-templates` Element
 Getting the Value of Nodes with `xsl:value-of`
 Handling Multiple Selections with
`xsl:for-each`
 Specifying Patterns for the `match` Attribute
 Specifying Patterns for the `select` Attribute
 Understanding XPath
 The Default XSLT Rules
 Altering Document Structure Based on Input
 Creating New Attributes
 Generating Comments with `xsl:comment`
 Copying Nodes
 Sorting Elements
 Using `xsl:if`
 Using `xsl:choose`
 Controlling Output Type

Chapter 14. XSL Formatting Objects

Formatting an XML Document
 Creating the XSLT Stylesheet
 Transforming a Document into Formatting
 Object Form
 Creating a Formatted Document
 The XSL Formatting Objects
 The XSL Formatting Properties
 Working with Formatting Objects
 The Document Root: `fo:root`
 The Master Set Layout: `fo:layout-master-set`
 Using a Page Master: `fo:simple-page-master`
 Creating Regions
 Creating Page Sequences: `fo:page-sequence`
 Creating Flows: `fo:flow`
 Creating Block-level Content: `fo:block`
 Inline-Level Formatting Objects
 Creating Page Numbers: `fo:page-number`
 Creating Tables
`fo:table`

Creating Lists

Chapter 15. XLinks and XPointers

XLinks

XPointers

All About XLinks

Declaring the XLink Attributes

The xlink:type Attribute

Locating Resources with xlink:href

Describing Resources: xlink:role and
xlink:title

The xlink:show Attribute

The xlink:actuate Attribute

The xlink:arcrole and xlink:label Attributes

Extended Links

Creating Arcs with the xlink:from and xlink:to
Attributes

Inline Versus Out-of-Line Links

All About XPointers

XPointer Axes

XPointer Node Tests

XPointer Predicates

XPointer Location Set Functions

Using XPointer Points

Using XPointer Ranges

XPointer Abbreviations

Chapter 16. Essential XHTML

XHTML Version 1.0

XHTML Version 1.1

XHTML Version 2.0

XHTML Basic

XHTML Checklist

Differences Between XHTML and HTML

Automatic Conversion from HTML to
XHTML

Validating Your XHTML Document

`<html>` The Document Element

`<head>` Creating a Web Page's Head

`<title>` The Document's Title

`<body>` The Document's Body

`<!-- -->` Comments

Headings: `<h1>` Through `<h6>`

Text Handling

`` Making Text Bold

`<i>` Making Text Italic

`<u>` Underlining Text

`` Specifying a Text Font

`
` Creating Line Breaks

`<p>` Organizing Text into Paragraphs

`<hr>` Creating Horizontal Rules

`<center>` Centering Displayed Text

`<div>` Formatting Text Blocks

`` Formatting Text Inline

Chapter 17. XHTML at Work

- Displaying an Image
- <a> Creating a Hyperlink or Anchor
- <link> Setting Link Information
- <table> Creating Tables
- <tr> Creating Table Rows
- <th> Creating Table Headings
- <td> Creating Table Data
- <frameset> Creating Documents with Frames
- <frame> Creating Individual Frames
- Using Stylesheets in XHTML
- <style> Creating Embedded Stylesheets in XHTML
- Using Inline Styles in XHTML
- <script> Using Script Programming
- <form> Creating XHTML Forms
- <input type = "button"> Creating Controls
- <input type="text"> Creating Text Fields
- Extending XHTML Creating New Elements and Attributes
- Extending Public XHTML DTDs
- Extending XHTML with a Driver DTD
- All About XHTML 1.1 Modules
- Modifying XHTML Content Models

