

# The Most Complete Reference

HTML 3.2 • Netscape® Communicator • Internet Explorer® 4  
HTML Layout Control • Cougar (HTML 4.0) • ActiveX® Controls  
Plug-ins • Dynamic HTML • Font & Document Formatting  
Frames & Tables • Style Sheets • Java® • JavaScript®  
VBScript • Multimedia • VRML • WebTV® • Netscape Gold®  
ActiveX Control Pad • CGI • PageMill® • FrontPage®

**Bestseller!**  
**Over 175,000**  
**Copies Sold**

*Special Edition*

USING

# HTML<sup>3.2</sup>

## Third Edition

CD-ROM includes Internet Explorer 3.0, HTML editors and utilities, plug-ins, Java tools, ForeFront Web printer, source code from the book, and four Que books in HTML format!

Mark R. Brown  
& Jerry Honeycutt

**que**®

*Special Edition*

USING

HTML 3.2

THIRD EDITION

**que**<sup>®</sup>

Before using any of the software on this disc, you need to install the software you plan to use. See Appendix C, "What's on the CD?" for directions. If you have problems with this CD, please contact Macmillan Technical Support at (317) 581-3833. We can be reached by e-mail at [support@mcp.com](mailto:support@mcp.com) or by CompuServe at GO QUEBOOKS.

## Read this Before Opening Software

By opening this package, you are agreeing to be bound by the following:

This software is copyrighted and all rights are reserved by the publisher and its licensors. You are licensed to use this software on a single computer. You may copy the software for backup or archival purposes only. Making copies of the software for any other purpose is a violation of United States copyright laws. THIS SOFTWARE IS SOLD AS IS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Neither the publisher nor its dealers and distributors nor its licensors assume any liability for any alleged or actual damages arising from the use of this software. (Some states do not allow exclusion of implied warranties, so the exclusion may not apply to you.)

The entire contents of this disc and the compilation of the software are copyrighted and protected by United States copyright laws. The individual programs on the disc are copyrighted by the authors or owners of each program. Each program has its own use permissions and limitations. To use each program, you must follow the individual requirements and restrictions detailed for each. Do not use a program if you do not agree to follow its licensing agreement.

**Technical**

**Support:**

If you need assistance with the information in this book or with a CD/Disk accompanying the book, please access the Knowledge Base on our Web site at <http://www.superlibrary.com/general/support>. Our most Frequently Asked Questions are answered there. If you do not find the answer to your questions on our Web site, you may contact Macmillan Technical Support (317) 581-3833 or e-mail us at [support@mcp.com](mailto:support@mcp.com).

*Special Edition*

# USING HTML 3.2 THIRD EDITION

*Written by Jerry Honeycutt and Mark R. Brown  
with*

*Jim O'Donnell • Eric Ladd • Robert Meegan • Bill Bruns  
Robert Niles • David Wall • Mathew Brown • Rob Falla*

**que**<sup>®</sup>

## Special Edition Using HTML 3.2, Third Edition

Copyright© 1997 by Que® Corporation.

All rights reserved. Printed in the United States of America.

No part of this book may be used or reproduced in any form or by any means, or stored in a database or retrieval system, without prior written permission of the publisher except in the case of brief quotations embodied in critical articles and reviews. Making copies of any part of this book for any purpose other than your own personal use is a violation of United States copyright laws. For information, address Que Corporation, 201 W. 103rd Street, Indianapolis, IN 46290. You may reach Que's direct sales line by calling 1-800-428-5331.

Library of Congress Catalog No.: 97-65026

ISBN: 0-7897-1097-8

This book is sold *as is*, without warranty of any kind, either express or implied, respecting the contents of this book, including but not limited to implied warranties for the book's quality, performance, merchantability, or fitness for any particular purpose. Neither Que Corporation nor its dealers or distributors shall be liable to the purchaser or any other person or entity with respect to any liability, loss, or damage caused or alleged to have been caused directly or indirectly by this book.

99 98 97 6 5 4 3 2 1

Interpretation of the printing code: the rightmost double-digit number is the year of the book's printing; the rightmost single-digit number, the number of the book's printing. For example, a printing code of 97-1 shows that the first printing of the book occurred in 1997.

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Que cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Screen reproductions in this book were created with Collage Plus from Inner Media, Inc., Hollis, NH.

# Credits

## **PRESIDENT**

Roland Elgey

## **PUBLISHER**

Joseph B. Wikert

## **PUBLISHING MANAGER**

Jim Minatel

## **EDITORIAL SERVICES DIRECTOR**

Elizabeth Keaffaber

## **DIRECTOR OF MARKETING**

Lynn E. Zingraf

## **ACQUISITIONS MANAGER**

Cheryl D. Willoughby

## **ACQUISITIONS EDITOR**

Philip Wescott

## **PRODUCT DIRECTORS**

Mark Cierzniak

Steven M. Schafer

## **PRODUCTION EDITOR**

Jim Bowie

## **EDITORS**

Patricia R. Kinyon

Kelli M. Brooks

Kate O. Givens

## **PRODUCT MARKETING MANAGER**

Kristine Ankney

## **ASSISTANT PRODUCT MARKETING MANAGERS**

Karen Hagen

Christy M. Miller

## **STRATEGIC MARKETING MANAGER**

Barry Pruett

## **TECHNICAL EDITORS**

Andy Angrick

Matthew Brown

Jim Hofman

Eric Ladd

Jim O'Donnell

Tony Schafer

Jon Steever

## **TECHNICAL SUPPORT SPECIALIST**

Nadeem Muhammed

## **ACQUISITIONS COORDINATOR**

Jane K. Brownlow

## **SOFTWARE RELATIONS COORDINATOR**

Susan D. Gallaher

## **EDITORIAL ASSISTANTS**

Jennifer L. Condon

Andrea Duvall

## **BOOK DESIGNER**

Ruth Harvey

## **COVER DESIGNER**

Dan Armstrong

## **PRODUCTION TEAM**

Marcia Brizendine

Bob LaRoche

Tim Neville

Paul Wilson

## **INDEXER**

Kevin Fulcher

## About the Authors

**Jerry Honeycutt** provides business-oriented technical leadership to the Internet community and software development industry. He has served companies such as The Travelers, IBM, Nielsen North America, IRM, Howard Systems International, and NCR. Jerry has participated in the industry since before the days of Microsoft Windows 1.0, and is completely hooked on Windows and the Internet.

Jerry is a leading author in the Internet field. He is the author of *Using Microsoft Plus!*; *Using the Internet with Windows 95*; *Windows 95 Registry & Customization Handbook*; *Special Edition Using the Windows 95 Registry*; *VBScript by Example*; *Special Edition Using the Internet*, Third Edition; *Using the Internet*, Second Edition; and *Windows NT and Windows 95 Registry Customization Handbook*, all of which are published by Que. Many of his books are sold internationally and have been translated into a variety of languages, including French, Italian, Korean, Japanese, Portuguese, Russian, Spanish, and Turkish.

Jerry is also a contributing author on *Special Edition Using Windows 95*, *Special Edition Using Netscape 2*, *Platinum Edition Using Windows 95*, *Visual Basic for Applications Database Solutions*, *Special Edition Using Netscape 3*, *Windows 95 Exam Guide*, *Netscape Navigator 3 Starter Kit*, *Using Java Workshop*, *Using JScript*, *Internet Explorer ActiveX and Plug-Ins Companion*, and *Windows NT Server 4.0 Advanced Technical Reference*, all of which are published by Que. He has been printed in *Computer Language* magazine and is a regular speaker at Windows World, Comdex, and other industry trade shows on topics related to software development, Windows, and the Internet.

Jerry graduated from the University of Texas at Dallas in 1992 with a B.S. degree in Computer Science. He currently lives in the Dallas suburb of Frisco, Texas with two Westies: Corky and Turbo, and a bird called Opie. Jerry is an avid golfer, and has a passion for fine photography and international travel. Feel free to contact Jerry on the Internet at [jerry@honeycutt.com](mailto:jerry@honeycutt.com).

