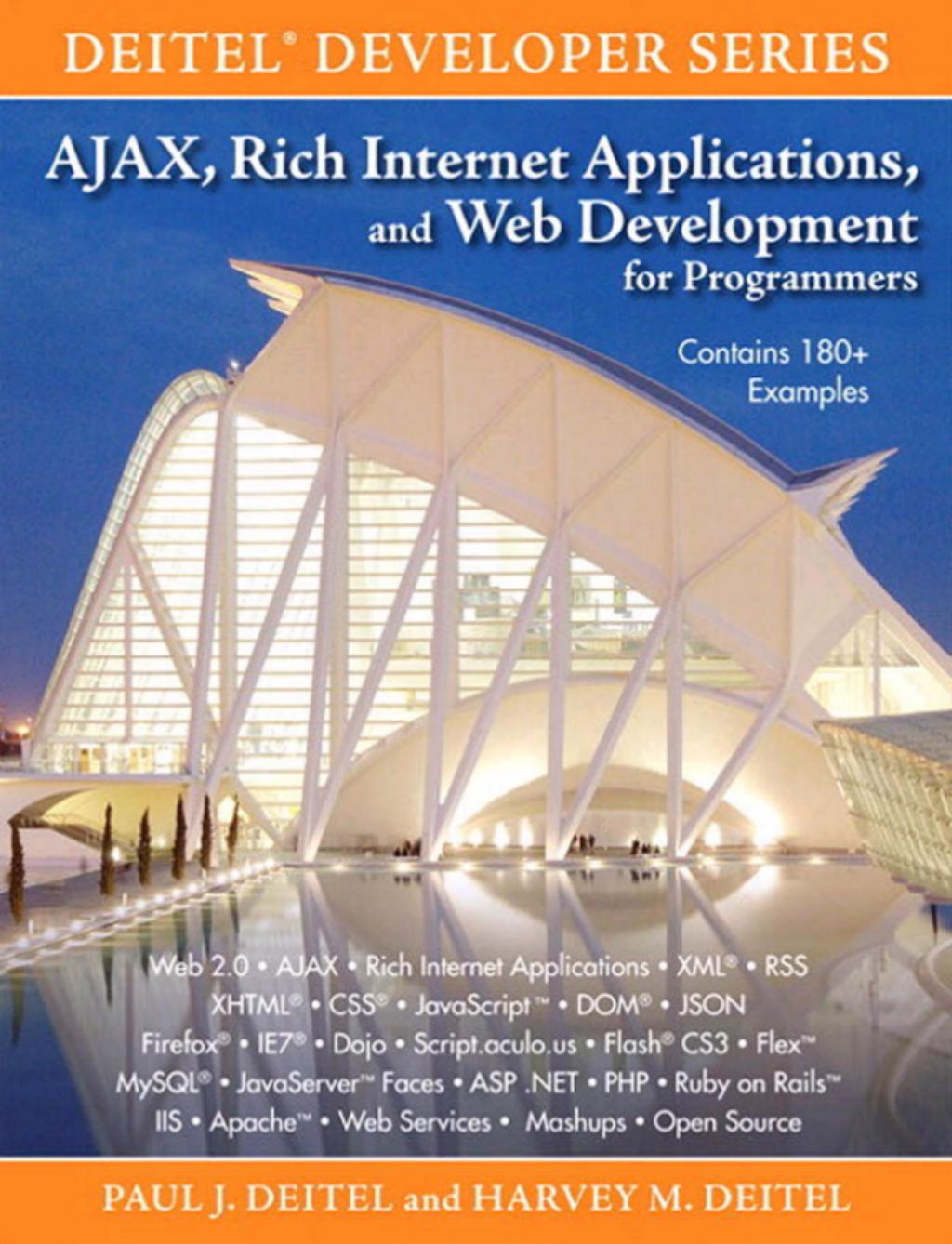


DEITEL® DEVELOPER SERIES

# AJAX, Rich Internet Applications, and Web Development for Programmers

Contains 180+  
Examples



Web 2.0 • AJAX • Rich Internet Applications • XML® • RSS  
XHTML® • CSS® • JavaScript™ • DOM® • JSON  
Firefox® • IE7® • Dojo • Script.aculo.us • Flash® CS3 • Flex™  
MySQL® • JavaServer™ Faces • ASP .NET • PHP • Ruby on Rails™  
IIS • Apache™ • Web Services • Mashups • Open Source

PAUL J. DEITEL and HARVEY M. DEITEL

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U. S. Corporate and Government Sales  
(800) 382-3419  
[corpsales@pearsontechgroup.com](mailto:corpsales@pearsontechgroup.com)

For sales outside the U.S., please contact:

International Sales  
[international@pearsoned.com](mailto:international@pearsoned.com)

Visit us on the Web: [www.phptr.com](http://www.phptr.com)



#### This Book Is Safari Enabled

The Safari® Enabled icon on the cover of your favorite technology book means the book is available through Safari Bookshelf. When you buy this book, you get free access to the online edition for 45 days.

Safari Bookshelf is an electronic reference library that lets you easily search thousands of technical books, find code samples, download chapters, and access technical information whenever and wherever you need it.

To gain 45-day Safari Enabled access to this book:

- Go to <http://www.prenhallprofessional.com/safarienabled>
- Complete the brief registration form
- Enter the coupon code M8IP-VNEG-CZHV-DIG8-1652

If you have difficulty registering on Safari Bookshelf or accessing the online edition, please e-mail [customer-service@safaribooksonline.com](mailto:customer-service@safaribooksonline.com).

---

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

On file

© 2008 Pearson Education, Inc.

All rights reserved. Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, write to:

Pearson Education, Inc.  
Rights and Contracts Department  
501 Boylston Street, Suite 900  
Boston, MA 02116  
Fax (617) 671-3447

ISBN 0-13-158738-2

Text printed in the United States on recycled paper at R.R. Donnelley in Crawfordsville, Indiana.  
First printing, January 2008

# Preface

---

*... the challenges are for the designers of these applications: to forget what we think we know about the limitations of the Web, and begin to imagine a wider, richer range of possibilities. It's going to be fun.*

—Jesse James Garrett, Adaptive Path  
“Ajax: A New Approach to Web Applications”  
(adaptivepath.com/ideas/essays/archives/000385.php)

## Introduction

Welcome to Internet and web programming and Web 2.0! And welcome to a walkthrough of the Web 2.0 phenomenon from the technical, business and social perspectives. We've worked hard to create what we hope you'll find to be an informative, entertaining and challenging learning experience. At Deitel & Associates, we write programming language professional books and textbooks for Prentice Hall, deliver corporate training worldwide and develop Web 2.0 Internet businesses. This book reflects today's Web 2.0, Ajax-based, Rich Internet Application-development methodologies.

The technologies you'll learn here are appropriate for experienced professionals who build substantial information systems. You'll find "industrial-strength" code examples. We have attempted to write in a clear and straightforward manner using best practices.

Perhaps most important, the book presents hundreds of working code examples and shows the outputs produced when these examples are rendered in browsers or run on computers. We present all concepts in the context of complete working programs. We call this the "live-code approach." All of the source code is available for download from

[www.deitel.com/books/ajaxfp/](http://www.deitel.com/books/ajaxfp/)

under the heading "Download Code Examples and Other Premium Content for Registered Users," once you've registered and logged on to our site.

The early chapters present a carefully paced introduction to "client-side" web programming, using the popular JavaScript language and the closely related technologies of XHTML (Extensible HyperText Markup Language), CSS (Cascading Style Sheets) and the DOM (Document Object Model). The material in the JavaScript chapters presents a solid foundation for the deeper treatment of scripting in the Adobe Flash, Adobe Flex, PHP and Ruby on Rails chapters later in the book. Experienced programmers will read the early chapters quickly and find the treatment of scripting in the later chapters to be rigorous and challenging.

Today's users are accustomed to desktop applications with rich graphical user interfaces (GUIs), such as those used on Apple's Mac OS X systems, Microsoft Windows systems, various Linux systems and more. Users want applications that employ the

multimedia capabilities of graphics, images, animation, audio and video. They want applications that can run on the Internet and the web and communicate with other applications. Users want to apply database technologies for storing and manipulating their business and personal data. They want applications that are not limited to the desktop or even to some local computer network, but that can integrate Internet and web components, and remote databases. Programmers want to use all these capabilities in a truly portable manner so that applications will run without modification on a variety of platforms.

In this book, we present a number of powerful software technologies that will enable you to build these kinds of systems. Early in the book we focus on building the portions of web-based applications that reside on the client side (i.e., the portions of applications that typically run in web browsers such as Mozilla's Firefox or Microsoft's Internet Explorer), using technologies such as the XHTML, JavaScript, CSS, Flash, Flex and Extensible Markup Language (XML). Later in the book we concentrate on using technologies such as web servers, databases, PHP, Ruby on Rails, ASP.NET, ASP.NET Ajax and JavaServer Faces (JSF) to build the server side of web-based applications. These portions of applications typically run on "heavy-duty" computer systems on which organizations' business-critical websites reside. By mastering the technologies in this book, you'll be able to build substantial web-based, client/server, database-intensive, "multitier" applications.

If you have questions as you read this book, send an e-mail to [deitel@deitel.com](mailto:deitel@deitel.com)—we'll respond promptly. For updates on the book and the status of all supporting software, and for the latest news on Deitel publications and services, visit [www.deitel.com](http://www.deitel.com). Sign up at [www.deitel.com/newsletter/subscribe.html](http://www.deitel.com/newsletter/subscribe.html) for the free *Deitel® Buzz Online* e-mail newsletter and check out [www.deitel.com/ResourceCenters.html](http://www.deitel.com/ResourceCenters.html) for our growing list of Internet and web programming, Internet business, Web 2.0 and related Resource Centers. Each week we announce our latest Resource Centers in the newsletter.

## Key Features

Here's some of the key features of *Ajax, Rich Internet Applications and Web Development for Programmers*:

- Reflects today's Web 2.0, Ajax-based, Rich Internet Application-development methodologies.
- Coverage of the two leading web browsers—Internet Explorer and Firefox. All client-side applications in the book run correctly on both browsers.
- Focus on Web 2.0 technologies and concepts.
- Chapter on Web 2.0 and Internet Business (reviewed by leaders in the Web 2.0 community).
- Focus on building Rich Internet Applications with the interactivity of desktop applications.
- Chapter on building Ajax-enabled web applications with "raw" Ajax and with the Dojo JavaScript libraries. Applications in this chapter demonstrate partial-page updates and type-ahead capabilities.
- Chapter on Adobe Flex—a Rich Internet Application framework for creating scalable, cross-platform, multimedia-rich applications for delivery within the enterprise or across the Internet.

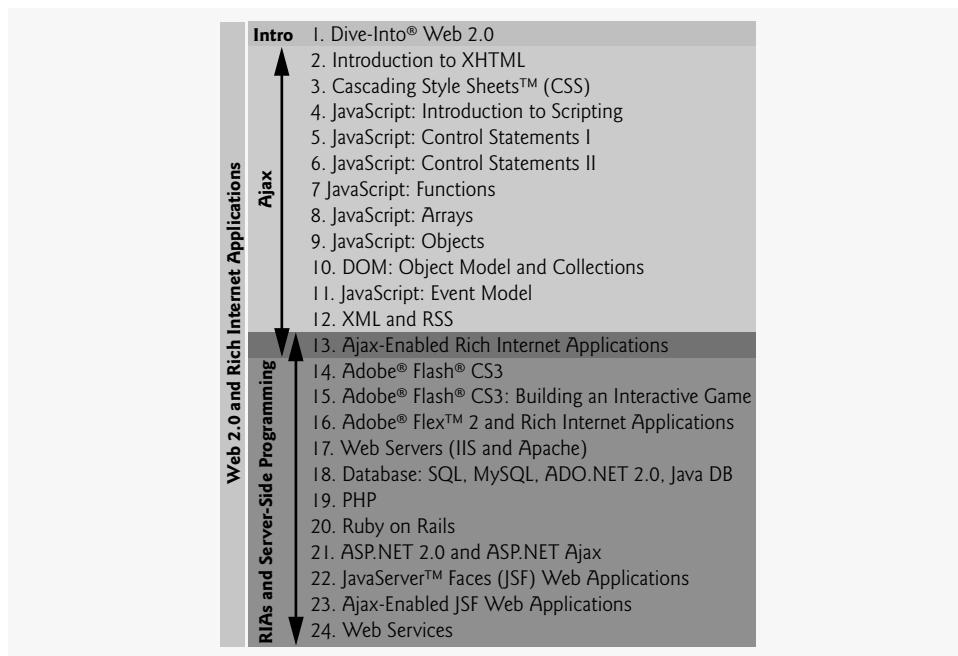
- Chapter on rapid applications development of database-driven web applications with Ruby on Rails; also discusses developing Ajax applications with the Prototype and Script.aculo.us libraries.
- Two chapters on Adobe Flash CS3, including building a computer game.
- Significant treatment of client-side scripting with JavaScript.
- Significant treatments of XHTML DOM manipulation and JavaScript events.
- Significant treatment of XML DOM manipulation with JavaScript.
- Chapter on building SOAP-based web services with Java and REST-based web services with ASP.NET (using Visual Basic).
- Chapter on PHP 5.
- Coverage of ASP.NET, featuring ASP.NET Ajax.
- JavaServer Faces (JSF) coverage emphasizing building Ajax-enabled JSF applications.
- Client-side case studies that enable you to interact with preimplemented server-side applications and web services that we host at `test.deitel.com`.
- Case studies including Deitel Cover Viewer (JavaScript/DOM), Address Book (Ajax), Cannon Game (Flash), Weather/Yahoo! Maps Mashup (Flex), Mailing List (PHP/MySQL), Message Forum and Flickr Photo Viewer (Ruby on Rails), Guest Book and Secure Books Database (ASP.NET), Address Book with Google Maps (JavaServer Faces) and Blackjack (JAX-WS web services).

All of this has been carefully reviewed by a team of 38 distinguished industry developers and academics.

## **AJAX, Rich Internet Applications and Web Development for Programmers Architecture**

This book focuses on Web 2.0 and Rich Internet Application (RIA) development. Our goal is to develop webtop applications that have the responsiveness, look and feel of traditional desktop applications. Deitel & Associates, Inc. has evolved into a development organization, while continuing its focus on programming languages textbook and professional book authoring, and corporate training. We're building the infrastructure for the Internet businesses we're designing and developing as part of our Web 2.0 Internet Business Initiative. This book includes discussions of many practical issues we've encountered in developing that infrastructure.

Figure 1 shows the architecture of *AJAX, Rich Internet Applications and Web Development for Programmers*. The book is divided into several parts. Chapter 1 introduces Web 2.0 from the technical, business and social perspectives, and provides a foundation for understanding Rich Internet Application development. If you are a serious web developer, you'll want to test your web applications across many browsers and platforms. The examples for the book execute correctly on both Microsoft's Internet Explorer 7 (IE7) and Mozilla's Firefox 2 (FF2) browsers. Most of the examples will also work in other browsers such as Opera and Safari, but may not work on earlier browsers. Microsoft Windows users should upgrade to IE7 and install Firefox; readers with other operating systems should install Firefox.



**Fig. 1 |** Architecture of *AJAX, Rich Internet Applications and Web Development for Programmers*.

The second part of the book, Chapters 2–13, presents a detailed treatment of Ajax component technologies, including a comprehensive treatment of JavaScript that spans nine chapters and concludes with Chapter 13’s treatment of Ajax development. Ajax is not a new technology—we’ve been writing about all but one of its component technologies since 1999, and many of the technologies existed before that. However, Ajax is one of the key technologies of Web 2.0 and RIAs. Chapters 2–13 cover “raw” Ajax programming, where you’ll handle the details yourself—several later chapters in the book demonstrate technologies that encapsulate Ajax functionality to help you easily build Ajax-based applications that operate across a wide variety of browsers and browser versions without your having to be concerned with the low-level details.

The third part of the book, Chapters 13–24, focuses on both the client and server sides of the GUI and the graphical part of RIA development. Here we cover client-side technologies such as Adobe Flash and Adobe Flex that use, or can be combined with, Ajax or Ajax-like capabilities to develop RIAs. Each of these technologies also can consume web services. Next, we present the server side of web application development with discussions of web servers (IIS and Apache), databases, server-side scripting languages such as PHP and Ruby on Rails, and several server-side frameworks such as ASP.NET 2.0 and Java-Server Faces. We complete our server-side discussion with a chapter on building both SOAP-based and REST-based web services.

You may have noticed that Chapter 13, Ajax-Enabled Rich Internet Applications, overlaps the second and third parts of the book. Chapter 13 serves as a bridge from “raw” Ajax development to “encapsulated” Ajax development with the Dojo libraries.

## Dependency Chart

Figure 2 illustrates the dependencies that exist among chapters in the book. An arrow pointing into a chapter indicates that it depends on the content of the chapter from which the arrow points. For example, Chapter 24, Web Services, depends on both Chapters 21 and 23. We recommend that you read all of a given chapter's dependencies before reading that chapter, though other orders are possible. We've also commented on some additional dependencies in the diagram's footnotes.

## Teaching Approach

*AJAX, Rich Internet Applications and Web Development for Programmers* contains a rich collection of examples. The book concentrates on the principles of good software engineering and stresses program clarity. We are educators who teach leading-edge topics in industry classrooms worldwide. Dr. Harvey M. Deitel has 20 years of college teaching experience and 18 years of industry teaching experience. Paul Deitel has 16 years of industry teaching experience. The Deitels have taught courses at all levels to government, industry, military and academic clients of Deitel & Associates.

**Live-Code Approach.** *AJAX, Rich Internet Applications and Web Development for Programmers* is loaded with “live-code” examples—each new concept is presented in the context of a complete working web application that is immediately followed by one or more screen captures showing the application’s functionality. This style exemplifies the way we teach and write about programming; we call this the “live-code approach.”

**Syntax Shading.** We syntax shade all the code, similar to the way most integrated-development environments and code editors syntax color code. This improves code readability—an important goal, given that this book contains about 18,000 lines of code in complete, working programs. Our syntax-shading conventions are as follows:

*comments appear in italic*  
**keywords appear in bold italic**  
**PHP, Ruby, ASP.NET, JSP delimiters and errors appear in bold black**  
**constants and literal values appear in bold gray**  
all other code appears in black

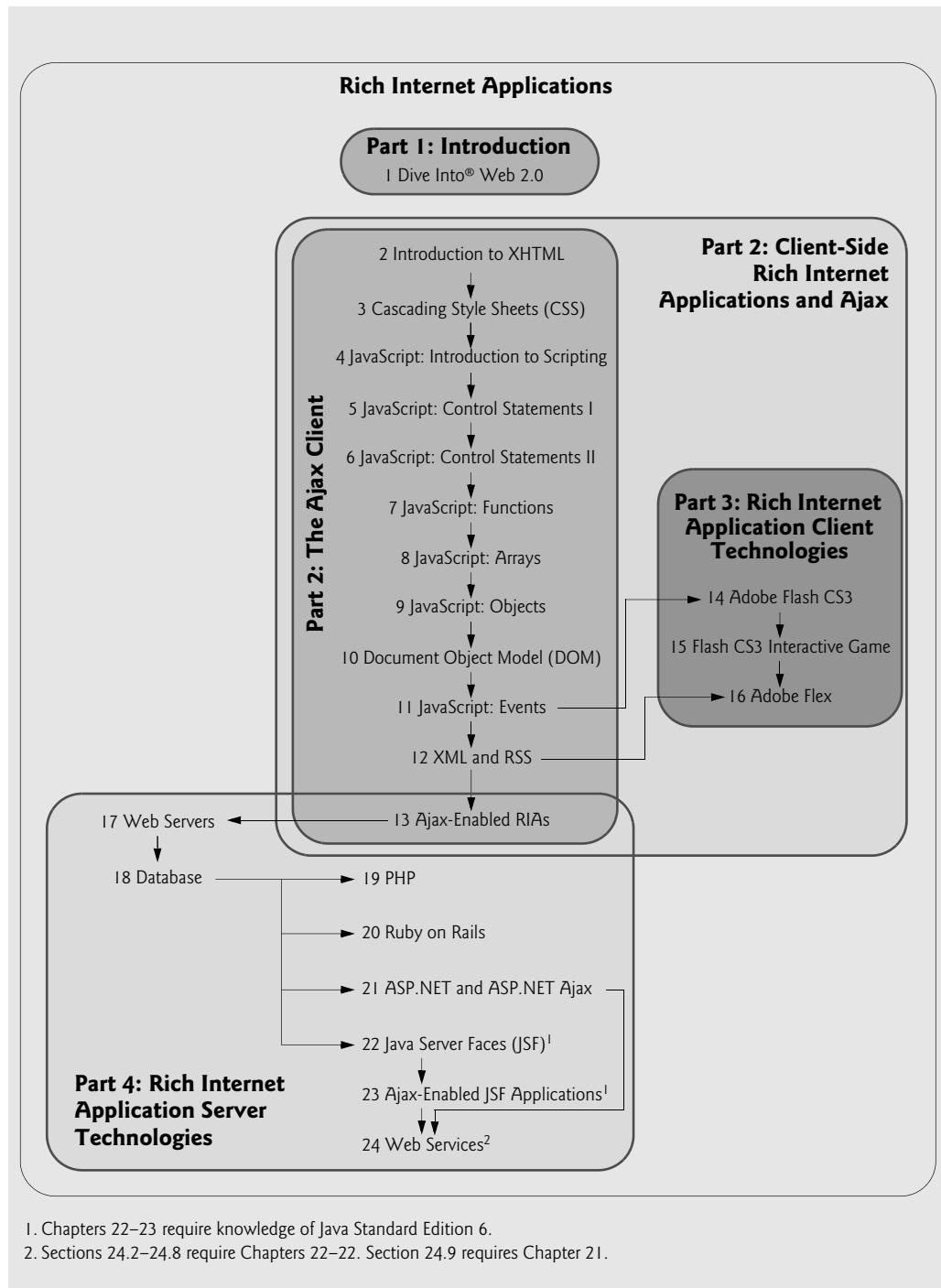
**Code Highlighting.** We place white rectangles around each program’s key code segments.

**Using Fonts for Emphasis.** We place the key terms and the index’s page reference for each defining occurrence in ***bold italic*** text for easier reference. We emphasize on-screen components in the **bold Helvetica** font (e.g., the **File** menu) and emphasize program text in the Lucida font (e.g., `int x = 5`).

**Web Access.** All of the source-code examples for *AJAX, Rich Internet Applications and Web Development for Programmers* are available for download from:

[www.deitel.com/books/ajaxfp/](http://www.deitel.com/books/ajaxfp/)

Site registration is quick, easy and free. Download all the examples, then run each program as you read the corresponding text discussions. Making changes to the examples and seeing the effects of those changes is a great way to enhance your Internet and web programming learning experience.



**Fig. 2 |** *AJAX, Rich Internet Applications and Web Development for Programmers* chapter dependency chart.

**Objectives.** Each chapter begins with a statement of objectives. This lets you know what to expect and gives you an opportunity to determine if you have met the objectives after reading the chapter.

**Quotations.** The learning objectives are followed by quotations. Some are humorous; some are philosophical; others offer interesting insights. We hope that you enjoy relating the quotations to the chapter material.

**Outline.** The chapter outline helps you approach the material in a top-down fashion, so you can anticipate what is to come and set a comfortable learning pace.

**Illustrations/Figures.** Abundant charts, tables, line drawings, programs and program output are included.

**Programming Tips.** We include programming tips to help you focus on important aspects of program development. These tips and practices represent the best we have gleaned from a combined six decades of programming and teaching experience. One of our readers told us that she feels this approach is like the highlighting of axioms, theorems and corollaries in mathematics books—it provides a basis on which to build good software.



## Good Programming Practices

---

Good Programming Practices *call attention to techniques that will help you produce programs that are clearer, more understandable and more maintainable.*



## Common Programming Errors

---

Programmers tend to make certain kinds of errors frequently. Pointing out these Common Programming Errors reduces the likelihood that you'll make the same mistakes.



## Error-Prevention Tip

---

These tips contain suggestions for exposing bugs and removing them from your programs; many describe aspects of programming that prevent bugs from getting into programs in the first place.



## Performance Tip

---

These tips highlight opportunities for making your programs run faster or minimizing the amount of memory that they occupy.



## Portability Tip

---

We include Portability Tips to help you write code that will run on a variety of platforms and to explain how to achieve a high degree of portability.



## Software Engineering Observation

---

The Software Engineering Observations highlight architectural and design issues that affect the construction of software systems, especially large-scale systems.

**Thousands of Index Entries.** We have included an extensive index which is especially useful when you use the book as a reference.

**“Double Indexing” of Live-Code Examples.** For every source-code program in the book, we index the figure caption both alphabetically and as a subindex item under “Examples.” This makes it easier to find examples using particular features.

## AJAX, Rich Internet Applications and Web Development for Programmers Software Downloads and Additional Resources

Many Internet and web development tools are available. We wrote *AJAX, Rich Internet Applications and Web Development for Programmers* using Internet Explorer 7, Firefox 2 and other free-for-download software. Links to additional resources and software downloads are available in our Internet and Web programming related Resource Centers:

[www.deitel.com/resourcecenters.html/](http://www.deitel.com/resourcecenters.html/)

and at the website for this book:

[www.deitel.com/books/ajaxfp/](http://www.deitel.com/books/ajaxfp/)

## Deitel® Buzz Online Free E-mail Newsletter

Each week, the free *Deitel® Buzz Online* newsletter announces our latest Resource Center(s) and includes commentary on industry trends and developments, links to free articles and resources from our published books and upcoming publications, product-release schedules, errata, challenges, anecdotes, information on our corporate instructor-led training courses and more. It's also a good way for you to keep posted about issues related to *AJAX, Rich Internet Applications and Web Development for Programmers*. To subscribe, visit

[www.deitel.com/newsletter/subscribe.html](http://www.deitel.com/newsletter/subscribe.html)

## The Deitel Online Resource Centers

Our website, [www.deitel.com](http://www.deitel.com), provides scores of Resource Centers on various topics including programming languages, software, Web 2.0, Internet business and open source projects (Fig. 3). The Resource Centers have evolved out of the research we do to support our books and business endeavors. We've found many exceptional resources including tutorials, documentation, software downloads, articles, blogs, podcasts, videos, code samples, books, e-books and more. Most of them are free. In the spirit of Web 2.0, we share these resources with the worldwide community. Each week we announce our latest Resource Centers in the *Deitel® Buzz Online* ([www.deitel.com/newsletter/subscribe.html](http://www.deitel.com/newsletter/subscribe.html)).

## Acknowledgments

It is a great pleasure to acknowledge the efforts of many people whose names may not appear on the cover, but whose hard work, cooperation, friendship and understanding were crucial to the production of the book. Many people at Deitel & Associates, Inc. devoted long hours to this project—thanks especially to Abbey Deitel and Barbara Deitel.

We'd also like to thank the participants in our Honors Internship program who contributed to this publication—Ilana Segall, a mathematical and computational science major at Stanford University; Jennifer Fredholm, an English and computer science major at New York University; Scott Wehrwein, a computer science major at Middlebury College; Mark Kagan, a computer science, economics and math major at Brandeis University; Andrew Faden, a computer engineering major at Northeastern University; Jessica Henkel, a psychology major and business minor at Northeastern University; and Kyle Banks, a computer science and business major at Northeastern University.

Deitel Resource Centers			
<b>Programming</b>	Visual Studio Team	Java EE 5	MySQL
ASP.NET 3.5	System	Java SE 6	Open Source
Adobe Flex	Web 3D Technologies	Java SE 7 (Dolphin)	Search Engines
Ajax	Web Services	Resource Center	SharePoint
Apex	Windows Presentation	JavaFX	Skype
ASP.NET Ajax	Foundation	<b>Microsoft</b>	Web Servers
ASP.NET	XHTML	ASP.NET	Wikis
C	XML	ASP.NET 3.5	Windows Vista
C++		ASP.NET Ajax	<b>Web 2.0</b>
C++ Boost Libraries	<b>Computer Science</b>	C#	Alert Services
C++ Game Programming	Regular Expressions	DotNetNuke (DNN)	Attention Economy
C#		Internet Explorer 7 (IE7)	Blogging
Code Search Engines and Code Sites	<b>Games and Game Programming</b>	Microsoft LINQ	Building Web Communities
Computer Game Programming	Computer Game Programming	.NET	Community Generated Content
CSS 2.1	Computer Games	.NET 3.0	Facebook Developer Platform
Dojo	Mobile Gaming	.NET 3.5	Facebook Social Ads
Facebook Developer Platform	Sudoku	SharePoint	Google Base
Flash 9		Silverlight	Google Video
Java	<b>Internet Business</b>	Visual Basic	Google Web Toolkit (GWT)
Java Certification and Assessment Testing	Affiliate Programs	Visual C++	Internet Video
Java Design Patterns	Competitive Analysis	Visual Studio Team System	Joost
Java EE 5	Facebook Social Ads	Windows Presentation Foundation	Location-Based Services
Java SE 6	Google AdSense	Windows Vista	Mashups
Java SE 7 (Dolphin) Resource Center	Google Analytics	Microsoft Popfly	Microformats
JavaFX	Google Services		Recommender Systems
JavaScript	Internet Advertising	<b>Open Source &amp; LAMP Stack</b>	RSS
JSON	Internet Business Initiative	Apache	Social Graph
Microsoft LINQ	Internet Public Relations	DotNetNuke (DNN)	Social Media
Microsoft Popfly	Link Building	Eclipse	Social Networking
.NET	Location-Based Services	Firefox	Software as a Service (SaaS)
.NET 3.0	Online Lead Generation	Linux	Virtual Worlds
.NET 3.5	Podcasting	MySQL	Web 2.0
OpenGL	Search Engine Optimization	Open Source	Web 3.0
Perl	Selling Digital Content	Perl	Widgets
PHP	Sitemaps	PHP	<b>Dive Into Web 2.0 eBook</b>
Programming Projects	Web Analytics	Python	<b>Other Topics</b>
Python	Website Monetization	Ruby	Computer Games
Regular Expressions	YouTube and AdSense		Computing Jobs
Ruby	<b>Java</b>	<b>Software</b>	Gadgets and Gizmos
Ruby on Rails	Java	Apache	Ring Tones
Silverlight	Java Certification and Assessment Testing	DotNetNuke (DNN)	Sudoku
Visual Basic	Java Design Patterns	Eclipse	
Visual C++		Firefox	
		Internet Explorer 7 (IE7)	
		Linux	

**Fig. 3** | Deitel Resource Centers at [www.deitel.com/resourcecenters.html](http://www.deitel.com/resourcecenters.html).

We are fortunate to have worked on this project with the talented and dedicated team of publishing professionals at Prentice Hall. We appreciate the extraordinary efforts of Marcia Horton, Editorial Director of Prentice Hall's Engineering and Computer Science Division, Mark Taub, Editor-in-Chief of Prentice Hall Professional, and John Fuller, Managing Editor of Prentice Hall Professional. Carole Snyder and Dolores Mars did a remarkable job recruiting the book's large review team and managing the review process. Sandra Schroeder and Chuti Prasertsith did a wonderful job designing the book's cover. Bob Engelhardt and Marta Samsel did a marvelous job managing the book's production.

This book was adapted from our book *Internet & World Wide Web How to Program*, 4/e. We wish to acknowledge the efforts of our reviewers on that book. Adhering to a tight time schedule, they scrutinized the text and the programs, providing countless suggestions for improving the accuracy and completeness of the presentation.

### **Reviewers**

Roland Bouman (MySQL AB), Peter Brandano (KoolConnect Technologies, Inc.), Matt Chotin (Adobe), Chris Cornutt (PHPDeveloper.org), Phil Costa (Adobe), Umachitra Damodaran (Sun Microsystems), Vadiraj Deshpande (Sun Microsystems), Justin Erenkrantz (The Apache Software Foundation), Christopher Finke (Netscape), Jesse James Garrett (Adaptive Path), Kevin Henrikson (Zimbra.com), Tim Heuer (Microsoft), Molly E. Holtzschlag (W3C), Ralph Hooper (University of Alabama, Tuscaloosa), Johnvey Hwang (Splunk, Inc.), Joe Kromer (New Perspective and the Pittsburgh Adobe Flash Users Group), Eric Lawrence (Microsoft), Billy B. L. Lim (Illinois State University), Shobana Mahadevan (Sun Microsystems), Patrick Mineault (Freelance Flash Programmer), Anand Narayanaswamy (Microsoft), John Peterson (Insync and V.I.O., Inc.), Jennifer Powers (University of Albany), Robin Schumacher (MySQL AB), José Antonio González Seco (Parlamento de Andalucía), Dr. George Semeczko (Royal & SunAlliance Insurance Canada), Steven Shaffer (Penn State University), Tim Sneath (Microsoft), Karen Tegtmeyer (Model Technologies, Inc.), Paul Vencill (MITRE), Raymond Wen (Microsoft), Eric M. Wendelin (Auto-trol Technology Corporation), Raymond F. Wisman (Indiana University) and Daniel Zappala (Brigham Young University).