Chapter 18. SOAP and RDF

- SOAP
- A SOAP Example Using Java
- RDF
- RDF Syntax
- RDF Root Element
- RDF Namespace
- RDF Description Element
- RDF Property Elements
- Dublin Core
- Describing Multiple Properties
- Describing Multiple Resources
- Nesting Resources
- Referring to Resources by Reference
- Using XML in Property Elements
- Using Abbreviated RDF Syntax
- RDF Containers
- Using the Bag Container
- Using the Seq Container
- Using the Alt Container
- Making Statements About Containers
- Making Statements About the Items in a Container
- Selecting Container Items by Prefix
- Creating RDF Schemas

Chapter 19. Vector Markup Language

- Creating VML Documents

- The VML Elements
- The <shape> Element
- Using Predefined Shapes
- Coloring Shapes
- Scaling Shapes
- Positioning Shapes
- The absolute Position Style
- The <shadow> Element
- The <fill> Element
- Using the <shapetype> Element
- More Advanced VML

Chapter 20. WML, ASP, JSP, Servlets, and Perl

- XML and ASP
- XML and Java Servlets
- JSP
- XML and Perl
- WML
- Getting Starting with WML
- Setting Text Alignment
- Basic Text Styling
- The <do> Element
- The <prev> Element
- Hyperlinks
- Tables
- Text Input
- Select Elements
- Timers
- Connecting to the Server
- Images

Appendix A. The XML 1.0 Recommendation (Second Edition)

- Extensible Markup Language (XML) 1.0
(Second Edition)
- Abstract
- Status of this Document
- Table of Contents
- Section 1. Introduction
- Section 2. Documents
- Section 3. Logical Structures
- Section 4. Physical Structures
- Section 5. Conformance
- Section 6. Notation
- Section A. References
- Section B. Character Classes
- Section C. XML and SGML
(Non-Normative)
- Section D. Expansion of Entity and Character
References (Non-Normative)
- Section E. Deterministic Content Models
(Non-Normative)
- Section F. Autodetection of Character
Encodings (Non-Normative)

Section G. W3C XML Working Group
 (Non-Normative)
 Section H. W3C XML Core Group
 (Non-Normative)
 Section I. Production Notes (Non-Normative)

Index
 [Team LiB]
 [Team LiB]

Copyright

Copyright © New Riders Publishing is a division of Peachpit Publishing Group

210 W. 103rd St.

Indianapolis, IN 46290

Find us on the World Wide Web at: <http://www.newriders.com>

For resources mentioned in this book, see: <http://www.newriders.com>

Credits

Associate Publisher: Stephanie Wall

Production Manager: Gina Kanouse

Senior Product Marketing Manager: Tammy Detrich

Publicity Manager: Susan Nixon

Senior Project Editor: Lori Lyons

Copy Editor: Krista Hansing

Indexers: Chris Morris and Lisa Stumpf

Composition: Wil Cruz and Amy Parker

Interior Design: Jeff Carlson

Cover Illustration: Alan Clements

Colophon

This book was written and edited in Microsoft Word, and laid out in QuarkXPress. The fonts used for the body text are Minion and Mono. It was printed on 50# Husky Offset Smooth paper at Edwards Brothers, Inc. in Ann Arbor, Michigan. Prepress consisted of PostScript computer-to-plate technology (filmless process). The cover was printed at Moore Langen Printing in Terre Haute, Indiana, on 12 pt., coated on one side.

Notice of Rights

All rights reserved. No part of this book may be reproduced or transmitted in any form by any means electronic, mechanical, photocopying, recording, or otherwise without the prior written permission from the publisher, except for the inclusion of brief quotations in a review. For information on getting permission for reprints and excerpts, contact staceybeheler@newriders.com.

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. New Riders nor Peachpit Press cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

W3C® Massachusetts Institute of Technology (MIT), Institut National de Recherche en Informatique et en Automatique (INRIA), Keio University (Keio)

Notice of Liability

The information in this book is distributed on an "As Is" basis, without warranty. While every precaution has been taken in the preparation of this book, neither the authors nor New Riders nor Peachpit Press shall have any liability nor responsibility to any person or entity with respect to any loss or damages caused or alleged to be caused directly or indirectly from the information contained in this book or by the computer software or hardware products described in it.

