



### **Core Web Programming, Second Edition**

By [Marty Hall](#), [Larry Brown](#)

Publisher : Prentice Hall PTR  
 Pub Date : May 01, 2001  
 ISBN : 0-13-089793-0  
 Pages : 1440

777

One-stop shopping for serious Web developers!

The authoritative guide to every technology that enterprise Web developers need to master, from HTML 4 to Java 2 Standard Edition 1.3, servlets to JavaServer Pages, and beyond. *Core Web Programming, Second Edition* brings them all together in the ultimate Web development resource for experienced programmers.

**HTML 4:** In-depth, practical coverage of HTML document structure, block-level and text-level elements, frames, cascading style sheets, and beyond.

**Java 2:** Basic syntax, object-oriented design, applets and animation, the Java Plug-In, user interface development with Swing, layout managers, Java2D, multithreading, network programming, database connectivity, and more.

**Server-Side Java:** Servlets, JSP, XML, and JDBC—the foundations of enterprise development with Java. Advanced topics include JSP custom tag libraries, combining servlets and JSP (MVC), database connection pooling, SAX, DOM, and XSLT processing, and detailed coverage of HTTP 1.1.

**JavaScript:** Dynamic creation of Web page content, user event monitoring, HTML form field validation, and more. Includes a complete quick reference guide.

This book's first edition is used in leading computer science programs worldwide, from MIT to Stanford, UC Berkeley to Princeton, UCLA to Johns Hopkins. Now, it's been 100% updated for today's hottest Web development technologies—with powerful new techniques, each with complete working code examples!

#### Copyright

#### Introduction

[Real Code for Real Programmers](#)

[How This Book Is Organized](#)

[Conventions](#)

[About the Web Site](#)

[About the Authors](#)

#### Acknowledgments

#### Part 1: The HyperText Markup Language

##### Chapter 1. Designing Web Pages with HTML 4.0

- [1.1 The HyperText Markup Language](#)
- [1.2 HTML 4.0 and Other HTML Standards](#)
- [1.3 Steps to Publish a Document on the Web](#)

- 1.4 The Basic Structure of HTML Documents
- 1.5 HEAD—High-Level Information About the Page
- 1.6 BODY—Creating the Main Document
- 1.7 Summary

#### Chapter 2. Block-Level Elements in HTML 4.0

- 2.1 Headings
- 2.2 Basic Text Elements
- 2.3 Numbered, Bulleted, and Indented Lists
- 2.4 Tables
- 2.5 Fill-Out Forms
- 2.6 Miscellaneous Block-Level Elements
- 2.7 Summary

#### Chapter 3. Text-Level Elements in HTML 4.0

- 3.1 Physical Character Styles
- 3.2 Logical Character Styles
- 3.3 Specifying Hypertext Links
- 3.4 Embedded Images
- 3.5 Client-Side Image Maps
- 3.6 Embedding Other Objects in Documents
- 3.7 Controlling Line Breaks
- 3.8 Summary

#### Chapter 4. Frames

- 4.1 Frame Document Template
- 4.2 Specifying Frame Layout
- 4.3 Specifying the Content of Frame Cells
- 4.4 Targeting Frame Cells
- 4.5 Solving Common Frame Problems
- 4.6 Inline Frames
- 4.7 Summary

#### Chapter 5. Cascading Style Sheets

- 5.1 Specifying Style Rules
- 5.2 Using External and Local Style Sheets
- 5.3 Selectors
- 5.4 Cascading: Style Sheet Precedence Rules
- 5.5 Font Properties
- 5.6 Foreground and Background Properties
- 5.7 Text Properties
- 5.8 Properties of the Bounding Box
- 5.9 Images and Floating Elements
- 5.10 List Properties
- 5.11 Standard Property Units
- 5.12 Layers
- 5.13 Summary

### Part 2: Java Programming

#### Chapter 6. Getting Started with Java

- 6.1 Unique Features of Java
- 6.2 Myths About Java
- 6.3 Java Versions
- 6.4 Getting Started: Nuts and Bolts
- 6.5 Some Simple Java Programs
- 6.6 Summary