Well, there you have it! We hope you enjoy this look at the exciting world of Ajax, Rich Internet Applications and web development in a Web 2.0 world. As you read the book, we would sincerely appreciate your comments, criticisms, corrections and suggestions for improving the text. Please address all correspondence to:

[deitel@deitel.com](mailto:deitel@deitel.com)

We'll respond promptly, and post corrections and clarifications at:

[www.deitel.com/books/ajaxfp/](http://www.deitel.com/books/ajaxfp/)

We hope you enjoy reading *Ajax, Rich Internet Applications and Web Development for Programmers* as much as we enjoyed writing it!

*Paul J. Deitel*

*Dr. Harvey M. Deitel*

Maynard, Massachusetts

## About the Authors

**Paul J. Deitel**, CEO and Chief Technical Officer of Deitel & Associates, Inc., is a graduate of MIT's Sloan School of Management, where he studied Information Technology. He holds the Java Certified Programmer and Java Certified Developer certifications, and has been designated by Sun Microsystems as a Java Champion. Through Deitel & Associates, Inc., he has delivered Internet and web programming, Java, C, C++, C# and Visual Basic courses to industry clients, including IBM, Sun Microsystems, Dell, Lucent Technologies, Fidelity, NASA at the Kennedy Space Center, the National Severe Storm Laboratory, White Sands Missile Range, Rogue Wave Software, Boeing, Stratus, Cambridge Technology Partners, Open Environment Corporation, One Wave, Hyperion Software, Adra Systems, Entergy, CableData Systems, Nortel Networks, Puma, iRobot, Invensys and many more. He has also lectured on Java and C++ for the Boston Chapter of the Association for Computing Machinery. He and his father, Dr. Harvey M. Deitel, are the world's best-selling programming language textbook authors.

**Dr. Harvey M. Deitel**, Chairman and Chief Strategy Officer of Deitel & Associates, Inc., has 46 years of experience in the computer field. Dr. Deitel earned B.S. and M.S. degrees from MIT and a Ph.D. from Boston University. He has 20 years of college teaching experience, including earning tenure and serving as the Chairman of the Computer Science Department at Boston College before founding Deitel & Associates, Inc., with his son, Paul J. Deitel. He and Paul are the co-authors of several dozen books and multimedia packages and they are writing many more. With translations published in Japanese, German, Russian, Spanish, Traditional Chinese, Simplified Chinese, Korean, French, Polish, Italian, Portuguese, Greek, Urdu and Turkish, the Deitels' texts have earned international recognition. Dr. Deitel has delivered hundreds of professional seminars to major corporations, academic institutions, government organizations and the military.

## About Deitel & Associates, Inc.

Deitel & Associates, Inc., is an internationally recognized corporate training and content-creation organization specializing in computer programming languages, Internet and web software technology, object technology education and Internet business development through its Web 2.0 Internet Business Initiative. The company provides instructor-led courses on major programming languages and platforms, such as C++, Java, C, C#, Visual C++, Visual Basic, XML, object technology and Internet and web programming. The founders of Deitel & Associates, Inc. are Paul J. Deitel and Dr. Harvey M. Deitel. The company's clients include many of the world's largest companies, government agencies, branches of the military, and academic institutions. Through its 32-year publishing partnership with Prentice Hall, Deitel & Associates, Inc. publishes leading-edge programming professional books, textbooks, interactive multimedia *Cyber Classrooms*, *Complete Training Courses*, web-based training courses, online and offline *LiveLessons* video courses, and e-content for the popular course management systems WebCT, Blackboard and Pearson's CourseCompass. Deitel & Associates, Inc., and the authors can be reached via e-mail at:

[deitel@deitel.com](mailto:deitel@deitel.com)

To learn more about Deitel & Associates, Inc., its publications and its worldwide *Dive Into®* Series Corporate Training curriculum, visit:

[www.deitel.com](http://www.deitel.com)

and subscribe to the free *Deitel® Buzz Online* e-mail newsletter at:

[www.deitel.com/newsletter/subscribe.html](http://www.deitel.com/newsletter/subscribe.html)

Check out the growing list of online Resource Centers at:

[www.deitel.com/ResourceCenters.html](http://www.deitel.com/ResourceCenters.html)

Individuals wishing to purchase Deitel publications can do so through:

[www.deitel.com/books/index.html](http://www.deitel.com/books/index.html)

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U. S. Corporate and Government Sales  
(800) 382-3419  
[corpsales@pearsontechgroup.com](mailto:corpsales@pearsontechgroup.com)

For sales outside the U. S., please contact:

International Sales  
[international@pearsoned.com](mailto:international@pearsoned.com)

Visit the publisher on the Web at [www.phptr.com](http://www.phptr.com).

# 10

# Document Object Model (DOM): Objects and Collections

## OBJECTIVES

In this chapter you will learn:

- How to use JavaScript and the W3C Document Object Model to create dynamic web pages.
- The concepts of DOM nodes and DOM trees.
- How to traverse, edit and modify elements in an XHTML document.
- How to change CSS styles dynamically.
- To create JavaScript animations.

*Our children may learn about heroes of the past. Our task is to make ourselves architects of the future.*

—Jomo Mzee Kenyatta

*Though leaves are many, the root is one.*

—William Butler Yeats

*The thing that impresses me most about America is the way parents obey their children.*

—Duke of Windsor

*Most of us become parents long before we have stopped being children.*

—Mignon McLaughlin

*To write it, it took three months; to conceive it three minutes; to collect the data in it—all my life.*

—F. Scott Fitzgerald

*Sibling rivalry is inevitable. The only sure way to avoid it is to have one child.*

—Nancy Samalin

**Outline**

- 10.1** Introduction
- 10.2** Modeling a Document: DOM Nodes and Trees
- 10.3** Traversing and Modifying a DOM Tree
- 10.4** DOM Collections
- 10.5** Dynamic Styles
- 10.6** Summary of the DOM Objects and Collections
- 10.7** Web Resources

## 10.1 Introduction

In this chapter we introduce the *Document Object Model (DOM)*. The DOM gives you access to all the elements on a web page. Inside the browser, the whole web page—paragraphs, forms, tables, etc.—is represented in an *object hierarchy*. Using JavaScript, you can create, modify and remove elements in the page dynamically.

Previously, both Internet Explorer and Netscape had different versions of Dynamic HTML, which provided similar functionality to the DOM. However, while they provided many of the same capabilities, these two models were incompatible with each other. In an effort to encourage cross-browser websites, the W3C created the standardized Document Object Model. Firefox 2, Internet Explorer 7, and many other major browsers implement *most* of the features of the W3C DOM.

This chapter begins by formally introducing the concept of DOM nodes and DOM trees. We then discuss properties and methods of DOM nodes and cover additional methods of the document object. We also discuss how to dynamically change style properties, which enables you to create many types of effects, such as user-defined background colors and animations. Then, we present a diagram of the extensive object hierarchy, with explanations of the various objects and properties, and we provide links to websites with further information on the topic.



### Software Engineering Observation 10.1

---

*With the DOM, XHTML elements can be treated as objects, and many attributes of XHTML elements can be treated as properties of those objects. Then, objects can be scripted (through their id attributes) with JavaScript to achieve dynamic effects.*

## 10.2 Modeling a Document: DOM Nodes and Trees

As we saw in previous chapters, the document's `getElementById` method is the simplest way to access a specific element in a page. In this section and the next, we discuss more thoroughly the objects returned by this method.

The `getElementById` method returns objects called *DOM nodes*. Every element in an XHTML page is modeled in the web browser by a DOM node. All the nodes in a document make up the page's *DOM tree*, which describes the relationships among elements. Nodes are related to each other through child-parent relationships. An XHTML element inside another element is said to be a *child* of the containing element. The containing element is known as the *parent*. A node may have multiple children, but only one parent. Nodes with the same parent node are referred to as *siblings*.

Some browsers have tools that allow you to see a visual representation of the DOM tree of a document. When installing Firefox, you can choose to install a tool called the **DOM Inspector**, which allows you to view the DOM tree of an XHTML document. To inspect a document, Firefox users can access the **DOM Inspector** from the **Tools** menu of Firefox. If the DOM inspector is not in the menu, run the Firefox installer and choose **Custom** in the **Setup Type** screen, making sure the **DOM Inspector** box is checked in the **Optional Components** window.

Microsoft provides a *Developer Toolbar* for Internet Explorer that allows you to inspect the DOM tree of a document. The toolbar can be downloaded from Microsoft at [go.microsoft.com/fwlink/?LinkId=92716](http://go.microsoft.com/fwlink/?LinkId=92716). Once the toolbar is installed, restart the browser, then click the » icon at the right of the toolbar and choose **IE Developer Toolbar** from the menu. Figure 10.1 shows an XHTML document and its DOM tree displayed in Firefox's DOM Inspector and in IE's Web Developer Toolbar.

The XHTML document contains a few simple elements. We explain the example based on the Firefox DOM Inspector—the IE Toolbar displays the document with only minor differences. A node can be expanded and collapsed using the + and - buttons next to the node's name. Figure 10.1(b) shows all the nodes in the document fully expanded. The document node (shown as **#document**) at the top of the tree is called the *root node*, because it has no parent. Below the document node, the **HTML** node is indented from the document node to signify that the **HTML** node is a child of the **#document** node. The **HTML** node represents the `html` element (lines 7–24).

The **HEAD** and **BODY** nodes are siblings, since they are both children of the **HTML** node. The **HEAD** contains two **#comment** nodes, representing lines 5–6. The **TITLE** node

```

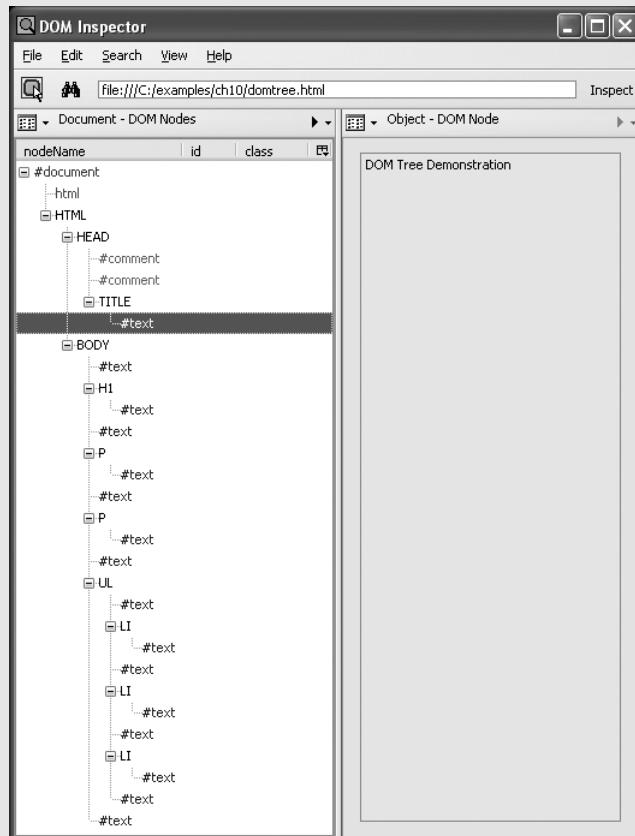
1 <?xml version = "1.0" encoding = "utf-8"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 10.1: domtree.html -->
6 <!-- Demonstration of a document's DOM tree. -->
7 <html xmlns = "http://www.w3.org/1999/xhtml">
8   <head>
9     <title>DOM Tree Demonstration</title>
10   </head>
11   <body>
12     <h1>An XHTML Page</h1>
13     <p>This page contains some basic XHTML elements. We use the Firefox
14       DOM Inspector and the IE Developer Toolbar to view the DOM tree
15       of the document, which contains a DOM node for every element in
16       the document.</p>
17     <p>Here's a list:</p>
18     <ul>
19       <li>One</li>
20       <li>Two</li>
21       <li>Three</li>
22     </ul>
23   </body>
24 </html>
```

**Fig. 10.1 |** Demonstration of a document's DOM tree. (Part I of 3.)

a) The XHTML document is rendered in Firefox.



b) The Firefox DOM inspector displays the document tree in the left panel. The right panel shows information about the currently selected node.



**Fig. 10.1** | Demonstration of a document's DOM tree. (Part 2 of 3.)

- c) The Internet Explorer Web Developer Toolbar displays much of the same information as the DOM inspector in Firefox in a panel at the bottom of the browser window.



**Fig. 10.1 |** Demonstration of a document's DOM tree. (Part 3 of 3.)

has a child text node (#text) containing the text **DOM Tree Demonstration**, visible in the right pane of the DOM inspector when the text node is selected. The **BODY** node contains nodes representing each of the elements in the page. Note that the **LI** nodes are children of the **UL** node, since they are nested inside it.

Also, notice that, in addition to the text nodes representing the text inside the body, paragraphs and list elements, a number of other text nodes appear in the document. These text nodes contain nothing but white space. When Firefox parses an XHTML document into a DOM tree, the white space between sibling elements is interpreted as text and placed inside text nodes. Internet Explorer ignores white space and does not convert it into empty text nodes. If you run this example on your own computer, you will notice that the **BODY** node has a **#comment** child node not present above in both the Firefox and Internet Explorer DOM trees. This is a result of the copyright line at the end of the example file that you downloaded.

This section introduced the concept of DOM nodes and DOM trees. The next section considers DOM nodes in more detail, discussing methods and properties of DOM nodes that allow you to modify the DOM tree of a document using JavaScript.

## 10.3 Traversing and Modifying a DOM Tree

The DOM gives you access to the elements of a document, allowing you to modify the contents of a page dynamically using event-driven JavaScript. This section introduces

properties and methods of all DOM nodes that enable you to traverse the DOM tree, modify nodes and create or delete content dynamically.

Figure 10.2 shows some of the functionality of DOM nodes, as well as two additional methods of the document object. The program allows you to highlight, modify, insert and remove elements.

Lines 117–132 contain basic XHTML elements and content. Each element has an `id` attribute, which is also displayed at the beginning of the element in square brackets. For example, the `id` of the `h1` element in lines 117–118 is set to `bigheading`, and the heading text begins with `[bigheading]`. This allows the user to see the `id` of each element in the page. The body also contains an `h3` heading, several `p` elements, and an unordered list.

A `div` element (lines 133–162) contains the remainder of the XHTML body. Line 134 begins a `form` element, assigning the empty string to the required `action` attribute (because we're not submitting to a server) and returning `false` to the `onsubmit` attribute. When a form's `onsubmit` handler returns `false`, the navigation to the address specified in the `action` attribute is aborted. This allows us to modify the page using JavaScript event handlers without reloading the original, unmodified XHTML.

```

1  <?xml version = "1.0" encoding = "utf-8"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 10.2: dom.html -->
6  <!-- Basic DOM functionality. -->
7  <html xmlns = "http://www.w3.org/1999/xhtml">
8      <head>
9          <title>Basic DOM Functionality</title>
10         <style type = "text/css">
11             h1, h3      { text-align: center;
12                         font-family: tahoma, geneva, sans-serif }
13             p           { margin-left: 5%;
14                         margin-right: 5%;
15                         font-family: arial, helvetica, sans-serif }
16             ul          { margin-left: 10% }
17             a           { text-decoration: none }
18             a:hover     { text-decoration: underline }
19             .nav         { width: 100%;
20                           border-top: 3px dashed blue;
21                           padding-top: 10px }
22             .highlighted { background-color: yellow }
23             .submit       { width: 120px }
24         </style>
25         <script type = "text/javascript">
26             <!--
27             var currentNode; // stores the currently highlighted node
28             var idcount = 0; // used to assign a unique id to new elements
29
30             // get and highlight an element by its id attribute
31             function getById()
32             {

```

**Fig. 10.2** | Basic DOM functionality. (Part 1 of 8.)

```
33             var id = document.getElementById( "gb1" ).value;
34             var target = document.getElementById( id );
35
36             if ( target )
37                 switchTo( target );
38         } // end function byId
39
40         // insert a paragraph element before the current element
41         // using the insertBefore method
42         function insert()
43     {
44             var newNode = createNewNode(
45                 document.getElementById( "ins" ).value );
46             currentNode.parentNode.insertBefore( newNode, currentNode );
47             switchTo( newNode );
48         } // end function insert
49
50         // append a paragraph node as the child of the current node
51         function appendNode()
52     {
53             var newNode = createNewNode(
54                 document.getElementById( "append" ).value );
55             currentNode.appendChild( newNode );
56             switchTo( newNode );
57         } // end function appendNode
58
59         // replace the currently selected node with a paragraph node
60         function replaceCurrent()
61     {
62             var newNode = createNewNode(
63                 document.getElementById( "replace" ).value );
64             currentNode.parentNode.replaceChild( newNode, currentNode );
65             switchTo( newNode );
66         } // end function replaceCurrent
67
68         // remove the current node
69         function remove()
70     {
71             if ( currentNode.parentNode == document.body )
72                 alert( "Can't remove a top-level element." );
73             else
74             {
75                 var oldNode = currentNode;
76                 switchTo( oldNode.parentNode );
77                 currentNode.removeChild( oldNode );
78             } // end else
79         } // end function remove
80
81         // get and highlight the parent of the current node
82         function parent()
83     {
84             var target = currentNode.parentNode;
85         }
```

Fig. 10.2 | Basic DOM functionality. (Part 2 of 8.)

```
86         if ( target != document.body )
87             switchTo( target );
88         else
89             alert( "No parent." );
90     } // end function parent
91
92     // helper function that returns a new paragraph node containing
93     // a unique id and the given text
94     function createNewNode( text )
95     {
96         var newNode = document.createElement( "p" );
97         nodeId = "new" + idcount;
98         ++idcount;
99         newNode.id = nodeId;
100        text = "[" + nodeId + "] " + text;
101        newNode.appendChild(document.createTextNode( text ) );
102        return newNode;
103    } // end function createNewNode
104
105    // helper function that switches to a new currentNode
106    function switchTo( newNode )
107    {
108        currentNode.className = ""; // remove old highlighting
109        currentNode = newNode;
110        currentNode.className = "highlighted"; // highlight new node
111        document.getElementById( "gbi" ).value = currentNode.id;
112    } // end function switchTo
113    // -->
114    </script>
115</head>
116<body onLoad = "currentNode = document.getElementById( 'bigheading' )">
117    <h1 id = "bigheading" class = "highlighted">
118        [bigheading] DHTML Object Model</h1>
119        <h3 id = "smallheading">[smallheading] Element Functionality</h3>
120        <p id = "para1">[para1] The Document Object Model (DOM) allows for
121            quick, dynamic access to all elements in an XHTML document for
122            manipulation with JavaScript.</p>
123        <p id = "para2">[para2] For more information, check out the
124            "JavaScript and the DOM" section of Deitel's
125            <a id = "link" href = "http://www.deitel.com/javascript">
126                [link] JavaScript Resource Center.</a></p>
127        <p id = "para3">[para3] The buttons below demonstrate:(list)</p>
128        <ul id = "list">
129            <li id = "item1">[item1] getElementById and parentNode</li>
130            <li id = "item2">[item2] insertBefore and appendChild</li>
131            <li id = "item3">[item3] replaceChild and removeChild </li>
132        </ul>
133        <div id = "nav" class = "nav">
134            <form onsubmit = "return false" action = "">
135                <table>
136                    <tr>
137                        <td><input type = "text" id = "gbi"
138                            value = "bigheading" /></td>
```

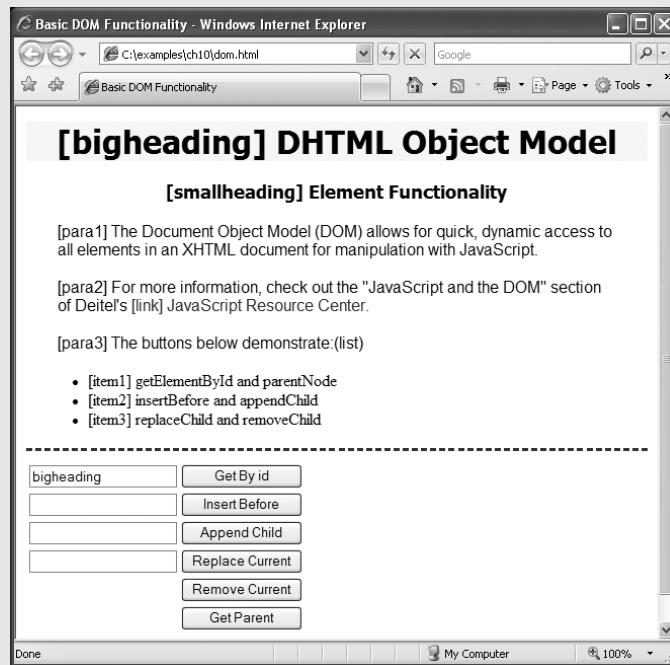
**Fig. 10.2** | Basic DOM functionality. (Part 3 of 8.)

```

139         <td><input type = "submit" value = "Get By id"
140             onclick = "byId()" class = "submit" /></td>
141     </tr><tr>
142         <td><input type = "text" id = "ins" /></td>
143         <td><input type = "submit" value = "Insert Before"
144             onclick = "insert()" class = "submit" /></td>
145     </tr><tr>
146         <td><input type = "text" id = "append" /></td>
147         <td><input type = "submit" value = "Append Child"
148             onclick = "appendNode()" class = "submit" /></td>
149     </tr><tr>
150         <td><input type = "text" id = "replace" /></td>
151         <td><input type = "submit" value = "Replace Current"
152             onclick = "replaceCurrent()" class = "submit" /></td>
153     </tr><tr><td />
154         <td><input type = "submit" value = "Remove Current"
155             onclick = "remove()" class = "submit" /></td>
156     </tr><tr><td />
157         <td><input type = "submit" value = "Get Parent"
158             onclick = "parent()" class = "submit" /></td>
159     </tr>
160   </table>
161 </form>
162 </div>
163 </body>
164 </html>

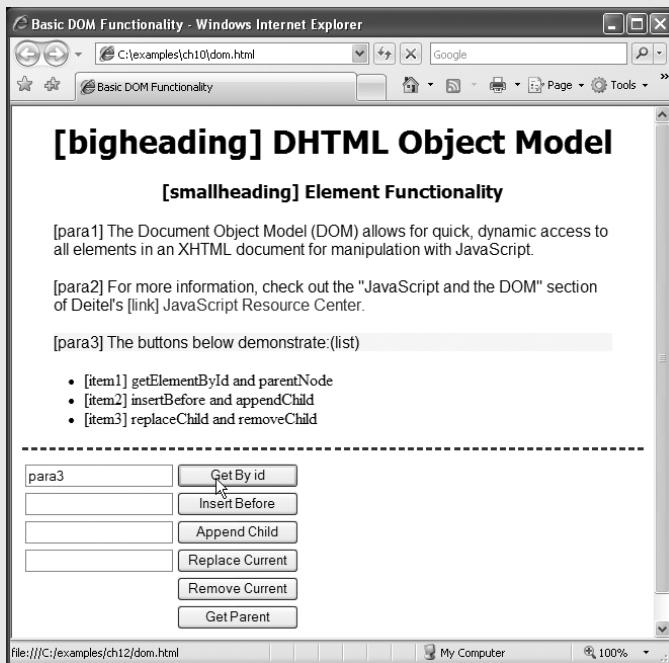
```

a) This is the page when it first loads. It begins with the large heading highlighted.

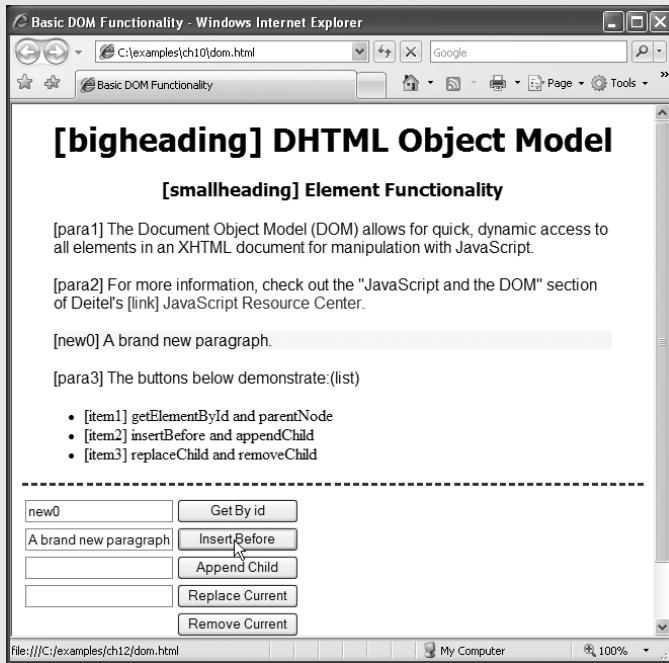


**Fig. 10.2** | Basic DOM functionality. (Part 4 of 8.)

b) This is the document after using the **Get By id** button to select para3.

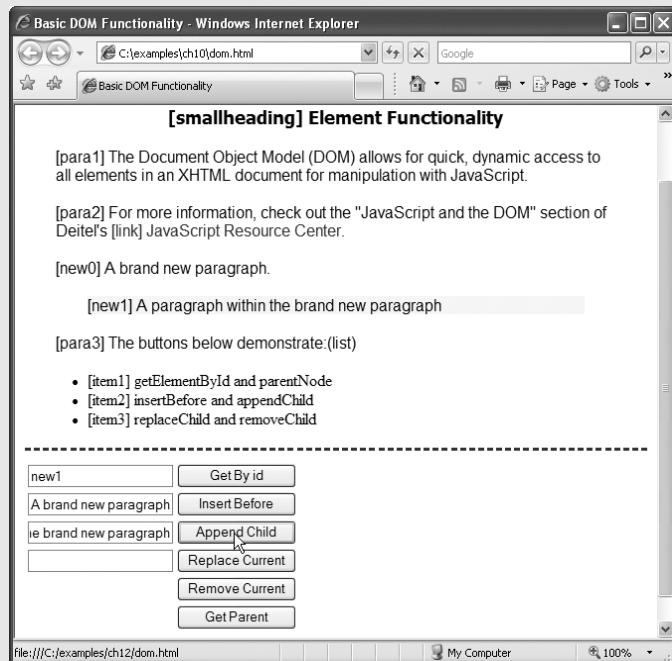


c) This is the document after inserting a new paragraph before the selected one.

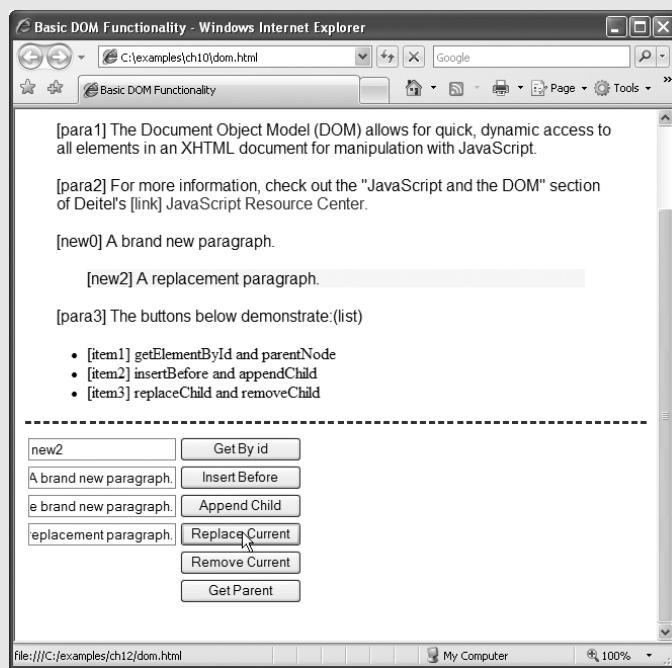


**Fig. 10.2** | Basic DOM functionality. (Part 5 of 8.)

d) Using the Append Child button, a child paragraph is created.

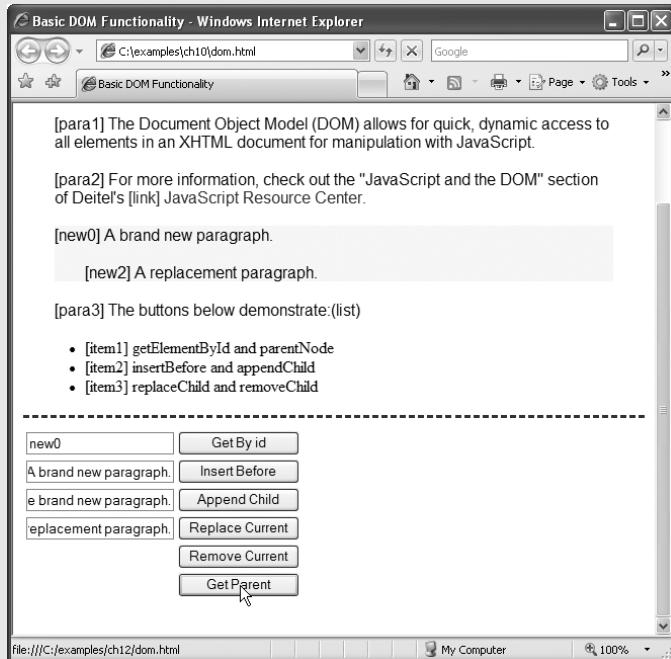


e) The selected paragraph is replaced with a new one.



**Fig. 10.2** | Basic DOM functionality. (Part 6 of 8.)

f) The **Get Parent** button gets the parent of the selected node.



g) Now we select the first list item.

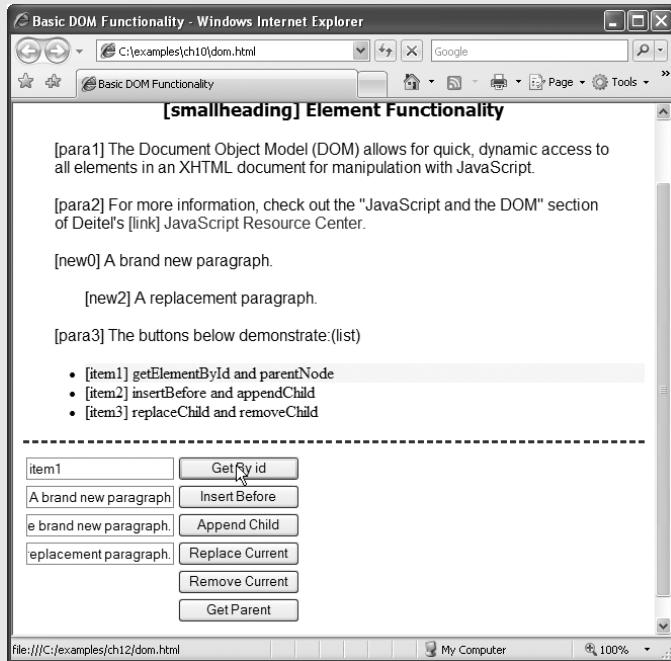
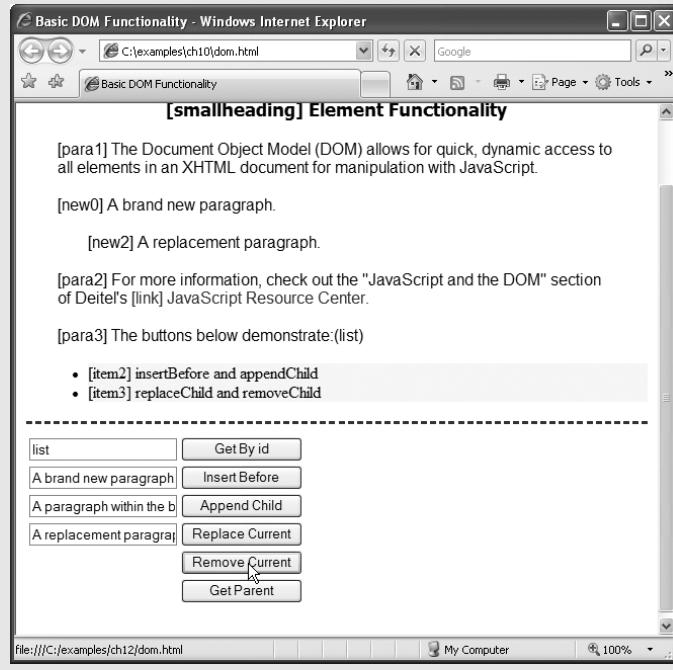


Fig. 10.2 | Basic DOM functionality. (Part 7 of 8.)

h) The **Remove Current** button removes the current node and selects its parent.



**Fig. 10.2** | Basic DOM functionality. (Part 8 of 8.)

A **table** (lines 135–160) contains the controls for modifying and manipulating the elements on the page. Each of the six buttons calls its own event-handling function to perform the action described by its **value**.

The JavaScript code begins by declaring two variables. The variable **currentNode** (line 27) keeps track of the currently highlighted node, because the functionality of the buttons depends on which node is currently selected. The body's **onload** attribute (line 116) initializes **currentNode** to the **h1** element with **id bigheading**. Variable **idcount** (line 28) is used to assign a unique **id** to any new elements that are created. The remainder of the JavaScript code contains event handling functions for the XHTML buttons and two helper functions that are called by the event handlers. We now discuss each button and its corresponding event handler in detail.

### **Finding and Highlighting an Element Using `getElementById` and `className`**

The first row of the table (lines 136–141) allows the user to enter the **id** of an element into the text field (lines 137–138) and click the **Get By Id** button (lines 139–140) to find and highlight the element, as shown in Fig. 10.2(b) and (g). The **onclick** attribute sets the button's event handler to function **byId**.