Library of Congress Catalog Card Number: 2002100320

06 05 04 03 7 6 5 4 3 2 1

Printed and bound in the United States of America

Dedication

To Nancy, of course!

[[Team LiB](#)]

[[Team LiB](#)]

Reviewers Praise Real World XML

The author's approach is definitely bottom up, written in a highly personable tone. He makes efficient use of example code, which sets this book apart from many I have read in the past. His examples bring to life the code without overwhelming the reader, and he does not present any examples for which the reader has not been prepared. In addition, no prior knowledge of XML is assumed. As such, this is an excellent book for both beginners and intermediate level web designers and programmers. Experts, too, will find this book of value, due to its emphasis on real world applicability. Overall, this book will benefit all web developers and programmers, with a special emphasis on beginner and intermediate developers.

Donna A. Dulo, MS, MA, Senior Systems Engineer, U.S. Department of Defense

This book will provide a brilliant basis for anyone wishing to keep up to speed with the new XML developments.

Mr. Andrew Madden, Department of Computer Science, University of Wales

I found this book's strengths to be: its exhaustive specification reference for the conscientious developer; access to the official specs, which is key; the wide variety of choices provided for all aspects of XML; several alternatives provided for each editor, browser, parser, stylesheet transform engine, and programming language; and working examples that show the power of the tools used.

Jaime Ryan, Software Developer/Documentation Manager, Blue Titan Software

[Team LiB]

[Team LiB]

Comments from the Previous Edition

Steve Holzner has successfully given each topic good, detailed treatment each chapter builds on the last providing the reader a solid understanding of XML, its related technologies, and how it fits into the current application development arena. This book will appeal to the novice and experienced XML programmers alike.

Beth Breidenbach, Senior Software Engineer, MCSD Getronics

Inside XML's building block approach, clear explanations, and extensive examples are perfectly targeted for an introduction to XML.

Tom Comerford, Director Supratext, LLC

Readers will find this book an excellent companion for XML development. Well-constructed examples are ample throughout, making this book not only a genuine learning tool, but a fine resource as well.

Andrew J. Indovina, Software Engineer/e-Commerce Developer

[Team LiB]

[Team LiB]

About the Author



Steven Holzner is an award-winning author who has been writing about XML for as long as it's been around. He's the author of 75 books on programming, and a former contributing editor to PC Magazine. A number of his books have been programming bestsellers, and he's had books translated into 16 languages around the world, selling more than 1.5 million copies. He received his Ph.D. at Cornell University, and has been on the faculty of both Cornell and MIT. His hobbies include travel, chess, classical music, and writing books on philosophy.

[Team LiB]

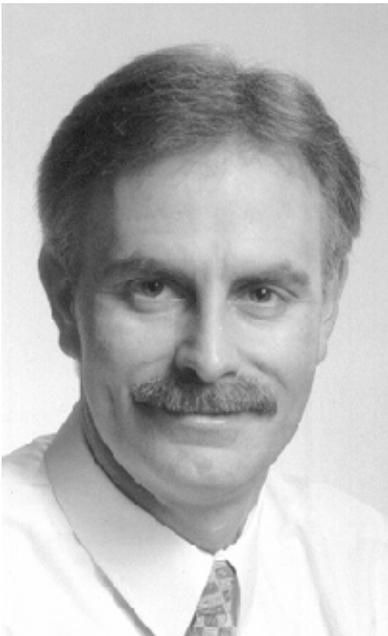
[Team LiB]

About the Technical Reviewers

These reviewers contributed their considerable hands-on expertise to the entire development process for Real World XML. As the book was being written, these dedicated professionals reviewed all the material for technical content, organization, and flow. Their feedback was critical to ensuring that Real World XML fits our reader's need for the highest-quality technical information.