#### Chapter 7. Object-Oriented Programming in Java

- 7.1 Instance Variables

- 7.2 Methods
- 7.3 Constructors and the "this" Reference
- 7.4 Destructors
- 7.5 Overloading
- 7.6 Public Version in Separate File
- 7.7 Javadoc
- 7.8 Inheritance
- 7.9 Interfaces and Abstract Classes
- 7.10 Packages, Classpath, and JAR Archives
- 7.11 Modifiers in Declarations
- 7.12 Summary

#### Chapter 8. Basic Java Syntax

- 8.1 Rules of Syntax
- 8.2 Primitive Types
- 8.3 Operators, Conditionals, Iteration
- 8.4 The Math Class
- 8.5 Input and Output
- 8.6 Execution of Non-Java Programs
- 8.7 Reference Types
- 8.8 Strings
- 8.9 Arrays
- 8.10 Vectors
- 8.11 Example: A Simple Binary Tree
- 8.12 Exceptions
- 8.13 Summary

#### Chapter 9. Applets and Basic Graphics

- 9.1 What Are Applets?
- 9.2 Creating an Applet
- 9.3 An Example Applet
- 9.4 The Applet Life Cycle
- 9.5 Other Applet Methods
- 9.6 The HTML APPLET Element
- 9.7 Reading Applet Parameters
- 9.8 HTML OBJECT Element
- 9.9 The Java Plug-In
- 9.10 Graphical Applications
- 9.11 Graphics Operations
- 9.12 Drawing Images
- 9.13 Preloading Images
- 9.14 Controlling Image Loading: Waiting for Images and Checking Status
- 9.15 Summary

#### Chapter 10. Java 2D: Graphics in Java 2

- 10.1 Getting Started with Java 2D
- 10.2 Drawing Shapes
- 10.3 Paint Styles
- 10.4 Transparent Drawing
- 10.5 Using Local Fonts
- 10.6 Stroke Styles
- 10.7 Coordinate Transformations
- 10.8 Other Capabilities of Java 2D
- 10.9 Summary

#### Chapter 11. Handling Mouse and Keyboard Events

- 11.1 Handling Events with a Separate Listener
- 11.2 Handling Events by Implementing a Listener Interface
- 11.3 Handling Events with Named Inner Classes

- 11.4 Handling Events with Anonymous Inner Classes
- 11.5 The Standard Event Listeners
- 11.6 Behind the Scenes: Low-Level Event Processing
- 11.7 A Spelling-Correcting Textfield
- 11.8 A Whiteboard Class
- 11.9 Summary

#### Chapter 12. Layout Managers

- 12.1 The FlowLayout Manager
- 12.2 The BorderLayout Manager
- 12.3 The GridLayout Manager
- 12.4 The CardLayout Manager
- 12.5 GridBagLayout
- 12.6 The BoxLayout Manager
- 12.7 Turning Off the Layout Manager
- 12.8 Effective Use of Layout Managers
- 12.9 Summary

#### Chapter 13. AWT Components

- 13.1 The Canvas Class
- 13.2 The Component Class
- 13.3 Lightweight Components in Java 1.1
- 13.4 The Panel Class
- 13.5 The Container Class
- 13.6 The Applet Class
- 13.7 The ScrollPane Class
- 13.8 The Frame Class
- 13.9 Serializing Windows
- 13.10 The Dialog Class
- 13.11 The FileDialog Class
- 13.12 The Window Class
- 13.13 Handling Events in GUI Controls
- 13.14 The Button Class
- 13.15 The Checkbox Class
- 13.16 Check Box Groups (Radio Buttons)
- 13.17 Choice Menus
- 13.18 List Boxes
- 13.19 The TextField Class
- 13.20 The TextArea Class
- 13.21 The Label Class
- 13.22 Scrollbars and Sliders
- 13.23 Pop-up Menus
- 13.24 Summary

#### Chapter 14. Basic Swing

- 14.1 Getting Started with Swing
- 14.2 The JApplet Component
- 14.3 The JFrame Component
- 14.4 The JLabel Component
- 14.5 The JButton Component
- 14.6 The JPanel Component
- 14.7 The JSlider Component
- 14.8 The JColorChooser Component
- 14.9 Internal Frames
- 14.10 The JOptionPane Component
- 14.11 The JToolBar Component
- 14.12 The JEditorPane Component
- 14.13 Other Simple Swing Components
- 14.14 Summary