The **byId** function is defined in lines 31–38. Line 33 uses **getElementById** to assign the contents of the text field to variable **id**. Line 34 uses **getElementById** again to find the element whose **id** attribute matches the contents of variable **id**, and assign it to variable **target**. If an element is found with the given **id**, **getElementById** returns an object rep-

resenting that element. If no element is found, `getElementById` returns `null`. Line 36 checks whether `target` is an object—recall that any object used as a boolean expression is `true`, while `null` is `false`. If `target` evaluates to `true`, line 37 calls the `switchTo` function with `target` as its argument.

The `switchTo` function, defined in lines 106–112, is used throughout the program to highlight a new element in the page. The current element is given a yellow background using the style class `highlighted`, defined in line 22. Line 108 sets the current node's `className` property to the empty string. The `className` property allows you to change an XHTML element's `class` attribute. In this case, we clear the `class` attribute in order to remove the `highlighted` class from the `currentNode` before we highlight the new one.

Line 109 assigns the `newNode` object (passed into the function as a parameter) to variable `currentNode`. Line 110 adds the `highlighted` style class to the new `currentNode` using the `className` property.

Finally, line 111 uses the `id` property to assign the current node's `id` to the input field's `value` property. Just as `className` allows access to an element's `class` attribute, the `id` property controls an element's `id` attribute. While this isn't necessary when `switchTo` is called by `byId`, we will see shortly that other functions call `switchTo`. This line makes sure that the text field's value is consistent with the currently selected node's `id`. Having found the new element, removed the highlighting from the old element, updated the `currentNode` variable and highlighted the new element, the program has finished selecting a new node by a user-entered `id`.

### ***Creating and Inserting Elements Using `insertBefore` and `appendChild`***

The next two table rows allow the user to create a new element and insert it before the current node or as a child of the current node. The second row (lines 141–145) allows the user to enter text into the text field and click the `Insert Before` button. The text is placed in a new paragraph element, which is then inserted into the document before the currently selected element, as in Fig. 10.2(c). The button in lines 143–144 calls the `insert` function, defined in lines 42–48.

Lines 44–45 call the function `createNewNode`, passing it the value of the input field (whose `id` is `ins`) as an argument. Function `createNewNode`, defined in lines 94–103, creates a paragraph node containing the text passed to it. Line 96 creates a `p` element using the `document` object's `createElement` method. The `createElement` method creates a new DOM node, taking the tag name as an argument. Note that while `createElement` creates an element, it does not *insert* the element on the page.

Line 97 creates a unique `id` for the new element by concatenating "new" and the value of `idcount` before incrementing `idcount` in line 98. Line 99 assigns the `id` to the new element. Line 100 concatenates the element's `id` in square brackets to the beginning of `text` (the parameter containing the paragraph's text).

Line 101 introduces two new methods. The document's `createTextNode` method creates a node that can contain only text. Given a string argument, `createTextNode` inserts the string into the text node. In line 101, we create a new text node containing the contents of variable `text`. This new node is then used (still in line 101) as the argument to the `appendChild` method, which is called on the paragraph node. Method `appendChild` is called on a parent node to insert a child node (passed as an argument) after any existing children.

After the `p` element is created, line 102 returns the node to the calling function `insert`, where it is assigned to variable `newNode` in lines 44–45. Line 46 inserts the newly created node before the currently selected node. The `parentNode` property of any DOM node contains the node's parent. In line 46, we use the `parentNode` property of `currentNode` to get its parent.

We call the `insertBefore` method (line 46) on the parent with `newNode` and `currentNode` as its arguments to insert `newNode` as a child of the parent directly before `currentNode`. The general syntax of the `insertBefore` method is

```
parent.insertBefore( newChild, existingChild );
```

The method is called on a parent with the new child and an existing child as arguments. The node `newChild` is inserted as a child of `parent` directly before `existingChild`. Line 47 uses the `switchTo` function (discussed earlier in this section) to update the `currentNode` to the newly inserted node and highlight it in the XHTML page.

The third table row (lines 145–149) allows the user to append a new paragraph node as a child of the current element, demonstrated in Fig. 10.2(d). This features uses a similar procedure to the `insertBefore` functionality. Lines 53–54 in function `appendNode` create a new node, line 55 inserts it as a child of the current node, and line 56 uses `switchTo` to update `currentNode` and highlight the new node.

### ***Replacing and Removing Elements Using `replaceChild` and `removeChild`***

The next two table rows (lines 149–156) allow the user to replace the current element with a new `p` element or simply remove the current element. Lines 150–152 contain a text field and a button that replaces the currently highlighted element with a new paragraph node containing the text in the text field. This feature is demonstrated in Fig. 10.2(e).

The button in lines 151–152 calls function `replaceCurrent`, defined in lines 60–66. Lines 62–63 call `createNewNode`, in the same way as in `insert` and `appendNode`, getting the text from the correct input field. Line 64 gets the parent of `currentNode`, then calls the `replaceChild` method on the parent. The `replaceChild` method works as follows:

```
parent.replaceChild( newChild, oldChild );
```

The `parent`'s `replaceChild` method inserts `newChild` into its list of children in place of `oldChild`.

The Remove Current feature, shown in Fig. 10.2(h), removes the current element entirely and highlights the parent. No text field is required because a new element is not being created. The button in lines 154–155 calls the `remove` function, defined in lines 69–79. If the node's parent is the body element, line 72 alerts an error—the program does not allow the entire body element to be selected. Otherwise, lines 75–77 remove the current element. Line 75 stores the old `currentNode` in variable `oldNode`. We do this to maintain a reference to the node to be removed after we've changed the value of `currentNode`. Line 76 calls `switchTo` to highlight the parent node.

Line 77 uses the `removeChild` method to remove the `oldNode` (a child of the new `currentNode`) from its place in the XHTML document. In general,

```
parent.removeChild( child );
```

looks in `parent`'s list of children for `child` and removes it.

The final button (lines 157–158) selects and highlights the parent element of the currently highlighted element by calling the `parent` function, defined in lines 82–90. Function `parent` simply gets the parent node (line 84), makes sure it is not the body element, (line 86) and calls `switchTo` to highlight it (line 87). Line 89 alerts an error if the parent node is the body element. This feature is shown in Fig. 10.2(f).

This section introduced the basics of DOM tree traversal and manipulation. Next, we introduce the concept of collections, which give you access to multiple elements in a page.

## 10.4 DOM Collections

Included in the Document Object Model is the notion of *collections*, which are groups of related objects on a page. DOM collections are accessed as properties of DOM objects such as the document object or a DOM node. The document object has properties containing the *images collection*, *links collection*, *forms collection* and *anchors collection*. These collections contain all the elements of the corresponding type on the page. Figure 10.3 gives an example that uses the `links` collection to extract all of the links on a page and display them together at the bottom of the page.

```

1  <?xml version = "1.0" encoding = "utf-8"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 10.3: collections.html -->
6  <!-- Using the links collection. -->
7  <html xmlns = "http://www.w3.org/1999/xhtml">
8      <head>
9          <title>Using Links Collection</title>
10         <style type = "text/css">
11             body           { font-family: arial, helvetica, sans-serif }
12             h1            { font-family: tahoma, geneva, sans-serif;
13                           text-align: center }
14             p             { margin: 5% }
15             p a           { color: #aa0000 }
16             .links        { font-size: 14px;
17                           text-align: justify;
18                           margin-left: 10%;
19                           margin-right: 10% }
20             .link a        { text-decoration: none }
21             .link a:hover { text-decoration: underline }
22         </style>
23         <script type = "text/javascript">
24             <!--
25             function processlinks()
26             {
27                 var linkslist = document.links; // get the document's links
28                 var contents = "Links in this page:\n<br />| ";
29
30                 // concatenate each link to contents
31                 for ( var i = 0; i < linkslist.length; i++ )
32                 {

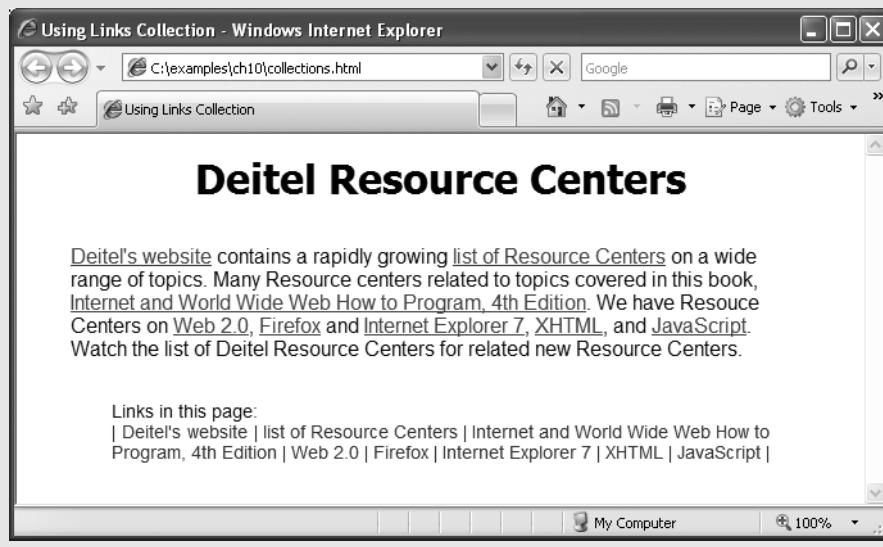
```

**Fig. 10.3 |** Using the `links` collection. (Part I of 2.)

```

33         var currentLink = linkslist[ i ];
34         contents += "<span class = 'link'>" +
35             currentLink.innerHTML.link( currentLink.href ) +
36             "</span> | ";
37     } // end for
38
39     document.getElementById( "links" ).innerHTML = contents;
40 } // end function processlinks
41 // -->
42 </script>
43 </head>
44 <body onload = "processlinks()">
45     <h1>Deitel Resource Centers</h1>
46     <p><a href = "http://www.deitel.com/">Deitel's website</a> contains
47         a rapidly growing
48         <a href = "http://www.deitel.com/ResourceCenters.html">list of
49             Resource Centers</a> on a wide range of topics. Many Resource
50             centers related to topics covered in this book,
51             <a href = "http://www.deitel.com/iw3htp4">Internet and World Wide
52                 Web How to Program, 4th Edition</a>. We have Resouce Centers on
53             <a href = "http://www.deitel.com/Web2.0">Web 2.0</a>,
54             <a href = "http://www.deitel.com/Firefox">Firefox</a> and
55             <a href = "http://www.deitel.com/IE7">Internet Explorer 7</a>,
56             <a href = "http://www.deitel.com/XHTML">XHTML</a>, and
57             <a href = "http://www.deitel.com/JavaScript">JavaScript</a>.
58             Watch the list of Deitel Resource Centers for related new
59             Resource Centers.</p>
60     <div id = "links" class = "links"></div>
61 </body>
62 </html>

```



**Fig. 10.3** | Using the `links` collection. (Part 2 of 2.)

The XHTML body contains a paragraph (lines 46–59) with links at various places in the text and an empty div (line 60) with id `links`. The body's `onLoad` attribute specifies that the `processLinks` method is called when the body finishes loading.

Method `processLinks` declares variable `linksList` (line 27) to store the document's `links` collection, which is accessed as the `links` property of the `document` object. Line 28 creates the string (`contents`) that will contain all the document's links, to be inserted into the `links` div later. Line 31 begins a `for` statement to iterate through each link. To find the number of elements in the collection, we use the collection's ***length* property**.

Line 33 inside the `for` statement creates a variable (`currentLink`) that stores the current link. Note that we can access the collection stored in `linksList` using indices in square brackets, just as we did with arrays. DOM collections are stored in objects which have only one property and two methods—the `length` property, the ***item method*** and the ***namedItem method***. The `item` method—an alternative to the square bracketed indices—can be used to access specific elements in a collection by taking an index as an argument. The `namedItem` method takes a name as a parameter and finds the element in the collection, if any, whose `id` attribute or `name` attribute matches it.

Lines 34–36 add a `span` element to the `contents` string containing the current link. Recall that the `link` method of a string object returns the string as a link to the URL passed to the method. Line 35 uses the `link` method to create an `a` (anchor) element containing the proper text and `href` attribute.

Notice that variable `currentLink` (a DOM node representing an `a` element) has a specialized ***href property*** to refer to the link's `href` attribute. Many types of XHTML elements are represented by special types of nodes that extend the functionality of a basic DOM node. Line 39 inserts the contents into the empty div with id "links" (line 60) in order to show all the links on the page in one location.

Collections allow easy access to all elements of a single type in a page. This is useful for gathering elements into one place and for applying changes across an entire page. For example, the `forms` collection could be used to disable all form inputs after a submit button has been pressed to avoid multiple submissions while the next page loads. The next section discusses how to dynamically modify CSS styles using JavaScript and DOM nodes.

## 10.5 Dynamic Styles

An element's style can be changed dynamically. Often such a change is made in response to user events, which we discuss in Chapter 11. Such style changes can create many effects, including mouse hover effects, interactive menus, and animations. Figure 10.4 is a simple example that changes the `background-color` style property in response to user input.

```

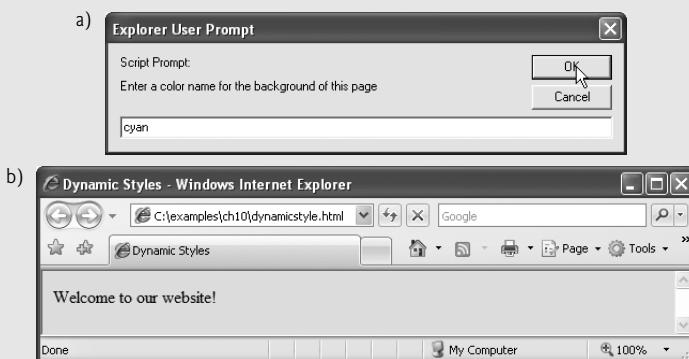
1 <?xml version = "1.0" encoding = "utf-8"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 10.4: dynamicstyle.html -->
6 <!-- Dynamic styles. -->
7 <html xmlns = "http://www.w3.org/1999/xhtml">
8   <head>
```

**Fig. 10.4 |** Dynamic styles. (Part 1 of 2.)

```

9      <title>Dynamic Styles</title>
10     <script type = "text/javascript">
11       <!--
12       function start()
13       {
14         var inputColor = prompt( "Enter a color name for the " +
15           "background of this page", "" );
16         document.body.style.backgroundColor = inputColor;
17       } // end function start
18       // -->
19     </script>
20   </head>
21   <body id = "body" onload = "start()">
22     <p>Welcome to our website!</p>
23   </body>
24 </html>

```



**Fig. 10.4** | Dynamic styles. (Part 2 of 2.)

Function `start` (lines 12–17) prompts the user to enter a color name, then sets the background color to that value. [Note: An error occurs if the value entered is not a valid color.] We refer to the background color as `document.body.style.backgroundColor`—the **body** *property* of the `document` object refers to the `body` element. We then use the `style` *property* (a *property* of most XHTML elements) to set the `background-color` CSS *property*. This is referred to as `backgroundColor` in JavaScript—the hyphen is removed to avoid confusion with the subtraction (-) operator. This naming convention is consistent for most CSS *properties*. For example, `borderWidth` correlates to the `border-width` CSS *property*, and `fontFamily` correlates to the `font-family` CSS *property*. In general, CSS *properties* are accessed in the format `node.style.styleable`.

Figure 10.5 introduces the `setInterval` and `clearInterval` methods of the `window` object, combining them with dynamic styles to create animated effects. This example is a basic image viewer that allows you to select a Deitel book cover and view it in a larger size. When one of the thumbnail images on the right is clicked, the larger version grows from the top-left corner of the main image area.

The `body` (lines 66–85) contains two `div` elements, both floated `left` using styles defined in lines 14 and 17 in order to present them side by side. The `left` `div` contains the full-size image `iw3htp4.jpg`, the cover of this book, which appears when the page loads.

The right `div` contains six thumbnail images which respond to the click event by calling the `display` method and passing it the filename of the corresponding full-size image.

The `display` function (lines 46–62) dynamically updates the image in the left `div` to the one corresponding to the user's click. Lines 48–49 prevent the rest of the function from executing if `interval` is defined (i.e., an animation is in progress.) Line 51 gets the left `div` by its `id`, `imgCover`. Line 52 creates a new `img` element. Lines 53–55 set its `id` to `imgCover`, set its `src` to the correct image file in the `fullsize` directory, and set its required `alt` attribute. Lines 56–59 do some additional initialization before beginning the animation in line 61. To create the growing animation effect, lines 57–58 set the image `width` and `height` to 0. Line 59 replaces the current `bigImage` node with `newNode` (created in line 52), and line 60 sets `count`, the variable that controls the animation, to 0.

Line 61 introduces the `window` object's ***setInterval method***, which starts the animation. This method takes two parameters—a statement to execute repeatedly, and an integer specifying how often to execute it, in milliseconds. We use `setInterval` to call

```

1 <?xml version = "1.0" encoding = "utf-8"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 10.5: coverviewer.html -->
6 <!-- Dynamic styles used for animation. -->
7 <html xmlns = "http://www.w3.org/1999/xhtml">
8   <head>
9     <title>Deitel Book Cover Viewer</title>
10    <style type = "text/css">
11      .thumbs { width: 192px;
12                  height: 370px;
13                  padding: 5px;
14                  float: left }
15      .mainimg { width: 289px;
16                  padding: 5px;
17                  float: left }
18      .imgCover { height: 373px }
19      img { border: 1px solid black }
20    </style>
21    <script type = "text/javascript">
22      <!--
23      var interval = null; // keeps track of the interval
24      var speed = 6; // determines the speed of the animation
25      var count = 0; // size of the image during the animation
26
27      // called repeatedly to animate the book cover
28      function run()
29      {
30        count += speed;
31
32        // stop the animation when the image is large enough
33        if ( count >= 375 )
34        {

```

Fig. 10.5 | Dynamic styles used for animation. (Part 1 of 4.)

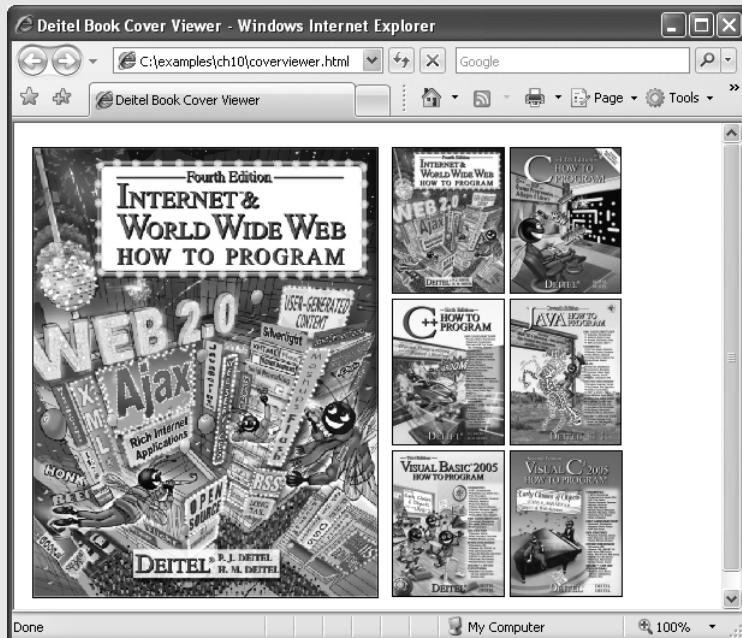
```

35         window.clearInterval( interval );
36         interval = null;
37     } // end if
38
39     var bigImage = document.getElementById( "imgCover" );
40     bigImage.style.width = .7656 * count + "px";
41     bigImage.style.height = count + "px";
42 } // end function run
43
44 // inserts the proper image into the main image area and
45 // begins the animation
46 function display( imgfile )
{
47     if ( interval )
        return;
50
51     var bigImage = document.getElementById( "imgCover" );
52     var newNode = document.createElement( "img" );
53     newNode.id = "imgCover";
54     newNode.src = "fullsize/" + imgfile;
55     newNode.alt = "Large image";
56     newNode.className = "imgCover";
57     newNode.style.width = "0px";
58     newNode.style.height = "0px";
59     bigImage.parentNode.replaceChild( newNode, bigImage );
60     count = 0; // start the image at size 0
61     interval = window.setInterval( "run()", 10 ); // animate
62 } // end function display
63 // -->
64 </script>
65 </head>
66 <body>
67     <div id = "mainimg" class = "mainimg">
68         <img id = "imgCover" src = "fullsize/iw3http4.jpg"
69             alt = "Full cover image" class = "imgCover" />
70     </div>
71     <div id = "thumbs" class = "thumbs" >
72         <img src = "thumbs/iw3http4.jpg" alt = "iw3http4"
73             onclick = "display( 'iw3http4.jpg' )" />
74         <img src = "thumbs/chtp5.jpg" alt = "chtp5"
75             onclick = "display( 'chtp5.jpg' )" />
76         <img src = "thumbs/cpphttp6.jpg" alt = "cpphttp6"
77             onclick = "display( 'cpphttp6.jpg' )" />
78         <img src = "thumbs/jhttp7.jpg" alt = "jhttp7"
79             onclick = "display( 'jhttp7.jpg' )" />
80         <img src = "thumbs/vbhttp3.jpg" alt = "vbhttp3"
81             onclick = "display( 'vbhttp3.jpg' )" />
82         <img src = "thumbs/vcsharphttp2.jpg" alt = "vcsharphttp2"
83             onclick = "display( 'vcsharphttp2.jpg' )" />
84     </div>
85 </body>
86 </html>

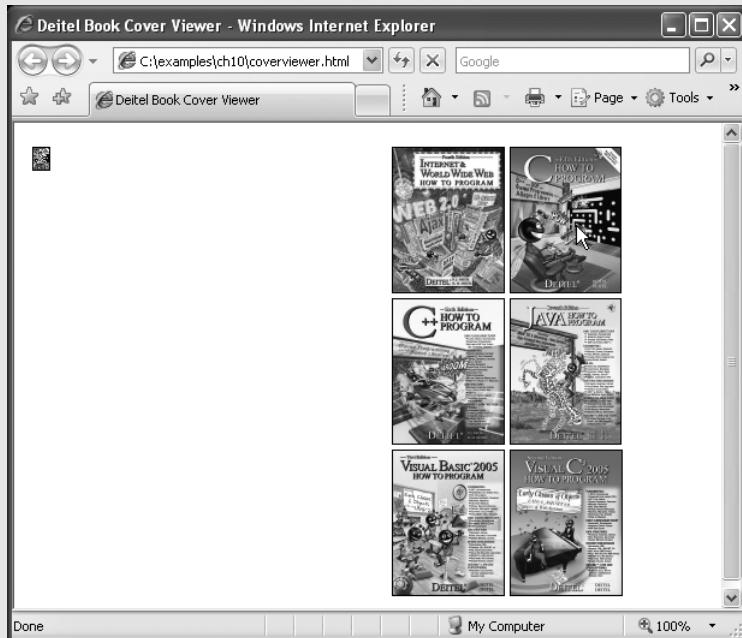
```

Fig. 10.5 | Dynamic styles used for animation. (Part 2 of 4.)

a) The cover viewer page loads with the cover of this book.

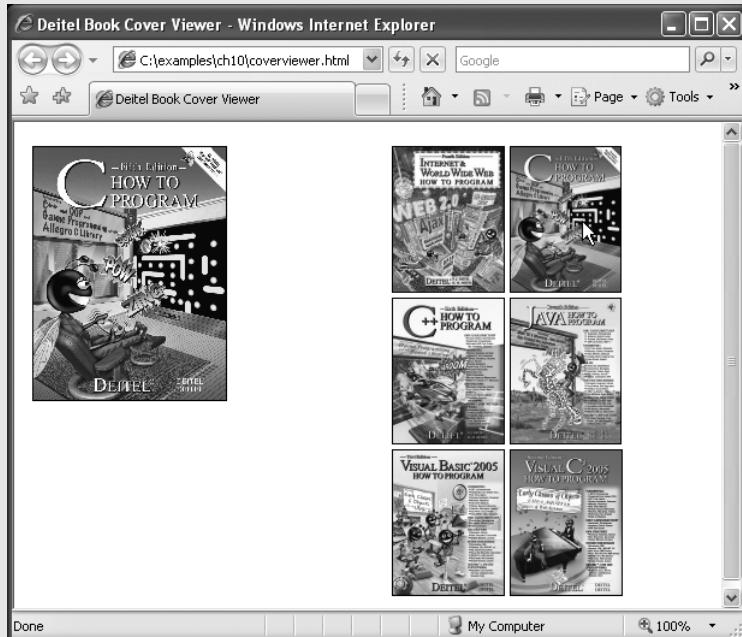


b) When the user clicks the thumbnail of *C How to Program*, the full-size image begins growing from the top-left corner of the window.

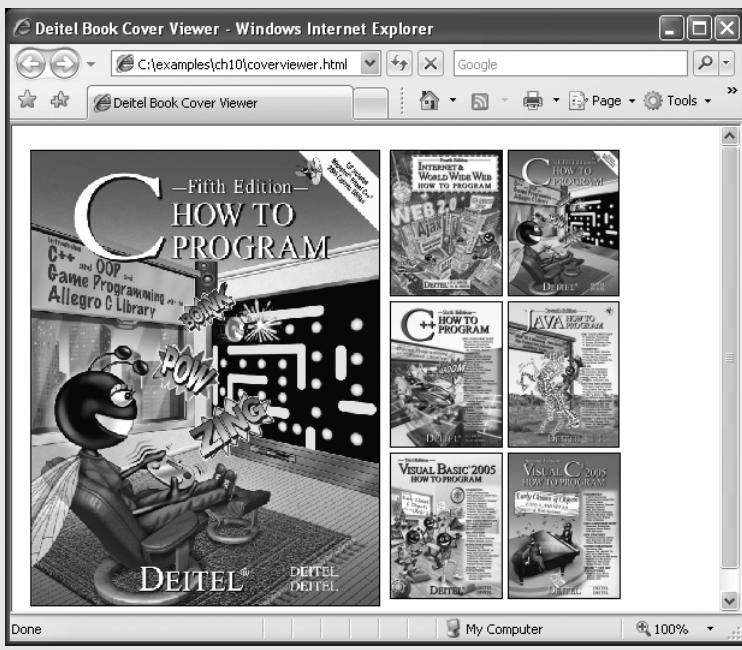


**Fig. 10.5** | Dynamic styles used for animation. (Part 3 of 4.)

c) The cover continues to grow.



d) The animation finishes when the cover reaches its full size.



**Fig. 10.5** | Dynamic styles used for animation. (Part 4 of 4.)

function run every 10 milliseconds. The `setInterval` method returns a unique identifier to keep track of that particular interval—we assign this identifier to the variable `interval`. We use this identifier to stop the animation when the image has finished growing.

The `run` function, defined in lines 28–42, increases the height of the image by the value of `speed` and updates its width accordingly to keep the aspect ratio consistent. Because the `run` function is called every 10 milliseconds, this increase happens repeatedly to create an animated growing effect. Line 30 adds the value of `speed` (declared and initialized to 6 in line 24) to `count`, which keeps track of the animation's progress and dictates the current size of the image. If the image has grown to its full height (375), line 35 uses the window's `clearInterval` method to stop the repetitive calls of the `run` method. We pass to `clearInterval` the interval identifier (stored in `interval`) that `setInterval` created in line 61. Although it seems unnecessary in this script, this identifier allows the script to keep track of multiple intervals running at the same time and to choose which interval to stop when calling `clearInterval`.

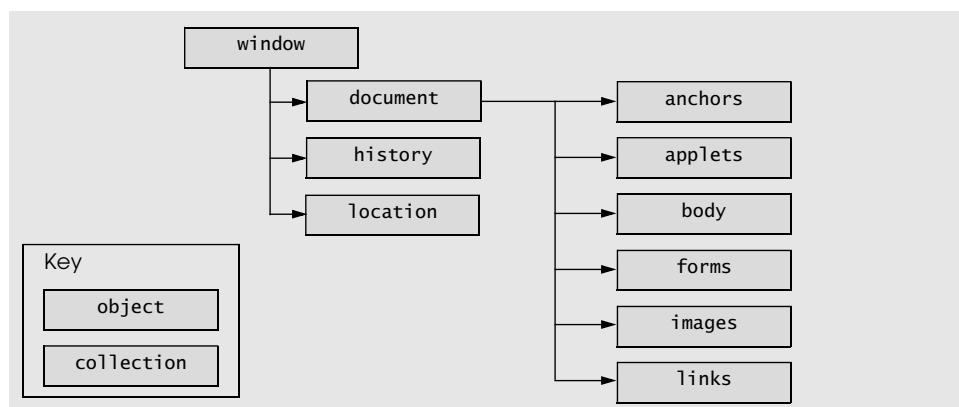
Line 39 gets the image and lines 40–41 set its `width` and `height` CSS properties. Note that line 40 multiplies `count` by a scaling factor of .7656 in order to keep the ratio of the image's dimensions consistent with the actual dimensions of the image. Run the code example and click on a thumbnail image to see the full animation effect.

This section demonstrated the concept of dynamically changing CSS styles using JavaScript and the DOM. We also discussed the basics of how to create scripted animations using `setInterval` and `clearInterval`.

## 10.6 Summary of the DOM Objects and Collections

As you've seen in the preceding sections, the objects and collections in the W3C DOM give you flexibility in manipulating the elements of a web page. We've shown how to access the objects in a page, how to access the objects in a collection, and how to change element styles dynamically.

The W3C DOM allows you to access every element in an XHTML document. Each element in a document is represented by a separate object. The diagram in Fig. 10.6 shows many of the important objects and collections provided by the W3C DOM. Figure 10.7 provides a brief description of each object and collection in Fig. 10.6.



**Fig. 10.6** | W3C Document Object Model.

Object or collection	Description
<i>Objects</i>	
window	Represents the browser window and provides access to the window's document object. Also contains history and location objects.
document	Represents the XHTML document rendered in a window. Provides access to every element in the document and allows dynamic modification of it. Contains collections for accessing all elements of a given type.
body	Provides access to the body element of an XHTML document.
history	Keeps track of the sites visited by the browser user. The object provides a script programmer with the ability to move forward and backward through the visited sites.
location	Contains the URL of the rendered document. When this object is set to a new URL, the browser immediately navigates to the new location.
<i>Collections</i>	
anchors	Collection contains all the anchor elements (a) that have a name or id attribute. The elements appear in the collection in the order in which they were defined in the XHTML document.
forms	Contains all the form elements in the XHTML document. The elements appear in the collection in the order in which they were defined in the XHTML document.
images	Contains all the img elements in the XHTML document. The elements appear in the collection in the order in which they were defined in the XHTML document.
links	Contains all the anchor elements (a) with an href property. The elements appear in the collection in the order in which they were defined in the XHTML document.

**Fig. 10.7** | Objects and collections in the W3C Document Object Model.

For a complete reference on the W3C Document Object Model, see the DOM Level 3 recommendation from the W3C at <http://www.w3.org/TR/DOM-Level-3-Core/>. The DOM Level 2 HTML Specification (the most recent HTML DOM standard), available at <http://www.w3.org/TR/DOM-Level-2-HTML/>, describes additional DOM functionality specific to HTML, such as objects for various types of XHTML elements. Keep in mind that not all web browsers implement all features included in the specification.

## 10.7 Web Resources

[www.deitel.com/javascript/](http://www.deitel.com/javascript/)

The Deitel JavaScript Resource Center contains links to some of the best JavaScript resources on the web. There you'll find categorized links to JavaScript tools, code generators, forums, books, libraries, frameworks, tutorials and more. Check out the section specifically dedicated to the Document Object Model. Be sure to visit the related Resource Centers on XHTML ([www.deitel.com/xhtml/](http://www.deitel.com/xhtml/)) and CSS 2.1 ([www.deitel.com/css21/](http://www.deitel.com/css21/)).