Steve Heckler is President of Accelebrate, an IT training and technical writing firm in Atlanta. An avid ASP.NET, Java, and XML developer and trainer, Steve served more than six years as a senior manager and trainer at a leading east-coast IT training firm prior to founding Accelebrate. He holds a Bachelor and Masters degree from Stanford University.



Carl Burnham is a web developer, technology specialist, and author, with 17 years of experience supporting IT operations within the private/public sectors. He founded Southpoint.com, a popular travel destination guide, in 1996. Carl is author of the book, *Web Hosting, a Complete Strategy*, and has coauthored several technology books.

[[Team LiB](#)]

[[Team LiB](#)]

Acknowledgments

A book like the one you're holding is the work of a great many people, not just the author. The people at New Riders have been great, and I'd like to thank Stephanie Wall, Associate Publisher extraordinaire; Lori Lyons and Krista Hansing, Project/Copy Editors, who kept things moving along; and finally, the Technical Reviewers, Steve Heckler and Carl Burnham, who did a great job of checking everything. Thanks, everyone,

for all your much-appreciated hard work.

[[Team LiB](#)]

[[Team LiB](#)]

Preface

Welcome to Real World XML. This book is designed to be as comprehensive and as accessible as possible for a single book on XML. XML is a standard, not an implementation, and it's become an umbrella for a great number of topics. You'll find XML just about everywhere you look on the Internet today, and even in many places behind the scenes (such as internally in Netscape Navigator 6). I believe this book provides more complete coverage of what's going on in XML than any other XML book today.

You'll find coverage of all the official XML standards here. I'll also take a look at many of the most popular and important implementations of XML that are out there, and put them to work in this book.

That's just part of the story we'll also put XML to work in depth, pushing the envelope as far as it can go. The best way to learn any topic like XML is by example, and this is an example-oriented book. You'll find hundreds of tested examples here, ready to be used.

Writing XML is not some ordinary and monotonous task: it inspires artistry, devotion, passion, exaltation, and eccentricity not to mention exasperation and frustration. I'll try to be true to that spirit and capture as much of the excitement and power of XML in this book as I can.

[[Team LiB](#)]

[[Team LiB](#)]

What Is Inside This Book

This book is designed to give you as much of the whole XML story as one book can hold. We'll not only see the full XML syntax from the most basic to the most advanced but also dig into many of the ways in which XML is used.

There are hundreds of real-world topics covered in this book, like connecting XML to databases both locally and on Web servers styling XML for viewing in today's browsers, reading and using XML documents in browsers, creating our own graphically based browsers, and a great deal more.

Here's a sample of some of the topics in this book note that each of these topics themselves have many subtopics (too many to list here):

- The complete XML syntax
- Well-formed XML documents
- Valid XML documents
- Document type definitions (DTDs)
- Namespaces
- The XML Document Object Model (DOM)
- Canonical XML
- XML schemas
- Parsing XML with JavaScript
- XML and data binding
- XML and Cascading Style Sheets (CSS)
- XML and Java
- XML and .NET

- DOM parsers
- SAX parsers
- Extensible Stylesheet Language (XSL) transformations
- XSL formatting objects
- XLinks
- XPointers
- XPath
- XBase
- XHTML 1.0 and 1.1
- Resource Description Framework (RDF)
- Simple Object Access Protocol (SOAP)
- Vector Markup Language (VML)
- Wireless Markup Language (WML)
- Server-side XML with JavaServer Pages (JSP), Active Server Pages (ASP), Java servlets, and Perl

This book starts with the basics. I do assume that you have some knowledge of HTML, but not necessarily much. We'll see how to create XML documents from scratch in this book, starting at the very beginning.