### Chapter 15. Advanced Swing

- [15.1 Using Custom Data Models and Renderers](#)
- [15.2 JList](#)
- [15.3 JTree](#)
- [15.4 JTable](#)
- [15.5 Swing Component Printing](#)
- [15.6 Swing Threads](#)
- [15.7 Summary](#)

### Chapter 16. Concurrent Programming with Java Threads

- [16.1 Starting Threads](#)
- [16.2 Race Conditions](#)
- [16.3 Synchronization](#)
- [16.4 Creating a Multithreaded Method](#)
- [16.5 Thread Methods](#)
- [16.6 Thread Groups](#)
- [16.7 Multithreaded Graphics and Double Buffering](#)
- [16.8 Animating Images](#)
- [16.9 Timers](#)
- [16.10 Summary](#)

### Chapter 17. Network Programming

- [17.1 Implementing a Client](#)
- [17.2 Parsing Strings by Using StringTokenizer](#)
- [17.3 Example: A Client to Verify E-Mail Addresses](#)
- [17.4 Example: A Network Client That Retrieves URLs](#)
- [17.5 The URL Class](#)
- [17.6 WebClient: Talking to Web Servers Interactively](#)
- [17.7 Implementing a Server](#)
- [17.8 Example: A Simple HTTP Server](#)
- [17.9 RMI: Remote Method Invocation](#)
- [17.10 Summary](#)

## Part 3: Server-Side Programming

### Chapter 18. HTML Forms

- [18.1 How HTML Forms Transmit Data](#)
- [18.2 The FORM Element](#)
- [18.3 Text Controls](#)
- [18.4 Push Buttons](#)
- [18.5 Check Boxes and Radio Buttons](#)
- [18.6 Combo Boxes and List Boxes](#)
- [18.7 File Upload Controls](#)
- [18.8 Server-Side Image Maps](#)
- [18.9 Hidden Fields](#)
- [18.10 Grouping Controls](#)
- [18.11 Tab Order Control](#)
- [18.12 Summary](#)

### Chapter 19. Server-Side Java: Servlets

- [19.1 The Advantages of Servlets Over "Traditional" CGI](#)
- [19.2 Server Installation and Setup](#)
- [19.3 Basic Servlet Structure](#)
- [19.4 The Servlet Life Cycle](#)
- [19.5 An Example Using Initialization Parameters](#)
- [19.6 The Client Request: Form Data](#)
- [19.7 The Client Request: HTTP Request Headers](#)
- [19.8 The Servlet Equivalent of the Standard CGI Variables](#)
- [19.9 The Server Response: HTTP Status Codes](#)

- [19.10 The Server Response: HTTP Response Headers](#)
- [19.11 Cookies](#)
- [19.12 Session Tracking](#)
- [19.13 Summary](#)

#### Chapter 20. JavaServer Pages

- [20.1 JSP Overview](#)
- [20.2 Advantages of JSP](#)
- [20.3 JSP Scripting Elements](#)
- [20.4 The JSP page Directive](#)
- [20.5 Including Files and Applets in JSP Documents](#)
- [20.6 Using JavaBeans with JSP](#)
- [20.7 Defining Custom JSP Tags](#)
- [20.8 Integrating Servlets and JSP](#)
- [20.9 Summary](#)

#### Chapter 21. Using Applets As Front Ends to Server-Side Programs

- [21.1 Sending Data with GET and Displaying the Resultant Page](#)
- [21.2 A Multisystem Search Engine Front End](#)
- [21.3 Using GET and Processing the Results Directly \(HTTP Tunneling\)](#)
- [21.4 A Query Viewer That Uses Object Serialization and HTTP Tunneling](#)
- [21.5 Using POST and Processing the Results Directly \(HTTP Tunneling\)](#)
- [21.6 An Applet That Sends POST Data](#)
- [21.7 Bypassing the HTTP Server](#)
- [21.8 Summary](#)