# Index

## Symbols

- range separator 621  
-- operator 170, 171  
! logical NOT or logical negation operator 198, 199  
!= not equal to operator 148, 619  
" double quote 609  
# single-line comment 609  
\$ for PHP variables 609  
\$ metacharacter 621  
\$\$ notation for variable variables 647  
\$\_COOKIE superglobal 623, 638, 640  
\$\_ENV superglobal 623  
\$\_GET superglobal 623, 626  
\$\_POST superglobal 623, 626  
\$\_SERVER superglobal 623  
\$\_GLOBALS superglobal 623  
% operator 147  
% SQL wildcard character 593  
%= operator 169  
%> Ruby delimiter 663  
&& logical AND operator 197, 198  
&# entity reference 80  
&copy; entity reference 80  
&delta; entity reference (MathML) 375  
&fract; entity reference 81  
&int; entity reference (MathML) 375  
&lt; special character 79, 81  
&nbsp; entity reference 395  
\* multiplication 384  
\* multiplication operator 147  
\* quantifier 621  
\* SQL wildcard character 592  
\*/ ending comment delimiter 609  
\*/ multiline comment delimiter 143  
\*= multiplication assignment operator 169, 612  
+ addition operator 147  
+ quantifier 621  
+ string concatenation 144  
++ increment operator 169, 169, 170, 171  
. PHP concatenation operator 612  
. PHP metacharacter 621  
/ forward slash in end tags 349  
/ XPath root selector 380  
/\* beginning comment delimiter 609  
/\* multiline comment delimiter 143  
// single-line comment 609  
/= division assignment operator 169  
< less than operator 619  
<! --> XML comment tags 352  
<%> 663  
<%-- --> ASP.NET comment 703  
<%\$ %> ASP.NET expression syntax 760  
<=% 663

<%@ and %> for delimiting ASP.NET directives 703  
<= less than or equal to operator 148, 619  
<> angle brackets for XML elements 349  
<? and ?> XML processing instruction delimiters 378  
<?php ?> PHP script delimiters 608  
= assignment operator 144  
-= subtraction assignment operator 169  
== is equal to operator 148, 619  
=> for Ruby hashes 657  
=> PHP associative array operator 617  
> greater than operator 619  
=> greater than or equal to operator 148, 619  
? PHP quantifier 621  
?: conditional operator 157  
@ XPath attribute symbol 380  
@@ for declaring Ruby class variable 659  
[[:<:]] bracket expression 621  
[[:>:]] bracket expression 621  
[] for array indexing 615  
[] for bracket expressions 621  
\ escape character 630  
\\" double-quote escape sequence 140  
\\\ backslash escape sequence 140  
\' single quote escape sequence 140  
\n newline escape sequence 140  
\r carriage return escape sequence 140  
\t tab escape sequence 140  
^ metacharacter 621  
\_ SQL wildcard character 593  
{m,n} quantifier 621  
{n,} quantifier 621  
{n} quantifier 621  
|| logical OR operator 197, 199  
' single quote 621

## Numerics

37Signals 653

## A

a (anchor) element 73, 79  
    href attribute 73, 98  
    href property 315, 322  
    target attribute 282  
abandoned blog 18  
abbreviating assignment expressions 169  
abort method of the XMLHttpRequest object 423  
About.com (content network) 13, 42  
abs 260  
absolute addressing (XPath) 380  
absolute attribute value (position) 113  
absolute positioning 111, 113, 713, 817  
Absolute positioning of elements 112  
absolute value 260  
absolute-length measurement 109, 109, 128  
abstraction 277  
**AbstractPageBean** class 806, 816, 937  
Accept request header 438  
access rule in ASP.NET 770  
access the objects in a collection 321  
accessible web pages 77  
**Accordion** element 535  
    creationPolicy attribute 535  
    selectedIndex attribute 537  
action 135, 801  
action attribute of a form element 626  
action attribute of form element 90  
action attribute of XHTML element form 708, 812  
action oriented 259  
action/decision model of programming 156  
action\_name method 665  
**ActionController** 661, 662  
ActionScript 3.0 32, 453, 495, 518  
ActionScript 3.0 elements 516  
ActionScript Virtual Machine 519  
**ActionView** 661, 662  
active database connection 603  
active tool 454  
**ActiveRecord** 661, 666, 671, 674  
**ActiveXObject**  
    load method 394  
**ActiveXObject** from Internet Explorer 394  
Ad XML element in an AdRotator advertisement file 724  
Adaptive Path 32, 417  
Add method of class Hashtable 743  
Add method of class  
    HttpSessionState 748  
addChild 496  
AddChild element 530  
addEventListerner function 471, 491, 496  
Adding a new message without reloading the page 690  
adding a web service reference to a project in Netbeans 905  
adding an intermediate color to a gradient 488  
Adding background images and indentation 115  
adding Frame Labels using the Property Inspector 479  
adding sound to a button 466  
adding text to a button 461

addition assignment operator ( $+=$ ) 169  
 addition operator (+) 147  
 addition script 145  
 ADF Faces 802  
 adjusting line spacing with the **Format Options** panel 478  
 ADO.NET 586, 602  
 ADO.NET 2.0 602  
 ADO.NET disconnected object model 603  
 ADO.NET object model 602  
 Adobe 6, 32, 44  
     Adobe Integrated Runtime (AIR) 34  
         Apollo 34  
         Flash 32  
         Flash CS3 xix, xx, 453, 493, 691  
         Flex xix, 6, 32, 518  
         Flex Platform 518  
         Flex Style Explorer 557  
 Adobe Flex  
     ActionScript Virtual Machine 519  
     E4X (ECMAScript for XML) 531  
     Flex Charting 519  
     Flex SDK (Software Development Kit) 519  
     Flex Software Development Kit (SDK) 519  
     mx prefix 522  
 Adobe Integrated Runtime (AIR) 566  
 AdRotator ASP.NET Web control 721, 722  
 AdSense advertising program (Google) 9  
 AdTech conference 7  
 advanced event models 324  
 advertisement 737  
 Advertisements XML element in an AdRotator advertisement file 724  
 advertising exchange 41  
     DoubleClick 41  
     Right Media Exchange 41  
 AdWords advertising program (Google) 9  
 Affiliate Marketing Summit conference 7  
 affiliate networks 40, 41  
     Commission Junction 40, 41  
     LinkShare 40, 41  
 affiliate programs 40  
     Amazon Associates program 40  
 Afternic.com 41  
 Aggregate Knowledge 42  
 agile software development 30  
 AIR (Adobe Integrated Runtime) 34, 566  
 Ajax 412, 414, 417, 653  
 Ajax (Asynchronous JavaScript and XML) 32  
 Ajax components (Java BluePrints) 872  
 Ajax-enabled component libraries 859  
 Ajax-enabled JSF components 872  
     AutoComplete Text Field 873  
     Buy Now Button 872  
     Map Viewer component 872, 882  
     Popup Calendar 872  
     Progress Bar 872  
     Rating 872  
     Rich Textarea Editor 872  
     Select Value Text Field 872  
 Ajax toolkits 412  
 Ajax web application 414  
 AjaxWorld Expo 7  
 alert dialog 139, 216  
 alert dialog displaying multiple lines 139  
 alert dialog displaying multiple lines 139  
 alert method of window object 139  
 Alert.show function 537  
     closeHandler parameter 537  
     flags parameter 537  
     parent parameter 537  
     text parameter 537  
     title parameter 537  
 Alexa 44  
 algebraic equation marked up with MathML and displayed in the Amaya browser 373  
 alias 581  
 All Things Digital conference 7  
 a11 XML Schema element 371  
 AllPeers 42  
 alpha value 487  
 alt attribute of img element 77  
 Alt key 333  
 AlternateText element in an AdRotator advertisement file 724  
 altKey property of an event object 333  
 Always On conference 7  
 Amaya (W3C browser) 372  
 Amazon 13, 14, 25, 34, 36, 40, 43  
     Associates program 6, 40  
     Auctions 42  
     E-Commerce Services (ECS) 36  
     Elastic Compute Cloud (EC2) 36  
     historical pricing data 36  
     Mechanical Turk 36  
     Simple Queue Service (SQS) 36  
     Simple Storage Service (S3) 36  
     Web Services 6  
 Amie Street 42  
 ancestor element 106  
 ancestor node 385  
 anchor 73, 470  
 anchor 271  
 anchor (a) element 263  
 anchor method 270, 271  
 anchor method of the String object 263, 271  
 anchors collection of the document object 313  
 AND 599  
 AND SQL keyword 599  
 Anderson, Chris (author of *The Long Tail*) 6  
 angle bracket (<>) for XML elements 349  
 angle brackets 552  
 animated button 456  
 Annotations  
     @Resource 919  
     @WebMethod 899  
     @WebParam 899  
     @WebService 899  
     dependency injection 919  
 anonymous function 343  
 anonymous String object 261  
 anonymous user 773  
 AOL 38  
 Apache 31, 43, 654  
 Apache Derby 604, 605  
 Apache HTTP Server 570, 581  
 Apache httpd.conf Configuration File 582  
 Apache Software Foundation 31, 581  
 Apache Tomcat 899  
 API 23, 28, 34  
 api.rubyonrails.org 653, 675  
 Apollo 34  
 appendChild method of a DOM node 311  
 appendChild method of a Node 396  
 appendChild method of the document object 427  
 appendRow method of class CachedRowSetDataProvider 871  
 Apple 22, 25  
     iPhone 22  
     iPod 25  
     iTunes 25  
     Safari web browser 567  
 Application element 519  
 application scope 803  
 Application servers  
     Apache Tomcat 899  
     BEA Weblogic Server 899  
     GlassFish 899  
     JBoss Application Server 899  
     Sun Java System Application Server 899  
 ApplicationBean 803  
 ApplicationControlBar element 535  
 ApplicationController 662  
 architecture of participation 4, 19  
 argument 135, 207, 260  
 arithmetic assignment operators:  $+=$ ,  $-=$ ,  $*=$ ,  $/=$  and  $\% =$  169  
 arithmetic calculation 147  
 arithmetic operator 147  
 array 614  
     elements 233  
     index 233  
     name 233  
     position number 233  
     subscript 233  
     zeroth element 233  
 array data structure 233  
 array function 615  
 array manipulation 615  
 Array object 204, 234, 235, 237, 238, 257, 258  
     join method 246  
     length property 233, 237, 240, 258  
     sort method 247, 248, 259  
 array of strings containing the token 269  
 Array with 12 elements 234  
 ArrayCollection  
     filterFunction attribute 540  
 ArrayCollection element 524  
 ArrayList class 920  
 Arrays 657  
 Arrays and hashes in Ruby 657  
 arrays.php 615  
 article.xml displayed by Internet Explorer 353  
 ArtViper 26  
 as keyword 618  
 AS SQL keyword 784  
 ascending order 384  
     ASC in SQL 594, 595  
 ASCII character set 226, 248

Ask 7, 10, 43  
**AskJeeves.com** 10  
**asp**  
  tag prefix 704  
**ASP.NET** xx, 413, 473, **701**  
  Page event life cycle 707  
**ASP.NET 2.0** 701  
**ASP.NET Ajax** xx, 33, 793  
**ASP.NET Ajax Control Toolkit** 793  
**ASP.NET comment** 703  
**ASP.NET expression** 760  
**ASP.NET login control** 765  
**ASP.NET server control** 701  
**ASP.NET validation control** 726  
**ASP.NET Web Service** project in Visual Web Developer 954  
**ASP.NET Web Site** in Visual Web Developer 708  
**ASPx** file 701  
**ASPx** file that displays the Web server's time 702  
**.aspx** filename extension 701  
assigning more than one class to an XHTML element 119  
assignment 144  
assignment operator 144, 148, 171  
assignment operator = associates right to left 152  
associate from left to right 152, 172  
associate from right to left 152, 172  
association (in the UML) 259  
**association** method 675  
associative array 617  
associativity of operators 152, 172, 201  
asterisk (\*) indicates multiplication 147  
asterisk (\*) occurrence indicator 363  
asterisk (\*) SQL wildcard character 592  
**Asynchronous JavaScript and XML (Ajax)** 32  
asynchronous page updates (Ajax) 417  
asynchronous programming model 519  
asynchronous request 414  
Atom 18, 39  
attention economy 8, 29  
**ATTLIST** attribute-list declaration (DTD) 363  
**Attr** object 396  
attribute 259  
  in the UML 258  
**attribute** element 371  
attribute in XML 356  
attribute-list declaration 363  
attribute node 380  
attribute value in XML 356  
attributes 70  
attributes (data) 135  
**attributes** property of a Node 396  
aural media type 122  
authenticating a user 768  
**authentication** element in **Web.config** 773  
author 106  
author style 126  
author style overriding user style 128  
**authorISBN** table of books database 588  
authorization element in **Web.config** 773  
authors table of books database 588

**AutoComplete Text Field** JSF component 873  
**AutoEventWireup** attribute of ASP.NET page 703  
autoincremented 588, 598  
**AutoPostBack** property of a DropDownList ASP.NET control 788  
**avatar** 5, 22  
average calculation 164

**B**

**B2B** (business-to-business) transactions 893  
**b5media** (blog network) 13, 18  
**background-attachment** property 116  
**Background Color** 456, 457  
**background-color** property 115  
**background-image** property 115, 116  
**background-position** property 115  
**background-repeat** property 116  
Backpack (37Signals) 30  
backslash (\) escape character 137  
badge 28  
Ba11 505  
**Bandwidth Profiler** 490  
banner ad 40  
base 227  
base 2 logarithm of  $e$  261  
**base** attribute of element **extension** 371  
base case 227  
base  $e$  260  
base of a natural logarithm 261  
base type (XML Schema) 371  
**baseAtZero** attribute 566  
Basecamp 653  
Basecamp (37Signals) 30, 33  
basic XHTML colors 338  
Battelle, John (author of *The Search: How Google and Its Rivals Rewrote the Rules of Business and Transformed Our Culture*) 8  
Searchblog 41  
BEA Weblogic Server 899  
Bebo 5, 43  
Beck, John (author of *The Attention Economy*) 8  
behavior 258  
behaviors (methods) 135  
behaviors in the UML 258  
**belongs\_to** association 675  
Berners-Lee, Tim 46  
binary format 227  
binary operator 144, 144, 147, 199  
**[Bindable]** metadata tag 538  
binding a JSF Table to a database table 862  
**binding** attribute 805, 811  
**Binding** element 523  
**BindingProvider** interface 921  
  **get RequestContext** method 921  
bitmapped graphics 458  
BitTorrent 42  
**Black and white** button 458  
black hat SEO 11  
blackjack 916

blank line 207  
**blink** value (**text-decoration**) 107  
Blinkx 10  
Blish 43  
block 159, 166, 195, 196, 225  
block dimension 114  
block display value 124  
block-level element 121  
**Blocker** 501  
block-level element 114  
blog 16, 23, 24, 39, 41  
  blog search engine 19, 41  
  blog software 18  
  reader comments 16  
blog advertising 40  
  Blogads 40  
  Federated Media 40  
Blog Business Summit conference 7  
blog networks 18, 41  
  b5media 18  
  Corante 18, 41  
  Gawker Media 18, 41  
  9rules 18, 41  
  Weblogs, Inc. 18, 41  
Blogads 40  
Blogdigger (blog search engine) 19  
Blogger 18  
blogger 16  
blogging 16, 25  
Bloglines 42  
blogosphere 18, 19  
blogroll 17  
Blue Dot 28, 43  
body DOM object 322  
body element 70, 71, 205  
body of a **for** 183  
body of a loop 181  
body property of the **document** object 316  
body section 70  
**<body>** tag 134  
BoingBoing 41  
bold value (**font-weight** property) 104  
bolder value (**font-weight** property) 104  
books database 588  
  table relationships 590  
boolean expression 156  
**Boolean** object 277  
border 118  
border attribute of **table** element 84  
border properties 119  
border-collapse property 332  
border-color property 118, 119  
border-left-color property 119  
Borders of block-level elements 118  
border-style property 118, 119  
border-top-style property 119  
border-width property 118, 119, 316  
bottom margin 113, 114, 115  
bottom tier 574  
**BoundField** ASP.NET element 759  
box model 118  
Box model for block-level elements 118  
**<br />** 138  
br (line break) element 94  
braces ({} ) 159  
bracket expressions 621

- brackets that enclose subscript of an array 234  
**braille** media type 122  
**break apart** 473  
**break out of a nested set of structures** 195  
**break statement** 190, 192, 195  
**break statement in a for statement** 192, 193  
**bricks-and-mortar store** 737, 833  
**Brightcove** 42  
Brin, Sergey 8  
*Britannica* 14  
broadband Internet 4, 6, 23  
brochure web 4  
**Browse** access permission 579  
browser request 88  
**Brush Mode** 474  
**Brush tool** 474  
bubbling 333  
**Build Page** command in Visual Web Developer 716  
**Build Project** option in Netbeans 900  
**Build Site** command in Visual Web Developer 716  
“building blocks” 259  
building blocks 154, 175  
built-in data types 369  
business letter marked up as XML 355  
business logic 575, 630, 660  
business object 786  
business rule 575  
business rules 630  
business-to-business (B2B) transactions 893  
Button ASP.NET Web control 721  
Button element 537  
  click attribute 537  
Button JSF component 820, 825  
  primary property 860  
  reset property 860  
button symbol 462  
button symbols 462  
button-click event 220  
buttons 461  
Buy Now Button JSF component 872  
buying and selling domain names 41
- C**
- C programming language 68  
C++ programming language 68  
cache 574  
cache of data stored temporarily in local memory 603  
CachedRowSet interface 863  
CachedRowSetDataProvider class 863, 871  
  appendRow method 871  
  commitChanges method 872  
  refresh method 872  
CafePress.com 40  
calculating the sum of the elements of an array 240  
calculation 147  
calculus expression marked up with MathML and displayed in the Amaya browser 374  
callback function 414, 422  
Campfire (37Signals) 30
- Cancel** button 146  
cancel event bubbling 343, 345  
cancelBubble property of an event object 333, 343, 345  
CandlestickChart element 559, 566  
  dataProvider attribute 566  
CandlestickSeries element 566  
  closeField attribute 566  
  highField attribute 566  
  lowField attribute 566  
  openField attribute 566  
capitalize method 657  
caption element (table) 84  
CareerBuilder 42  
caret metacharacter (^) 621  
carriage return 140  
Cascading Style Sheets (CSS) xix, 68, 101  
case label 188, 188  
case sensitive 136, 142  
cases in which switch cases would run together 189  
case-sensitive 160  
cash register 465  
Caslon Analytics 18  
casting 612  
**catch** block 421  
**catch** clause 421  
**catch** keyword 421  
CategoryAxis element 566  
  categoryField attribute 566  
categoryField attribute 566  
CDATA keyword (DTD) 363  
CDATA section 552  
ceil method 260, 656  
center horizontally 116  
center value (background-position property) 115  
center value (text-align property) 118  
centering an image on the stage with the Property Inspector 481  
chance 211  
change attribute 522, 536  
changing gradient colors with the Color Mixer panel 487  
channel child element of an rss element 402  
character 261  
character classes 622  
character data in XML 363  
character entity reference 79, 363  
character-processing capabilities 261  
character-processing methods of String object 264, 265  
character string 135  
charAt method of String object 262, 264  
charCodeAt method of String object 262, 264  
check winner and show Winner or Game Over 511  
checkbox 94  
“checkbox” input 94  
checked attribute 95  
checked property of a radio button object 255  
Chemical Markup Language (CML) 376  
chicagocrime.org 35  
child 106, 299
- child element 353, 356  
child node (DOM tree) 385  
child window 279  
childNodes property of a Node 396  
children (DOM tree) 385  
ChoiceStream 43  
choosing a gradient fill 460  
citizen journalism 17  
Citizendum 42  
class 258, 259, 659  
class attribute 105, 106  
class-average program with counter-controlled repetition 161  
class variables 659  
Classes  
  AbstractPageBean 806, 937  
  CachedRowSetDataProvider 863, 871  
  Control 705  
  Cookie 842  
  DataColumn 603  
  DataRow 603  
  DataSet 603  
  DataTable 603  
  HttpCookie 743  
  HttpCookieCollection 745  
  HttpServletRequest 844  
  HttpServletResponse 842  
  HttpSessionState 746, 747, 748, 750, 753  
  Option 845  
  Page 705, 732, 746  
  Properties 841, 850  
  SqlCommand 603  
  SqlConnection 603  
  SqlDataAdapter 604  
  UIViewRoot 805  
  ValidatorException 832, 833  
  WebControl 706  
classifieds 15  
className property of a DOM node 311  
**Clean and Build Project** option in Netbeans 900  
**Clean Project** option in Netbeans 900  
**clearInterval** method of the window object 316, 321  
CleverSet 43  
click attribute 537  
ClickBank 40, 43  
ClickTracks 44  
Client desktop application for the HugeInteger web service 909  
Client interacting with web server. *Step 1:* The GET request 572  
Client interacting with web server. *Step 2:* The HTTP response 572  
client of a class 259  
client of an object 135  
client side xx  
client-side artifacts 905, 906  
client-side scripting 575  
client tier 575  
clientX property of an event object 333  
clientY property of an event object 333  
close a dialog 140  
closed database connection 603

- c**closeField attribute 566  
**c**loseHandler parameter 537  
**cm** (centimeter) 109  
**CML** (Chemical Markup Language) 349  
**CML** (XML vocabulary) 38  
**CMP** 42  
**code-behind file** 702  
**code-behind file** for a page that updates the time every minute 705  
**code blocks** 658  
**code.google.com/apis/gears/** 566  
**CodeFile** attribute in a **Page** directive 703  
**coin tossing** 211  
**collaborative filtering** 15, 15, 28  
**collection** 313  
**collection\_select** method 681  
**collections** of data items 233  
**collective intelligence** 5, 14  
**collision detector** 514  
**colon** (: ) 102, 105, 196  
**color name** 102  
**Color** panel in Flash CS3 456  
**color** property 102, 105  
**color swatch** 457  
**cols** attribute (**table**) 94  
**colspan** attribute 87  
**column** 249, 587, 588  
**column number in a result set** 592  
**columns** element 536  
**com.sun.rave.web.ui.appbase** package 806  
**com.sun.webui.jsf.component** package 806, 810  
**Comedy Central** 24  
**comma** 276  
**comma-separated list** 150  
**comma-separated list of arguments** 260  
**comma-separated list of variable names** 143  
**command-line utilities** 519  
**comment** 69  
**Commission Junction** 40, 41  
**commitChanges** method of class **CachedRowSetDataProvider** 872  
**common actions after collision** 512  
**common ActionScript functions** 491  
**Common Programming Errors** overview xxv  
**Commonly used JSF components** 820  
**comparator function** 248  
**compare.php** 618  
**comparison operators** 618  
**Compete, Inc.** 41  
**competitive intelligence** 41  
  Compete, Inc. 41  
  Hitwise 41  
**compiler** 519  
**completeEffect** attribute 524  
**complex content in XML Schema** 369  
**Complex XHTML table** 87  
**complexType** XML Schema element 369  
**Component** element 524  
**component in the UML** 259  
**component tree** 805  
**compound assignment operator** 169  
**compound interest** 183  
**Compression type** 465  
**computer** xx  
**comScore** 7  
**concat** method 262  
**concatenation** 262  
**concatenation operator (+)** 262  
**concatenation operator (.)** 612  
**condition** 148, 197  
**condition is false** 148  
**condition is true** 148  
**conditional expression** 157  
**conditional operator (?)** 157, 172  
**Conditionals, loops, and codeblocks** 658  
**conferences** 42  
  AdTech 7  
  Affiliate Marketing Summit 7  
  AjaxWorld Expo 7  
  All Things Digital 7  
  Always On 7  
  Blog Business Summit 7  
  CMP 42  
  eBay Live 7  
  Emerging Technology 7, 29  
  Emerging Telephony 7  
  Freeware Open Source Summit 30  
  Future of Online Advertising 7  
  JavaOne 7  
  Jupiter 42  
  Location Intelligence conference 37  
  Microsoft MIX 7, 47  
  Microsoft Tech Ed 7  
  MySQL Conference and Expo 7  
  O'Reilly Media 42  
  Open Source (OSCON) 7  
  OSCON (O'Reilly Open Source Convention) 30  
  RailsConf 7  
  Search Engine Strategies 7, 12  
  Tools of Change for Publishing 7  
  Ubuntu Live 7  
  Web 2.0 Expo 7  
  Web 2.0 Summit 4, 7  
  Where 2.0 7, 37  
**config/database.yml** 666  
**confirm** method of the **window** object 343  
**ConformsTo** property of a **WebServiceBinding** attribute 951  
**confusing equality operator** == with assignment operator = 148  
**ConnectionString** property of a **SqLDataSource** ASP.NET control 759  
**constant** 220, 261  
**constrained aspect ratio** 459  
**constructor** 273, 277  
**consuming a web service** 894, 904  
**contact.html** 75, 80  
**container** element 353  
**containing block-level element** 113  
**content** 372  
**Content** ASP.NET control 777  
**content** attribute of **meta** element 98, 99  
**Content MathML** 372  
**content networks** 12, 42  
  *9rules* 13  
  About.com 13, 42  
  b5media 13  
  Corante 13  
**content networks** (cont.)  
  Deitel 13, 42  
  eHow 13  
  Gawker Media 13  
  HowStuffWorks 13  
  LifeTips 13, 42  
  Suite101 13, 42  
  Weblogs, Inc. 13  
**content page in ASP.NET** 769  
**ContentPlaceHolder** ASP.NET control 775  
**context node (XPath)** 381  
**context-sensitive Property Inspector** 456  
**contextual advertising** 40  
**continue** statement 192, 196  
**continue** statement in a **for** statement 193, 194  
**continue** statement in a nested **for** statement 196  
**Control** class 705  
**control structure** 175  
**control variable** 178  
**control variable's name** 178  
**ControlBar** element 530  
**Controller** for Flickr Photo Viewer 698  
**Controller generator** 662  
**controller logic** 574  
**controlling expression of a switch** 188  
**ControlParameter** ASP.NET element 763  
**control-statement stacking** 155  
**control-statement nesting** 155  
**ControlToValidate** property of an ASP.NET validation control 727  
**ControlToValidate** property of class **RegularExpressionValidator** 727  
**Convention over Configuration** 660  
**Convert to Symbol...** 462  
**converting strings to all uppercase or lowercase letters** 262  
**cookie** 634, 738, 739, 745, 834, 835, 844, 845  
  deletion 739, 835  
  domain 746, 846  
  expiration 739, 835  
  expiration date 739, 835  
  header 739, 834, 835  
**Cookie** class 842  
  getDomain method 846  
  getMaxAge method 846  
  getName method 846  
  getPath method 846  
  getSecure method 846  
  getValue method 846  
**cookies** 284  
**Cookies** array 844  
**Cookies** collection of the **Response** object 743  
**Cookies** directory 638  
**Cookies** property of **Request** class 745  
**Coordinated Universal Time (UTC)** 272, 276  
**coordinates of mouse cursor inside client area** 333  
**Copy Frames** 475  
**copyright infringement** 23  
**Corante** (blog network) 13, 18, 41  
**cos** method 260

- cosine 260  
cost-per-action (CPA) 40  
cost-per-click (CPC) 40  
cost-per-thousand impressions (CPM) 40  
count downward 182  
**count** function 615  
**Count** property of class  
  **HttpSessionState** 752  
counter-controlled repetition 161, 167, 175, 177, 179, 191  
Counter-controlled repetition to calculate a class average 161  
Counter-controlled repetition with the **for** statement 177  
**Counts** property of  
  **HttpSessionState** class 750  
**Courier** font 105  
CPA (cost-per-action) 40  
CPC (cost-per-click) 40  
CPM (cost-per-thousand impressions) 40  
Craigslist 15, 35, 42  
craps 215  
Craps game simulation 216  
create a desktop application in Netbeans 5.5 905  
create a web application in Netbeans 5.5 895  
**createAttribute** method 397  
**createDocument** method of the **document** object's **implementation** property 394  
**createElement** method (XML DOM) 397  
**createElement** method of the **document** object 311, 332, 427  
**createTextNode** method (XML DOM) 397  
**createTextNode** method of the **document** object 311  
**CreateUserWizard** ASP.NET login control 777  
creating a basic XHTML table 85  
creating a dynamic text field with the **Property Inspector** 470  
creating a frame-by-frame animation 484  
creating a mask layer 476  
creating a motion tween 469  
creating a new symbol with the **Convert to Symbol** dialog 462  
creating a rectangle with rounded corners 478  
creating a rotating object with the motion tween **Rotate** option 486  
creating a shape tween 481  
Creating a simple book cover viewer in Flex 2 520  
creating a text field 470  
creating a title with the Text tool 468  
creating an animation with a symbol 482  
creating an instance 235  
creating an oval graphic 476  
creating multiple instances of a symbol with the **Library** panel 479  
**creationComplete** attribute 538  
**creationPolicy** attribute 535  
Creative Commons licenses 31  
**Critter** font 105  
CRM (customer relationship management) 30  
cross-browser events 346  
cross-site scripting (XSS) 418  
CRUD (create, retrieve, update, destroy) 667, 668  
CSS (Cascading Style Sheets) xix, 32, 68, 101, 412, 683  
  comment 109  
  CSS 2 101  
  CSS 3 130  
  CSS 3 features 130  
  drop-down menu 124  
  property 102  
  rule 104  
  selector 103  
  stylesheet 805, 814  
**Ctrl/Key** 333  
**ctrlKey** property of an **event** object 332, 333  
**Cubic.easeIn** function 531  
curly brace ({} ) 105  
**cursive** font 105  
cursor 140  
custom map 37  
custom tag libraries 801  
custom tags 801  
custom type 939  
customer relationship management (CRM) 30  
customize proportions 480
- ## D
- dangling-else problem 158  
dashed value (**border-style** property) 119  
**data** 135, 603  
**data** cells 86  
**data** keyword 524  
**data** method of a **Text** node 398  
**data provider** 863, 882  
**data push** 519  
**data source** 603  
**data tier** 574  
**data.php** 610  
**database** xx, 68, 586, 591, 653

demonstrating the `onfocus` and `onblur` events 339  
demonstrating the `onload` event 327  
demonstrating the `onsubmit` and `onreset` events 341  
`deny` element in `Web.config` 773  
dependency injection 919  
**Deploy Project** option in Netbeans 900  
deploying a web service 900  
descendant elements 106  
descendant node 385  
descending sort (DESC) 594, 595  
deselect 460  
**Design** mode in Java Studio Creator 2 860  
**Design** mode in Netbeans 813, 817  
**Design** mode in Visual Web Developer 711  
**destroy** method (JavaServer Faces) 811, 816, 872  
detect collision using `hitTestObject` 514  
detecting a miss 511  
Developer Toolbar 300  
developer.yahoo.com 540  
developer.yahoo.com/flash/  
as3\_api\_libraries.html 540  
developer.yahoo.com/flash/  
as3webapis/docs/com/yahoo/  
webapis/maps/methodgroups/  
MapController.html 543  
developer.yahoo.com/flash/  
howto-maps-as3.html 540  
dialog 139  
dialog boxes 139  
diamond symbol 155, 156, 181  
Dice 42  
Dice-rolling program using arrays instead of `switch` 241  
die function 630, 634  
differences between preincrementing and postincrementing 171  
Digg 14, 15, 24, 43  
digit 261  
Digital Rights Management (DRM) 25  
DimDim 44  
**Dimensions** 456  
directive 801  
directive in ASP.NET 703  
disabling event bubbling 343  
disc (bullet for unordered lists) 81, 82  
disconnected object model 603  
discovery 12, 42  
dismiss (or hide) a dialog 140  
Disney Family Locator 37  
display property 124, 126  
**Display** property of an ASP.NET validation control 727  
Display the name of the current action in the title bar 673  
Displaying a list of messages without reloading the page 687  
displaying the cookie's contents 639  
displaying the `MailList` database 647  
div element 114, 116  
divide and conquer 204  
DNS (domain name system) server 571  
DNS lookup 571  
do...while repetition statement 154, 190, 191, 192, 491  
flowchart 191  
DOCTYPE parts 356  
document 278  
**Document** object 396  
**document** object 135, 147, 221, 257, 322  
anchors collection 313  
appendChild method 427  
body property 316  
createElement method 311, 332, 427  
createTextNode method 311  
forms collection 313, 322  
getElementById method 221, 222, 299, 310, 311, 421  
getElementsByTagName method 427  
images collection 313, 322  
implementation property 394  
links collection 313, 322  
setAttribute method 427  
write method 137  
**document** object methods and properties 278, 284  
**Document Object Model** 299  
**Document Object Model (DOM)** 135, 412  
innerHtml property 421  
**Document Object Model (DOM) tree** 385  
**document** object's cookie property 284  
**document** object's write method 136  
**document** object's writeln method 135, 137  
**DOCUMENT** property of a Web Form 712  
document root 380  
document type declaration 703, 812  
**Document Type Definition (DTD)** 38, 350, 356, 362  
for a business letter 362  
**document.writeln** method 171  
Dodgeball.com 23, 42  
Dojo 6, 32, 412, 447  
Dojo Offline 34  
Dojo widget 447  
dollar amount 185  
dollar sign (\$) 142  
dollar-sign metacharacter 621  
DOM 32, 33  
DOM (Document Object Model) 412  
**DOM (Document Object Model) tree** 385  
DOM API (Application Programming Interface) 386  
**DOM collection**  
  item method 315  
  length property 315  
  namedItem method 315  
**DOM Inspector** 300  
**DOM node** 299  
  appendChild method 311  
  className property 311  
  id property 311  
  innerHTML property 221  
  insertBefore method 312  
  parentNode property 312  
  removeChild method 312  
**DOM node** (cont.)  
  replaceChild method 312  
  setAttribute method 427  
**DOM node firstChild** property 395  
**DOM node lastChild** property 395  
**DOM node nextSibling** property 395  
**DOM node nodeName** property 395  
**DOM node nodeType** property 395  
**DOM nodenodeValue** property 395  
**DOM node parentNode** property 396  
**DOM parser** 385  
**DOM tree** 299  
**domain name system (DNS) server** 571  
**Domain** property of  
  HttpCookie class 746  
**domain registrars** 42  
  GoDaddy 42  
  Network Solutions 42  
  Register.com 42  
**Don't Repeat Yourself** 660  
**Dornfest, Rael** 29  
**dot (.)** for accessing object properties and methods 260  
**dot notation** 221  
**DotNetNuke** 31  
**dotted value (border-style property)** 119  
**double click** 346  
**double data type** 610  
**double equals** 148  
**double quotation ("") mark** 135, 140, 358  
**double-quote character** 137  
**double-quote ("") character as part of string literal** 177  
**Double Range Validator** JSF component 825  
**double-selection statement** 154  
**double value (border-style property)** 119  
**DoubleClick** 41, 42  
**Dougherty, Dale** (coined term Web 2.0) 3  
**down state** 463  
**draggable\_element** method of  
  Script.aculo.us library 696  
**DRM (Digital Rights Management)** 25  
**Drop Down List** JSF component 820, 824  
**drop\_receiving\_element** method of  
  Script.aculo.us library 696  
**DropDownList** ASP.NET Web control 720  
**DTD (Document Type Definition)** 38, 350, 356  
**.dtd** filename extension 356  
**DTD repository** 362  
**dummy value** 164  
**duplicate symbol** 490  
**duplicating movie-clip symbols with the Library panel** 483  
**duration** attribute 531  
**duration** parameter of  
  visual\_effect method of  
    Script.aculo.us Library 692  
**dynamic form using PHP** 640  
**dynamic memory allocation operator** 235  
**dynamic modification of an XHTML document** 322  
**dynamic style** 315  
**dynamic typing** 656

dynamic web pages 141, 584

## E

E4X (ECMAScript for XML) 531  
casing value 488  
**easingFunction** attribute 531  
Eastern Standard Time 276  
eBay 13, 19, 34, 39, 40, 43  
eBay Live conference 7  
EBNF (Extended Backus-Naur Form)  
  grammar 362  
eBuddy 44  
ECMA International 132  
ECMAScript 132  
  E4X (ECMAScript for XML) 531  
ECMAScript 4 518  
ECMAScript for XML (E4X) 531  
ECMAScript standard 132  
  ([www.ecma-international.org/publications/standards/ECMA-262.htm](http://www.ecma-international.org/publications/standards/ECMA-262.htm)) 132  
ecommerce 40, 43  
E-Commerce Services (ECS) 36  
edit-in-place 441  
editing button symbols 463  
editing stage 461, 463, 467, 488  
editing tools 473  
eHow (content network) 13  
Elance 43  
Elastic Compute Cloud (EC2) 36  
element (XML) 349  
Element dimensions and text alignment 117  
ELEMENT element type declaration (DTD) 363  
element gains focus 338  
element loses focus 338  
element name restrictions (XML) 352  
Element object 396  
element of chance 210  
element type declaration 363  
element XML Schema element 368  
elements 70  
elements of ActionScript 3.0 516  
elements of an array 233, 614  
**elements** property of a form object 255  
em (M-height of font) 108, 128  
em element 81  
em measurement for text size 129  
emacs text editor 69  
e-mail 75  
e-mail anchor 75  
@Embed directive 527  
embedded 453  
embedded Ruby 663  
embedded style sheet 103, 104, 105  
Emerging Technology conference 7, 29  
Emerging Telephony conference 7  
EMI 25  
Employee controller provides all of the functionality for the application 668  
empty array 237  
empty body 136  
empty element 77, 94, 358  
EMPTY keyword (DTD) 364  
empty statement 151, 160, 180  
empty string 201, 262, 264, 614  
[en.wikipedia.org/wiki/EBNF](http://en.wikipedia.org/wiki/EBNF) 362