**Mark R. Brown** has been writing computer books, magazine articles, and software manuals for over fourteen years. He was managing editor of *.info magazine*, when it was named one of the best computer magazines of the year by the Computer Press Association in 1991, and was nominated by the Software Publisher's Association for the Software Reviewer of the Year award in 1988. He's now a full-time freelance writer, who has contributed to over a half-dozen Que books, and is the author of Que's *Special Edition Using Netscape* series and the *WWW Plug-Ins Companion*. He is Webmaster of a Web site devoted to the topic of airships at <http://www2.giant.net/people/mbrown>, and can be reached via e-mail at [mbrown@avalon.net](mailto:mbrown@avalon.net).

**Jim O'Donnell** was born Oct. 17, 1963, in Pittsburgh, Pennsylvania (you may forward birthday greetings to [odonnj@rpi.edu](mailto:odonnj@rpi.edu)). After a number of unproductive years, he began his studies in electrical engineering at Rensselaer Polytechnic Institute. Jim liked that so much he spent 11 years at the school getting three degrees, graduating for the third (and final) time in the summer of 1992. When he isn't writing or researching for Que, he likes to run, row, play hockey, collect comic books and PEZ dispensers, and play the Best Board Game Ever, Cosmic Encounter. You can find Jim on the Web at <http://www.rpi.edu/~odonnj/>.

**Eric Ladd** ([erl1@access.digex.net](mailto:erl1@access.digex.net)) is a task leader for Advanced Technology Systems in McLean, Virginia, where he leads a group of Internet developers that support the FDIC.

By night, he toils endlessly for Macmillan Computer Publishing, coauthoring *Platinum Edition Using HTML 3.2, Java 1.1 and CGI* and *Special Edition Using Microsoft Internet Explorer 3* and contributing to several other titles. Eric lives in Washington, D.C., with his Boxer puppy, Zack.

**Robert Meegan** designs large software systems for manufacturing and research facilities. He has been working with computers for longer than he cares to recall. He can be reached at [rmeegan@ia.net](mailto:rmeegan@ia.net).

**Bill Bruns** is the Network Analyst at the Illini Student Union at the University of Illinois at Urbana-Champaign. He really wanted to work in television production, but got interested in computers while working on an undergraduate internship at Square One TV, a children's mathematics show produced by the Children's Television Workshop. Bill has been a technical editor for Que and has worked on such titles as *Platinum HTML 3.2, Java 1.1 and CGI*, *Running a Perfect Netscape Site*, and *10 Minute Guide to HTML Style Sheets*. Previously, he ran administrative computing at New York University's Tisch School of the Arts. Bill holds bachelors degrees in Telecommunications and English Literature from Indiana University, a Masters of Public Administration from New York University, and is a Certified Network Engineer. You can reach him at [brunsl@prairienet.org](mailto:brunsl@prairienet.org). He's also the Illini Union's Webmaster. Check out their pages at [www.union.uiuc.edu](http://www.union.uiuc.edu).

**Robert Niles** is a systems administrator and Web programmer for InCommand Inc., a company located in Yakima, Washington that specializes in Internet and intranet applications.

Robert loves all things Internet, especially the Web and CGI. He has been online since 1983, exploring the very nature of the online world. In 1984, he entered the military service as a communications specialist, taking a one year intensive course at the Presidio of Monterey as a Czech linguist. After completing military service, he returned to his home in the Yakima Valley. Currently, Robert can usually be found with his head almost stuck to a monitor—no matter where he is. He specializes in the UNIX environment, Perl, and SQL.

Robert lives in Selah, Washington (Apple Country) with his wife, Kimberly, his son, Michael, and his daughter, Shaela. You can find him on the Web at <http://www.selah.net/> or via e-mail at [rniles@selah.net](mailto:rniles@selah.net).

**David Wall** lives in Utah, where he distinguishes himself by belonging to neither an organized religious group nor a militia. Mainly, he skis, perfecting the fine points of the high-speed inverted face-plant.

**Matthew Brown** lives in the small town of Addison, TX with his fiancée Caroline, and their children Ramses, Cleo, and Bastian. Matthew currently works at National Knowledge Networks, Inc., as the Webmaster and local Windows NT expert. In his free time, Matthew enjoys listening to music, go-cart racing, and watching mindless television for hours.

**Rob Falla** ([rfalla@netroute.net](mailto:rfalla@netroute.net)) is a professional HTML developer and part-time writer. He authored *HTML Style Sheets Quick Reference*. When not writing, Rob is usually on the Internet, learning about the next big trends before they become popular.

# Acknowledgments

Wow. This book took a lot of work by many people. The authors, of course, contributed many hours of bone-racking work to this book. The editors made sure that this book's content was on the up-and-up. There are a couple of very special folks that I want you to be aware of, though:

- Philip Wescott put together a fine group of contributing authors in order to get this book done.
- Mark Cierznia and Steve Schafer are the development editors for this book. They kept their heads cool and allowed the authors to put together the best book possible. Thanks.
- Jim Bowie, the production editor for this book, is a joy. He works with almost unbearable details and schedules, but manages to keep his head screwed on tight.
- The technical editors—Jim O'Donnell, Eric Ladd, Andy Angrick, Jim Hofman, Matthew Brown, Jon Steever, and Tony Schafer—did a fabulous job of making sure the technical details were correct. They scoured over every line of HTML in this book, as well as the text content. They didn't just verify the facts, either; they often came up with great ideas for new content.
- The copy editors—Patricia R. Kinyon, Kelli M. Brooks, and Kate O. Givens—made sure the gibberish the authors banged out on the keyboard was readable and conformed to some sense of English structure.
- The contributing authors—O'Donnell, Ladd, Matthew Brown, Bill Bruns, Robert Meegan, David Wall, Rob Falla, and Robert Niles—were shooting at fast-moving targets. They did good.
- The Internet community contributed ideas for content and often provided solutions to problems for which we were stumped.

Last, but not least, I'd like to thank my coauthor, Mark R. Brown. His contribution to this book was significant. It was more fun with him than it would have been without him.

—Jerry Honeycutt

Thanks, first of all, to the small group of knowledgeable writers who did the research and wrote down the words that comprise this book. Professionals all, I think that they have done a marvelous job of documenting a potentially formidable subject in an entertaining and enlightening manner.

I also extend my deepest thanks to the entire crew at Que. Please read their names on the masthead and try to imagine how hard these people had to work to make a huge and detailed book like this happen in such a short period of time. From this talented group, I'd like to single out Philip Wescott, Mark Cierznia, and Jim Bowie for special thanks—Philip for his dogged discipline and diehard diligence in bringing together all the disparate elements of this tome; Mark for his sharp eye and controlled demeanor under incredible pressure, as always (it must be the cigars); and Jim for tying up all the loose ends. If you find yourself thinking that this book is pretty good, it's most likely due to something that these three did.

Closer to home, I'd like to thank Oran Sands for bringing me in the door at Que in the first place; Jim Oldfield for giving me my first chance (many, many years ago) to write professionally; and Benn Dunnington for an intense eight-year apprenticeship, which taught me to write improbably well under impossible deadlines. I'd also like to add to that list Mr. Frank Buston, my seventh grade English teacher, and Dr. Isaac Asimov, famed author of science books (both fact and fiction), for helping me to discover the seemingly disparate but surprisingly well-intertwined career paths of journalism and science.

Finally, I would like to thank my wife, Carol, for her continuing encouragement and support. We writers tend to make a big deal about how tough the writing life is, though it's often writers' spouses who suffer the most. She's suffered her share, and more.

—Mark R. Brown

## We'd Like to Hear from You!

As part of our continuing effort to produce books of the highest possible quality, Que would like to hear your comments. To stay competitive, we *really* want you, as a computer book reader and user, to let us know what you like or dislike most about this book or other Que products.