From there, we'll move up to see how to check the syntax of XML documents. The big attraction of XML is that you can define your own tags, like the `<DOCUMENT>` and `<GREETING>` tags in this document, which we'll see early in [Chapter 1](#):

```
<?xml version="1.0" encoding="UTF-8"?>
<DOCUMENT>
  <GREETING>
    Hello From XML
  </GREETING>
  <MESSAGE>
    Welcome to the wild and woolly world of XML.
  </MESSAGE>
</DOCUMENT>
```

Because you can create your own tags in XML, it's also important to specify the syntax you want those tags to obey (for example, can a `<MESSAGE>` appear inside a `<GREETING>`?). XML puts a big emphasis on this, too, and there are two main ways to specify the syntax you want your XML to follow with XML document type definitions (DTDs) and XML schemas. We'll see how to create both.

And because you can make up your own tags in XML, it's also up to you to specify how they should be used. Netscape Navigator won't know, for example, that a `<KILLER>` tag marks a favorite book in your collection. Because it's up to you to determine what a tag actually means, handling your XML documents in programming is an important part of learning XML, despite what some second-rate XML books try to claim. The two languages I'll use in this book are JavaScript and Java; before using them, I'll introduce them in special sections with plenty of examples, so even if you're not familiar with these languages, you won't have to go anywhere else to get the skills you need.

The major browsers today are becoming more and more XML-aware, and they use scripting languages to let you work with your XML documents. We'll be using the most popular and powerful of those scripting languages here, JavaScript. Using JavaScript, we'll be able to read XML documents directly in browsers like the Microsoft Internet Explorer.

It's also important to know how to handle XML outside browsers, because there are plenty of things that JavaScript can't handle. These days, most XML development is taking place in Java, and there is an endless arsenal of Java resources available for free on the Internet. In fact, the connection between Java and XML is a natural one, as we'll see. We'll use Java to read XML documents and interpret them, starting in [Chapter 11](#). That doesn't mean you have to be a Java expert far from it, in fact because I'll introduce all the Java we'll need right here in this book. And because most XML development is done in Java today, we're going to find a

wealth of tools here, ready for use.

You can also design your XML documents to be displayed directly in some modern browsers, and I'll take a look at doing that in two ways with Cascading Style Sheets (CSS) and the Extensible Stylesheet Language (XSL). Using CSS and XSL, you can indicate exactly how a tag that you make up, like `<PANIC>` or `<BIG_AND_BOLD>` or `<AZURE_UNDERLINED_TEXT>`, should be displayed. I'll take a look at both parts of XSL XSL transformations and formatting objects in depth.

In addition, we'll see various other XML specifications in this book as well, such as XLinks, XBase, and XPointers, which let you point to particular items in XML documents in very specific ways. The XML specifications are made by a body called the World Wide Web Consortium, abbreviated W3C, and we'll become very familiar with those specifications here, seeing what they say and seeing what they lack.

I'll wind up the book by taking a look at a number of the most popular uses of XML on the Internet in several chapters. XML is really a language for defining languages, and there are hundreds of such XML-based languages out there now. Some of them are gaining great popularity, and I'll cover them in some depth in this book.

There is an astonishing wealth of material on XML available on the Internet today, so I'm also going to fill this book with the URIs of dozens of those resources (in XML, you use Uniform Resource Identifiers, not URLs, although in practice they are the same thing for most purposes today). In nearly every chapter, you'll find lists of free online programs and other resources. (However, there's a hazard here that I should mention URIs change frequently on the Internet, so don't be surprised if some of these URIs have changed by the time you look for them.)

[[Team LiB](#)]

[[Team LiB](#)]

Who Is This Book For?

This book is designed for just about anyone who wants to learn XML and how it's used today in the real world. The only assumption that I make is that you have some knowledge of how to create documents using Hypertext Markup Language (HTML). You don't have to be any great HTML expert, but a little knowledge of HTML will be helpful. That's really all you need.