[en.wikipedia.org/wiki/](http://en.wikipedia.org/wiki/)  
  IP\_address 571  
**Enable Paging** setting for an ASP.NET GridView 789  
**Enable Sorting** setting for an ASP.NET GridView 789  
**EnableSessionState** property of a Web Form 703, 714  
**EnableViewState** property of an ASP.NET Web control 704, 714, 732  
enabling JavaScript in Internet Explorer 7 133  
encapsulation 258, 519  
enclose script code in an XHTML comment 134  
**enclosure** element (RSS) 403  
encrypted data 782  
encyclopedia and reference sources 42  
  Citizendium 42  
  Reference.com 42  
  Wikipedia 42  
end of a script 135  
“end of data entry” 164  
end tag 70, 349  
end value of queue parameter 693  
ending index 269  
engine optimization (SEO) 99  
entity  
  & 364  
  gt; 364  
  lt; 364  
entity reference &InvisibleTimes; 1 in MathML 374  
entity reference (MathML) 375  
entity-relationship diagram 590  
environment variable 623, 663  
equality and relational operators 148, 149  
equality operator 197, 198  
equality operators 148, 618  
equality operators and Strings 262  
equals equals 148  
**Eraser tool** 474  
erb 663  
ereg function 619, 621, 622  
ereg\_replace function 622, 623  
eregi function 621  
Error-Prevention Tip overview xxv  
**ErrorMessage** property of an ASP.NET validation control 727  
**Errors** 675  
escape character 137, 598  
escape early from a loop 192  
escape function 226, 285  
escape sequence 137, 630  
escape sequence \\" 176  
escaped property 826  
EST for Eastern Standard Time 276  
eval function 226  
evaluate method of a Firefox 2 XML document object 401  
event 220, 220  
event bubbling 343, 345  
event-driven programming 220  
event handler 220, 324  
event handling 219  
event-handling function 220  
event model 324  
event object 328, 332  
  altKey property 333  
event object (cont.)  
  cancelBubble property 333, 343  
  clientX property 333  
  clientY property 333  
  ctrlKey property 332, 333  
  keyCode property 333  
  screenX property 333  
  screenY property 333  
  shiftKey property 333  
  srcElement property (IE) 338  
  target property (FF) 338  
  type property 333  
event processing life cycle 810  
event processing lifecycle 806, 816  
  destroy method 811, 816, 872  
  init method 810, 816  
  preprocess method 810, 816  
  prerender method 810  
event processingcycle  
  prerender method 816  
event registration models 324  
events  
  onmouseout 417  
  onmouseover 417  
events onmouseover and onmouseout 333  
*Everything is Miscellaneous* (by David Weinberger) 26  
ex (“x-height” of the font) 108  
ex value 176  
Examination-results problem 167  
Examples 504  
  absolute positioning of elements 112  
  Adding a new message without reloading the page 690  
  Adding an intermediate color to a gradient 488  
  Adding background images and indentation 115  
  Adding Frame Labels using the Property Inspector 479  
  Adding sound to a button 466  
  Adding text to a button 461  
  Addition script “in action” 145  
  addition.html 145  
AddressBook JSP with an AutoComplete Text Field component 875  
Adjusting line spacing with the Format Options panel 478  
Alert dialog displaying multiple lines 139  
Algebraic equation marked up with MathML and displayed in the Amaya browser 373  
analysis.html 167  
Array manipulation 615  
Arrays and hashes in Ruby 657  
arrays.php 615  
article.xml displayed by Internet Explorer 353  
ASPX file that displays the Web server’s time 702  
average.html 161  
average2.html 164  
Basic DOM functionality 303  
borders of block-level elements 118  
box model for block-level elements 118

- Examples (cont.)
- `BreakLabelTest.html` 195
  - `BreakTest.html` 193
  - Business letter marked up as XML 355
  - Calculating the sum of the elements of an array 240
  - Calculus expression marked up with MathML and displayed in the Amaya browser 374
  - Centering an image on the stage with the **Property Inspector** 481
  - Changing gradient colors with the **Color Mixer** panel 487
  - Check winner and show **Winner or Game Over** 511
  - Choosing a gradient fill 460
  - Code-behind file for a page that updates the time every minute 705
  - Common actions after collision 512
  - `compare.php` 618
  - complex XHTML table 87
  - Conditionals, loops, and codeblocks 658
  - `contact.html` 75
  - `contact2.html` 80
  - `ContinueLabelTest.html` 196
  - `ContinueTest.html` 194
  - Controller for Flickr Photo Viewer 698
  - counter-controlled repetition 175
  - counter-controlled repetition to calculate a class average 161
  - Counter-controlled repetition with the **for** statement 177
  - `coverviewer.html` 317
  - Craps game simulation 216
  - Creating a basic table. 85
  - Creating a dynamic text field with the **Property Inspector** 470
  - Creating a frame-by-frame animation 484
  - Creating a mask layer 476
  - Creating a motion tween 469
  - Creating a new symbol with the **Convert to Symbol** dialog 462
  - Creating a rectangle with rounded corners 478
  - Creating a rotating object with the motion tween **Rotate** option 486
  - Creating a shape tween 481
  - Creating a simple book cover viewer in Flex 2 520
  - Creating a text field 470
  - Creating a title with the Text tool 468
  - Creating an animation with a symbol 482
  - Creating an oval graphic 476
  - Creating multiple instances of a symbol with the **Library** panel 479
  - CSS drop-down menu 124
  - `data.php` 610
  - Database migration script modified to add data to the table 667, 674, 676
  - Examples (cont.)
    - Database migration script modified to add data to the `users` table 670
    - Default layout of `Script.aculo.us demo` 693
    - Default namespaces demonstration 361
    - Default view of `Script.aculo.us demo` 692
    - Defining the hit area of a button 489
    - Deitel Resource Centers xxvii
    - Demonstrating date and time methods of the `Date` object 274
    - Demonstrating logical operators 200
    - Demonstrating the `onfocus` and `onblur` events 339
    - Demonstrating the `onload` event 327
    - Demonstrating the `onsubmit` and `onreset` events 341
    - Demonstration of a document's DOM tree 300
    - Detect collision using `hitTestObject` 514
    - Detecting a miss 511
    - Dice-rolling program using arrays instead of `switch` 241
    - Differences between preincrementing and postincrementing 171
    - Disabling event bubbling 343
    - Display the name of the current action in the title bar 673
    - Displaying a list of messages without reloading the page 687
    - Displaying the cookie's contents 639
    - Displaying the **MailingList** database 647
    - Document Type Definition (DTD) for a business letter 362
    - `dom.html` 303
    - `domtree.html` 300
    - `DoWhileTest.html` 191
    - Duplicating movie clip symbols with the **Library** panel 483
    - Dynamic form using PHP 640
    - Dynamic styles 315
    - Dynamic styles used for animation 317
    - `dynamicstyle.html` 315
    - element dimensions and text alignment 117
    - em measurement for text size 129
    - embedded style sheets 103
    - Employee** controller provides all of the functionality for the application 668
    - Event registration models 324
    - Events `onmouseover` and `onmouseout` 333
    - Examination-results problem 167
    - Expression marked up with MathML and displayed in the Amaya browser 373
    - `expression.php` 619
    - Factorial calculation with a recursive function 229
    - Fire ball on click event 507
    - Examples (cont.)
      - First program in JavaScript 133
      - first XHTML example 70
      - `first.php` 608
      - Floating elements 120
      - `ForCounter.html` 177
      - form including radio buttons and a drop-down list 91
      - Form that allows the user to enter a new message 685
      - Form to query a MySQL database 631
      - form with hidden fields and a text box 89
      - `form.html` 89
      - `form2.html` 91
      - Forum model that includes validation and initialization functionality 676
      - Forum that allows the user to add another message on the same page 688
      - Forums layout that uses the default JavaScript libraries 686
      - ForumsController** implements CRUD functionality 677
      - `fullsizeImage` view of Flickr photo viewer 699
      - Gathering data to be written as a cookie 635
      - Generated code for an **Employee** model 668
      - Generated code for the **User** model 671
      - Handle `playButton` click event 498
      - Handle the map functions for the Address Book application 541
      - Handle the map functions for the Weather application 553
      - heading elements h1 through h6 73
      - `heading.html` 73
      - HTML wrapper for `coverViewer.swf` 526
      - HTTPService, Validators, and new UI elements 531
      - HugeInteger** web service that performs operations on large integers 896
      - Images as link anchors 78
      - Images in XHTML files 76
      - Implement the actions of each button 544
      - Important methods of class `HttpCookie` 746
      - `increment.html` 171
      - inheritance in style sheets 107
      - Initializing multidimensional arrays 250
      - Initializing the elements of an array 235, 238
      - Input and password text field creation 479
      - Inserting a keyframe 464
      - Inserting special characters 80
      - Interaction between a web service client and a web service 894
      - `Interest.html` 183
      - internal hyperlinks to make pages more navigable 95

- Examples (cont.)
- `internal.html` 95
  - JSP file for the `AddressBook` page
    - with a search form and `Table` component 864
  - JSP file generated by Netbeans that displays the current time on the web server 803
  - JSP file that allows the user to select a programming language 835, 847
  - JSP file that displays book recommendations based on cookies 842
  - JSP file that displays book recommendations based on language selections stored in session scope 855
  - JSP for the `AddressBook` page with `Map Viewer` component 883
  - Labeled `break` statement in a nested `for` statement 195
  - Labeled `continue` statement in a nested `for` statement 196
  - Layout that displays a greeting 665
  - Layout that displays the logged in user for every `Forums` action 681
  - `Link` view of `Script.aculo.us` demo 693
  - linking an external style sheet 110
  - linking to an e-mail address 75
  - Linking to other Web pages 74
  - `links.html` 74
  - `links2.html` 82
  - `list.html` 83
  - `LogicalOperators.html` 200
  - Login form used to send data to the `user` controller 672
  - Main view for Flickr Photo Viewer 697
  - `main.html` 70
  - Making multiple selections with the Selection tool 459
  - Mashups 35
  - Media types 122
  - Message layout that links a style in and displays a message 686
  - Message** model containing validation and initialization functionality 675
  - `MessagesController` that implements CRUD functionality 683
  - `meta` elements provide keywords and a description of a page 98
  - `meta.html` 98
  - Method calls on numeric and string data 656
  - Modifying button states with a button's editing stage 464
  - Modifying the size of a shape with the **Property Inspector** 460
  - Move ball and set speed 506
  - Move target, set direction and speed, and keep track of number of blocks hit 499
  - `mxmlc` running in Windows XP **Command Prompt** 525
  - `nav.html` 78
  - Nested and ordered lists. 83
- Examples (cont.)
- Object animation in ActionScript 496
  - Obtaining user input through forms 626
  - Online Quiz graded with JavaScript 253
  - Online quiz in a child window 294
  - `operators.php` 613
  - Optimizing sound with the `Sound Properties` dialog 466
  - Page bean file that sets `clockText` to the time on the web server 806
  - Page bean for adding a contact to the `AddressBook` 870
  - Page bean for validating user input and redisplaying that input if valid 831
  - Page bean that displays book recommendations based on a `SessionBean` property 855
  - Page bean that displays book recommendations based on cookies storing user's selected languages 844
  - Page bean that gets a map to display in the `MapView` component 887
  - Page bean that stores language selections in a `SessionBean` property 852
  - Page bean that stores user's language selection as a cookie on the client 839
  - Page bean that suggests names in the `AutoComplete Text Field` 879
  - Partial that contains a form used to add a new forum 680
  - Passing arrays and individual array elements to functions 245
  - `picture.html` 76
  - `print` media type 122
  - Printing on multiple lines with a single statement 138
  - Printing on one line with separate statements 136
  - Printing one line with separate statements 136
  - Programmer-defined function `square` 206
  - Programmer-defined `maximum` function 209
  - Prompt box used on a welcome screen 141
  - `pt` measurement for text size 126
  - Querying a database and displaying the results 632
  - Random image generation using arrays 242
  - Random image generation using `Math.random` 222
  - Registration form that demonstrates JSF components 821
  - Regular expressions in PHP 619
  - relative positioning of elements 113
  - Renaming a layer 467
  - Rendering an RSS feed in a web page using XSLT and JavaScript 406
- Examples (cont.)
- Reset the ball to its original position 514
  - Resizing an image with the Free transform tool 475
  - Rich welcome page using several JavaScript concepts 288
  - Rolling a six-sided die 6000 times 213
  - RSS 2.0 sample feed 408
  - Ruby class 659
  - Running `mxmlc` with the `theme` parameter in Windows XP **Command Prompt** 558
  - Schema-valid XML document describing a list of books 366
  - Scoping example 224
  - `Script.aculo.us` Demo controller 692
  - Searching `Strings` with `indexOf` and `lastIndexOf` 266
  - selecting a background color 457
  - Selecting an object with the Selection tool 462
  - Sentinel-controlled repetition to calculate a class average 164
  - `SessionBean` file for storing language selections 851
  - `SessionBean` that initializes the data source for the `AddressBook` database 868
  - Setting text alignment with the **Property Inspector** 468
  - setting the fill and stroke colors 458
  - Setting the font face, size, weight and color with the **Property Inspector** 461
  - Shifted and scaled random integers 211
  - Simple controller that passes a parameter to the view 664
  - Simple controller that renders a message on the web page 662
  - Simple drawing program 329
  - Simple PHP program 608
  - Simple Ruby program 654
  - SOAP request for the `HugeInteger` web service 915
  - some common escape sequences 140
  - Sorting an array with `Array` method `sort` 247
  - Stops all timers and sends objects into hidden frame 510
  - String methods `charAt`, `charCodeAt`, `toLowerCase` and `toUpperCase` 264
  - `Sum.html` 182
  - Summation with `for` 182
  - `SwitchTest.html` 186
  - `table1.html` 85
  - `table2.html` 87
  - Template for a `List` action that displays a list of messages 684
  - Template for a `New` action that adds a `forum` to the `forums` table 680
  - Template for `Delete` action used to delete a `forum` 681
  - Template for the `List` action that displays a list of messages 679

- Examples (cont.)  
 Text Cloud of Deitel Web 2.0  
     Resource Center 27  
     thumbs view of Flickr photo viewer 698  
 Time class declaration maintains the time in 24-hour format 662, 664, 665, 671, 672, 673, 675, 681  
 Timer countdown 509  
 Traversing an XML document using the XML DOM 387  
 Tree structure for the document `article.xml` of Fig. 14.2 386  
 Type conversion 610  
 Unordered list containing hyperlinks. 81  
 user style sheet applied with `em` measurement 129  
 user style sheet applied with `pt` measurement 128  
 User style sheet in Internet Explorer 7 127  
**UserController** provides validation functionality for the table 671  
 Using a `filterFunction` for a live search feature 540  
 Using a `Style` element and inline styles 554  
 Using cookies to store user identification data 285  
 Using equality and relational operators 149  
 Using inline styles 102  
 Using PHP's arithmetic operators 613  
 Using `States` to dynamically modify a user interface 527  
 Using `String` object methods `split` and `substring` 268  
 Using the `break` statement in a `for` statement 193  
 Using the `continue` statement in a `for` statement 194  
 Using the `do...while` repetition statement 191  
 Using the `Links` collection 313  
 Using the string-comparison operators 618  
 Using the `switch` multiple-selection statement 186  
 Using the `window` object to create and modify child windows 279  
 Using XPath to locate nodes in an XML document 398  
 Validating an XML document with Microsoft's XML Validator 357  
 View that displays a list of `employees` 669  
 View that displays the name of the server 664  
 Weather Forecast application in Flex 2 546  
`welcome.html` 133  
`welcome2.html` 136  
`welcome3.html` 138  
`welcome4.html` 139  
`welcome5.html` 141  
`welcome6.html` 149
- Examples (cont.)  
`WhileCounter.html` 176  
 Writing a cookie to the client 636  
 XHTML form for gathering user input 624  
 XHTML markup methods of the `String` object 270  
 XHTML response generated when the browser requests `Time.jsp` 811  
 XHTML response when the browser requests `WebTime.aspx` 707  
 XML document containing book information 381  
 XML document that describes various sports 377, 401  
 XML document using the `laptop` element defined in `computer.xsd` 371  
 XML file containing contacts 538  
 XML namespaces demonstration 359  
 XML Schema document defining simple and complex types 370  
 XML Schema document for `book.xml` 366  
 XML used to mark up an article 351  
 XML Validator displaying an error message 364  
 XSL document that transforms `sorting.xml` into XHTML 382  
 XSLT that creates elements and attributes in an XHTML document 379  
 Examples of inline elements 114  
`examples.adobe.com/flex2/inproduct/sdk/explorer/explorer.html` 518  
 exception 421  
 exception handler 421  
 exception handling 421  
**Execute** (such as ISAPI applications or CGI) access permission 579  
`exp` method 260  
 Expat XML Parser 350  
 Expedia 10, 44  
 expiration date of a cookie 739, 835  
`expires` property of a cookie 284  
`Expires` property of `HttpCookie` class 746  
 exponential method 260  
 exponentiation 147  
 export to Flash Player 467  
 expression marked up with MathML and displayed in the Amaya browser 373  
`expression.php` 619  
 extend an XML Schema data type 371  
 Extended Backus-Naur Form (EBNF) grammar 362  
 extenders 797  
 Extensible Business Reporting Language (XBRL) 372  
 Extensible Hypertext Markup Language (XHTML) ix, xx, 68, 349, 570, 701, 799  
 Extensible Markup Language (XML) xx, 32, 38, 46, 800  
 Extensible Stylesheet Language (XSL) 351, 361, 376
- Extensible User Interface Language (XUL) 372, 376  
 extension 575  
**extension element**  
     `base` attribute 371  
**extension** XML Schema element 371  
 external DTD 356  
 external linking 109  
 external style sheet 109, 557  
**extract** function 626  
 eyeOS (webtop) 29
- F**
- f:view** element 805  
 Facebook 5, 19, 21, 21, 43  
 Factorial calculation with a recursive function 229  
 Fade effect 524, 551  
 false 148  
**false** 156  
**fault** attribute 551  
 FCC (Federal Communications Commission) 37  
 Federal Communications Commission (FCC) 37  
 Federated Media 40, 42  
 feed aggregator 42  
 Feedburner 18, 39, 41  
 feedmashr.com 35  
 Feedster (blog search engine) 19, 41  
 Filename extensions  
     `.jsp` 802  
 file sharing 42  
     AllPeers 42  
     BitTorrent 42  
     Kazaa 42  
     LimeWire 42  
     Shareaza 42  
 file size 457  
 Filename extensions  
     `.aspx` 701  
     `.master` 774  
 fill color 458  
 Filo, David 9  
**filterFunction** attribute 540  
**Filters** panel in Flash CS3 456  
 final value of the control variable 175, 178, 181  
 findability 7  
 fire an event 324  
 fire ball on click event 507  
 Firefox 2 31  
     `importstylesheet` method of the `XSLTProcessor` object 409  
     `transformToFragment` method of the `XSLTProcessor` object 409  
     `XSLTProcessor` object 409  
 Firefox 2 (FF2)  
     load XML document dynamically with JavaScript 394  
 firewall 914  
 first program in JavaScript 133  
 First XHTML example 70  
`first.php` 608  
**firstChild** property of a DOM node 395  
**firstChild** property of a Node 396  
 five-day forecast 545

- #FIXED keyword (DTD) 363  
**fixed** 270  
**fixed** method of the `String` object 263, 271  
**Fixedsys** font 105  
**fixed-template** data 801  
**fixed-template** text 801  
**Fixnum** 656  
`.fla` 456  
`.fla` file format 456  
flag value 164  
**flags** parameter 537  
Flash xix, xx, 32, 518  
  ActionScript Virtual Machine 519  
Flash CS3 493  
  Welcome Screen 454  
**flash hash table** 678  
Flash Player 519  
  plug-in 453, 467, 519  
Flex xix, xx, 6, 32  
Flex 3 Beta 567  
Flex Builder 520  
Flex Charting 520, 558  
Flex Style Explorer 557  
[flexwiki.adobe.com/confluence/display/ADOBESTRATEGY/Flex+3+Planning](http://flexwiki.adobe.com/confluence/display/ADOBESTRATEGY/Flex+3+Planning) 567  
Flickr 5, 14, 20, 27, 34, 43  
Flickr API key 697  
Flickr library for Ruby 696  
Flickr Live 27  
**float** property 121  
floated element 119  
floating 119  
Floating elements 120  
floating-point number 163  
**floor** 260  
**floor** method of the `Math` object 211, 215, 218  
flow text around `div` element 119  
flowchart 191  
flowcharting the `do...while` repetition statement 192  
flowcharting the double-selection `if...else` statement 157  
flowcharting the `for` repetition statement 181  
flowcharting the single-selection `if` statement 156  
flowcharting the `while` repetition statement 161  
**focus** 296, 338  
**focus** method of the `window` object 296  
folksonomy 26  
**font-family** property 105, 316  
**font-size** property 102, 105, 176  
**font-style** property 116  
**font-weight** property 104  
**fontFamily** property 316  
footer 664  
**for** repetition statement 154, 177, 179  
**for** repetition statement flowchart 181  
**for** statement 615  
**for** statement header 179  
**for...in** repetition statement 154, 239, 240, 252  
**foreach** statement 618  
foreign key 589, 590  
form 68, 88, 219, 338  
**Form** element 538  
**form** element 89  
  **action** attribute 90  
  **option** element 95  
**form** elements in an XHTML document 322  
**form** field 338  
**form** GUI component 144  
**form** handler 90  
**Form** including radio buttons and a dropdown list 91  
**form** object  
  **elements** property 255  
**form** resets 346  
**Form** that allows the user to enter a new message 685  
**form** to query a MySQL database 631  
**form** validation 630  
**Form** with hidden fields and a text box 89  
**form\_remote\_tag** 690  
**form\_tag** method 673  
**FormItem** element 538  
  **required** attribute 538  
**form-processing** events 324  
**forms** authentication 765  
**forms** collection of the `document` object 313, 322  
Forrester Research 25  
**Forum** model that includes validation and initialization functionality 676  
**Forum** that allows the user to add another message on the same page 688  
**Forms** layout that uses the default JavaScript libraries 686  
**FormsController** implements CRUD functionality 677  
forward slash character (/) 77  
forward slash character (/) in end tags 349, 380  
fractional parts of dollars 185  
frame 454  
frame label 479  
**Frame Rate** 456, 457  
frames per second 457  
**framesLoaded** property 491  
framework 802  
frameworks 32  
Free Software Foundation 31, 43  
Free transform tool 474  
Freeware Open Source Summit 30  
freshmeat.net 31, 42  
Friendster 20  
**FROM SQL** clause 591  
**fromCharCode** method of the `String` object 262, 264, 265  
**fromState** attribute 531  
**fscommand** function 491  
**fullsizeImage** view of Flickr photo viewer 699  
function 147, 204, 259, 491, 495  
function (or local) scope 224  
function body 208  
function call 207  
function definition 207  
function parameter 207  
function `parseInt` 146  
function-call operator 207  
Furl 28  
Future of Online Advertising conference 7  
**G**  
G.I.M.P. 76  
gadget 36  
Gaia Online 22, 41, 43  
gain focus 338  
gambling casino 210  
game of craps 216, 220  
game playing 210  
Garrett, Jesse James 32, 417  
Gates, William 5, 39, 47  
gathering data to be written as a cookie 668  
Gawker Media (blog network) 13, 18, 41  
gem 696  
General Motors 18  
Generated code for an `Employee` model 668  
Generated code for the `User` model 671  
generating XHTML dynamically during script processing 270  
generic font family 105  
geneva font 105  
**Geocoding Service Object** 882, 890  
  **geoCode** method 890  
**Geography Markup Language (GML)** 376  
**Georgia** font 105  
GeoRSS 38  
geotagging 38  
Gerstner, Louis 31  
GET HTTP request 571  
get method of interface `MessageContext` 919  
GET method of the `XMLHttpRequest` object 422  
get request 574  
get request type 89  
**getAllResponseHeaders** method of the `XMLHttpRequest` object 423  
**getAttribute** method of an `Element` 397  
**getAttributeNode** method of an `Element` 397  
**getDate** 272, 274  
**getDay** 272, 274  
**getDocumentElement** method 397  
**getElementById** method of `document` object 221, 222  
**getElementById** method of the `document` object 299, 310, 311, 395, 421  
**getElementsByTagName** method 397  
**getElementsByTagName** method of the `document` object 427  
**getFullYear** 272, 274, 275  
**getHours** 272, 274  
**getMilliseconds** 272, 274  
**getMinutes** 272, 274  
**getMonth** 272, 274  
**getRequestContext** method of interface `BindingProvider` 921  
**getResponseHeader** method of the `XMLHttpRequest` object 423  
gets 148  
gets the value of 148

- getSeconds** 272, 274  
**getTime** 272  
**getTimeZone** 275  
**getTimezoneOffset** 272, 274  
*Getting Real* (by 37Signals) 30  
**gettype** function 612  
**getUTCDate** 272  
**getUTCDay** 272  
**getUTCFullYear** 272  
**getUTHours** 272  
**getUTCMilliseconds** 272  
**getUTCMinutes** 272  
**getUTCMonth** 272  
**getUTCSeconds** 272  
Gigablast 11  
Gizmodo 41  
GlassFish 899  
global functions 226  
Global object 226  
global positioning system (GPS) 37  
global scope 223  
global variable 223  
GML (Geography Markup Language) 376  
GML website ([www.opengis.org](http://www.opengis.org)) 376  
GMT (Greenwich Mean Time) 272, 276  
GNU General Public License (GPL) 31  
GoDaddy 42  
Goldcorp 14  
Good Programming Practices overview xxv  
Google 5, 7, 8, 18, 19, 20, 23, 30, 35, 42, 43  
AdSense 9, 40  
AdWords 9  
Dodgeball.com 23  
gmail 44  
Google Analytics 44  
Google Base 42  
Google Blog Search 41  
Google Checkout 43  
Google Custom Search Engine 11  
Google Earth 35, 37, 38  
Google Gears 34  
Google Maps 6, 35, 37, 42  
Google Reader 34, 42  
Google Video 23  
iGoogle 43  
PageRank 8, 14  
Google Gears 566  
Google Maps 859  
Google Maps API 882  
**gotoAndPlay** function 491  
**gotoAndStop** function 480, 491  
GoToMeeting 44  
GPL (GNU General Public License) 31  
GPS (global positioning system) 37  
gradient color 487  
gradient fill 460  
gradient range 487  
gradients 460  
graphic symbols 462  
graphical user interface (GUI) xix, 32  
graphics 461  
Graphics Interchange Format (GIF) 76  
GreatDomains 41  
Greenwich Mean Time (GMT) 272, 276  
Grid Panel JSF component 820  
GridView ASP.NET data control 753  
**groove** value (**border-style** property) 119  
GROUP BY clause 591  
grouped object 467, 468  
grouping element 114  
groups.google.com 670  
guestbook on a website 753  
Guggenheim Museum 26  
GUI (graphical user interface) xix  
GUI component 144  
Guild Wars 42  
Guru.com 43
- H**
- h1** heading element 72  
**h6** heading element 72  
Habbo 41, 44  
Halo Classic theme 558  
Halo theme 558  
**Hand tool** 458  
**handheld** media type 122  
handle an event in a child element 343  
handle **playButton** click event 498  
Handle the map functions for the Address Book application 541  
Handle the map functions for the Weather application 553  
Hansson, David Heinemeier (creator of Ruby on Rails) 33, 653  
**has\_many** method 675  
**Hashes** 657  
**Hashtable** class 743  
**HBox** element 522  
    **hideEffect** attribute 551  
    **showEffect** attribute 551  
**</head>** tag 136  
**<head>** tag 134  
**head** element 70, 104, 703  
head section 70  
header 664  
header cell 86  
**headerText** attribute 536  
heading 72, 72  
heading element 72, 191  
Heading elements h1 through h6 73  
**heading.html** 73  
**height** attribute 522  
**height** attribute of **img** element 76  
**height** property 116  
helper 673  
Hewlett-Packard 22  
hex 80  
hexadecimal 227  
hexadecimal escape sequence 285  
hexadecimal notation 457  
hexadecimal value 80  
**hidden**  
    element 738, 834  
    field 738, 834  
**hidden** form element 738, 834  
**hidden** input 89  
**hidden** input in an XHTML form 707  
**hidden** value (**border-style** property) 119  
hide global variable names 224  
**hideEffect** attribute 551  
hierarchy 352  
**highField** attribute 566  
Highrise (37Signals) 30  
**history** object 322  
hit area 463  
hit state 463, 489  
**hitTestObject** ActionScript collision detection method 514  
Hitwise 19, 20, 23, 41  
horizontal positioning 115  
horizontal rule 81, 81  
horizontal tab 140  
**horizontalAlign** attribute 522  
**horizontalAxis** element 566  
**HorizontalList** element 524  
    **dataProvider** attribute 524  
    **itemRenderer** element 524  
    **selectedIndex** attribute 524  
    **selectedItem** attribute 524  
host 571  
hostname 571  
hours since midnight 272  
housingmaps.com 35  
**hover** pseudoclass 108, 124  
hovering 337  
HowStuffWorks (content network) 13  
**<hr />** element (horizontal rule) 81  
**href** attribute of a element 73, 98  
**href** property of an a node 315, 322  
**HSI** der element 524, 524  
    **change** attribute 522  
    **labels** attribute 523  
    **liveDragging** attribute 522  
    **maximum** attribute 522  
    **minimum** attribute 522  
    **showDataTip** attribute 524  
    **snapInterval** attribute 524  
    **tickInterval** attribute 524  
    **value** attribute 522  
**.htm** (XHTML filename extension) 69  
**.html** (XHTML filename extension) 69  
HTML (Hypertext Markup Language) 68  
**html** element 70, 703  
HTML markup methods of the String object 269  
HTML wrapper for **coverViewer.swf** 526  
HTTP (Hypertext Transfer Protocol) 538, 570, 738, 834  
    being used with firewalls 914  
    header 572  
    method 571  
    request type 573, 573  
    stateless protocol 834  
    transaction 571  
<http://www.adobe.com/2006/mxml> namespace 522  
<http://www.w3.org/2001/XMLSchema> (XML Schema URI) 367  
**HttpCookie** class 743, 745, 746  
    **Domain** property 746  
    **Expires** property 746  
    **Name** property 745, 746  
    **Path** property 746  
    **Secure** property 746  
    **Value** property 745, 746  
**HttpCookieCollection** class 745  
**httpd.conf** Apache web server configuration file 581