You can mail comments, ideas, or suggestions for improving future editions to the address below, or send us a fax at (317) 581-4663. Our staff and authors are available for questions and comments through our Internet site, at <http://www.quecorp.com>, and Macmillan Computer Publishing also has a forum on CompuServe (type **GO QUEBOOKS** at any prompt).

In addition to exploring our forum, please feel free to contact me personally to discuss your opinions of this book: I'm [mcierzniak@que.mcp.com](mailto:mcierzniak@que.mcp.com) on the Internet.

Thanks in advance—your comments will help us to continue publishing the best books available on new computer technologies in today's market.

Mark Cierzniak  
Product Development Specialist  
Que Corporation  
201 W. 103rd Street  
Indianapolis, Indiana 46290  
USA

**NOTE** Although we cannot provide general technical support, we're happy to help you resolve problems you encounter related to our books, disks, or other products. If you need such assistance, please contact our Tech Support department at 800-545-5914 ext. 3833.

To order other Que or Macmillan Computer Publishing books or products, please call our Customer Service department at 800-835-3202 ext. 666. ■

# Contents at a Glance

## I | Overview of WWW Publishing

---

- 1 The Wide World of HTML Publishing 9
- 2 HTML Page Design and Implementation 19

## II | Basic HTML

---

- 3 HTML Document Architecture 37
- 4 Document Formatting 59
- 5 Text Formatting 75
- 6 Graphics Formatting 87
- 7 Creating Document Links 105
- 8 Creating Lists 125
- 9 Creating Tables 143

## III | Advanced HTML

---

- 10 Frames 171
- 11 Forms 197
- 12 Imagemaps 235
- 13 Layers 259
- 14 Style Sheets 285
- 15 Netscape/Microsoft Extensions 311
- 16 Dynamic HTML 323

## IV | Incorporating Objects

---

- 17 Graphics 347
- 18 Browser Plug-Ins 375
- 19 Audio 391
- 20 Video, Animation, and Multimedia 419
- 21 VRML 457
- 22 ActiveX Controls 493
- 23 ActiveX Control Pad 517

## V | Programming and Scripting

---

- 24 Overview of Programming and Scripting 539
- 25 All About CGI Scripts 563
- 26 Java 583

- 27 JavaScript/JScript 599
- 28 Developing with LiveWire Pro 635
- 29 VBScript 663

## **VI | Web Site Management**

---

- 30 Putting Your Pages Online 699
- 31 Verification and File Management 713
- 32 Search Engines and Advertising 733
- 33 Security 757

## **VII | Web Site Tools**

---

- 34 HTML Tag Editors 775
- 35 WYSIWYG HTML Editors 801

## **VIII | Examples**

---

- 36 Accessing Databases 849
- 37 Creating a Web Site for Business 881
- 38 Setting Up a Company Intranet 899
- 39 Creating a Personal Web Site 917
- 40 Interactive Site 935

## **IX | Appendixes**

---

- A Looking Ahead to HTML 4.0 965
- B Creating Content for WebTV 969
- C What's on the CD? 979

Index 985

# Table of Contents

## **Introduction 1**

- What Is HTML? 2
- Why You Need This Book 2
- What's Changed in the Third Edition 3
- How This Book Is Organized 3
- Conventions Used in This Book 4

## **I | Overview of WWW Publishing**

---

### **1 The Wide World of HTML Publishing 9**

- The Birth of the World Wide Web 10
- The Web Explosion 11
- The Rise of the Corporate Intranet 14
- HTML in E-Mail and News 16
- HTML for GUIs 17

### **2 HTML Page Design and Implementation 19**

- What HTML Is 20
- The Only Tool You Really Need 21
- Image Is Not Everything 22
- The Right Stuff 24
  - Keep Focused 24
  - On the Links 24
  - Timeliness 25
  - Create a Vortex 27
- The Wrong Stuff 28
- Legal and Moral Issues 30
  - Be a Legal Eagle 31
  - Electronic Morality 32
- Sending Your Work Out into the World 33

## II Basic HTML

---

### 3 HTML Document Architecture 37

- Starting with the Basics 38
- The Document Heading 38
  - Naming Your Document 39
  - Indexing Your Document 41
  - Creating a Document Hierarchy 42
  - Customized Heading Information 45
  - Other Heading Elements 47
- The Document Body 47
  - Starting with the Required Elements 48
  - Attributes of the *BODY* Element 49
  - Coloring Your Documents 49
  - Filling In the Background 51
  - Setting the Margins 52
- Commenting Your HTML Documents 54
- The *Address* Element 54
- Special Characters 56
- Creating a Document Template 57

### 4 Document Formatting 59

- Breaking Text into Paragraphs 60
- Adding and Preventing Line Breaks 62
- Creating a Text Outline 65
  - Adding Headings 65
  - Adding Horizontal Lines 66
- Using Preformatted Text 70
- Layout Tips 72
- A Specialized Template 73

### 5 Text Formatting 75

- Text Formatting 76
- Logical Formatting 76
- The *<BLOCKQUOTE>* Element 77
- Physical Format Elements 79

|                                 |    |
|---------------------------------|----|
| Fonts                           | 81 |
| The <i>FONT</i> Element         | 81 |
| The <i>FACE</i> Attribute       | 82 |
| The <i>SIZE</i> Attribute       | 83 |
| The <i>COLOR</i> Attribute      | 84 |
| The <i>&lt;BASEFONT&gt;</i> Tag | 84 |
| Text Formatting Tips            | 86 |

## **6 Graphics Formatting 87**

|                                                 |     |
|-------------------------------------------------|-----|
| Understanding the Issues                        | 88  |
| What Should I Know When Using Inline Images?    | 88  |
| Do I Have to Worry About an Image's Copyright?  | 88  |
| Can I Get into Trouble for Pornographic Images? | 89  |
| Picking the Right Graphics File Format          | 90  |
| Formats                                         | 90  |
| File Speed                                      | 92  |
| Colors                                          | 92  |
| Loss                                            | 92  |
| Browser Support                                 | 93  |
| Adding Inline Images to Your HTML Document      | 93  |
| Aligning Text with an Inline Image              | 94  |
| Positioning an Image on the Web Page            | 95  |
| Giving the Browser Size Hints                   | 98  |
| Providing Alternative Text                      | 100 |
| Framing an Image with a Border                  | 101 |
| Giving an Image Space                           | 102 |
| Using an Image as an Anchor                     | 103 |

## **7 Creating Document Links 105**

|                                        |     |
|----------------------------------------|-----|
| Understanding Hypertext and Hypermedia | 106 |
| Understanding Links                    | 106 |
| Anchors                                | 107 |
| URL References                         | 109 |
| Linking to Documents and Files         | 110 |
| Internal Links                         | 112 |
| Files, Plug-In Content, and So On      | 113 |
| Linking to Other Net Resources         | 114 |

|                                                       |     |
|-------------------------------------------------------|-----|
| Creating a Link to E-Mail                             | 115 |
| Creating a Link to UseNet News                        | 116 |
| Making FTP Available on Your Site                     | 118 |
| Linking Your Home Page to a Gopher Site               | 119 |
| Providing Access to a Large Database with a WAIS Link | 120 |
| Accessing Remote Computers with Telnet Links          | 121 |
| How Links Work Together                               | 122 |

## **8 Creating Lists 125**

|                                                 |     |
|-------------------------------------------------|-----|
| Creating an Ordered List                        | 126 |
| Using the <code>&lt;OL&gt;</code> Tag           | 126 |
| New HTML 3.2 <code>&lt;OL&gt;</code> Attributes | 128 |
| Creating an Unordered List                      | 130 |
| Using the <code>&lt;UL&gt;</code> Tag           | 131 |
| New HTML 3.2 <code>&lt;UL&gt;</code> Attributes | 132 |
| Creating Menu Lists                             | 134 |
| Creating Directory Lists                        | 135 |
| Creating Definition Lists                       | 136 |
| Combining List Types                            | 138 |
| Manually Formatting Lists                       | 140 |