#### Chapter 22. JDBC

- [22.1 Basic Steps in Using JDBC](#)
- [22.2 Basic JDBC Example](#)
- [22.3 Some JDBC Utilities](#)
- [22.4 Applying the Database Utilities](#)
- [22.5 An Interactive Query Viewer](#)
- [22.6 Prepared Statements \(Precompiled Queries\)](#)
- [22.7 Summary](#)

#### Chapter 23. XML Processing with Java

- [23.1 Parsing XML Documents with DOM Level 2](#)
- [23.2 DOM Example: Representing an XML Document as a JTree](#)
- [23.3 Parsing XML Documents with SAX 2.0](#)
- [23.4 SAX Example 1: Printing the Outline of an XML Document](#)
- [23.5 SAX Example 2: Counting Book Orders](#)
- [23.6 Transforming XML with XSLT](#)
- [23.7 XSLT Example 1: XSLT Document Editor](#)
- [23.8 XSLT Example 2: Custom JSP Tag](#)
- [23.9 Summary](#)

### Part 4: JavaScript

#### Chapter 24. JavaScript: Adding Dynamic Content to Web Pages

- [24.1 Generating HTML Dynamically](#)
- [24.2 Monitoring User Events](#)
- [24.3 Mastering JavaScript Syntax](#)
- [24.4 Using JavaScript to Customize Web Pages](#)
- [24.5 Using JavaScript to Make Pages Dynamic](#)
- [24.6 Using JavaScript to Validate HTML Forms](#)
- [24.7 Using JavaScript to Store and Examine Cookies](#)
- [24.8 Using JavaScript to Interact with Frames](#)
- [24.9 Accessing Java from JavaScript](#)
- [24.10 Accessing JavaScript from Java](#)

- [24.11 Summary](#)

#### Chapter 25. JavaScript Quick Reference

- [25.1 The Array Object](#)
- [25.2 The Button Object](#)
- [25.3 The Checkbox Object](#)
- [25.4 The Date Object](#)
- [25.5 The Document Object](#)
- [25.6 The Element Object](#)
- [25.7 The FileUpload Object](#)
- [25.8 The Form Object](#)
- [25.9 The Function Object](#)
- [25.10 The Hidden Object](#)
- [25.11 The History Object](#)
- [25.12 The Image Object](#)
- [25.13 The JavaObject Object](#)
- [25.14 The JavaPackage Object](#)
- [25.15 The Layer Object](#)
- [25.16 The Link Object](#)
- [25.17 The Location Object](#)
- [25.18 The Math Object](#)
- [25.19 The MimeType Object](#)
- [25.20 The Navigator Object](#)
- [25.21 The Number Object](#)
- [25.22 The Object Object](#)
- [25.24 The Password Object](#)
- [25.25 The Plugin Object](#)
- [25.26 The Radio Object](#)
- [25.27 The RegExp Object](#)
- [25.28 The Reset Object](#)
- [25.29 The Screen Object](#)
- [25.30 The Select Object](#)
- [25.31 The String Object](#)
- [25.32 The Submit Object](#)
- [25.33 The Text Object](#)
- [25.34 The Textarea Object](#)
- [25.35 The Window Object](#)
- [25.36 Summary](#)

## Copyright

### Library of Congress Cataloging-in-Publication Data

Hall, Marty.

Core web programming / Marty Hall, Larry Brown.

p. cm.

Only Marty Hall's name appears on previous edition.

Includes index.

ISBN 0-13-089793-0

1. Internet programming. 2. HTML (Document markup language) 3. Java (Computer

program language) 4. CGI (Computer network protocol) 5. World Wide Web. I. Hall, Marty.  
II. Title.

QA76.625 .B757 2001

005.2'76--dc21

2001021692

© 2001 Sun Microsystems, Inc.—

Printed in the United States of America.

901 San Antonio Road, Palo Alto, California

94303-4900 U.S.A.

All rights reserved. This product and related documentation are protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

**RESTRICTED RIGHTS LEGEND:** Use, duplication, or disclosure by the United States Government is subject to the restrictions set forth in DFARS 252.227-7013 (c)(1)(ii) and FAR 52.227-19.