**HTTPService** element 538  
 URL attribute 538  
**HTTPService**, Validators, and new UI elements 531  
**HttpServletRequest** class 844  
**HttpServletResponse** class 842  
**HttpSession** 919  
**HttpSessionState** class 746, 747, 748, 750, 753  
 Counts property 750  
 IsCookieless property 750  
 IsNewSession property 750  
 IsReadOnly property 750  
 Keys property 751, 753  
 SessionID property 750, 751  
 Timeout property 750, 751  
**HugeInteger** web service that performs operations on large integers 896  
“human bandwidth” 8  
hyperlink 73, 271  
HyperLink ASP.NET Web control 721  
**Hyperlink** JSF component 820, 825  
hyperlink location 263  
hyperlink target 271  
hypertext link 488  
Hypertext Transfer Protocol (HTTP) 570, 573, 738, 834

**I**

IAC (InterActiveCorp) 10  
IBM 22, 31  
Ice theme 558  
IceRocket (blog search engine) 19, 41  
id attribute 114, 522  
ID attribute of an ASP.NET web control 704  
id property of a DOM node 311  
identifier 142, 154  
identifier element (MathML) 374  
identifier followed by a colon 196  
IDEs  
 Netbeans 893  
if control structure 491  
if selection statement 154, 155, 156, 159  
if single-selection statement 148  
if...else double-selection statement 154, 156, 169, 658  
IIS (Internet Information Services) 570  
Image ASP.NET web control 720  
Image element 522  
 completeEffect attribute 524  
percentHeight attribute 522  
percentWidth attribute 522  
source attribute 523  
image hyperlink 79  
Image JSF component 820, 824  
Image object 337  
 src property 338  
image/gif 104  
Images as link anchors 78  
images collection of the document object 313, 322  
images in Web pages 76  
images in XHTML files 76  
ImageUrl element in an AdRotator advertisement file 724

ImageUrl property of an Image web control 720  
img element 76, 77, 79, 111  
 alt attribute 77  
 height attribute 76  
img elements in an XHTML document 322  
Implement the actions of each button 544  
implementation property of the document object 394  
#IMPLIED keyword (DTD) 363  
import 496  
Import to Stage... 473  
Important methods of class  
 HttpCookie 746  
importStylesheet method of the XSLTProcessor object (Firefox) 409  
Impressions element in an AdRotator advertisement file 724  
IMVU 41  
in (inches) 109  
increment 175  
increment control variable 177, 181  
increment expression 180  
increment operator (++) 170  
increment section of for statement 180  
increment.html 171  
Indeed 10, 40, 42  
indentation body of control structure 177  
index 262  
index in an array 233  
index value 634  
indexOf method of the String object 262, 265, 266  
indices for the characters in a string 264  
infinite loop 160, 166, 179, 191  
infinity symbol 590  
information hiding 258  
information tier 574  
infrastructure for open source projects 42  
infrastructure web services 36  
inherit a style 106  
inheritance 106, 258, 519  
Inheritance in style sheets 107  
Inherits attribute of ASP.NET page 703  
Init event of an ASP.NET Web page 705  
\_init method 810, 841  
\_init method 810, 816  
initial value 175, 178  
initial value of control variable 175, 181  
initialization 179  
initialization expression in for statement header 180  
initialize method 659  
initializer list 237  
initializer method for an object 273  
Initializing multidimensional arrays 250  
Initializing the elements of an array 235, 238  
initiate a script 327  
Inkscape 76  
inline-level element 114  
inline model of event registration 324  
inline scripting 134  
inline style 102, 105  
inline styles override any other styles 102  
inner for statement 196, 197, 252  
INNER JOIN SQL clause 591, 596  
innerHTML property (DOM) 421  
innerHTML property of a DOM node 221  
innerText 294, 328  
input 738, 834  
input and password text field creation 479  
input element 90  
 maxLength attribute 91  
 name attribute 91  
input text field 478  
Insert Keyframe 464  
insert layer 467  
INSERT SQL statement 591, 597  
insertBefore method of a DOM node 312  
insertBefore method of a Node 396  
InsertCommand property of a SqlDataSource ASP.NET control 760  
inserting a keyframe 464  
Inserting special characters 80  
InsertQuery property of a SqlDataSource ASP.NET data control 758  
inset value (border-style property) 119  
installing Rails 661  
instance 461, 462  
instance name 471  
instance variables 659, 665  
Instant Rails 654  
instantiated 259  
instantiating an object 235  
Institutional theme 558  
integer 610  
integers 145  
integral symbol (MathML) 374  
interaction between a web service client and a web service 894  
interactive animated movies 453  
InterActiveCorp (IAC) 10  
interest rate 183  
interface 258, 586  
Interfaces  
 CachedRowSet 863  
 MessageContext 919  
 RequestContext 921  
 Servlet 800  
 WebServiceContext 919  
internal hyperlink 97  
Internal hyperlinks to make pages more navigable 95  
internal linking 68, 95  
internal pointer 617  
Internet bookmark 28  
Internet Explorer 68, 77, 353  
 Developer Toolbar 300  
 Developer Toolbar  
 (go.microsoft.com/fwlink/?LinkId=92716) 300  
Internet Explorer 7  
 load XML document dynamically with JavaScript 394  
Internet Information Services (IIS) 570, 576  
Internet Information Services (IIS) Manager 579

- Internet radio 25, 42  
     Last.fm 42  
     Pandora 42
- Internet Server Application Program Interface (ISAPI) 575
- Internet Services Manager 576
- Internet TV 24, 42  
     Brightcove 42  
     Joost 42
- Internet video 23, 42  
     Yahoo! Video 42  
     YouTube 42
- interpolation 609, 621, 656
- interpret 134
- interpret <body> 145
- interpret <head> 145
- interpreted language 653
- Interstitial ads 40
- in-text contextual advertising 40
- IP address 571
- iPhone 22
- iPod 25
- IRB (Interactive Ruby) 655
- ISAPI (Internet Server Application Program Interface) 575
- `IsCookieless` property of `HttpSessionState` class 750
- `isFinite` function 226
- `isNaN` 278
- `isNaN` function 226
- `isNewSession` property of `HttpSessionState` class 750
- `IsPostBack` property of `Page` class 732
- `IsReadOnly` property of `HttpSessionState` class 750
- `isset` function 640
- `italic` value (`font-style` property) 116
- `item` method of a DOM collection 315
- `item` method of a `NodeList` 397
- `item` subelement of `channel` child element of an `rss` element 402
- `itemRenderer` element 524
- iteration of the loop 175, 178, 180
- iteration through an array 617
- iterative solution 228
- iTunes 25, 42
- J**
- Java 519  
     Java Runtime Edition 1.4.2\_08 519
- Java BluePrints 33, 802, 872
- Java BluePrints Ajax component library 859
- Java DB 604, 859
- Java EE 5 (`java.sun.com/javaee`) 800
- Java Enterprise Edition (Java EE) 800
- Java programming language 68
- Java Runtime Edition 1.4.2\_08 519
- Java Studio Creator 2  
     binding a JSF `Table` to a database table 862
- Java Technology and Web Services  
     Overview (`java.sun.com/webservices/overview.html`) 893
- JavaBean 801, 803, 805
- JavaFX 33
- JavaOne conference 7
- JavaScript xix, xx, 32, 70, 412, 473, 575, 802
- JavaScript 2 518  
     E4X (ECMAScript for XML) 531
- JavaScript applying an XSL transformation 405
- JavaScript events 324
- JavaScript interpreter 133, 136
- JavaScript keywords 154, 155
- JavaScript Object Notation (JSON) 39
- JavaScript scripting language 132
- JavaScript single-line comment 134
- `javascrip_include_tag` helper method 687
- JavaServer Faces (JSF) xx, 33, 413, 802  
     Ajax-enabled components 872  
     `Button` component 820, 825  
     components 802  
     `Drop Down List` component 820, 824  
     `Grid Panel` component 820  
     `Hyperlink` component 820, 825  
     `Image` component 820, 824  
     `Label` component 820, 825  
     `Listbox` component 843  
     `Message` component 825  
     `Message Group` component 864  
     `Meta` component 819  
     `Radio Button Group` component 820, 824  
     `Static Text` component 804, 806, 820  
     `Table` component 860  
     `Text Field` component 820, 824
- JavaServer Pages (JSP) 801, 802  
     action 801  
     directive 801  
     root element 804  
     scripting element 801  
     XML declaration 804  
     `xmlns` attributes 804
- JavaServer Pages Standard Tag Library (JSTL) 801
- `javax.faces.validators` package 825
- `javax.servlet` package 800
- `javax.servlet.http` package 800, 842, 844
- `javax.servlet.jsp` package 801
- `javax.servlet.jsp.tagext` package 801
- JAX-WS 2.0 894
- JBoss Application Server 899
- job boards 42  
     CareerBuilder 42  
     Dice 42  
     Monster 42
- job search engines 42  
     Indeed 42  
     Jobster 42  
     SimplyHired 42
- Jobs, Steve 25
- Jobster 42
- `join` method 268
- `join` method of an `Array` object 246
- joining database tables 589, 596
- Joost 24, 42, 46
- journalism 17
- JPEG (Joint Photographic Experts Group) 76
- JSF (JavaServer Faces) xx, 802
- JSF components 802, 806
- JSF elements  
     `webjsf:staticText` 868  
     `webjsf:tableRowGroup` 868  
     `webuijsf:table` 868
- JSF Expression Language 805
- JSON 949
- JSON (JavaScript Object Notation) 39
- JSP (JavaServer Pages) 801
- JSP container 801, 802  
     .jsp filename extension 802
- `jsp:directive.page` element 804
- `jsp:root` element 804
- JSTL (JavaServer Pages Standard Tag Library) 801
- Jupiter 42
- K**
- Kayak 10, 44
- Kazaa 42
- KDE  
     Konqueror web browser 567
- key function 617
- key-value pair 743
- key/value pair 841, 850
- `keyCode` property of an `event` object 333
- keyframe 459, 463
- Keyframe added to the timeline 459
- Keys property of `HttpSessionState` class 751, 753
- keyword 142, 154
- Keywords  
     `catch` 421  
     `try` 421
- keywords in PHP 614
- Koders 44
- Konqueror web browser 567
- Kontera 40, 42
- Krugle 44
- L**
- label 196
- `Label` element 522, 524  
     `text` attribute 524
- `Label` JSF component 820, 825
- `Label` web control (ASP.NET) 706
- labeled `break` statement 195
- Labeled `break` statement in a nested `for` statement 195
- labeled compound statement 196
- labeled `continue` statement 196
- Labeled `continue` statement in a nested `for` statement 196
- labeled for statement 196, 197
- labeled repetition statement 196
- labeled statement 195
- `Labels` attribute 523
- `labs.adobe.com/technologies/air/` 567
- `labs.adobe.com/technologies/flex/` 567
- `labs.adobe.com/technologies/flex/sdk/flex3sdk.html` 567

- `1abs.adobe.com/wiki/index.php/AIR:Developer_FAQ` 567
- `1abs.adobe.com/wiki/index.php/Flex_3:Feature_Introductions` 567
- `1abs.adobe.com/wiki/index.php/Flex_3:Release_Notes#Installation_Instructions` 567
- Language attribute in a Page directive 703
- large relative font size 105
- larger 260
- larger relative font size 105
- Lasso tool 474
- Last.fm 14, 25, 42, 43
  - scrobbing 25
- lastChild property of a DOM node 395
- lastChild property of a Node 396
- lastIndexOf method of the String object 263, 265, 266, 267
- Laszlo Webtop 29
- LaTeX software package 372
- layer 467
- layer overlapping elements 113
- layout 664
- Layout that displays a greeting 665
- Layout that displays the logged in user for every Forums action 681
- lead generation 40
- left margin 113, 114, 115
- left value (`text-align` property) 118
- left-hand-side expression 172, 233
- LendingTree.com 10
- length method of a Text node 398
- length property 258
- length property of a DOM collection 315
- length property of a NodeList 397
- length property of an Array object 233, 237, 240
- Length Validator JSF component 825, 830
- Lerdorf, Rasmus (PHP creator) 607
- letter 261
- letters 142
- levels of nesting 177
- li (list item) element 81
- Library panel 462, 463
- licenses
  - BSD license 31
  - Creative Commons 31
  - GNU Free Documentation License 31
  - GNU Lesser General Public License 31
  - GPL (GNU General Public License) 31
  - MIT license 31
- life-cycle methods 810, 806, 816
  - destroy 811, 816
  - init 810, 816
  - preprocess 810, 816
  - prerender 810, 816
- LifeTips (content network) 13, 42
- lighter value (`font-weight` property) 104
- lightweight business model 6
- LIKE SQL clause 593, 594, 595
- LimeWire 42
- Linden Labs 22
- line break 94
- line break XHTML tag 138
- line type 471
- Lineage 42
- linear gradient 460
- linear gradients 460
- LinearAxis element 566
  - `baseAtZero` attribute 566
- LineChart element 559, 566
  - `dataProvider` attribute 566
  - `showDataTips` attribute 566
- LineSeries element 566
  - `showDataEffect` attribute 566
  - `yField` attribute 566
- line-through value (`text-decoration`) 106
- Lingo (VoIP) 39, 44
- link 271
- link 271
- link baiting 11
- link building 11
- link element 110
- link method of the String object 263
- link popularity 11
- Link view of Script.aculo.us demo 693
- link\_to method 678
- LinkButton element 530
- LinkedIn 5, 21, 43
- linking external style sheets 109, 110
- Linking to an e-mail address 75
- links 73
- links collection of the document object 313, 322
- links.html 74
- LinkShare 40, 41
- linrails.thembird.com 654
- Linux 14, 31, 581
- list item 81
- list.html 83
- Listbox JSF component 843
- ListBox web control 751
- listen for events 220
- ListItem ASP.NET control 720
- list-style-type CSS property 188
- literal 135
- literal characters 621
- live-code approach xix, xxiii
- Live Search (Microsoft) 9
- LiveCycle Data Services ES 520
- livedocs.adobe.com/flex/201/
  - html/
    - charts\_intro\_108\_03.html
- livedocs.adobe.com/flex/201/
  - html/wrapper\_131\_05.html
- livedocs.adobe.com/flex/201/
  - langref/mx/effects/easing/package-detail.html
- livedocs.adobe.com/flex/201/
  - langref/mx/effects/package-detail.html
- liveDragging attribute 522
- LiveJournal 16
- Load event of an ASP.NET Web page 706
- load method of the ActiveXObject object 394
- Loader class 491
- local 224
- local search 37
- local time zone method 272
- local variable 205, 225
- local variable names 224
- local web servers 576
- localhost 576, 709
- localization 37
- location in memory 142, 244
- Location Intelligence conference 37
- location object 322
- location-based search 10
- Location-Based Services (LBS) 37
- lock/unlock layer 508
- locomotive.raaum.org 654
- log 260
- LOG10E 261
- logarithm 260
- logic error 146, 147, 160, 180
- logical AND (`&&`) operator 197, 198, 199
- logical negation (!) operator 198, 199
- logical NOT 198
- logical operator 197, 198, 200, 201
- logical OR (`||`) 198, 199
- logical OR (`||`) operator 198
- Login ASP.NET control 765, 781
- login control in ASP.NET 765
- Login form used to send data to the user controller 672
- LoginName ASP.NET login control 783
- LoginStatus ASP.NET login control 783
- Long Range Validator JSF component 825
- Long Tail 6, 15
- loop body 180
- loop-continuation condition 177, 179, 182, 190, 191
- loop-continuation test 191, 192, 196
- loop counter 175
- loopback address 576
- loosely typed 610
- loosely typed language 147
- lose focus 346
- lowercase letter 136, 142
- lowField attribute 566
- Lulu 43
- lvalue 172

## M

- m-by-n array 249
- Ma.gnolia 5
  - ma.gnolia 29, 43
- .Mac 44
- Mac OS X 581
- Macintosh 31
- Macromedia 518
- Magic Wand 474
- Mahalo 43
- Mahalo (people-assisted search) 7
- mailto: URL 75
- Main view for Flickr Photo Viewer 697
- main.htm1 70
- mainstream media 17
- making multiple selections with the selection tool 459
- many-to-many relationship 591
- Map Marker JSF component 882

- Map Viewer** JSF component 872, 882  
**MapController** function 543  
**MapQuest** 38  
**margin** 118  
**margin** property (block-level elements) 121  
**margin-bottom** property 121  
**margin-left** property 121  
**margin-right** property 121  
**margins** for individual sides of an element 121  
**margin-top** property 121  
**markup** in XML 348, 351  
**markup language** 68  
**MarkupMethods.html** 270  
**Marriott International** 18  
**mashup** 6, 35, 42
  - chicagocrime.org 35
  - examples chart 35
  - feedmashr.com 35
  - housingmaps.com 35
  - paul.kedrosky.com/publicloos/ 35**Programmable Web** 42
  - secretprices.com 35**masking effect** 474, 476  
**masking layer** 474  
**mass media** 17, 24  
**massively multiplayer online games** 42  
**Master directive** 775  
**.master** filename extension 774  
**master page** in ASP.NET 769  
**MasterPageFile** property of a Page directive 777  
**match** attribute 380  
**Match.com** 10  
**Math class** 504, 944  
**Math object** 201, 257, 260, 261
  - floor** method 211, 215, 218
  - max** method 210
  - pow** method 204
  - random** method 211, 215, 226
  - round** method 204**Math object** methods 260  
**Math.E** 261  
**Math.floor** 241  
**Math.LN10** 261  
**Math.LN2** 261  
**Math.LOG10E** 261  
**Math.LOG2E** 261  
**Math.PI** 261  
**Math.random** 241  
**Math.random** function 471, 491  
**Math.sqrt** 260  
**Math.SQRT1\_2** 261  
**Math.SQRT2** 261  
**mathematical calculation** 204, 260  
**mathematical constant** 261  
**Mathematical Markup Language (MathML)** 372  
**MathML** 349, 372
  - &delta;** entity reference 375
  - &int;** entity reference 375
  - .mml** filename extension 373
  - entity reference 375
  - entity reference
    - &InvisibleTimes;** 374
  - identifier element 374
  - integral symbol 374
  - frac** element 374
- mi** element 374  
**mn** element 373  
**mo** element 373  
**mrow** element 375  
**msqrt** element 375  
**msubsup** element 375  
**msup** element 374  
**square-root symbol** 374  
**symbolic representation** 374  
**MathML (XML vocabulary)** 38  
**Matsunomo, Yukihiko "Matz"** 653  
**max** 260
  - max method of the **Math** object 210
  - maximum** attribute 522
  - maximum** function 208
  - maxLength** attribute of **input** element 91
  - maxOccurs** XML Schema attribute 369
  - meaningful variable names 221
  - Mechanical Turk** (Amazon) 36
  - media types** 122, 122
  - MediaWiki** 15
  - medium** relative font size 105
  - medium** value 119
  - Meebo** 44
  - message** 135, 258
  - message dialog** 216
  - message forum** 670
    - message forum case study 670
  - Message Group** JSF component 864
  - Message** JSF component 825
    - Message layout that links a style in and displays a message 686
  - Message** model containing validation and initialization functionality 675
  - MessageContext** interface 919
    - get** method 919
  - messages.yahoo.com** 670
  - MessagesController** that implements CRUD functionality 683
  - Meta component (JSF)** 819
  - meta** element 98, 99
    - content** attribute 98, 99
    - name** attribute 98
  - meta** tags provide keywords and a description of a page 98
  - metacharacters** 621
  - Metcalf's Law** 19
  - method** 135, 147, 204, 226, 259
    - method = "get"** 89
    - method = "post"** 89, 626
    - method** attribute 89
    - method** attribute of XHTML element form 708, 812
  - Method** calls on numeric and string data 656
  - method** **prompt** 147
  - method** UTC 276
  - method** **writeln** 147
  - frac** MathML element 374
  - mi** MathML element 374
  - microblogging** 23
  - microformats** 47
    - adr** 47
    - hresume** 47
    - xfolk** 47
  - Microsoft** 18, 30, 31, 33, 42, 43, 44
    - adCenter** 10
    - Live Search** 9
    - Windows** 31
  - Microsoft Internet Explorer** 7 xx
  - Microsoft MIX conference** 5, 7, 39, 47
  - Microsoft Office Live** 43
  - Microsoft Silverlight** 691
  - Microsoft SQL Server** 586, 604
  - Microsoft Tech Ed conference** 7
  - Microsoft Visual Studio** 33
  - Microsoft Visual Web Developer** 33
  - Microsoft Windows Live** 43
  - Microsoft XML Core Services (MSXML)** 350, 353
  - middle tier** 574
  - MIME (Multipurpose Internet Mail Extensions)** 104, 110, 572, 739, 835
  - minimum** attribute 522
  - minInclusive** XML Schema element 371
  - minOccurs** XML Schema attribute 369
  - minus sign (-)** for container elements in Internet Explorer 353
  - mm** (millimeters) 109
  - .mml** filename extension for MathML documents 373
  - mn** element 375
  - mn** MathML element 373
  - mo** MathML element 373
  - mobile blogging** 18
  - mobile device** 18, 22
  - mobile social networking** 42
    - Dodgeball.com 42
    - MocoSpace 42
    - Twitter 42
  - MobiTV** 24
  - moblogging** 18
  - MocoSpace** 42
  - mode** attribute of element
    - authentication** in **Web.config** 773
  - model generator** 670
  - Model-View-Controller (MVC)** 660
  - moderation** 15
  - modifying button states with a button's editing stage** 464
  - modifying the size of a shape with the Property Inspector** 460
  - module** 204, 575
  - modulus operator** 147
  - MOG** 42
  - monetization** 39
  - monospace** 105
  - Monster** 10, 13, 42
  - Moore's Law** 4
  - Mosaic browser** 3
  - Motion Tween**
    - Rotate** option 486
  - motion tween** 468
  - mouse button pressed down** 346
  - mouse button released** 346
  - mouse cursor** 139, 140
  - mouse cursor over an element** 108
  - mouse events** 324
  - mouse pointer** 140
  - MouseEvent** class (Flash) 496
  - Movable Type** 18
  - move ball and set speed** 506
  - move target, set direction and speed, and keep track of number of blocks hit** 499
  - movie clip** 462
  - movie clip symbol** 462, 480

movie clips 461  
 movie dimensions 458  
**Movie Explorer** 462  
**Movie Explorer for CeoAssistant.fla**  
 463  
**Movie Properties** dialog 456  
**MovieLens** 43  
 movies 453  
 moving the mouse 324  
**Mozilla Firefox** 68  
**Mozilla Firefox 2 xx**  
**Mozilla Foundation** 31, 43  
**Mozilla** project 376  
 mp3 25  
**MP3** audio compression format 465  
**mrow** MathML element 375  
[msdn.microsoft.com/vstudio/express/vwd/default.aspx](http://msdn.microsoft.com/vstudio/express/vwd/default.aspx) 702  
**MSN** 9, 43  
**MSN Hotmail** 44  
**MSN Virtual Earth** 38  
**msqrt** MathML element 375  
**msubsup** element 375  
**msubsup** MathML element 375  
**msup** MathML element 374  
**MSXML** (Microsoft XML Core Services) 350, 353  
**MSXML** parser 394  
**MTV** 24  
**Muinár** ([www.sounds.muinár.com](http://www.sounds.muinár.com)) 465  
 multidimensional array 249  
 multiline comment 609  
 multiline comment /\* and \*/ 143  
 multimedia xx  
 multiple conditions 197  
 multiple-selection statement 154  
 multiple-selection structure 189  
 multiplication assignment operator (\*=) 612  
**Multipurpose Internet Mail Extensions (MIME)** 572, 739, 835  
**Multipurpose Internet Mail Extensions (MIME)** type 104, 110  
 multitiier application xx, 574  
 music distribution sites 42  
**MVC** (Model-View-Controller) 660  
**mx** prefix 522  
**MXML** markup language 518  
**xmlc**  
 theme parameter 558  
**xmlc** command 525  
**xmlc** running in Windows XP  
 Command Prompt 525  
**mysimon** 10  
**MySpace** 5, 19, 20, 24, 43  
**MySpaceTV** 24  
**MySQL** 31, 586, 600, 607, 630, 654  
**MySQL 5.0 Community Edition** 600  
**MySQL** command-line client tool 601, 602  
**MySQL Conference and Expo** 7  
**MySQL** database 602  
**MySQL** reference manual 600  
**mysql\_close** function 634  
**mysql\_connect** function 633  
**mysql\_error** function 634  
**mysql\_fetch\_row** function 634  
**mysql\_num\_rows** function 634  
**mysql\_query** function 634  
**mysql\_select\_db** function 634  
**mysqld-nt.exe** 601  
**MyStrands** 43

## N

n-tier application 574  
 name attribute 530  
 name attribute (XPath) 384  
 name attribute of **input** element 91  
 name attribute of **meta** element 98  
 name node-set function 384  
 name of an array 233  
 name of an attribute 71  
 name of the anchor 271  
 name property of an **Attr** object 397  
 Name property of class **HttpCookie** 745  
 Name property of **HttpCookie** class 746  
 name XML Schema attribute 368  
 named constant 612  
**namedItem** method of a DOM collection 315  
 namespace 358  
 namespace prefix 359, 360, 370  
**Namespace** property of a **WebService** attribute 951  
**Namespaces**  
 System.Data 603  
 System.Data.OleDb 603  
 System.Data.SqlClient 603  
 System.Web.UI 705  
 System.Web.UI.WebControls 706  
 naming collision 358  
 NaN 226, 262, 265, 278  
 NaN (not a number) 146  
 NaN (not a number) 163  
 natural linking 11  
 natural logarithm 260  
 natural logarithm of 10 261  
 natural logarithm of 2 261  
**nav.html** 78  
**navigateToURL** function 543  
**navigateToUrl** function 491  
**NavigateUrl** element in an **AdRotator** advertisement file 724  
**NavigateUrl** property of a **HyperLink** control 721  
 navigation bar on a website 774  
**NAVTEQ** 38  
 Nested and ordered lists. 83  
 nested element 71, 352  
 nested for statement 195, 196, 252  
 nested for...in statement 251  
 nested if or if...else statement 197  
 nested if statement 151, 158  
 nested if...else statements 157  
 nested list 82, 109  
 nesting 177  
 nesting multi-line comments 143  
**Netbeans** 799  
 Visual Web Pack 799  
**Netbeans 5.5 IDE** 893  
 create a desktop application 905  
 create a web application 895  
**New JFrame Form** dialog 908  
**New Project** dialog 895  
**New Web Service Client** dialog 905  
 Netbeans 5.5 IDE (cont.)  
 New Web Service dialog 895  
 Project Properties dialog 901  
 Web Application project 895  
**Netflix** 6, 43  
**NetNewsWire** 42  
**Netscape** 25, 32  
 LiveScript 32  
**Netvibes** 43  
 network effect 19  
**Network Solutions** 42  
 new **Date** object 273, 276  
**New Document** dialog 456  
**New JFrame Form** dialog 908  
 new layer 467  
 new operator 150, 234, 237, 273  
**New Project** dialog 895  
 new symbol 480  
**New Web Service Client** dialog 905  
**New Web Service** dialog 895  
 newline character (\n) 140  
 newline escape sequence 654  
 Newmark, Craig 15  
 News Corp 20  
 Newsvine 25  
*Newsweek* 36  
 next function 617  
 nextInt method of class **Random** 944  
**nextSibling** property of a DOM node 395  
**nextSibling** property of a Node 396  
**nil** 655  
**9rules** (blog network) 13, 18, 41  
 no color 458, 475  
**No color** option 475  
 no-repeat property 116  
 node (DOM tree) 385  
 Node object 396  
 Node object methods 396  
 node-set function 384  
 node set of an **xsl**  
 for-each element 380  
**NodeList** object 396, 397  
**NodeList** object methods 397  
**nodeName** property of a DOM node 395  
**nodeName** property of a Node 396  
**nodeType** property of a DOM node 395  
**nodeType** property of a Node 396  
**nodeValue** property of a DOM node 395  
**nodeValue** property of a Node 396  
 nonbreaking space (&nbsp;) 395  
 none value (**border-style** property) 119  
 none value (**font-style** property) 116  
 nonvalidating XML parser 350  
**normal** value (**font-weight** property) 104  
 not a number 146  
 not a number (NaN) 163  
 Notepad text editor 69  
 nouns in a system specification 259  
**null** 144, 147  
**Number** object 146, 185, 277  
 toFixed method 185  
**Number.MAX\_VALUE** 278  
**Number.MIN\_VALUE** 278  
**Number.NaN** 278  
**Number.NEGATIVE\_INFINITY** 226, 278

Number.POSITIVE\_INFINITY 226, 278  
 numbered list 188  
 numeric character reference 80

**O**

O'Reilly Media 3, 29, 42  
 O'Reilly, Tim 4, 5, 34  
 object 135, 226, 257  
 object (or instance) 257, 258  
 object animation in ActionScript 496  
 object-based programming language 257  
 Object class 524  
 object hierarchy 299  
 object orientation 257  
 object-oriented design (OOD) 258  
 object-oriented language 259  
 object-oriented programming (OOP) 259  
 Object-Relational Mapping (ORM) 665  
 ObjectDataSource ASP.NET data control 783  
 oblique value (`font-style` property) 116  
 obtaining user input through forms 626  
 occurrence indicator 363  
 octal 227  
 off-by-one error 179  
 Offline Problem 33  
 OK button 140  
 ol element 82  
`omit-xml-declaration` attribute 380  
 ON clause 596  
 onabort event 346  
 onblur event 338, 339  
 onchange event 346  
 onclick event 220, 265, 266, 268, 269, 283, 327, 345, 346  
 ondblclick event 346  
 1000tags.com 41  
 one-to-many relationship 590  
 onfocus event 338, 339, 346  
 onfocusblur.html 339, 341  
 onkeydown event 346  
 onkeypress event 346  
 onkeyup event 346  
 online advertising 42  
   DoubleClick 42  
   Federated Media 42  
   Google 42  
   Kontera 42  
   Microsoft 42  
   Quigo 42  
   Tribal Fusion 42  
   ValueClick 42  
   Vibrant Media 42  
   Yahoo! 42  
 online auctions 42  
   Amazon Auctions 42  
   eBay 42  
   Overstock.com 42  
 online classifieds 42  
 Online Quiz graded with JavaScript 253  
 online quiz in a child window 294  
 online quizzes 253  
 online survey sites 43  
   Survey Monkey 43  
 onload event 224, 327, 346  
 onload.html 327

onmousedown event 346  
 onmousemove event 328, 332, 333, 346  
 onmouseout event 333, 338, 346, 417  
 onmouseover event 333, 337, 338, 346, 417  
 onmouseup event 346  
 onReadyStateChange property of the XMLHttpRequest object 422  
 onreadystatechange property of the XMLHttpRequest object 421  
 onreset event 341, 346  
 onresize event 346  
 onselect event 346  
 onsubmit event 255, 341, 346  
 ontology 47  
 onunload event 346  
 OOD (object-oriented design) 258  
 OOP (object-oriented programming) 259  
 open method of the XMLHttpRequest object 422  
 open source 4, 5, 6, 25, 30, 43, 581, 600, 607  
   Apache 43  
   Free Software Foundation 43  
   Mozilla Foundation 43  
   Zend 43  
 Open Source (OSCON) conference 7  
 Open Source Initiative 31  
 open technology 348  
 openField attribute 566  
 OpenGIS Consortium 376  
 OpenLaszlo 29  
 operand 144  
 operation element 551  
 operation in the UML 258  
 operator precedence chart 649  
 Operators  
   function-call operator 207  
   new 234  
 operators  
   ! (logical NOT or logical negation) operator 198  
   && (logical AND) operator 197  
   || (logical OR) operator 197  
 operators.php 613  
 optimizing sound with the Sound Properties dialog 466  
 Option 845  
 option element (form) 95  
 Oracle Corporation 586  
 Orbitz 44  
 order attribute 384  
 ORDER BY SQL clause 591, 594, 595  
 ordered list 82  
 ordering of records 591  
 ORM (Object-Relational Mapping) 665  
 OSCON (O'Reilly Open Source Convention) 30  
 outer for loop 196, 197  
 Outline window 819  
 outset value (`border-style` property) 119  
 outsourcing marketplaces 43  
   Elance 43  
   Guru.com 43  
 Oval tool 458  
 over state 463  
 overflow boundaries 118  
 overflow property 118  
 overlapping text 114  
 overline value (`text-decoration`) 106  
 Overstock.com 42  
 Overture 9  
 OWL (Web Ontology Language) 47