## **9 Creating Tables 143**

|                                      |     |
|--------------------------------------|-----|
| Introducing Tables                   | 144 |
| Understanding the Basic Tags         | 144 |
| Aligning Table Elements              | 146 |
| Working with Advanced Tables         | 149 |
| Creating Borderless Tables           | 149 |
| Spanning Rows and Columns            | 152 |
| Understanding Empty Cells            | 154 |
| Controlling Table Layout             | 155 |
| Using Color in Tables                | 157 |
| Using a Table Alternative            | 159 |
| Table Examples                       | 162 |
| Using Tables as a Layout Tool        | 162 |
| Combining Text and Lists             | 164 |
| Nesting HTML Tables                  | 165 |
| Using an Image as a Table Header     | 167 |
| Using a Table to Lay Out a Home Page | 168 |

### III | Advanced HTML

---

#### 10 Frames 171

- The Frames Concept 172
  - Netscape's DevEdge Site 172
  - The CyberDungeon 173
  - The Mondrian Machine 174
- Frame Creation Basics 175
- The *FRAMESET* Container 176
  - Defining Rows and Columns 177
- The *FRAME* Tag 178
  - Going to the Source 179
  - Providing Alternate Content 180
- A Few Simple Frame Examples 181
  - A Simple Page with Two Frames 181
  - A Simple Rectangular Grid of Frames 182
  - Creating a Complex Grid of Frames 183
- Modifying a Frame's Look and Feel 186
  - Frame Margins 186
  - Frame Scrollbars 187
  - Frame Resizing 187
  - Frame Borders 188
- Targeted Hyperlinks 190
  - Naming and Targeting Frames 190
  - Legitimate Names 191
  - Updating More Than One Frame at a Time 192

#### 11 Forms 197

- Introducing HTML Forms 198
- Working with HTML Forms Tags 198
  - <FORM>* 199
  - <TEXTAREA>* 199
  - <SELECT>* 201
  - <INPUT>* 204
    - Setting the *<INPUT>* Tag's *TYPE* 205
- Formatting and Designing Forms 212
  - Using Line Break Tags 213
  - Using List Tags 220

|                                    |     |
|------------------------------------|-----|
| Check Box and Radio Button Layouts | 223 |
| Multiple Forms in a Document       | 227 |
| Combining Forms with Tables        | 229 |
| Final Notes on Form Layouts        | 233 |

## 12 Imagemaps 235

|                                                  |     |
|--------------------------------------------------|-----|
| Introducing Imagemaps                            | 236 |
| When to Use Imagemaps                            | 237 |
| When Not to Use Imagemaps                        | 238 |
| Using Server-Side Imagemaps                      | 239 |
| Creating the Imagemap Graphic                    | 239 |
| Understanding Imagemap Definition Files          | 240 |
| Using a CGI Program to Look Up Mouse Clicks      | 245 |
| Putting the HTML on the Web Page                 | 246 |
| Putting Your Imagemap Through a Dry Run          | 246 |
| Building Client-Side Imagemaps                   | 247 |
| Defining a Client-Side Imagemap                  | 247 |
| Referencing the Client-Side, Imagemap Definition | 248 |
| Combining Client-Side and Server-Side Imagemaps  | 249 |
| Working with Mapping Programs                    | 249 |
| Working with Mapedit                             | 249 |
| Using WebMap                                     | 253 |
| Using Alternatives to Imagemaps                  | 255 |
| Image Chunks                                     | 255 |
| Scripting                                        | 255 |
| Textual Alternatives                             | 256 |
| Example: Building an Imagemap                    | 256 |

## 13 Layers 259

|                                                           |     |
|-----------------------------------------------------------|-----|
| Creating a Basic Layer                                    | 260 |
| Positioning a Layer                                       | 261 |
| Changing the Size of a Layer                              | 264 |
| Using a Background Image or Color                         | 266 |
| Using Multiple Layers                                     | 268 |
| Changing a Layer's <i>Z-INDEX</i>                         | 270 |
| Overlapping Two Layers with <i>ABOVE</i> and <i>BELOW</i> | 272 |
| Attaching Scripts to Layers                               | 274 |

|                                                |     |
|------------------------------------------------|-----|
| Using a Script to Hide or Show a Layer         | 275 |
| Moving a Layer with a Script                   | 278 |
| Nesting Layers                                 | 280 |
| Putting Layers to Practical Use: Fly-Over Tips | 282 |

## **14 Style Sheets 285**

|                                                         |     |
|---------------------------------------------------------|-----|
| Attaching a Style Sheet to Your HTML Document           | 286 |
| Linking a Style Sheet                                   | 286 |
| Embedding a Style Sheet                                 | 287 |
| Defining Styles Inline                                  | 288 |
| Importing Style Sheets                                  | 289 |
| Understanding What Style Sheets Look Like (Syntax)      | 289 |
| Setting Multiple Properties                             | 290 |
| Grouping Selectors                                      | 290 |
| Adding Comments to Your Style Sheet                     | 290 |
| Exploring the Properties of a Style                     | 291 |
| Background Properties                                   | 293 |
| Box Properties                                          | 296 |
| List Properties                                         | 300 |
| Text Properties                                         | 301 |
| Grouping Properties to Simplify Style Definitions       | 306 |
| Border Properties                                       | 306 |
| Background Properties                                   | 306 |
| Font Properties                                         | 306 |
| List Properties                                         | 307 |
| Margin Properties                                       | 307 |
| Padding Properties                                      | 307 |
| Using Inheritance                                       | 307 |
| Using Contextual Selectors                              | 308 |
| Understanding the Cascade                               | 308 |
| Working with Classes in Your Style Sheets               | 309 |
| Taking Care of Browsers That Don't Support Style Sheets | 309 |

## **15 Netscape/Microsoft Extensions 311**

|                       |     |
|-----------------------|-----|
| The Browser Wars      | 312 |
| Are Differences Good? | 312 |
| Joining the Fracas    | 315 |

## The HTML "Standard" and the W3C 316

### Netscape Extensions 317

Blink 317

Newspaper-Style Columns 317

Whitespace 318

Embedding Technologies 318

### Microsoft Extensions 319

Background Music 319

Fixed Backgrounds 319

Marquees 320

AVI Video 320

Style Sheets 320

The `<OBJECT>` Tag 321

## 16 Dynamic HTML 323

### Introducing Dynamic HTML 324

Reduce Download Time 325

Tools of Dynamic HTML 325

Here's How Dynamic HTML Works 326

### The Basics of Dynamic HTML 326

Understanding the Components 327

Data Binding 328

2-D Layout 328

### Using the Document Structure 328

Why Is Structure So Important in Dynamic HTML? 328

How Collections Work 329

*cancelBubble* versus *returnValue* 330

### Using Events with Dynamic HTML 331

Using Elements of Events 331

Using the Events 332

### Using the CSSOM 339

Containment in HTML 339

Style Model 340

styleRule Model and styleRule Collections 341

CSSOM Naming Rules 343

## **IV | Incorporating Objects**

---

### **17 Graphics 347**

- How Graphic Information Is Stored 348
  - Bitmapped Graphics 348
  - Vector Graphics 348
- Web Graphic Formats 349
  - GIF 349
  - JPEG 353
  - When to Use GIF or JPEG 354
  - Up and Coming: PNG 355
- Making Good Use of Images 355
  - Main Page Graphics 355
  - Icons 356
  - Navigation Bars 356
  - Backgrounds 357
  - Finding Graphics Online 357
- Bandwidth Issues 358
  - Think of Your Audience 358
  - Tips for Reducing File Size 359
- Working with Scanned Images 360
  - When to Use Them and Where to Get Them 360
  - Manipulating Scanned Images 360
- Useful Graphics Tools 360
  - LView Pro 361
  - Paint Shop Pro 364
  - Adobe Photoshop 367
  - Microsoft Image Composer 370
  - GIF Construction Set 372