However, it's a fact of life that most XML software these days is targeted at Windows. Among other things, this means you should have access to Windows for many of the topics covered in this book, and in [Chapters 7](#) and [8](#), we'll be taking a look at the XML support in Microsoft Internet Explorer. I wish there was more support for the other operating systems I like, such as Unix, but right now a lot of it is Windows-only. I'll explore alternatives when I can. One hopeful note for the future is that there are more and more Java-based XML tools appearing daily, and those tools are platform-independent.

[[Team LiB](#)]

[[Team LiB](#)]

At What Level This Book Is Written

This book is written at several different levels from basic to advanced because the XML spectrum is so broad. The rule of thumb is that this book was written to follow HTML books in level. We start at the basic level and gradually get more advanced in a slow, steady way.

I'm not going to assume that you have any programming knowledge (at least until we get to the advanced topics in [Chapter 20](#), such as JavaServer Pages and using Perl with XML) when you start this book. We'll be using both JavaScript and Java in this book, but all you need to know about those languages will be introduced before we use them, and it won't be hard to pick up.

Because there are so many uses of XML available today, this book involves many different software packages; all the ones I'll put to work in the text are free to download from the Internet; I'll tell you where to get them.

[[Team LiB](#)]

[[Team LiB](#)]

Conventions Used

There are several conventions that I use in this book that you should be aware of. The most important one is that when I add new sections of code, I'll mark them with gray highlighting to point out the actual lines I'm discussing so that they stand out. (This sample is written in one of the languages built on XML, the Wireless Markup Language (WML), which is targeted at "micro-browsers" in cellular phones and personal digital assistants, or PDAs.)

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.
xml">
<wml>
  <card id="Card1" title="First WML Example">
    <!-- This is a comment -->
    <p>
      Greetings from WML.
    </p>
  </card>
</wml>
```

Also, where there's something worth noting or some additional information that adds something to the discussion, I'll add a sidebar:

More on SOAP

With a common name like SOAP, it's hard to search the Internet for more information about the Simple Object Access Protocol unless you're really into pages on personal cleanliness and daytime television. For more information, you might check out this starter list:

<http://msdn.microsoft.com/xml/general/soapspec.asp>, www.oasis-open.org/cover/soap.html, www.develop.com/soap/, and www.develop.com/soap/soapfaq.xml.

Well, we're ready to go. If you have comments, I encourage you to write to me, care of New Riders. This book is designed to be the new standard in XML coverage, truly more complete and more accessible than ever before. Please do keep in touch with me with ways to improve it and keep it on the forefront. If you think the book lacks anything, let me know I'll add it, because I want to make sure this book stays on top.

[[Team LiB](#)]

[[Team LiB](#)]

Chapter 1. Essential XML

Welcome to the world of Extensible Markup Language, XML. This book is your guided tour to that world, so have no worries you've come to the right place. That world is large and expanding in unpredictable ways every minute, but we're going to become familiar with the lay of the land in detail here. And there's a lot of territory to cover because XML is getting into the most amazing places, and in the most amazing ways, these days.

XML is a language defined by the World Wide Web Consortium (W3C, www.w3c.org), the body that sets the standards for the Web, and this first chapter is all about getting a solid overview of that language and how you can use it. For example, you probably already know that you can use XML to create your own elements, thus creating a customized markup language for your own use. In this way, XML supercedes other markup languages such as Hypertext Markup Language (HTML); in HTML, all the elements you use are predefined and there are not enough of them. In fact, XML is a met markup language because it lets you create your own markup languages.

[Team LiB]

[Team LiB]

Markup Languages

Markup languages are all about describing the form of the document that is, the way the content of the document should be interpreted. The markup language that most people are familiar with today is, of course, HTML, which you use to create standard Web pages. Here's an example HTML page:

Listing ch01_01.html