The products described may be protected by one or more U.S. patents, foreign patents, or pending applications.

**TRADEMARKS**—HotJava, Java, Java Development Kit, Solaris, SPARC, SunOS, and Sunsoft are trademarks of Sun Microsystems, Inc. All other products or services mentioned in this book are the trademarks or service marks of their respective companies or organizations

The publisher offers discounts on this book when ordered in bulk quantities. For more information, contact Corporate Sales Department, Prentice Hall PTR , One Lake Street, Upper Saddle River, NJ 07458. Phone: 800-382-3419; FAX: 201- 236-7141. E-mail: [corpsales@prenhall.com](mailto:corpsales@prenhall.com).

10 9 8 7 6 5 4 3 2 1

## **Sun Microsystems Press**

### **A Prentice Hall Title**

Production Editor and Compositor: *Vanessa Moore*

Project Coordinator: *Anne Trowbridge*

Acquisitions Editor: *Gregory G. Doench*

Editorial Assistant: *Brandt Kenna*

Cover Design Director: *Jerry Votta*

Cover Designer: *Nina Scuderi*

Cover Illustration: *Karen Strelecki*

Art Director: *Gail Cocker-Bogusz*

Manufacturing Manager: *Alexis R. Heydt*

Marketing Manager: *Debby vanDijk*

Sun Microsystems Press:

Marketing manager: *Michael Llwyd Alread*

Publisher: *Rachel Borden*

## Introduction

In late 1995, Marty Hall proposed a new course for the part-time graduate program in Computer Science at the Johns Hopkins University. The idea was to bring together the major Web-related topics in a single course dubbed "Distributed Development on the World Wide Web," with Java technology as a unifying theme. Students would look at HTML, Java, HTTP, CGI programming, and JavaScript, with lots of hands-on projects and no exams. Little did Marty know what he was getting himself into. By the time the first section was offered in the summer of 1996, the Java tidal wave had swept through the university and the companies that the students represented. Shortly after enrollment opened, the class was filled. There were more students on the waiting list than in the course. Marty got frantic phone calls from students insisting that they absolutely *had* to be in the course. Several local companies called, asking for on-site courses. What fun!

However, when Marty went shopping for texts over the next semester or two, he got a rude surprise. Despite the availability of good books in most of the individual areas he wanted to cover, Marty found that he needed three, four, or even five separate books to get good coverage of the overall material. Similarly, for his day job, Marty was constantly switching back and forth among the best of the huge stack of books he had accumulated and various on-line references. Surely there was a better way. Shouldn't it be possible to fit 85 percent of what professional programmers use in about 35 percent of the space, and get it all in one book?

That was the genesis of the first edition of *Core Web Programming*. The book was very popular, but the industry has been rapidly moving since the book's release. Browsers moved from HTML 3.2 to 4.0. The Java 2 platform was released, providing greatly improved performance and graphics libraries suitable for commercial-quality applications. JSP 1.0 came along, resulting in an explosion of interest in both servlets and JSP as an alternative to CGI and to proprietary solutions like ASP and ColdFusion. XML burst upon the scene. The server equalled or even surpassed the desktop as the biggest application area for the Java programming language.

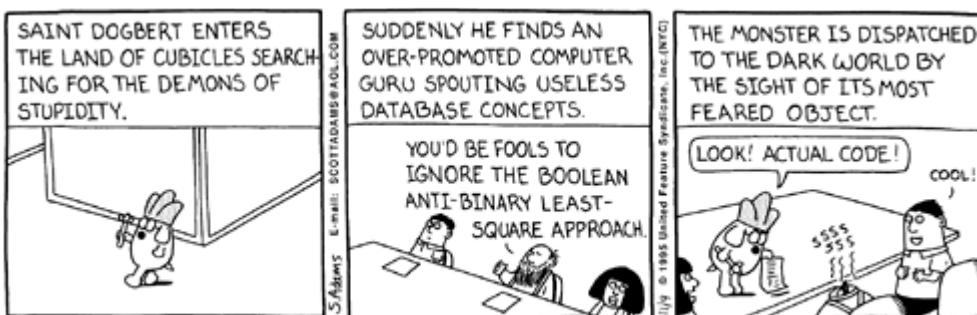
Wow. And demand has only been growing since then. Although readers were clamoring for a new edition of the book, it was just too much for Marty to handle alone. Enter Larry Brown, with broad development and teaching experience in Java and Web technologies,