### **18 Browser Plug-Ins 375**

- How Plug-Ins "Plug In" 376
- The Three Kinds of Plug-Ins 379
- Running a Plug-In 381
- Using Plug-Ins with Netscape Navigator 381
- Identifying Installed Netscape Navigator Plug-Ins 382

Adding Plug-In Content to Your Pages 383

The *Embed* Tag 384

Server Issues 385

Setting MIME Types 385

Installing Content Servers 387

Issues of Space and Time 388

Bandwidth Limitations 388

Storage Problems 389

## **19 Audio 391**

Audio Hardware—What You Need 392

LiveAudio 392

LiveAudio Controls 393

Using the *EMBED* Tag with LiveAudio 394

Other Audio Plug-Ins 396

TrueSpeech 396

Crescendo and Crescendo Plus 397

ToolVox 397

EchoSpeech 398

Talker and Other Macintosh Speech Plug-Ins 398

More Sound Plug-Ins 399

RealAudio 400

How RealAudio Works 400

The RealAudio Player 403

Installation and Setup 404

Controls and Menus 405

The RealAudio Encoder 406

Installing the RealAudio Encoder 407

Encoding 407

Using RealAudio Content on Your Web Pages 409

The RealAudio Servers 413

The RealAudio Personal Server 413

Setting MIME Types 415

The RealAudio Server 3.0 415

Synchronized Multimedia 417

**20 Video, Animation, and Multimedia 419**

- Limitations on Multimedia Content 420
  - File Size and Dialup Connections 420
  - Browser Compatibility 421
  - What Are Plug-Ins Best Used For? 423
  - Plug-In Content-Creation Programs 424
  - Keep It Small 425
- Plug-Ins and Intranets 425
- Video and Animation Plug-Ins 426
- When to Use Video Content 426
  - Video Speed 427
  - Content Considerations 427
- Creating Video Content 429
- Video Frame Digitizers 430
  - Snappy 430
  - Connectix QuickCam 431
  - WinCam.One 431
- Animation 432
- Video for Windows Plug-Ins 433
  - LiveVideo 433
  - VDOLive 435
  - Other AVI Video Plug-Ins 435
- QuickTime Plug-Ins 436
  - The Apple QuickTime Plug-In 436
  - Other QuickTime Plug-Ins 438
- MPEG Plug-Ins 439
  - InterVU MPEG Plug-In 439
- Animation Plug-Ins 441
  - Enliven 441
  - A Few More Animation Plug-Ins 443
- Multimedia Plug-Ins 443
- ASAP WebShow 444
  - Other Multimedia Plug-Ins 448
- Shockwave for Macromedia Director 449
- Director versus Java and JavaScript 449

|                                             |     |
|---------------------------------------------|-----|
| What You Can Do with Shockwave for Director | 450 |
| Creating Director Movies for Shockwave      | 452 |
| Lingo Network Extensions                    | 453 |
| Director Limitations on HTML Pages          | 454 |
| Page Design Considerations                  | 454 |
| Optimizing Director Movies                  | 454 |
| More Information                            | 455 |

## **21 VRML 457**

|                                             |     |
|---------------------------------------------|-----|
| The Virtual Reality Modeling Language       | 458 |
| VRML Objects                                | 458 |
| VRML Worlds                                 | 459 |
| Adding Motion and More to VRML              | 459 |
| Why (and How) Should You Use VRML?          | 459 |
| Basic VRML Syntax                           | 460 |
| Nodes and Fields                            | 462 |
| Objects, Hierarchies, Separators            | 463 |
| Simple VRML Objects                         | 463 |
| Geometric Shapes                            | 463 |
| ASCII Text                                  | 464 |
| General VRML Shapes                         | 465 |
| Adding Color and Texture to VRML Objects    | 467 |
| <i>Material</i>                             | 467 |
| <i>Texture2</i>                             | 467 |
| Creating VRML Worlds and Inline VRML Scenes | 468 |
| Design Considerations                       | 469 |
| Size and Detail                             | 469 |
| Design and Layout                           | 470 |
| VRML World Design Steps                     | 470 |
| Mapping Your VRML Environment               | 470 |
| Putting Your Design on Paper                | 472 |
| Moving Things Around                        | 473 |
| Creating Object Hierarchies                 | 475 |
| Adding Realism with Textures                | 479 |
| Linking to the Web                          | 481 |
| Bringing VRML to Life                       | 482 |
| Live3D Extensions for Dynamic Worlds        | 482 |

Example Live3D VRML World 484

Features of VRML 2.0 489

Enhanced Static Worlds 490

Interaction 490

Animation 490

Scripting 490

Prototyping 490

VRML Resources 491

VRML Software 491

VRML Object Libraries 491

## **22 ActiveX Controls 493**

The Evolution of ActiveX Controls 494

OLE 494

ActiveX Controls 495

What ActiveX Controls Mean to You 496

Working with Data 496

Integrating with Windows 497

Provide Building Blocks 498

ActiveX versus Everything Else 499

ActiveX and Forms 499

ActiveX and Plug-Ins 500

ActiveX and Java 500

Getting Microsoft's ActiveX Controls 501

Label Control 503

Preloader Control 503

Timer Control 504

Menu Control 504

Inserting Controls with the *OBJECT* Tag 505

Setting a Control's Properties with the *PARAM* Tag 506

More About the *OBJECT* Tag 507

Using ActiveX Controls in Netscape (NCompass) 510

Connecting Controls to Scripts 511

Handling an Event 511

Changing an Object's Properties 512

Tying It All Together with an Example 513

## **23 ActiveX Control Pad 517**

- Downloading and Installing the ActiveX Control Pad 518
- Getting Acquainted with the HTML Editor 518
- Placing Objects into Your HTML File 519
- Editing Scripts with the Control Pad's Script Wizard 521
  - List View 521
  - Code View 523
- Controlling Page Layout with the HTML Layout Control 524
  - Understanding the Layout Control 525
  - Inserting the Layout Control into an HTML File 525
  - Editing the Layout 526
- Learning by Doing 530
  - Building a Navigational Menu 530
  - Pop-Up Help (Tooltips) 534

# **V | Programming and Scripting**

---

## **24 Overview of Programming and Scripting 539**

- Picking the Best Tool 540
  - Programming and Scripting Options 540
  - Scripting Advantages 540
  - Scripting Disadvantages 541
- Understanding Event-Driven Programming 541
  - What Are Events? 542
  - How Events Work in Windows (and Other GUIs) 543
  - How Events Work on a Web Page (Scripting) 543
- Learning to Program: An Overview 544
  - Variables 544
  - Mathematical Expressions 544
  - Boolean Expressions 544
  - Decisions 546
  - Loops 547
  - Functions and Subprocedures 547
- Understanding the Scripting Object Model 548
  - window* 549
  - document* 552

|                                     |     |
|-------------------------------------|-----|
| Working with Scripts                | 556 |
| Embedding Scripts in Your Web Pages | 556 |
| Managing a VBScript Project         | 557 |

## **25 All About CGI Scripts 563**

|                                     |     |
|-------------------------------------|-----|
| What Is CGI?                        | 564 |
| How the CGI Works                   | 564 |
| The Process                         | 564 |
| Characteristics of the CGI          | 565 |
| The MIME Content-Type Output Header | 567 |
| Using a Content-Type Output Header  | 568 |
| Environment Variables               | 569 |
| Can You Write CGI Scripts?          | 571 |
| Can You Write CGI Scripts?          | 571 |
| Which Language Should You Use?      | 572 |
| Common CGI Scripting Languages      | 573 |
| Creating CGI Scripts                | 573 |
| An E-Mail Script                    | 573 |
| A Simple Guestbook Script           | 576 |
| A Redirection Script                | 577 |
| Simple Count                        | 579 |
| An Advertisement Banner             | 581 |