```
<HTML>
  <HEAD>
    <TITLE>Hello From HTML</TITLE>
  </HEAD>
  <BODY>
    <CENTER>
      <H1>
        Hello From HTML
      </H1>
    </CENTER>
    Welcome to the wild and woolly world of HTML.
  </BODY>
</HTML>
```

You can see the results of this HTML in [Figure 1-1](#) in Netscape Navigator. Note that the HTML markup in this page that is, tags such as <HEAD>, <CENTER>, <H1>, and so on is there to give directions to the browser. That's what markup does; it specifies directions on the way the content is to be interpreted.

Figure 1-1. An HTML page in a browser.



When you think of markup in terms of specifying how the content of a document is to be handled, it's easy to see that there are many kinds of markup languages all around already. For example, if you use a word processor to save a document in Rich Text Format (RTF), you'll find all kinds of markup codes embedded in the document. Here's an example; in this case, I've just created an RTF file with the letters `abc` underlined and in bold using Microsoft Word try searching for the actual text (hint: it's near the very end):

```
{\rtf1\ansi\ansicpg1252\uc1 \deff0\deflang1033
\deflangfe1033{\fonttbl{\f0\froman\charset0\prq2{\*\panose
02020603050405020304}Times New Roman;}}{\colortbl;\red0
\green0\blue0;\red0\green0\blue255;\red0\green255\blue255;
\red0\green255\blue0;\red255\green0\blue255;\red255\green0
\blue0;\red255\green255\blue0;\red255\green255\blue255;\red0
\green0\blue128;\red0\green128\blue128;\red0\green128\blue0;
\red128\green0\blue128;\red128\green0\blue0;\red128\green128
\blue0;\red128\green128\blue128;\red192\green192\blue192;}
{\stylesheet{\widctlpar\adjustright \fs20\cgrid \snext0 Normal;}
{\*\cs10 \additive Default Paragraph Font;}}{\info{\title }
{\author Steven Holzner}{\operator Steven Holzner}{\creati
m\yr2000\mo\dy\hr\min}{\revtim\yr2000\mo4\dy17\hr13\min55}
{\version1}{\edmins1}{\nofpages1}{\nofwords0}{\nofchars1}
{\*\company SteveCo}{\nofcharsws1}{\vern89}}\widowctrl\ftnbj
\ænddoc\formshade\viewkind4\viewscale100\pgbrdrhead\pgbrdrfoot
\fet0\sectd \pszl\linex0\endnhere\sectdefaultcl {\*\pnseclvl1
\pnucrm\pnstart1\pnindent720\pnhang{\pntxta .}}{\*\pnseclvl2
\pnucltr\pnstart1\pnindent720\pnhang{\pntxta .}}{\*\pnseclvl3
\pndec\pnstart1\pnindent720\pnhang{\pntxta .}}{\*\pnseclvl4
\pnlcltr\pnstart1\pnindent720\pnhang{\pntxta )}}{\*\pnseclvl5
\pndec\pnstart1\pnindent720\pnhang{\pntxtb (}{\pntxta )}}
{\*\pnseclvl6\pnlcltr\pnstart1\pnindent720\pnhang{\pntxtb (}
{\pntxta )}}{\*\pnseclvl7\pnlcrm\pnstart1\pnindent720\pnhang
{\pntxtb (}{\pntxta )}}{\*\pnseclvl8\pnlcltr\pnstart1
\pnindent720\pnhang{\pntxtb (}{\pntxta )}}{\*\pnseclvl9\pnlcrm
\pnstart1\pnindent720\pnhang{\pntxtb (}{\pntxta )}}\pard\plain
\s1480\slmult1\widctlpar\adjustright \fs20\cgrid {\b\fs24\u1 abc }{\b\u1 \par }}
```

The markup language that most people are familiar with these days is HTML, but it's easy to see how that language doesn't provide enough power for anything beyond creating standard Web pages.

HTML 1.0 consisted of only a dozen or so tags, but the most recent version, HTML 4.01, consists of almost 100 and if you include the other tags added by the major browsers, that number is closer to 120. But as handling data on the Web and other nets intensifies, it's clear that 120 tags isn't enough and, in fact, you can