and with particular expertise in the Java Foundation Classes, multithreaded programming, RMI, and XML processing with Java. Larry teamed up with Marty to totally update the existing material to HTML 4, CSS/1, HTTP 1.1, and the Java 2 platform; to replace the CGI sections with chapters on servlets 2.2 and JSP 1.1; and to add completely new sections on Swing, Java 2D, and XML processing with JAXP, DOM Level 2, SAX 2.0, and XSLT. They even got a little bit of sleep along the way.

We—Marty and Larry—hope you find the result enjoyable and useful!

## Real Code for Real Programmers

This book is aimed at serious software developers. If you are looking for a book that shows you how to use a browser, lists the current hottest Web sites, and pontificates about how Web-enabled applications will revolutionize your business, you've come to the wrong place. If you're already a programmer of some sort and want to get started with HTML, XML, Java applets, desktop applications in Java, servlets, JavaServer Pages, and JavaScript as quickly as possible, this is the book for you. We illustrate the most important approaches and warn you of the most common pitfalls. To do so, we include plenty of working code: over 250 complete Java classes, for instance. We try to give detailed examples of the most important and frequently used features, summarize the lesser-used ones, and refer you to the API (available on-line) for a few of the rarely used ones.



DILBERT © UFS. Reprinted by permission.

A word of caution, however. Nobody becomes a great developer just by reading. You have to write some real code too. The more, the better. In each chapter, we suggest that you start by making a simple program or a small variation of one of the examples given, then strike off on your own with a more significant project. Skim the sections you don't plan on using right away, then come back when you are ready to try them out.

If you do this, you should quickly develop the confidence to handle the real-world problems that brought you here in the first place. You should be able to balance the demand for the latest features in Web pages with the need for multiplatform support. You should be comfortable with frames, style sheets, and layered HTML. You should be able to make portable stand-alone graphical applications. You should have no qualms about developing Web interfaces to your corporate database through JDBC. You should be able to connect these applications to remote systems over the network. You should understand how to easily distribute computation among multiple threads, or even spin it off to separate systems by using RMI. You should be able to decide where servlets apply well, where JSP is better, and where a combination is best. You should understand HTTP 1.1 well enough to use its capabilities to enhance the effectiveness of your pages. You should be able to spin off complex server-side behaviors into JavaBeans components or custom JSP tag libraries. You should be able to use JavaScript to validate HTML forms or to animate Web

pages. You should get a raise. A big one, preferably.

## How This Book Is Organized

This book is divided into four parts: HTML, Java programming, server-side programming, and JavaScript.

### Part 1: The HyperText Markup Language

Web pages are created with HTML, the HyperText Markup Language. HTML lets you mix regular text with special tags that describe the content, layout, or appearance of the text. These tags are then used by Web browsers like Netscape Navigator or Microsoft Internet Explorer to format the page. This first part of the book covers the following topics in HTML.

- **HTML 4.01.** Full coverage of all the elements in the latest official HTML standard. Hypertext links, fonts, images, tables, client-side image maps, and more.
- **Major Netscape and Internet Explorer extensions.** Forwarding pages, using custom colors and font faces, embedding audio, video, and ActiveX components.
- **Frames.** Dividing the screen into rectangular regions, each associated with a separate HTML document. Borderless frames. Floating frames. Targeting frame cells from hypertext links.
- **Cascading style sheets.** Level-one style sheets for customizing fonts, colors, images, text formatting, indentation, lists, and more.

### Part 2: Java Programming

Java is a powerful general-purpose programming language that can be used to create stand-alone programs as well as ones that are embedded in Web pages. The following Java topics are covered.