## **26 Java 583**

|                                       |     |
|---------------------------------------|-----|
| How Java Got Started                  | 584 |
| Getting Started with Java             | 584 |
| Using Java Applets in Web Pages       | 585 |
| Java Security                         | 592 |
| Notable Java Applets                  | 593 |
| Java Resources                        | 595 |
| Development Tools                     | 595 |
| Web Resources                         | 595 |
| Internet Newsgroups and Mailing Lists | 596 |

## **27 JavaScript/JScript 599**

|                               |     |
|-------------------------------|-----|
| Introduction to JavaScript    | 600 |
| Why Use a Scripting Language? | 600 |
| What Can JavaScript Do?       | 601 |

|                                          |            |
|------------------------------------------|------------|
| What Does JavaScript Look Like?          | 601        |
| JavaScript Programming Conventions       | 603        |
| The JavaScript Language                  | 604        |
| Using Identifiers                        | 604        |
| Using Functions, Objects, and Properties | 604        |
| JavaScript and the Browser               | 606        |
| When Scripts Get Executed                | 606        |
| Where to Put Your Scripts                | 607        |
| Objects and Events                       | 607        |
| HTML Objects and Events                  | 610        |
| Programming with JavaScript              | 614        |
| Expressions                              | 614        |
| Operators                                | 614        |
| Controlling Your JavaScripts             | 617        |
| JavaScript Reserved Words                | 618        |
| Command Reference                        | 619        |
| Sample JavaScript Code                   | 623        |
| Dumping an Object's Properties           | 623        |
| Building a Link Table                    | 624        |
| Updating Data in Form Fields             | 625        |
| Validating Data in Form Fields           | 626        |
| A Pseudo-Scientific Calculator           | 626        |
| JavaScript Style Sheets (JSSS)           | 631        |
| Including Style in HTML                  | 632        |
| The Contextual Function                  | 633        |
| The <i>Tags</i> Object                   | 633        |
| The <i>Classes</i> Object                | 633        |
| The <i>ids</i> object                    | 634        |
| The <i>Tags.__.apply</i> Property        | 634        |
| <b>28 Developing with LiveWire Pro</b>   | <b>635</b> |
| The "Pro" in LiveWire Pro                | 636        |
| How LiveWire Works                       | 637        |
| A Brief HTTP Tutorial                    | 638        |
| What Options Does Netscape Offer?        | 639        |
| What Does LiveWire Do with a Request?    | 640        |
| Understanding SQL                        | 640        |
| The Relational Model                     | 641        |
| A SQL Primer                             | 643        |

|                                                       |            |
|-------------------------------------------------------|------------|
| Understanding Transactions                            | 644        |
| Understanding Cursors                                 | 645        |
| Introduction to Crystal Reports                       | 646        |
| The Database Connectivity Library                     | 647        |
| Opening and Closing the Connection                    | 647        |
| Inserting Data into Tables                            | 649        |
| Deleting Rows                                         | 649        |
| Accessing Data a Row at a Time                        | 650        |
| Accessing Data as a Set                               | 650        |
| Using BLOBs                                           | 651        |
| Transactions in LiveWire Pro                          | 652        |
| Error Handling                                        | 653        |
| JavaScript and the Second Generation Netscape Servers | 653        |
| Understanding Java and JavaScript                     | 653        |
| Server-Side JavaScript                                | 654        |
| The Virtual Java Machine                              | 655        |
| Putting It All Together—a Database Example            | 655        |
| <br>                                                  |            |
| <b>29 VBScript</b>                                    | <b>663</b> |
| Visual Basic Meets the Internet                       | 664        |
| Visual Basic                                          | 664        |
| The Internet                                          | 664        |
| VBScript                                              | 665        |
| Learning the VBScript Language                        | 666        |
| VBScript Variables                                    | 666        |
| VBScript Operators                                    | 668        |
| VBScript Keywords                                     | 670        |
| VBScript Runtime Library                              | 675        |
| VBScript Event Handlers                               | 679        |
| VBScript Event Procedures                             | 679        |
| Inline Event Handler (Scriptlet)                      | 680        |
| FOR/EVENT Attributes                                  | 681        |
| Handling Common Events                                | 681        |
| Connecting Scripts to Forms                           | 683        |
| Handling an Element's Events                          | 683        |
| Setting and Getting an Element's Value                | 685        |
| Element Properties, Methods, and Events               | 686        |

|                                            |     |
|--------------------------------------------|-----|
| Using the Coding Conventions               | 689 |
| Using the Naming Convention                | 689 |
| Formatting Conventions                     | 693 |
| Commenting Conventions                     | 694 |
| Examples                                   | 695 |
| Validating a Form                          | 695 |
| Scrolling a Message Across the Status Line | 696 |
| Creating Enhanced Client-Side Imagemaps    | 696 |

## VI | Web Site Management

---

### 30 Putting Your Pages Online 699

|                                       |     |
|---------------------------------------|-----|
| What Are Your Options?                | 700 |
| Uploading Files to an ISP             | 700 |
| Run Your Own Server                   | 700 |
| Choosing an Option                    | 701 |
| Uploading to an ISP                   | 702 |
| Preparing Your Site                   | 703 |
| Preparing the UNIX Directory          | 705 |
| Uploading the Files                   | 706 |
| Running Your Own Server               | 708 |
| The Right Server                      | 708 |
| Putting Your Site Online              | 709 |
| Web Servers                           | 710 |
| Microsoft Internet Information Server | 710 |
| Luckman Web Commander                 | 711 |
| WebSite Professional                  | 712 |

### 31 Verification and File Management 713

|                                |     |
|--------------------------------|-----|
| Validating Your HTML Documents | 714 |
| Doctor HTML                    | 714 |
| Weblint                        | 718 |
| WebTechs                       | 720 |
| Other Verification Services    | 723 |
| Managing Your Source Files     | 724 |
| Visual SourceSafe              | 725 |
| Spidersoft WebGal              | 726 |

|                                 |     |
|---------------------------------|-----|
| Using Microsoft FrontPage       | 727 |
| Using NetObjects Fusion         | 728 |
| Working Directly with Files     | 729 |
| Mirroring Your Web Site Locally | 730 |

## **32 Search Engines and Advertising 733**

|                                                 |     |
|-------------------------------------------------|-----|
| Understanding What Kind of Advertising You Need | 734 |
| Listing with WWW Search Engines                 | 735 |
| The Major Search Engines                        | 736 |
| Using Submission Services                       | 742 |
| Putting Your Site at the Top of the List        | 744 |
| Good Planning Gets You on Top                   | 744 |
| Other Tips for Getting on Top                   | 744 |
| Categorize Your Site with the <META> Tag        | 745 |
| Getting Noticed by Using Other Ways             | 746 |
| Best of the Web Listings                        | 746 |
| Links from Other Web Sites                      | 747 |
| Specialized Index Pages                         | 749 |
| Using Newsgroups Effectively                    | 751 |
| Using a Publicity Consultant                    | 754 |
| Advertise Your Site in Print                    | 754 |