**P**

p element 71  
 Packages  
   com.sun.rave.web.ui.appbase 806  
   com.sun.webui.jsf.component 806, 810  
   javax.faces.validators 825  
   javax.servlet 800  
   javax.servlet.http 800, 842, 844  
   javax.servlet.jsp 801  
   javax.servlet.jsp.tagext 801  
 Packet (VoIP) 39, 44  
 padding 118  
 padding-bottom property 121  
 padding for individual sides of an element 121  
 padding-left property 121  
 padding property (block-level elements) 121  
 padding-right property 121  
 padding-top property 121  
 page bean 803, 811  
 Page class 705, 732, 746  
   Session property 746  
 Page directive in ASP.NET 703  
 Page event lifecycle (ASP.NET) 707  
 Page Navigation file 814  
 Page, Larry 8  
 Page\_Init event handler 705  
 Page\_Load event handler 706  
 Page\_PreInit event handler 706  
 Page\_Unload event handler 707  
 Pageflakes 43  
 PageRank 8  
 PageSize property of a GridView ASP.NET control 789  
 PageSize property of a GridView control 789  
 paid blog posts 40  
   PayPerPost 40  
   ReviewMe 40  
   SponsoredReviews 40  
 Paint Behind 474  
 Paint Inside 474  
 Paint Selection 474  
 Palette 814, 817, 820  
 Pandora 42, 43  
 Panel element 522  
   height attribute 522  
   horizontalAlign attribute 522  
   title attribute 522  
   verticalAlign attribute 522  
   width attribute 522  
 panels 454  
 PanTool element 543  
 paragraph element 71  
 Parallel element 551  
 parameter 205

**Parameter** ASP.NET element 763  
**Parameters** panel in Flash CS3 456  
 parent 299  
 parent element 106, 353  
 parent node 385  
**parent** parameter 537  
 parent window 279  
 parent/child relationships between data 352  
 parentheses 198  
**parentNode** property of a DOM node 312, 396  
**parentNode** property of a Node 396  
 parse 276, 277  
 parsed character data 363  
**parseFloat** function 205, 208, 227  
**parseInt** function 145, 146, 163, 165, 227  
 parser 350  
 parsing 288  
 parsing a SOAP message 914  
 partial 680  
 partial class 705  
**partial** modifier 705  
 partial page update 414  
 partial RSS feed 403  
 Partial that contains a form used to add a new forum 680  
 participant in a virtual form 874  
 participatory journalism 17  
 pass-by-reference 243, 244  
 pass-by-value 243, 244  
 passing arrays 245  
 Passing arrays and individual array elements to functions 245  
 password 478  
**password** input 94  
**password\_field** method 672  
**Paste Frames** 475  
 patent  
     social networking 20  
 Path property of **HttpCookie** class 746  
 Path system variable 525  
 path to a resource 571  
 pattern matching 593  
 paul.kedrosky.com/publicloos / 35  
 payments 43  
     Google Checkout 43  
     PayPal 43  
 PayPal 43  
 PayPerPost 40  
 pc (picas—1 pc = 12 pt) 109  
**#PCDATA** keyword (DTD) 363  
 PDML (Product Data Markup Language) 372  
 people-assisted search 43  
 percent (%) SQL wildcard character 593  
 percent sign (%) remainder operator 147  
 percentage 108  
**percentHeight** attribute 522  
**percentWidth** attribute 522  
 Performance Tips overview xxv  
 performance-based advertising 40  
 performance-intensive situation 195  
 Perl-compatible regular expressions 619  
 permalink 17  
 perpetual beta 30  
 persistent cookie 637  
 persistent database connection 603  
 persistent information 834  
 personalization 737, 833  
 personalized start page 43  
 Pew Internet & American Life Project 4, 16, 18, 26  
 photo sharing 43  
     Flickr 43  
     Photobucket 43  
 Photobucket 43  
 Photoshop Elements 76  
 PHP xix, xx, 413, 473, 654  
 PHP (PHP: Hypertext Preprocessor) 607  
     .php extension 608  
 PHP: Hypertext Preprocessor (PHP) 607  
 PHP quantifiers 621  
 PHPNuke 31  
 PI 261  
 PI (processing instruction) 378  
 picture element (pixel) 108  
**picture.html** 76  
 piracy 25  
 pixel 76, 458  
 pixels 458  
 place holder in an initializer list 237  
 platform xx  
**play** function 491  
 playhead 454, 480  
 plus sign (+) for container elements in Internet Explorer 353  
 plus sign (+) occurrence indicator 363  
 PM 276  
 PNG (Portable Network Graphics) 76  
 podcast 25  
 podcasting 25, 28  
 POJO (Plain Old Java Object) 898  
 Policy Aware Web Project 46  
 polling 519  
**Polygon Mode of Lasso tool** 474  
 polymorphism 519  
 populating a **DataSet** 603  
**Popup Calendar** JSF component 872  
 pop-up window 328  
 Portability Tips overview xxv  
 Portable Network Graphics (PNG) 76  
 Portable Operating System Interface (POSIX) 619  
 position number 233  
**position** property 111  
 POSIX extended regular expressions 619  
**post** 677  
 post request type 626  
 post request type 89  
 postback 810  
 postback event of an ASP.NET page 732  
 postdecrement operator 170, 170  
 PostgreSQL 586  
 postincrement operator 170, 172  
 Pow method of class **Math** (Java) 944  
 pow method of **Math** object 185, 204, 260  
 power 260  
 Powerset (semantic search) 7  
 Practical Extraction and Report Language (Perl) 608  
**pre** element 140  
 precedence 172  
 precedence and associativity of operators 152, 173, 201  
 predecrement operator 170, 170  
 predicate 592  
**preg\_match** function 619  
 preincrement operator 170, 172  
**PreInit** event of an ASP.NET Web page 706  
 preloader 485  
 preloading images 338  
 premium content 40  
     “packaged” function 204  
**preprocess** method 810, 816  
**prerender** method 810, 816  
 presentation 372  
 presentation logic 575  
 Presentation MathML 372  
 presentation of a document 68  
 press a key 346  
 pressing keys 324  
**previousSibling** property of a Node 396  
 primary key 587, 590  
**primary** property of a JSF **Button** 860  
 principal 183  
**print** function 609  
**print** media type 122, 122, 124  
 printing on multiple lines with a single statement 138  
 printing one line with separate statements 136  
 privacy 19, 21, 23  
 privacy protection 738, 833  
 probability 211  
 Problogger 41  
 procedural programming language 259  
 processing instruction (PI) 378  
 processing instruction target 378  
 processing instruction value 378  
 processor 350  
 Product Data Markup Language (PDML) 372  
 program 132, 133  
 program control 132  
 program development 132  
 program-construction principles 175  
 Programmable Web 35, 42  
     Mashup Matrix 35  
 programmer-defined function 205  
 Programmer-defined function **square** 205, 206  
 Programmer-defined **maximum** function 209  
 Programming languages  
     Ruby 653  
**Progress Bar** JSF component 872  
**ProgressBar** element 551  
**Project Properties** dialog 901  
**Projects** window 814  
 prolog (XML) 352  
 promotion 737  
 prompt box used on a welcome screen 141  
 prompt dialog 141, 144, 145, 146, 208, 216  
 prompt method of **window** object 143, 144, 145  
 prompt to a user 143  
**Properties** class 841, 850  
     **getProperty** method 841  
     **setProperty** method 841  
 properties of the **Math** object 261

**P**roperties panel in Flash CS3 456  
properties separated by a semicolon 102  
**P**roperties window 712, 713, 817  
property 259, 495  
**P**roperty Inspector 456, 497  
protocol 89  
Protopage 43  
Prototype 413  
Prototype Ajax library 412  
prototype Ajax Updater object 688  
Prototype JavaScript framework 33  
proxy class for a web service 894  
proxy object handles the details of communicating with the web service 894  
pseudoclass 108  
pt (points) 105, 109  
pt measurement for text size 126  
publish 472  
Published Flash files 473  
published state 466  
publishing a web service 894, 900  
put method of interface RequestContext 921  
puts 654

**Q**

qualified name 596  
quantifier 621  
query 586, 587  
**Q**uery Builder dialog 957  
query string 573  
querying a database and displaying the results 632  
question mark (?) occurrence indicator 363  
queue parameter of visual\_effect method of Script.aculo.us Library 693  
Quigo 42  
quotation ('') mark 135  
quotemeta function 647

**R**

radial 460  
radial gradient 460  
radio button 255  
**R**adio Button Group JSF component 820, 824  
radio button object checked property 255 "radio" input 95  
**R**adioButtonList ASP.NET Web control 721  
radix 227, 278  
Rails 653  
rails application generator 661  
RailsConf (conference) 7  
Random class 944  
Random image generation using arrays 242  
Random image generation using Math.random 222  
random image generator 222  
random method 218, 241  
random method of the Math object 211, 215, 226  
raster graphic 458

**R**ating JSF component 872  
"raw" Ajax 412  
RDBMS (relational database management system) 574, 600  
RDF (Resource Description Framework) 47, 402  
RDF Site Summary (RSS) 402  
RDF triples 47  
**R**ead access permission 578  
readability 69  
readyState property of the XMLHttpRequest object 421, 422  
real estate 43  
RealEstate.com 10  
Really Simple Syndication 402  
reciprocal linking 11  
recognizing clients 738, 834  
recommender systems 43  
ChoiceStream 43  
CleverSet 43  
Last.fm 43  
MovieLens 43  
MyStrands 43  
Netflix 43  
Pandora 43  
StumbleUpon 43  
rectangle tool 459  
recursion step 227  
recursive base case 227  
recursive call 227  
recursive descent 384  
Recursive evaluation of 5! 228  
recursive function 227  
Red Hat 31  
Reddit 5, 25, 43  
Redfin 43  
redgreenblu.com/flickr/ 696  
refactoring 816  
Reference.com 42  
Refresh button 145  
refresh method of class CachedRowSetDataProvider 872  
Register mouseInHandler as MOUSE\_MOVE event handler 504  
register mouseInHandler as MOUSE\_MOVE event handler 504  
Register.com 42  
registering an event handler 324  
    inline model 324  
    traditional model 324, 327  
registering the event handler 220  
regular expression 618, 619, 830  
regular expressions in PHP 619  
**R**egularExpressionValidator  
    ASP.NET validation control 727  
relational database 586, 587  
relational database management system (RDBMS) 574, 586, 600  
relational database table 587  
relational operator 148, 149, 197, 198, 619  
relational operators and strings 262  
relationship between documents 110  
relative addressing (XPath) 380  
relative-length measurement 108, 116, 128  
relative positioning 113  
Relative positioning of elements 113

**r**elative value (position property) 114  
release a key 346  
reload an entire web page 413  
reload an XHTML document 145  
Reload button 145  
reload method of browser's location object 288  
remainder after division 147  
remainder operator (%) 147  
remixing 31  
remote web servers 576  
removeAttribute method of an Element 397  
RemoveChild element 530  
removeChild method of a DOM node 312  
removeChild method of a Node 396  
Renaming a layer 467  
render\_text 662  
rendered property of a JSF component 826  
rendering a view 663  
rendering an RSS feed in a web page using XSLT and JavaScript 406  
repeat value (background-repeat property) 116  
repeat-x value (background-repeat property) 116  
repeat-y value (background-repeat property) 116  
repeating infinitely 163  
repetition statement 160  
repetition structure 160, 191, 192  
replace method of the String object 263  
replaceChild method of a DOM node 312  
replaceChild method of a Node 396  
Representational State Transfer (REST) 36, 892, 949  
reputation systems 14, 43  
    Amazon 43  
    eBay 43  
**R**equest class  
    Cookies property 745  
request method 573  
request object 663  
Request object in ASP.NET 745  
request scope 803  
RequestBean 803  
RequestContext interface 921  
    put method 921  
#REQUIRED keyword (DTD) 363  
required attribute 538  
required property 829  
RequiredFieldValidator ASP.NET validation control 727  
Research Information Exchange Markup Language (RIXML) 376  
reset function 617  
reset input 91  
reset property of a JSF Button 860  
reset the ball to its original position 514  
reset\_session 672  
reset\_session 672  
Resize element 531  
    duration attribute 531  
    easingFunction attribute 531

Resizing an image with the free transform tool 475  
**@Resource** annotation 919  
 Resource Description Framework (RDF) 47, 402  
 resources 69  
 responding to user interaction 324  
**responseText** property of the XMLHttpRequest object 421, 422  
**responseXML** property of the XMLHttpRequest object 422  
 REST (Representational State Transfer) 36, 892  
 RESTful web services 36, 37, 949  
 restriction on built-in XML Schema data type 369, 371  
**result** attribute 551  
 result of a database query 592  
 result tree (XSLT) 377  
 return 140  
 return by reference 244  
**return** statement 207, 208  
 return value of an event handler 343  
 reuse 259  
**reverse!** 657  
 ReviewMe 40  
 Rhapsody 42  
`.rhtml` filename extension 663  
 RIA (Rich Internet Application) 32, 412, 518  
 Rich Internet Applications (RIAs) 32, 412, 518  
 Rich Site Summary 402  
 Rich Textarea Editor JSF component 872  
 rich welcome page using several JavaScript concepts 288  
**ridge** value (`border-style` property) 119  
**right margin** 113, 114, 115  
 Right Media Exchange 41  
**right** value (`text-align` property) 118  
 RIXML (Research Information Exchange Markup Language) 376  
 RIXML website ([www.rixml.org](http://www.rixml.org)) 376  
 Rocketboom 18  
 rolling a six-sided die 211  
 Rolling a six-sided die 6000 times 213  
 rollover 462  
 rollover effect 333, 337  
 rollover images 333  
 Rollyo 10  
 root element 804  
 root element (XML) 349, 352, 356  
 root node 300, 385  
 RoR 653  
 Rose, Kevin 24  
**Rotate** 486  
 Rotate option of Motion Tween 486  
 rotating the **cannon** object 504  
**rotation** property 504  
**rotation property** 504  
 round 211, 260  
 round method of a Ruby numeric value 656  
 round method of the Math object 204  
 row 587, 590, 591, 592, 593, 597, 598  
 rows 249  
**rows** attribute (`textarea`) 94  
 rows to be retrieved 591  
**rowspan** attribute (`tr`) 87  
 RSS 5, 18, 25, 35, 36, 39  
   ads 41  
   aggregator 39  
 RSS (RDF Site Summary) 402  
 RSS 2.0 sample feed 408  
 RSS aggregator 405  
**rss** root element 402  
 Ruby 33, 653  
 Ruby class 659  
 Ruby classes  
   **Array** 657  
   **Fixnum** 656  
   **Float** 656  
   **Hash** 657  
   **Object** 657  
   **String** 656  
   **Time** 683  
 Ruby interpreter 655  
 Ruby keywords  
   **class** 659  
   **def** 658  
   **end** 658  
   **nil** 655  
 Ruby method  
   **to\_s** method 659  
 Ruby on Rails xix, xx, 33, 413, 653  
   generators 666  
 Ruby on Rails methods  
   **find** 682  
   **save** 684  
   **server\_software** 663  
 Ruby operators  
   **=>** 657  
 ruby symbol 662  
 Ruby syntax  
   **@@** 659  
   dynamic typing 656  
   **if...else** 658  
   parentheses 654  
   semicolon 654  
 Rule of Entity Integrity 590  
 Rule of Referential Integrity 589, 591  
 Run Project option in Netbeans 900  
**Run scripts (such as ASP) access**  
   permission 578  
**runat** ASP.NET attribute 704  
 running a Ruby script 655  
 Running mxm1c with the theme parameter in Windows XP **Command Prompt** 558  
 running Ruby on Rails 663  
 runtime error 136

## S

SaaS (Software as a Service) 29, 43  
   Google 43  
   Microsoft Office Live 43  
   Microsoft Windows Live 43  
   Salesforce.com 43  
   37Signals 43  
   Zoho 43  
 Safari ([www.apple.com/safari](http://www.apple.com/safari)) 567  
 Safari Books Online 43  
 Salesforce.com 30, 43  
 same origin policy (SOP) 418  
**sans-serif** fonts 105, 124  
**sans-serif** font 460  
 savings account 183  
 scaffold (Ruby) 668, 676, 677  
**scaffold generator** 676  
 Scalable Vector Graphics (SVG) 372  
 scalars (scalar quantities) 246  
 scale factor 482  
 scaling factor 211, 215  
 scaling the range of random numbers 211  
 scenes 461  
 schema 38, 365  
 schema invalid document 365  
 schema repository 362  
 schema valid XML document 365  
 schema-valid XML document describing a list of books 366  
**schema** XML Schema element 367  
 Scoble, Robert 16  
 Scobleizer 41  
 scope 180, 223, 224  
 scope rules 224  
 scoping 224  
 Scoping example 224  
 screen coordinate system 333  
**screen** media type 122  
 screen resolution 108  
**screenX** property of an **event** object 333  
**screenY** property of an **event** object 333  
**<script>** tag 134, 135  
 script 70, 133  
**Script** element (ActionScript) 531, 540  
   **source** attribute 540  
 script element in ASP.NET 775  
**script** font 105  
 Script.aculo.us 6, 33  
 Script.aculo.us Ajax library 412, 413  
 Script.aculo.us Demo controller 692  
 Script.aculo.us JavaScript library 691  
 scripting element 801  
 scripting host 575  
 scripting language 132, 134, 653  
 script-level variables 223  
**ScriptManager** control 796  
 scrobbing (Last.fm) 25  
 scroll up or down the screen 115  
**scroll** value (**background-position** property) 116  
**scroll** value (**overflow** property) 118  
 scrolling the browser window 116  
 search engine 7, 8, 71, 98, 574  
   vertical search 10  
 Search Engine Land 12  
 search engine marketing (SEM) 11  
 Search Engine Marketing Professional Organization 12  
 search engine optimization (SEO) 11  
 search engine results page (SERP) 7  
 Search Engine Strategies conference 7, 12  
 Search Engine Watch 11, 12, 40  
 search engines 43  
   Ask 43  
   Google 43  
   MSN 43  
   Yahoo! 43  
 Searchblog 41  
 searching **Strings** with **indexOf** and **lastIndexOf** 266  
 SearchingStrings.html 266

Second Life 5, 22, 35, 41, 43, 44  
 secretprices.com 35  
**Secure** property of  
*HttpCookie* class 746  
 secure protocol 746, 846  
 security level 578  
**select** attribute (XPath) 384  
**select** attribute of *xsl:for-each*  
 element 380  
**select** element 95  
**SELECT** SQL keyword 591, 592, 594,  
 595  
**Select Value Text Field** JSF component  
 872  
**SelectCommand** property of a  
*SqlDataSource* ASP.NET control  
 760  
**selected** attribute 95  
**selectedIndex** attribute 524, 537  
**selectedItem** attribute 524  
 selecting a background color 457  
 Selecting an object with the Selection tool  
 462  
 selecting data from a table 587  
 selection box 462  
 selection criteria 592  
 selection tool 459, 462  
**selectNodes** method of *MSXML* 401  
 selector 104, 105  
 self-documenting 142  
 selling digital content 43  
     Blish 43  
     ClickBank 43  
     Lulu 43  
     Safari Books Online 43  
 semantic technology 24  
 Semantic Web 46  
 semicolon 609  
 semicolon (;) 102, 105, 135, 137, 143  
 semicolon on line by itself 151  
 semicolon resulting logic error 151  
**send** method of *HTTPService* element  
 538  
**send** method of the *XMLHttpRequest*  
 object 422  
 sentinel value 164  
 sentinel-controlled repetition 166  
 sentinel-controlled repetition to calculate  
     a class average 164  
 separation of structure from content 101  
 separator 247  
**series** element 566  
**SeriesEffect** element 566  
 serif fonts 105, 124  
 SERP (search engine results page) 7  
 server 894  
 server response 573  
 server side xx  
 server-side artifacts 898  
 server-side form handler 573  
 server-side script 575  
**server\_software** 663  
 server-side form handler 573  
 servlet 800  
 servlet container 800  
**Servlet** interface 800  
     **destroy** method 801  
     **init** method 800  
     **service** method 800  
 session 738, 834  
     tracking 738  
 session cookie 637  
 session item 747  
**Session** property of *Page* class 746  
 session scope 803  
 session tracking 738, 834  
     in web services 916  
 session variable 671  
**SessionBean** 803  
**SessionID** property of  
*HttpSessionState* class 751  
**SessionID** property of  
*HttpSessionState* class 750  
**SET** SQL clause 599  
**setAttribute** method of a DOM node  
 427  
**setAttribute** method of an *Element*  
 397  
**setAttribute** method of interface  
*HttpSession* 748  
**setAttribute** method of the  
*document* object 427  
**setAttributeNode** method of an  
*Element* 397  
**setcookie** function 637  
**setDate** 272, 274  
**SetEventHandler** element 530  
**setFullYear** 272, 274, 276  
**setHours** 273, 274  
**setInterval** method of the *window*  
     object 317, 327  
**setMilliseconds** 273  
**setMinutes** 273, 274  
**setMonth** 273, 274  
**SetProperty** element 530  
**setRequestHeader** method of the  
     *XMLHttpRequest* object 422  
**setSeconds** 273, 274  
**SetStyle** element 530  
**setTime** 273  
 Setting text alignment with the **Property**  
     **Inspector** 468  
 setting the fill and stroke colors 458  
 setting the font face, size, weight and color  
     with the **Property Inspector** 461  
**settype** function 610, 612  
**setUTCDate** 272  
**setUTCFullYear** 272  
**setUTCHours** 273  
**setUTCMilliseconds** 273  
**setUTCMinutes** 273  
**setUTCMonth** 273  
**setUTCSeconds** 273  
 shape tween 468, 481  
 Shareaza 42  
 Shift key 333  
 shift the range of numbers 211  
 shifted and scaled random integers 211  
 shifting value 215  
**shiftKey** property of an event object  
 333  
 Shockwave Flash (SWF) File 518  
 shopzilla 10  
 short-circuit evaluation 199  
 shorthand assignments of borders 121  
 show/hide layers 505  
**showDataEffect** attribute 566  
**showDataTip** attribute 524  
**showDataTips** attribute 566  
**showEffect** attribute 551  
 sibling node 385  
 siblings 299, 353  
 side effect 244  
**SideStep** 44  
 signal value 164  
 simple condition 197  
 simple content in XML Schema 369  
 Simple controller that passes a parameter  
     to the view 664  
 Simple controller that renders a message  
     on the web page 662  
 simple drawing program 329  
 Simple Object Access Protocol (SOAP)  
 37, 892  
 simple PHP program 608  
 Simple Queue Service (SQS) 36  
 Simple Ruby program 654  
 Simple Storage Service (S3) 36  
 simple type 369  
**simpleContent** XML Schema element  
 371  
**simpleType** XML Schema element 370  
 SimplyHired 42  
 Simpy 28  
**Simulate Download** 490  
 simulation and game playing 210  
**sin** method 260  
 single quote ('') 135, 177, 358, 593  
 single-entry/single-exit control statement  
 155, 156  
 single-line comment 609  
 single-line comment (//) 143, 145  
 single-selection **if** statement 155  
 single-selection statement 154  
**SingleSelectOptionsList** class 824  
**SingleSelectOptionsList** object 824  
 sites visited by the browser user 322  
 Six Apart 44  
**size** attribute (*input*) 91  
 size selection slider 460  
 skip remainder of a **switch** statement  
 192  
 Skype 39, 44  
**slice** method of the *String* object 263  
 small relative font size 105  
 smaller value 260  
 smart tag menu in Visual Web Developer  
 721  
**SMIL** (Synchronized Multimedia  
     Integration Language) 376  
 SMIL website ([www.w3.org/](http://www.w3.org/)  
     **Audio**/**Video**) 376  
 Smoke theme 558  
**snapInterval** attribute 524  
**SOAP** 894  
     envelope 914  
     message 914  
     parsing a SOAP message 914  
     SOAP request for the **HugeInteger**  
         web service 915  
         [www.w3.org/TR/soap/](http://www.w3.org/TR/soap/) 914  
**SOAP** (Simple Object Access Protocol)  
 37, 892  
 social bookmarking 5, 14, 15, 28, 43  
     Blue Dot 43  
     del.icio.us 43  
     ma.gnolia 43

- social media 5, 23, 43  
 Digg 43  
 Flickr 43  
 Reddit 43  
 Wikipedia 43  
 YouTube 43
- social networking 5, 19, 24, 28, 43  
 Bebo 43  
 Facebook 43  
 Gaia Online 43  
 LinkedIn 43  
 MySpace 43  
 Second Life 43
- social networking patent 20
- social networks  
 Gaia Online 22
- SocialText 15, 44
- Software as a Service (SaaS) 29, 43  
 Google 43  
 Microsoft Office Live 43  
 Microsoft Windows Live 43  
 Salesforce.com 43  
 37Signals 43  
 Zoho 43
- software asset 260
- Software Engineering Observations  
 overview xxv
- solid** value (*border-style* property) 119
- Some common escape sequences 140
- Some useful global arrays. 623
- sort** method 259
- sort** method of an *Array* object 247, 248
- sortable\_element** method of  
*Script.aculo.us* library 696
- Sorting an array with *Array* method  
**sort** 247
- sorting in XSL 384
- SoundMixer.stopAll** function 491
- source** attribute 523, 538, 540
- Source** mode in Visual Web Developer 711
- source string 262
- source tree (XSLT) 377
- source-code form 69
- SourceForge 31, 42
- Spacer** element 525
- spam 17, 23, 24, 29
- span as a generic grouping element 114
- span element 114, 806, 812
- span XHTML element 704, 708
- SPARQL (SPARQL Protocol and RDF Query Language) 47
- special character 79, 80, 621
- special characters 261
- specificity 106, 124
- speech device 84
- speech synthesizer 77
- splash screen 485
- split 268
- split** 269
- split** function 288
- split** method of the *String* object 263
- SplitAndSubString.html** 268
- splitting a statement in the middle of an identifier 142
- SponsoredReviews 40
- Sprite** class (Flash) 496
- SQL 586, 587, 591, 597  
 alias 784  
 AS keyword 784  
 DELETE statement 591, 599  
 FROM clause 591  
 GROUP BY clause 591  
 INNER JOIN clause 591, 596  
 INSERT statement 591, 597  
 LIKE clause 593, 594  
 ON clause 596  
 ORDER BY clause 591, 594, 595  
 SELECT query 591, 592, 594, 595  
 SET clause 599  
 UPDATE statement 591  
 VALUES clause 597  
 WHERE clause 592
- .sql 602
- SQL keyword 591
- SQL School ([www.w3schools.com/sql](http://www.w3schools.com/sql)) 604
- SqlCommand** class 603
- SqlConnection** class 603
- SqlDataAdapter** class 604
- SqlDataSource** ASP.NET data control 755
- SQLite database 567
- sqrt** 260
- SQRT1\_2 261
- SQRT2 261
- square brackets [] 233
- square root 260
- square-root symbol (MathML) 374
- src** attribute 76, 79
- src** property of an *Image* object 338
- srcElement** property (IE) of an event object 338
- stacked control structures 166
- stage 454, 455
- stage** 504
- stage.mouseX** 503
- stage.mouseY** 503
- “standardized, interchangeable parts” 259
- Stanford University 8, 9
- start page 820
- start tag 70, 349, 356
- StartForce (webtop) 29
- starting index 269
- State** element 530  
 name attribute 530
- stateless protocol 738, 834
- statement 135, 207
- statement terminator 135
- states** element 530
- StaticText** JSF component 804, 806, 820
- static web page 583
- static web pages 583
- status** property of the  
*XMLHttpRequest* object 422
- statusText** property of the  
*XMLHttpRequest* 422
- steve.museum project 26
- stop** 497
- stop** function 469, 477, 491
- stops all timers and sends objects into  
 hidden frame 510
- strcmp** function 619
- strike** 270
- strike** method of the *String* object 263
- string** 135
- string** assigned to a variable in a declaration 262
- string** comparison 248
- string** concatenation 144, 163
- string** constants 261
- string** data type 610
- string** literal 135, 261
- string** manipulation 204
- String** methods *charAt*, *charCodeAt*, *fromCharCode*, *toLowerCase* and *toUpperCase* 264
- String** object 261, 262, 264  
 anchor method 263, 271  
*charAt* method 262, 264, 264  
*charCodeAt* method 262, 264  
*fixed* method 263, 271  
*fromCharCode* method 262, 264, 265  
*indexOf* method 262, 265, 266  
*lastIndexOf* method 263, 265, 266, 267  
*link* method 263  
*replace* method 263  
*slice* method 263  
*split* method 263  
*strike* method 263  
*sub* method 263  
*substr* method 263  
*substring* method 263  
*sup* method 263  
*toLowerCase* method 263  
*toUpperCase* method 263
- String** object methods 262
- String** object's methods that generate XHTML markup tags 269
- string** representation of the number 278
- string** XML Schema data type 369
- string's length** 264
- string-concatenation operator (+)** 262
- stroke color** 458
- strong** element 73
- struck-out text** 271
- structure** 68
- structure** of a document 101
- structured programming** 154
- Structured Query Language (SQL) 586, 587, 591
- StumbleUpon 12, 28, 42, 43
- style** attribute 102, 176
- style** class 105, 105
- Style** element 554
- style sheet** 70, 353, 805, 814
- stylesheet** start tag 379
- sub** 270
- sub** element 81
- sub** method of the *String* object 263
- sub** initializer list 249
- submit** input 91
- submitter in a virtual form 874
- subscript** 81, 249
- subscript** of an array 233
- subscription sites 43

**substr** method of the *String* object 263  
**substring** 268  
**substr** method of the *String* object 263  
 substrings of a string 262  
*Suite101* (content network) 13, 42  
*Sullivan*, Danny 11, 12  
**sum** function (XSL) 384  
**summary** attribute 84  
 Summation with for 182  
*Sun Java System Application Server* 895, 899  
   starting on Windows 902  
   stopping on Windows 903  
*Tester* web page 901  
*Sun Java System Application Server (SJSAS)* 799, 819  
*Sun Microsystems, Inc.* 18, 33, 34  
   Java API 34  
**sup** element 81, 270, 271  
**sup** method of the *String* object 263  
 superglobal array 623  
 superscript 81  
*Suwowiecki*, James (author of *Wisdom of Crowds*) 15  
*Survey Monkey* 43  
*SVG* (Scalable Vector Graphics) 372  
 Swap colors 458  
 swatch 457  
**Swatches** panel in Flash CS3 456  
*SWF* (Shockwave Flash) File 518  
 .swf file format 466  
**switch** multiple-selection statement 189, 188  
*Sybase, Inc.* 586  
 symbol 461  
 symbol behavior 462  
 symbolic representation (MathML) 374  
**Sync** 465  
 synchronization 520  
 Synchronized Multimedia Integration Language (SMIL) 376  
 synchronous request 413  
 syntax error 159, 172  
*SYSTEM* keyword in XML 356  
*System.Data* namespace 603  
*System.Data.OleDb* namespace 603  
*System.Data.SqlClient* namespace 603  
*System.Web.UI* namespace 705  
*System.Web.UI.WebControls* namespace 706

**T**

tab 140  
*Tab* key 338  
 tab stop 140  
 table (database) 587  
*Table Adapter Configuration Wizard* 957  
 table body 86  
 table column heading 185  
 table data 86  
**table** data cells 86  
**table** element 84, 185  
   border attribute 84  
   caption element 84  
 table foot 86

table head 86  
 table head element 86  
**Table** JSF component 860  
 table of event object properties 333  
 table row 86  
**table** XHTML element 68  
*Ta-da* (37Signals) 30  
 tag 23, 26, 28  
 tag cloud 26  
 tag extension mechanism 801  
 tag library 801, 804  
 tagging 5, 25, 26, 28, 41  
 tagging for profit 100tags.com 41  
**tagName** property of an *Element* 397  
 tagscape 28  
 tahoma font 105  
**tan** method 260  
 tangent 260  
*Tapscott*, Don (author of *Wikinomics*) 14  
**Target** 498  
**target** attribute of an *a* (anchor) element 282  
 target of a hyperlink 271  
**target** property (FF) of an *event* object 338  
**Target** property of a *HyperLink* control 721  
**targetControlID** property of *theValidatorCalloutExtender* control 797  
**targetNamespace** XML Schema attribute 367  
**tbody** (table body) element 86  
*TcX* 600  
**td** element 86  
*TechCrunch* 17, 41  
*Technorati* 19, 26, 28, 41  
*Tele Atlas* 38  
 template 663, 668  
 Template for a *list* action that displays a list of messages 684  
 Template for a new action that adds a forum to the forums table 680  
 Template for delete action used to Delete a forum 681  
 Template for the list action that displays a list of messages 679  
 terminate a loop 163  
 terminate nested looping structures 195  
 terminator 135  
 ternary operator 157  
 test a web service 900  
**Test Movie** 467  
 testing a web service from another computer 903  
*TcX* software package 372  
 "text" value of type attribute 218  
**text** 460  
 text alignment 467  
**Text and Comment** methods 398  
 text area 94  
 text attribute 524  
 text box 91  
*Text Cloud of Deitel Web 2.0 Resource Center* 27  
**text-decoration** property 106  
 text editor 69  
**Text** element 566

text field 470  
**Text Field** JSF component 820, 824  
 text file 385  
**text input** 91  
*Text Link Ads* 40  
**text node-set** function 384  
 Text object 396, 398  
**text parameter** 537  
 text tool 460  
**text/javascript** 104  
**text\_field** method 672  
**text\_field\_with\_auto\_complete** method of *Script.aculo.us* library 696  
**text-align** property 118  
**textarea** element 94  
 text-based browser 77  
*TextBox* ASP.NET Web control 720  
*TextEditor* text editor 69  
**text-indent** property 116  
*TextInput* element 536, 538  
   change attribute 536  
*Textpattern* 18  
**tfoot** (table foot) element 86  
**th** (table header column) element 86, 185  
*The Attention Economy: Understanding the New Currency of Business* (by Thomas Davenport and John Beck) 8  
 The Digital Millennium Copyright Act of 1998 23  
 The Game Neverending 27  
 The Metropolitan Museum of Art 26  
 The NewPR Wiki 18  
*The Search: How Google and Its Rivals Rewrote the Rules of Business and Transformed Our Culture* (by John Battelle) 8  
*The Wall Street Journal Online* 40, 43  
**thead** element 86  
**theme** parameter 558  
 themes 558  
 There (virtual world) 41, 44  
**thick** border width 119  
**thin** border width 119  
 37Signals 30, 33, 43, 44  
   Backpack 30  
   Basecamp 30  
   Campfire 30  
   Hansson, David Heinemeier 33  
   Highrise 30  
   Ta-da 30  
   Writeboard 30  
**this** keyword 328, 332, 333  
 three-tier web application 574  
 thumbs view of Flickr photo viewer 698  
*Ticketmaster* 10  
**tickInterval** attribute 524  
 tier in a multtier application 574  
 tile an image only horizontally 116  
 tile an image vertically and horizontally 116  
 tiling no-repeat 116  
 tiling of the background image 116  
 time 149  
 Time class declaration maintains the time in 24-hour format 662, 664, 665, 671, 672, 673, 675, 681  
**time** function 637  
*TIME Magazine* 5  
 time manipulation 204