- **Unique features of Java.** What's different about Java? The truth about Java myths and hype.
- **Object-oriented programming in Java.** Variables, methods, constructors, overloading, and interfaces. Modifiers in class declarations. Packages, the CLASSPATH, and JAR files.
- **Java syntax.** Primitive types, operators, strings, vectors, arrays, input/output and the Math class.
- **Graphics.** Applets. Applications. Drawing, color, font, and clipping area operations. Loading and drawing images. Java Plug-In.
- **Java 2D.** Creating professional, high-quality 2D graphics. Creating custom shapes, tiling images, using local fonts, creating transparent shapes, and transforming coordinates.
- **Mouse and keyboard events.** Processing events. Event types, event listeners, and low-level event handlers. Inner classes. Anonymous classes.

- **Layout managers.** FlowLayout, BorderLayout, GridLayout, CardLayout, GridBagLayout, and BoxLayout. Positioning components by hand. Strategies for using layout managers effectively.
- **AWT components.** Canvas, Panel, Applet, ScrollPane, Frame, Dialog, FileDialog, and Window. Component and Container. Buttons, check boxes, radio buttons, combo boxes, list boxes, textfields, text areas, labels, scrollbars, and pop-up menus. Saving and loading windows with object serialization.
- **Basic Swing components.** Building Swing applets and applications. Changing the GUI look and feel. Adding custom borders to components. Using HTML in labels and buttons. Sending dialog alerts for user input. Adding child frames to applications. Building custom toolbars. Implementing a Web browser in Swing.
- **Advanced Swing.** JList, JTree, and JTable. Using custom data models and renderers. Printing Swing components. Updating Swing components in a thread-safe manner.
- **Multithreaded programming.** Threads in separate or existing objects. Synchronizing access to shared resources. Grouping threads. Multithreaded graphics and double buffering. Animating images. Controlling timers.
- **Network programming.** Clients and servers using sockets. The URL class. Implementing a generic network server. Creating a simple HTTP server. Invoking distributed objects with RMI.

## Part 3: Server-Side Programming

Programs that run on a Web server can generate dynamic content based on client data. Servlets are Java technology's answer to CGI programming and JSP is Java's answer to Active Server Pages or ColdFusion. The following server-side topics are discussed.

- **HTML forms.** Sending data from forms. Text controls. Push buttons. Check boxes and radio buttons. Combo boxes and list boxes. File upload controls. Server-side image maps. Hidden fields. Tab ordering.
- **Java servlets.** The advantages of servlets over competing technologies. Servlet life cycle. Servlet initialization parameters. Accessing form data. Using HTTP 1.1 request headers, response headers, and status codes. Using cookies in servlets. Session tracking.
- **JavaServer Pages (JSP).** The benefits of JSP. JSP expressions, scriptlets, and declarations. Using JavaBeans components with JSP. Creating custom JSP tag libraries. Combining servlets and JSP.
- **Using applets as servlet front ends.** Sending GET and POST data. HTTP tunneling. Using object serialization to exchange high-level data structures between applets and servlets. Bypassing the HTTP server altogether.
- **Java Database Connectivity (JDBC).** The seven basic steps in connecting to databases. Some utilities that simplify JDBC usage. Formatting a database result as plain text or HTML. An interactive graphical query viewer. Precompiled queries.

- **XML processing with Java.** Representing an entire XML document by using the Document Object Model (DOM) Level 2. Responding to individual XML parsing events with the Simple API for XML Parsing (SAX) 2.0. Transforming XML with XSLT. Hiding vendor-specific details with the Java API for XML Processing (JAXP).

## Part 4: JavaScript

JavaScript is a scripting language that can be embedded in Web pages and interpreted as the pages are loaded. The final part covers the following JavaScript topics.

- **JavaScript syntax.** Fields, methods, functions, strings, objects, arrays, and regular expressions.
- **Customizing Web pages.** Adapting to different browsers, JavaScript releases, and screen sizes.
- **Making pages dynamic.** Animating images. Manipulating layers. Responding to user events.
- **Validating HTML forms.** Checking form entries as they are changed. Checking data when form is submitted.
- **Handling cookies.** Reading and setting values. The `Cookie` object.
- **Controlling frames.** Sending results to specific frames. Preventing documents from being framed. Updating multiple frame cells. Giving frame cells the focus automatically.
- **Integrating Java and JavaScript.** `LiveConnect` and the `JSObject` class.
- **JavaScript quick reference.** Major classes in JavaScript 1.2. All fields, methods, and event handlers. `Document`, `Window`, `Form`, `Element`, `String`, `Math`, `RegExp`, and so forth.