## **33 Security 757**

|                                                  |     |
|--------------------------------------------------|-----|
| Security on the Internet                         | 758 |
| Play at Your Own Risk                            | 758 |
| Safe Serving                                     | 759 |
| Viruses                                          | 760 |
| What Are They?                                   | 760 |
| How to Protect Your System                       | 760 |
| General Security Precautions                     | 761 |
| If It's Not There, They Can't Steal It           | 761 |
| Get Rid of Guest Accounts                        | 761 |
| Rename the Administrator Account                 | 762 |
| Choose Passwords Intelligently                   | 762 |
| Disable Unneeded Protocols                       | 763 |
| Be Miserly with Write Privileges                 | 763 |
| Disable Your Server's Directory Browsing Feature | 764 |
| Watch Your System Logs                           | 764 |

|                                      |     |
|--------------------------------------|-----|
| Scripting, Programming, and Security | 765 |
| Java                                 | 765 |
| JavaScript and VBScript              | 766 |
| CGI                                  | 766 |
| Plug-Ins and ActiveX Controls        | 766 |
| Secure Transactions                  | 767 |
| Secure Sockets Layer (SSL)           | 767 |
| Secure HTTP (SHTTP)                  | 767 |
| Certificates                         | 768 |
| Pretty Good Privacy (PGP)            | 768 |
| Firewalls                            | 769 |
| How Firewalls Work                   | 769 |
| Implementing a Firewall              | 769 |
| Checking Your Security               | 770 |
| SATAN                                | 770 |
| Cracking Consultants                 | 770 |

## **VII | Web Site Tools**

---

### **34 HTML Tag Editors 775**

|                                         |     |
|-----------------------------------------|-----|
| What Is an HTML Tag Editor?             | 776 |
| Tag Editor Features                     | 776 |
| Why Use a Tag Editor?                   | 776 |
| What Is the Future of HTML Tag Editing? | 777 |
| Popular HTML Tag Editors                | 777 |
| HotDog Web Editor by Sausage Software   | 777 |
| Downloading and Installation            | 777 |
| HotDog User Interface                   | 778 |
| Basic HotDog Operations                 | 781 |
| Configuration and Customizing           | 782 |
| Advanced Features                       | 784 |
| HoTMetaL by SoftQuad                    | 784 |
| Downloading and Installation            | 785 |
| HoTMetaL User Interface                 | 785 |
| Basic HoTMetaL Operations               | 786 |
| Advanced Features                       | 788 |
| HomeSite by Nick Bradbury               | 789 |
| Downloading and Installation            | 789 |
| HomeSite User Interface                 | 789 |

|                               |     |
|-------------------------------|-----|
| Basic HomeSite Operations     | 791 |
| Configuration and Customizing | 793 |
| Advanced Features             | 795 |
| Program Information Summary   | 798 |
| Other HTML Editors            | 799 |
| ActiveX Control Pad           | 799 |
| Web Resources                 | 799 |

### **35 WYSIWYG HTML Editors 801**

|                                   |     |
|-----------------------------------|-----|
| A Word About WYSIWYG HTML Editors | 802 |
| Netscape Communicator             | 803 |
| Getting Started                   | 804 |
| Building Your Document            | 805 |
| Publishing Your Document          | 811 |
| Microsoft FrontPage 97            | 812 |
| The FrontPage Explorer            | 813 |
| The FrontPage Editor              | 817 |
| Using WebBots                     | 826 |
| Adobe PageMill                    | 827 |
| The PageMill Window               | 827 |
| The PageMill Toolbar              | 829 |
| The Link Location Bar             | 829 |
| The Attribute Inspector           | 830 |
| The Image Viewer                  | 831 |
| The Pasteboard                    | 832 |
| Creating Tables                   | 833 |
| Creating Forms                    | 838 |
| Creating Framed Documents         | 840 |
| Creating Targets                  | 844 |
| Saving All This Stuff             | 845 |

## **VIII | Examples**

---

### **36 Accessing Databases 849**

|                     |     |
|---------------------|-----|
| Available Databases | 850 |
| Oracle              | 850 |
| Sybase              | 850 |
| mSQL                | 851 |

|                                                  |     |
|--------------------------------------------------|-----|
| Illustra                                         | 851 |
| Microsoft SQL                                    | 851 |
| Postgres95                                       | 851 |
| Ingres                                           | 852 |
| FoxPro                                           | 852 |
| Microsoft Access                                 | 852 |
| Side-by-Side Comparison                          | 853 |
| Database Tools                                   | 853 |
| PHP/FI                                           | 854 |
| Cold Fusion                                      | 854 |
| W3-mSQL                                          | 855 |
| MsqlPerl                                         | 855 |
| MsqlJava                                         | 855 |
| Microsoft's dbWeb                                | 855 |
| WDB                                              | 856 |
| Web/Genera                                       | 856 |
| MORE                                             | 856 |
| DBI                                              | 857 |
| DBGateway                                        | 857 |
| Databases for Everyone                           | 858 |
| Flat File Databases                              | 858 |
| dbm Databases                                    | 863 |
| Using SQL Databases                              | 870 |
| Using Microsoft's Internet Database Connector    | 871 |
| IDC's Required Directives                        | 873 |
| Additional IDC Directives                        | 873 |
| The <i>BeginDetail</i> and <i>EndDetail</i> Tags | 874 |
| The <i>CurrentRecord</i> Directive               | 874 |
| Conditional Operators                            | 874 |
| The <i>MaxRecords</i> Variable                   | 875 |
| Fields                                           | 875 |
| HTTP Variables                                   | 876 |
| Using Oracle                                     | 877 |
| Web Listener                                     | 877 |
| PL/SQL                                           | 878 |
| Additional Information                           | 879 |

**37 Creating a Web Site for Business 881**

- Why Be on the Web? 882
  - Direct Sales 882
  - Name Brand Recognition 883
  - Customer Support 884
  - Promoting Investment 885
  - More Examples Online 887
- Nonprofit Organizations 887
- Creating a Business Site 889
  - Focusing the Pitch 889
  - Organizing the Site 889
  - Joe's Home Page 890
  - Up-Front Material 892
  - The Navigational Imagemap 893
  - Establishing Site Identity 894
  - The Teasers 894
  - The Wrap Up 895
  - More Work Ahead 895
- Putting Existing Documents on the Web 896
  - What to Include 896
  - What to Omit 896
  - How to Reorganize 896
  - The Mechanics 897
  - Templates for Word Processors 897

**38 Setting Up a Company Intranet 899**

- Defining the Intranet 900
  - Uses of an Intranet 900
  - Getting Started 900
  - Creating the Intranet Skeleton 902
  - Sharing Information 904
- Addressing Security Concerns 905
  - Incorporating a Firewall into Your LAN 905
  - Implementing Security Through a Proxy Server 906
  - Restricting Access on Your Web Server 907
- Targeting Your Audience 909
  - Creating Functional Content 909
  - Indexing Your Site 910

|                                  |     |
|----------------------------------|-----|
| Getting Users Connected          | 911 |
| Training Your Users              | 911 |
| Group Participation              | 912 |
| Advanced Intranet Uses           | 913 |
| Hosting Discussion Groups        | 913 |
| Using E-Mail to Communicate      | 914 |
| Collaboration with Whiteboarding | 915 |

### **39 Creating a Personal Web Site 917**

|                               |     |
|-------------------------------|-----|
| Getting Started               | 918 |
| Choosing a Focused Topic      | 918 |
| Selecting Content             | 919 |
| Organizing the Site           | 920 |
| Creating the Bones            | 921 |
| UpShip.htm                    | 922 |
| Title.htm                     | 922 |
| Frameset.htm                  | 922 |
| Navbar.htm                    | 923 |
| Photos.htm                    | 923 |
| Making Adjustments            | 923 |
| Bringing Together the Content | 925 |
| Creating the Photo Album      | 926 |
| Photos.htm                    | 927 |
| Thumbs.htm                    | 928 |
| Some Useful Links             | 929 |
| Links.htm                     | 930 |
| Adding Some Articles          | 931 |
| Texts.htm                     | 932 |
| Under Construction            | 932 |
| Putting It on the Web         | 933 |
| Your Personal Web Site        | 933 |