- Time Warner Network 7  
 timeline 454, 461, 468, 480  
**Timeout** property of  
*HttpSessionState* class 750, 751  
 timer 327  
 timer countdown 509  
 Times New Roman font 105  
**title** attribute 522  
 title bar 71, 139  
 title bar of a dialog 139  
**title** element 71  
 title of a document 70  
**title** parameter 537  
**Title** property of a *Page* directive 777  
**Title** property of a Web Form 712  
**titles** table of books database 588, 589  
**to\_s** method 659  
**toFixed** method of *Number* object 185  
 tokenization 268  
 tokenize a string 269  
 tokenizing 262  
 tokens 268  
**toLocaleString** 273, 274, 275  
**toLowerCase** 264  
**toLowerCase** 264, 265  
**toLowerCase** method of the *String*  
 object 263  
**Toolbox** 712  
**ToolkitScriptManager** control 796  
 Tools bar 454  
 Tools of Change for Publishing  
 conference 7  
 tooltip 724  
 top margin 113, 114, 115  
 top tier 575  
**toState** attribute 531  
**toString** 273, 274, 275, 277, 278  
**toUpperCase** 264  
**toUpperCase** 264, 265  
**toUpperCase** method of the *String*  
 object 263  
**toUCString** 273, 274, 275  
**tr** (table row) element 86  
**trace** function 491  
 trackback 17  
 tracking customers 737, 833  
 traditional model of event registration  
 324, 327  
 traditional web application 413  
**transformNode** method of an MSXML  
 document object 405  
**transformToFragment** method of the  
*XSLTProcessor* object (Firefox) 409  
**Transition** element 531  
*fromState* attribute 531  
*toState* attribute 531  
**transitions** element 531  
 travel sites 44  
 Expedia 44  
 Orbitz 44  
 Travelocity 44  
 Travelocity 44  
 traverse an array 250  
 traversing an XML document using the  
 XML DOM 387  
 tree structure 354  
 tree structure for the document  
*article.xml* of Fig. 14.2 386  
 Tribal Fusion 40, 42
- trigger** attribute 538  
**triggerEvent** attribute 538  
 trigonometric cosine 260  
 trigonometric sine 260  
 true 148  
**true** 156  
 Trulia 10, 43  
 truncate 163  
 trust 5  
 truth table 198, 200  
 truth table for the && logical AND  
 operator 198  
 truth table for the || (logical OR)  
 operator 199  
 truth table for the logical negation  
 operator 199  
**try** block 421  
**try** keyword 421  
 Tucows 31, 42  
 tween 468  
 24-hour clock format 276  
*24 Hours on Craigslist* 15  
 Twitter 23, 42  
 two-dimensional array 249, 250  
 type 104  
 type = "text/javascript" 134  
 type attribute 90, 134  
 type attribute in a processing instruction  
 378  
 type casting 612  
 type conversion 610  
 type property of an *event* object 333  
 type XML Schema attribute 368  
 type-ahead 428  
 TypePad 18
- U**
- Ubuntu 31  
 Ubuntu Live conference 7  
**ui:staticText** JSF element 868  
**ui:table** JSF element 868  
**ui:tableRowGroup** JSF element 868  
**UIviewRoot** class 805  
**ul** (unordered list) element 81  
**ul** element 81  
 unary operator 170, 199  
 decrement (--) 170  
 increment (+) 170  
 unbound value 369  
**undef** 614  
**undefined** 147  
**underline** value (*text-decoration*)  
 106  
 underscore (\_) SQL wildcard character  
 593  
**unescape** function 227, 287  
 Unicode 261, 264  
 Unicode value 264  
 Uniform Resource Identifier (URI) 360,  
 570  
 Uniform Resource Locator (URL) 360,  
 570  
 Uniform Resource Name (URN) 360  
 unique session ID of an ASP.NET client  
 750  
 United States Department of Defense 37  
 UNIX 581  
**unload** event of an ASP.NET page 707  
 unordered list 81  
 Unordered list containing hyperlinks. 81  
 unordered list element (*ul*) 81  
 up state 463  
 UPDATE SQL statement 591, 598  
**UpdateCommand** property of a  
*SqlDataSource* ASP.NET control  
 760  
**UpdatePanel** control 797  
 uppercase letters 136, 142  
 URI (Uniform Resource Identifier) 360,  
 570  
 URL (Uniform Resource Locator) 360,  
 570  
 rewriting 738, 834  
**URL** attribute 538  
 URL mapping 663  
 URL parameter 682  
**url(fileLocation)** 115  
**URLRequest** object 543  
 URN (Uniform Resource Name) 360  
 user agent 106  
 user input 88  
 user interaction 324  
 user interface 575  
 user style sheet 126  
 User style sheet applied with *em*  
 measurement 129  
 User style sheet applied with *pt*  
 measurement 128  
 User style sheet in Internet Explorer 7 127  
 user styles 126, 128, 129  
**UserController** provides validation  
 functionality for the table 671  
 user-defined types 259, 369  
 user-generated content 5, 13, 14, 23, 28  
 Using a *filterFunction* for a live  
 search feature 540  
 Using a **Style** element and inline styles  
 554  
 using cookies to store user identification  
 data 285  
 using equality and relational operators  
 149  
 Using inline styles 102  
 using internal hyperlinks to make pages  
 more navigable 95  
 using **meta** to provide keywords and a  
 description 98  
 using PHP's arithmetic operators 613  
 Using **States** to dynamically modify a  
 user interface 527  
 using **String** object methods *split* and  
*substring* 268  
 Using the **break** statement in a **for**  
 statement 193  
 Using the **continue** statement in a **for**  
 statement 194  
 Using the **do...while** repetition  
 statement 191  
 using the string-comparison operators  
 618  
 Using the **switch** multiple-selection  
 statement 186  
 using the **window** object to create and  
 modify child windows 279  
 using **XPath** to locate nodes in an XML  
 document 398

UTC (Coordinated Universal Time) 272, 275, 276

## V

valid attribute 538  
 valid XML document 350, 362  
 validate 672  
**validates\_format\_of** 675  
**validates\_presence\_of** method 675  
 validating an XML document with Microsoft's XML Validator 357  
 validating parser 38  
 validating XML parser 350  
 validation 825  
 validation control 726  
 validation service 72  
**ValidationExpression** property of a RegularExpressionValidator control 730  
**ValidationExpression** property of class RegularExpressionValidator 730  
 validator 726  
 Validator element  
   source attribute 538  
   trigger attribute 538  
   triggerEvent attribute 538  
   valid attribute 538  
 validator.w3.org 72  
 ValidatorCalloutExtender control 797  
**ValidatorException** class 832, 833  
 Validators  
   Double Range 825  
   Length 825, 830  
   Long Range 825  
 validators 675  
 value attribute 91, 522  
 value of an array element 233  
 value of an attribute 71  
 value property of a text field 221  
 value property of an Attr object 397  
 Value property of **HttpCookie** class 745  
 ValueClick 42  
 valueOf 273  
 VALUES 598  
 values of n (company) 29  
 VALUES SQL clause 597  
 Vander Wal, Thomas 26  
 var keyword 142, 207  
 var to declare variables is not required in JavaScript 142  
 variable 142  
 variable name 142  
 variable variables 647  
 variable's scope 180  
 various markup languages derived from XML 376  
 VBox element 522  
   hideEffect attribute 551  
   showEffect attribute 551  
 vector graphic 458  
 vectors 458  
 Veoh 24  
 verbs in a system specification 259  
 verify 677

**version** attribute (XSL) 379  
**version** in XML declaration 351  
 vertical and horizontal positioning 115  
 vertical search engines 7, 10, 44

  Kayak 44  
   Koders 44  
   Krugle 44  
   SideStep 44  
     Yahoo! Fare Finder 44

vertical spacing 177

**verticalAlign** attribute 522

**verticalAxis** element 566

VH1 24

vi text editor 69

Viacom 23, 24

Vibrant Media 40, 42

video blogging 18

**View In Browser** command in Visual Web Developer 715

view states 527

View that displays a list of employees 669

View that displays the name of the server 664

\_\_VIEWSTATE hidden input 732

viral effect 23

virtual directory 571, 577, 579, 954

**Virtual Directory Creation Wizard** 577

virtual forms 863, 874

  participant 874

  submitter 874

  There 44

virtual worlds monetization 41

**Visible** property of an ASP.NET Web control 727

Visual Editor 804, 813

visual inheritance 769

Visual Web Developer

**ASP.NET Web Service** project 954

  Visual Web Pack (Netbeans) 799

**visual\_effect** method of

  Script.aculo.us Library 692

  duration parameter 692

  queue parameter 693

  end value 693

vlogging 18

vocabulary (XML) 349

VoiceXML 349, 376

  www.voicexml.org 376

VoIP 39, 44

  Lingo 44

  Packet8 44

  Skype 44

  Vonage 44

Vonage (VoIP) 39, 44

**VS1lider** element 522

  change attribute 522

  labels attribute 523

  liveDragging attribute 522

  maximum attribute 522

  minimum attribute 522

  showDataTip attribute 524

  snapInterval attribute 524

  tickInterval attribute 524

  value attribute 522

## W

W3C (World Wide Web Consortium) 38, 47, 68, 348

Weather Forecast application in Flex 2 546

Web 2.0 4

Web 2.0 3

  Web 2.0 Expo 7

  Web 2.0 Summit 4, 7

Web 2.0 software 44

  Adobe 44

  Microsoft 44

  Six Apart 44

  37Signals 44

web analytics 44

  Alexa 44

  ClickTracks 44

  Google Analytics 44

  WebSideStory 44

  WebTrends 44

web and mobile messaging 44

web application

  Ajax 414

  traditional 413

Web application development 701

web application development 799

web application framework 802

**Web Application** project 895

web as a platform 29, 30

web-based application 68

web conferencing 44

  DimDim 44

  GoToMeeting 44

  WebEx 44

Web control 701

Web Form 701, 748

web framework 653

web of meaning 46

Web Ontology Language (OWL) 47

web page 68

web server xx, 69, 88, 570, 571, 635, 663

web service 892

  implemented as a class 894

  publishing a web service 900

Web Service Description Language (WSDL) 904

web service reference 905

Web services

**@Resource** annotation 919

**@WebMethod** annotation 899

**@WebParam** annotation 899

**@WebService** annotation 899

  adding a web service reference to an application 905

Build Project option in Netbeans 900

Clean and Build Project option in Netbeans 900

Clean Project option in Netbeans 900

Client desktop application for the HugeInteger web service 909

client-side artifacts 905, 906

consuming a web service 894, 904

Deploy Project option in Netbeans 900

deploying a web service 900

Web service (cont.)  
 Java Technology and Web Services  
 Overview ([java.sun.com/webservices/](http://java.sun.com/webservices/)  
*overview.html*) 893  
 JAX-WS 2.0 894  
 MessageContext interface 919  
 name element of @WebService  
 annotation 899  
 parsing a SOAP message 914  
 POJO (Plain Old Java Object) 898  
 proxy class 894  
 publishing a web service 894  
 RequestContext interface 921  
 REST 892  
**Run Project** option in Netbeans 900  
 serviceName element of  
 @WebService annotation 899  
 SOAP 894  
 SOAP envelope 914  
 SOAP message 914  
 Sun Java System Application Server's  
 Tester web page 901  
 test a web service 900  
 testing a web service from another  
 computer 903  
 user-defined types 939  
 Web Service Description Language  
 (WSDL) 904  
 web service reference 905  
 WebServiceContext interface 919  
 wire format 914  
 wire protocol 914  
[www.w3.org/TR/soap/](http://www.w3.org/TR/soap/) 914  
 server-side artifacts 898  
 web services 6, 9, 23, 28, 34, 519, 653  
 Web Services Description Language  
 (WSDL) 551  
 Web Services Interoperability  
 Organization (WS-I) 951, 952  
 Web Services Resource Center at  
[www.deitel.com/WebServices/](http://www.deitel.com/WebServices/)  
 892  
**Web Site Administration Tool** 770  
 Web.config ASP.NET configuration  
 file 715  
[web.eesite.com/forums](http://web.eesite.com/forums) 670  
 WebControl class 706  
 WebEx 44  
 WebKit rendering engine 567  
 weblog 16  
 Weblogs, Inc. (blog network) 13, 18, 41  
[weblogs.macromedia.com/mc/archives/](http://weblogs.macromedia.com/mc/archives/)  
[FlexStyleExplorer.html](http://FlexStyleExplorer.html) 557  
 webmail 44  
@WebMethod annotation 899  
 operationName element 899  
 WebMethod attribute 952  
@WebParam annotation 899  
 name element 899  
 web-safe palette 457  
 web-scale computing 36  
@WebService annotation 899  
 name element 899  
 serviceName element 899  
WebService attribute 951, 955  
 WebService class 952

WebService element  
 fault attribute 551  
 result attribute 551  
 WebServiceBinding attribute 951, 955  
 WebServiceContext interface 919  
 WebServiceX.net 545  
 WebSideStory 44  
 Websites  
[api.rubyonrails.org](http://api.rubyonrails.org) 653, 675  
[code.google.com/apis/gears/](http://code.google.com/apis/gears/)  
 566  
[db.apache.org/derby](http://db.apache.org/derby) 604  
[demo.quietlyscheming.com/](http://demo.quietlyscheming.com/)  
*ChartSampler/app.html* 520  
[demo.script.aculo.us/ajax/](http://demo.script.aculo.us/ajax/)  
*autocomplete* 696  
[demo.script.aculo.us/ajax/](http://demo.script.aculo.us/ajax/)  
*sortable\_elements* 696  
[demo.script.aculo.us/shop](http://demo.script.aculo.us/shop)  
 696  
[dev.mysql.com/doc/mysql/en/index.html](http://dev.mysql.com/doc/mysql/en/index.html) 604  
[developer.yahoo.com](http://developer.yahoo.com) 540  
[developer.yahoo.com/flash/as3\\_api\\_libraries.html](http://developer.yahoo.com/flash/as3_api_libraries.html)  
 540  
[developer.yahoo.com/flash/as3webapis/docs/com/yahoo/webapis/maps/methodgroups/MapController.html](http://developer.yahoo.com/flash/as3webapis/docs/com/yahoo/webapis/maps/methodgroups/MapController.html) 543  
[developer.yahoo.com/flash/howto-maps-as3.html](http://developer.yahoo.com/flash/howto-maps-as3.html) 540  
[developers.sun.com/javadb/604](http://developers.sun.com/javadb/604)  
[en.wikipedia.org/wiki/IP\\_address](http://en.wikipedia.org/wiki/IP_address) 571  
[examples.adobe.com/flex2/inproduct/sdk/explorer/explorer.html](http://examples.adobe.com/flex2/inproduct/sdk/explorer/explorer.html) 518  
[flexwiki.adobe.com/confluence/display/AD0BE/Flex+3+Planning](http://flexwiki.adobe.com/confluence/display/AD0BE/Flex+3+Planning) 567  
[go.microsoft.com/fwlink/?LinkId=92716](http://go.microsoft.com/fwlink/?LinkId=92716) 300  
[httpd.apache.org/docs-2.2/platform/windows.html](http://httpd.apache.org/docs-2.2/platform/windows.html) 581  
[httpd.apache.org/download.cgi](http://httpd.apache.org/download.cgi) 581  
[labs.adobe.com/technologies/air/](http://labs.adobe.com/technologies/air/) 567  
[labs.adobe.com/technologies/flex/](http://labs.adobe.com/technologies/flex/) 567  
[labs.adobe.com/technologies/flex/sdk/flex3sdk.html](http://labs.adobe.com/technologies/flex/sdk/flex3sdk.html) 567  
[labs.adobe.com/wiki/index.php/AIR:Developer\\_FAQ](http://labs.adobe.com/wiki/index.php/AIR:Developer_FAQ) 567  
[labs.adobe.com/wiki/index.php/Flex\\_3:Feature\\_Introductions](http://labs.adobe.com/wiki/index.php/Flex_3:Feature_Introductions) 567  
[labs.adobe.com/wiki/index.php/Flex\\_3:Release\\_Notes#Installation\\_Instructions](http://labs.adobe.com/wiki/index.php/Flex_3:Release_Notes#Installation_Instructions) 567  
[linrails.thembid.com](http://linrails.thembid.com) 654

Websites (cont.)  
[livedocs.adobe.com/flex/2/docs/wwhelp/wwhimpl/common/html/wwhelp.htm?context=LiveDocs\\_Parts&file=00001582.html#178239](http://livedocs.adobe.com/flex/2/docs/wwhelp/wwhimpl/common/html/wwhelp.htm?context=LiveDocs_Parts&file=00001582.html#178239) 525  
[livedocs.adobe.com/flex/201/html/charts\\_intro\\_108\\_03.html](http://livedocs.adobe.com/flex/201/html/charts_intro_108_03.html) 559  
[livedocs.adobe.com/flex/201/html/wrapper\\_131\\_05.html](http://livedocs.adobe.com/flex/201/html/wrapper_131_05.html) 525  
[livedocs.adobe.com/flex/201/langref/mx/effects/easing/package-detail.html](http://livedocs.adobe.com/flex/201/langref/mx/effects/easing/package-detail.html) 531  
[livedocs.adobe.com/flex/201/langref/mx/effects/package-detail.html](http://livedocs.adobe.com/flex/201/langref/mx/effects/package-detail.html) 524  
[locomotive.raaum.org](http://locomotive.raaum.org) 654  
[redgreenblu.com/flickr/](http://redgreenblu.com/flickr/) 696  
[weblogs.macromedia.com/mc/archives/](http://weblogs.macromedia.com/mc/archives/)  
[FlexStyleExplorer.html](http://FlexStyleExplorer.html) 557  
WebServiceX.net 545  
[www.adobe.com/devnet/flex/articles/flex3\\_whatsnew.html](http://www.adobe.com/devnet/flex/articles/flex3_whatsnew.html) 567  
[www.adobe.com/go/flex2\\_apiref](http://www.adobe.com/go/flex2_apiref) 519  
[www.adobe.com/go/flexsdk2\\_download](http://www.adobe.com/go/flexsdk2_download) 519  
[www.adobe.com/go/getflashplayer](http://www.adobe.com/go/getflashplayer) 453  
[www.adobe.com/go/tryflex](http://www.adobe.com/go/tryflex) 520, 558  
[www.adobe.com/go/trylivecycle\\_dataservices](http://www.adobe.com/go/trylivecycle_dataservices) / 520  
[www.adobe.com/products/flashplayer/download/detection\\_kit/](http://www.adobe.com/products/flashplayer/download/detection_kit/) 454  
[www.adobe.com/products/player\\_census/flashplayer/version\\_penetration.html](http://www.adobe.com/products/player_census/flashplayer/version_penetration.html) 454, 518  
[www.adobe.com/support/documentation/en/flex/2/install.html#installingcharts](http://www.adobe.com/support/documentation/en/flex/2/install.html#installingcharts) 558  
[www.deitel.com/flex](http://www.deitel.com/flex) 519, 567  
[www.deitel.com/RubyOnRails](http://www.deitel.com/RubyOnRails) 653  
[www.deitel.com/webservices](http://www.deitel.com/webservices) 545  
[www.flashkit.com](http://www.flashkit.com) 465  
[www.flickr.com/services/api/misc.api\\_keys.html](http://www.flickr.com/services/api/misc.api_keys.html) 697  
[www.linuxheadquarters.com/howto/basic/path.shtml](http://www.linuxheadquarters.com/howto/basic/path.shtml) 525  
[www.mysql.com](http://www.mysql.com) 600, 604  
[www.mysql.com/company/legal/licensing/](http://www.mysql.com/company/legal/licensing/) 600

- Websites (cont.)
- [www.mysql.com/products/enterprise/server.html](http://www.mysql.com/products/enterprise/server.html) 604
  - [www.mysql.com/why-mysql/topreasons.html](http://www.mysql.com/why-mysql/topreasons.html) 600
  - [www.sounds.muinar.com](http://www.sounds.muinar.com) 465
  - [www.w3.org/Protocols/rfc2616/rfc2616-sect10.html](http://www.w3.org/Protocols/rfc2616/rfc2616-sect10.html) 572
  - webtop xxi, 29, 32, 65, 417
  - WebTrends 44
  - webuijsf:body element 805
  - webuijsf:form element 805
  - webuijsf:head element 805
  - webuijsf:html element 805
  - webuijsf:label element 805
  - webuijsf:link element 805
  - webuijsf:message element 826
  - webuijsf:meta element 805
  - webuijsf:page element 805
  - webuijsf:staticText element 805
  - Weinberger, David (author of *Everything is Miscellaneous*) 26
  - Welcome Screen 454
  - well-formed XML document 350
  - Wesabe 14
  - Where 2.0 conference 7, 37
  - WHERE SQL clause 591, 592, 593, 594, 595, 598, 599
  - while control structure 491
  - while repetition statement 154, 166, 169, 179
  - while statement 622
  - white hat SEO 11
  - white-space character 268, 276, 155, 276
  - white-space characters in strings 135, 152
  - widget 36
  - Widgets element 543
  - Wikipedia 36
  - width attribute 76, 522
  - width attribute (table) 84
  - width property 116
  - width-to-height ratio 76
  - wiki 25
  - Wikia 15, 44
  - Wikinomics* (by Don Tapscott and Anthony D. Williams) 14
  - Wikimedia 5, 14, 15, 42, 43, 44, 47
  - wikis 14, 44
    - SocialText 44
    - Wikia 44
    - Wikipedia 44
  - Williams, Anthony D. (author of *Wikinomics*) 14
  - window object 139, 147, 257, 279, 316, 322
    - clearInterval method 316, 321
    - confirm method 343
    - setInterval method 317, 327
  - window object's open method 282
  - window object's prompt method 143, 145
  - window.alert method 139
  - window.closed 283
  - window.opener 293
  - window.prompt method 165, 186
  - Windows Projector 472
  - Windows Vista 903
  - Windows XP Service Pack 2
    - allow web requests 903
    - HTTP access from other computers on the network 903
  - wire format 914
  - wire protocol 914
  - Wireless Markup Language (WML) 372
  - wisdom of crowds 15, 24
  - Wisdom of Crowds* (by James Surowiecki) 15
  - WML (Wireless Markup Language) 372
  - Wojcicki, Susan (Google) 9
  - Wooden theme 558
  - word processor 265
  - WordPress 18
  - World Live Web 19
  - World of Warcraft 42
  - World Wide Wait 32
  - “World Wide Wait” 414
  - World Wide Web (WWW) 68
  - World Wide Web Consortium (W3C) 38, 47, 68, 348
  - wrappers 277
  - Write access permission 579
  - write method of the document object 137
  - Writeboard 30
  - writeln method 135, 137
  - writing a cookie to the client 636
  - WSDL (Web Service Description Language) 904
  - WSDL (Web Services Description Language) 551
  - WS-I (Web Services Interoperability Organization) 951, 952
  - WS-I Basic Profile 1.1 (BP 1.1) 951
  - WsProfiles\_BasicProfile1\_1 951
  - WWW (World Wide Web) 68
  - [www.adobe.com/devnet/flex/articles/flex3\\_whatsnew.html](http://www.adobe.com/devnet/flex/articles/flex3_whatsnew.html) 567
  - [www.adobe.com/go/flex2\\_apiref](http://www.adobe.com/go/flex2_apiref) 519
  - [www.adobe.com/go/flexsdk2\\_download](http://www.adobe.com/go/flexsdk2_download) 519
  - [www.adobe.com/go/getflashplayer](http://www.adobe.com/go/getflashplayer) 453
  - [www.adobe.com/go/tryflex](http://www.adobe.com/go/tryflex) 520, 558
  - [www.adobe.com/go/trylvilifecycle.dataservices](http://www.adobe.com/go/trylvilifecycle.dataservices) 520
  - [www.adobe.com/products/flashplayer/download/detection\\_kit/](http://www.adobe.com/products/flashplayer/download/detection_kit/) 454
  - [www.adobe.com/products/player\\_census/flashplayer/version\\_penetration.html](http://www.adobe.com/products/player_census/flashplayer/version_penetration.html) 454, 518
  - [www.adobe.com/support/documentation/en/flex/2/install.html#installingcharts](http://www.adobe.com/support/documentation/en/flex/2/install.html#installingcharts) 558
  - [www.deitel.com](http://www.deitel.com) 74
  - [www.deitel.com/books/ajaxfp/xix](http://www.deitel.com/books/ajaxfp/xix)
  - [www.deitel.com/flex/519,567](http://www.deitel.com/flex/519,567)
  - [www.deitel.com/RubyOnRails/653](http://www.deitel.com/RubyOnRails/653)
  - [www.deitel.com/webservices/545](http://www.deitel.com/webservices/545)
  - [www.ecma-international.org/publications/standards/ECMA-262.htm](http://www.ecma-international.org/publications/standards/ECMA-262.htm) (ECMAScript standard) 132
  - Flashkit 465
  - [www.flashkit.com](http://www.flashkit.com) 465
  - [www.flickr.com/services/api/misc.api\\_keys.html](http://www.flickr.com/services/api/misc.api_keys.html) 697
  - [www.garshol.priv.no/download/text/bnf.html](http://www.garshol.priv.no/download/text/bnf.html) 362
  - [www.linuxheadquarters.com/howto/basic/path.shtml](http://www.linuxheadquarters.com/howto/basic/path.shtml) 525
  - [www.microsoft.com/sql](http://www.microsoft.com/sql) 604
  - [www.mysql.com](http://www.mysql.com) 600, 604
  - [www.oasis-open.org](http://www.oasis-open.org) 362
  - [www.opengis.org](http://www.opengis.org) 376
  - [www.rixml.org](http://www.rixml.org) 376
  - [www.rubycentral.com/ref/](http://www.rubycentral.com/ref/) 657
  - [www.sounds.muinar.com](http://www.sounds.muinar.com) 465
  - [www.sql.org](http://www.sql.org) 604
  - [www.sqlmag.com](http://www.sqlmag.com) 605
  - [www.voicexml.org](http://www.voicexml.org) 376
  - [www.w3.org/2001/XMLSchema](http://www.w3.org/2001/XMLSchema) 367
  - [www.w3.org/Amaya/User/BinDist.html](http://www.w3.org/Amaya/User/BinDist.html) 372
  - [www.w3.org/AudioVideo](http://www.w3.org/AudioVideo) 376
  - [www.w3.org/Math](http://www.w3.org/Math) 375
  - [www.w3.org/XML/Schema](http://www.w3.org/XML/Schema) 356, 365
  - [www.w3schools.com/schema/default.asp](http://www.w3schools.com/schema/default.asp) 365
  - [www.w3schools.com/sql](http://www.w3schools.com/sql) 604
  - [www.WS-I.org](http://www.WS-I.org) 951, 952
  - [www.xml.org](http://www.xml.org) 362
  - [www.zend.com](http://www.zend.com) 607
  - WYSIWYG (What You See Is What You Get) editor 713, 817

**X**

- x-large relative font size 105
- x-small relative font size 105
- Xalan XSLT processor 377
- Xanga 16
- XBRL (Extensible Business Reporting Language) 349, 372
- XBRL (XML vocabulary) 38
- Xerces parser from Apache 350
- XHR (abbreviation for XMLHttpRequest) 412
- XHTML 32, 38, 68, 412, 415, 662, 663
- XHTML (Extensible Hypertext Markup Language) xix, xx, 68, 349, 377, 570, 701, 799
  - document type declaration 812
  - span element 806, 812
- XHTML 1.0 Strict 68
- XHTML colors 338
- XHTML comment 69
- XHTML documents 69
- XHTML form 88, 219
- XHTML form for gathering user input 624
- XHTML markup methods of the String object 270
- XHTML registration form 415
- XHTML response when the browser requests WebTime.aspx 707
- Xing 22

XML 412, 800  
 attribute 356  
 attribute value 356  
 child element 353  
 container element 353  
 declaration 351, 355  
 element 349  
 empty element 358  
 end tag 349  
 markup 351  
 node 354  
 parent element 353  
 prolog 352  
 root 354  
 root element 349  
 start tag 349  
 vocabulary 349  
 XML (Extensible Markup Language) xx,  
 32, 38, 46, 348  
 XML declaration 804  
 XML document containing book  
 information 381  
 XML Document Object Model 386  
 XML document that describes various  
 sports 377, 401  
 XML document using the `laptop`  
 element defined in `computer.xsd`  
 371  
 XML DOM 387  
 XML element name requirements 352  
 XML file containing contacts 538  
 .xml file extension 350  
 XML instance document 370, 371  
 xml namespace prefix 359  
 XML namespaces demonstration 359  
 XML parser 350  
 XML Path Language (XPath) 376, 398  
 XML processor 350  
 XML Resource Center  
 ([www.deitel.com/XML/](http://www.deitel.com/XML/)) 396  
 XML Schema 361, 365, 369  
 complex types 369  
 simple types 369  
 XML Schema document defining simple  
 and complex types 370  
 XML Schema document for book.xml  
 366  
 XML Schema URI ([http://  
 www.w3.org/2001/XMLSchema](http://www.w3.org/2001/XMLSchema))  
 367  
 XML serialization 939  
 XML used to mark up an article 351  
 XML Validator 356  
 XML Validator displaying an error  
 message 364

XML vocabularies 38  
 Chemical Markup Language (CML)  
 376  
 Extensible User Interface Language  
 (XUL) 376  
 Geography Markup Language  
 (GML) 376  
 Research Information Exchange  
 Markup Language (RIXML) 376  
 Synchronized Multimedia  
 Integration Language (SMIL) 376  
 VoiceXML 376  
 XML Working Group of the W3C 348  
**Xm1DataSource** ASP.NET data control  
 723  
**XM1HttpRequest** 32  
**XM1HttpRequest** object 412, 417, 418,  
 421  
`abort` method 423  
 GET method 422  
`getAllResponseHeaders` method  
 423  
`getResponseHeader` method 423  
`onreadystatechange` property  
 421, 422  
`open` method 422  
 properties and methods 421  
`readyState` property 422  
`responseText` property 421, 422  
`responseXML` property 422  
`send` method 422  
`setRequestHeader` method 422  
`status` property 422  
`statusText` property 422  
**xm1ns** attribute in XML 360  
**xm1ns** attributes 804  
 XPath 377  
 XPath (XML Path Language) 376, 398  
 XPath expression 398  
**XPathResult** object 401  
 .xsd filename extension 365  
 XSL (Extensible Stylesheet Language)  
 351, 361, 376  
 XSL document that transforms  
`sorting.xml` into XHTML 382  
 .xsl filename extension 379  
 XSL-FO (XSL Formatting Objects) 376  
 XSL Formatting Objects (XSL-FO) 376  
 XSL style sheet 377, 384  
 XSL template 380  
 xsl template element 380  
 XSL transformation 405  
 XSL Transformations (XSLT) 377  
 XSL variable 384  
**xsl:for-each** element 380

**xsl:output** element 380  
**xsl:value-of** element 380  
 XSLT processor 377  
 XSLT that creates elements and attributes  
 in an XHTML document 379  
**XSLTProcessor** object (Firefox) 409  
 XSS (cross-site scripting) 418  
 XUL (Extensible User Interface  
 Language) 372, 376  
 xx-large relative font size 105  
 xx-small relative font size 105

**Y**

Yahoo! 7, 9, 12, 27, 34, 42, 43  
 Classifieds 42  
 Fare Finder 44  
 Pipes 35  
 Publisher Network 40  
 Search Marketing 41  
 Video 42  
 Yahoo! Mail 44  
 Yahoo! Map 531  
 Yahoo! Maps API 540  
**YahooMapService** 537  
**YahooMapService** element 551  
 Yang, Jerry 9  
`yField` attribute 566  
`yield` 664  
 YouTube 5, 13, 23, 42, 43

**Z**

**z-index** property 113  
 Zend Engine 607  
 Zend Technologies ([www zend com](http://www zend com)) 43,  
 607  
 Zephiera 46  
 zero-based counting 179  
 zeroth element of an array 233  
 Zillow 10, 43  
 Zoho 43  
 Zoom effect 551  
**Zoom tool** 458