## Conventions

Throughout the book, concrete programming constructs or program output is presented in a monospaced font. For example, when abstractly describing Java programs that can be embedded in Web pages, we refer to "applets," but when we refer to `Applet` we are talking about the specific Java class from which all applets are derived.

User input is indicated in boldface, and command-line prompts are either generic (`Prompt>`) or indicate the operating system to which they apply (`Unix>`). For instance, the following indicates that "Some Output" is the result when "`java SomeProgram`" is executed.

```
Prompt> java SomeProgram
Some Output
```

Important standard techniques are indicated by specially marked entries, as in the following example.

## Core Approach



*Pay particular attention to items in "Core Approach" sections. They indicate techniques that should always or almost always be used.*

Notes and warnings are called out in a similar manner.

## About the Web Site

The book has a companion Web site at

<http://www.corewebprogramming.com/>

This free site includes:

- Documented source code for all examples shown in the book; this code can be downloaded for unrestricted use.
- On-line versions of all HTML pages, Java applets, and JavaScript examples.
- Links to all URLs mentioned in the text of the book.
- Information on book discounts.
- Reports on Java short courses.
- Book additions, updates, and news.
- A free Ronco combination paring knife and e-commerce tool. OK, maybe not.

## About the Authors

Marty Hall is a Senior Computer Scientist in the Research and Technology Development Center at the Johns Hopkins University Applied Physics Lab, where he specializes in the application of Java and Web technology to customer problems. He also teaches Java and Web programming in the Johns Hopkins part-time graduate program in Computer Science, where he directs the Distributed Computing and Web Technology concentration areas. When he gets a chance, he also teaches industry short courses on servlets, JavaServer Pages, and other Java technology areas. He is the author of *Core Servlets and JavaServer Pages* and the first edition of *Core Web Programming*. Marty can be reached at the following address:

*Research and Technology Development Center  
The Johns Hopkins University Applied Physics Laboratory  
11100 Johns Hopkins Road  
Laurel, MD 20723  
[hall@corewebprogramming.com](mailto:hall@corewebprogramming.com)*

Larry Brown is a Senior Network Engineer at the Naval Surface Warfare Center, Carderock

Division, where he specializes in developing and deploying network and Web solutions in an enterprise environment. He is also a Computer Science faculty member at the Johns Hopkins University, where he teaches server-side programming, distributed Web programming, and Java user interface development for the part-time graduate program in Computer Science. Larry can be reached at the following address:

*Naval Surface Warfare Center, Carderock Division  
9500 MacArthur Boulevard  
West Bethesda, MD 20817  
[brown@corewebprogramming.com](mailto:brown@corewebprogramming.com)*

## Acknowledgments

Many people have helped us out with this book. Without their assistance, we would still be on the fourth chapter. Those that provided valuable technical feedback, pointed out errors, and gave useful suggestions include Don Aldridge, Chris Bennett, Camille Bell, Pete Clark, Maria Dimalanta, Nguyen-Khoa Duy, Denise Evans, Amy Karlson, Paul McNamee, Toddi Norum, Walter Pasquinni, Rich Slywczak, Bob Tinker, and Kim Topley. This book would not be a success without their contributions. Mary Lou "Eagle Eye" Nohr spotted our errant commas, awkward sentences, typographical errors, and grammatical inconsistencies. She improved the result immensely. We hope that we learned from her advice. Vanessa Moore produced the final version; she did a great job despite our last-minute changes and crazy travel schedules. Ralph Semmel and Julie Wessel both provided supportive work environments and flexible schedules. Greg Doench of Prentice Hall believed in a second edition and encouraged us to write the book. Thanks to all.

Most of all, I—Marty—thank B.J., Lindsay, and Nathan for their patience with my long hours and funny schedule. I—Larry—thank Lee for her loving support and patience while I disappeared to the computer room every weekend.

God has blessed us both with great families.