### **40 Interactive Site 935**

|                                           |     |
|-------------------------------------------|-----|
| Creating a Feedback Form                  | 936 |
| Auto-Expanding the Form with Layers       | 938 |
| Validating User Input with Scripts        | 941 |
| Saving the Form's Contents with Cookies   | 943 |
| Dressing Up and Adding Value to Your Site | 946 |
| Add a Yahoo! Search Form                  | 946 |

|                                               |     |
|-----------------------------------------------|-----|
| Add a Yahoo! News Page                        | 947 |
| Provide Guidance by Using the Microsoft Agent | 949 |
| Providing Advanced Navigation                 | 951 |
| Advancing the Page Automatically              | 952 |
| Using Forms and Scripts for Navigation        | 952 |
| Using Forms and CGI for Navigation            | 954 |
| Creating a Remote Control                     | 954 |
| Searching Your Site                           | 957 |
| CGI Scripts                                   | 957 |
| Netscape                                      | 960 |
| Microsoft Index Server                        | 960 |
| FrontPage WebBots                             | 960 |

## **IX | Appendixes**

---

### **A Looking Ahead to HTML 4.0 965**

|                                    |     |
|------------------------------------|-----|
| Where HTML Is and How It Got There | 966 |
| Where HTML Is Going                | 966 |

### **B Creating Content for WebTV 969**

|                                      |     |
|--------------------------------------|-----|
| Reviewing WebTV's Content Philosophy | 970 |
| Working Within WebTV's Limitations   | 971 |
| Unsupported HTML Tags and Attributes | 971 |
| WebTV HTML Extensions                | 973 |
| Understanding WebTV's Style Guide    | 978 |

### **C What's on the CD? 979**

|                                                                |     |
|----------------------------------------------------------------|-----|
| Web Browsers                                                   | 980 |
| HTML Editors and Utilities                                     | 980 |
| Web Plug-Ins                                                   | 980 |
| Web Servers                                                    | 981 |
| Java                                                           | 981 |
| Web Utilities, Helper Applications, and Other Useful Utilities | 981 |
| Free HTML Versions of Popular Que Books                        | 983 |

### **Index 985**

# Frames

by Mark R. Brown

Sometimes, just the right frame can make a painting look perfect—and the wrong frame can ruin it. Likewise, prudent use of frames can help make your HTML documents better organized, more easily navigable, and more impressive looking. But used incorrectly or to excess, frames can ruin your site by making information inaccessible, fragmented, and just plain ugly.

You might wonder if you should even use frames on your pages at all, since they aren't supported by all browsers and are not a part of the current HTML standard. While it's true that not all browsers support frames, the top two—Netscape Navigator and Microsoft Internet Explorer—certainly do. That's well over 90 percent of your audience on the Web. If you can live with potentially alienating the other 10 percent who use nonframes-capable browsers (like the all-text UNIX browser, Lynx), frames can add a touch of class to your Web site. You'll find later in this chapter that this isn't necessarily an "either/or" decision, anyway—the HTML syntax for creating frames lets you provide alternate content for those who cannot display them. And if you're working on a corporate intranet that has standardized on Navigator or Explorer as the browser for all your users, of course compatibility isn't an issue at all. ■

## Create navigational menus with frames

If having a stay-put navigational menu would be of benefit to your site, then frames are the answer.

## How do you associate a link with a frame?

Frames accomplish much of their magic through targeted hyperlinks, which identify in which frame an URL is loaded.

## Learn how to create complex frames

Though simple frames are easy to create, more complex arrangements of frames can be difficult to understand and to implement.

## The Frames Concept

First introduced in Netscape Navigator 2.0, HTML frames create independently changeable and (sometimes) scrollable windows that tile together to break up and organize a display so that it's not only more visually appealing, but easier to work with. Frames are similar in many ways to HTML tables. If you understand how tables work, you'll have a jump start on how to work with frames.

- If you want to check out how tables work before starting with frames, **see** Chapter 9, "Creating Tables," p. 143

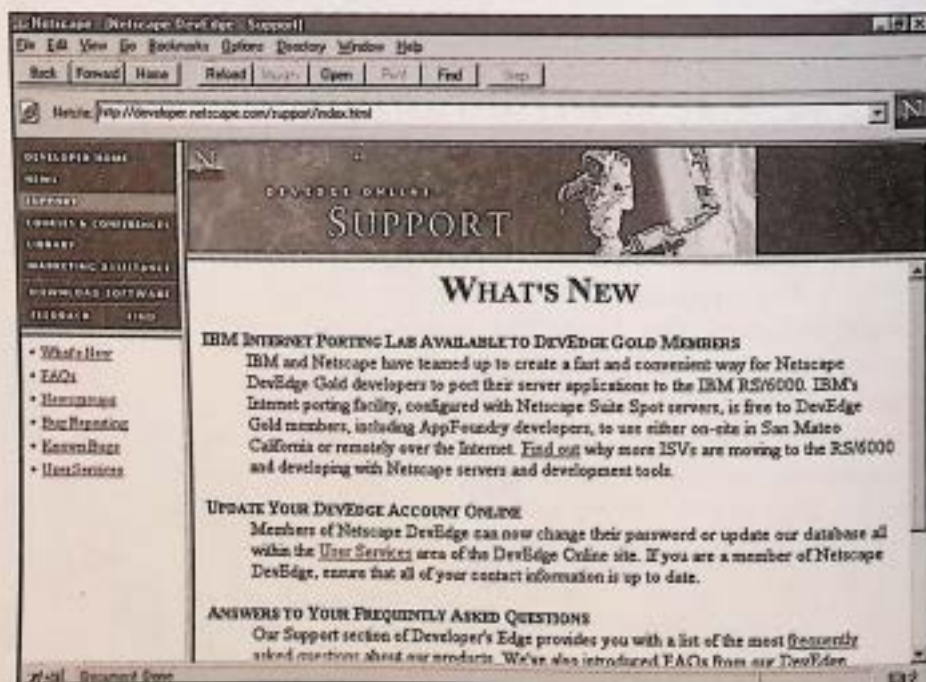
However, unlike tables, frames not only organize data, they organize your browser's display window, too. In fact, they break up the window into individual, independent panes or frames. Each frame holds its own HTML file as content, and the content of each frame can be scrolled or changed independently of the others. In a way, it's almost as though each frame becomes its own "mini-browser."

Perhaps the best way to get a feel for what you can do with frames is to look at a few real-world examples.

### Netscape's DevEdge Site

As you might expect, Netscape—the inventor of frames—has some excellent examples of frames on its Web sites. Figure 10.1 is taken from its DevEdge developer's site and shows a window that is broken into four separate frames.

**FIG. 10.1**  
Netscape's DevEdge site at <http://developer.netscape.com> showcases some excellent examples of using frames to separate information from navigation.



The frames on this page show how Netscape has split information display into the two frames on the right, while reserving navigation functions for the two frames on the left.

The top-right frame—which occupies about 80 percent of the width of the screen, but only about 20 percent of its height—holds a title graphic, which serves as a “landmark” to help you remember where you are. This is an important function, as HTML documents created using frames can get very complex very quickly. “Road signs” like this header graphic can help you get your bearings.

The top-left frame—which takes up about 20 percent of the horizontal real estate, and approximately 30 percent of the screen height—contains a top-level navigation menu, which stays in place wherever you go on the DevEdge site. Making a selection from this menu moves you to a new information category such as the support area or the library. This graphic menu also serves as a placeholder, since it shows the currently selected area as highlighted.

The bottom-left frame—about 20 percent of the screen width and 70 percent of its height—is a list of text-based hyperlinks, which makes up the information menu for the currently selected category. A new text menu is displayed in this frame whenever the user selects a new category from the graphic category menu in the frame above it.

**TIP**

Note how Netscape has saved itself a great deal of time and development work by making only the category-level menus graphic, while using much easier-to-create, text-only lists of links for the more numerous subcategory menus.

Finally, the bottom-right frame—which occupies the majority of the screen area, about 80 percent of its width and 70 percent of its height—contains all of the actual information displayed on this site. The information in this window is selected from the category-specific text link menu in the frame to its left.

This site can definitely serve as a template for good frames-based HTML document design for any information that is hierarchically organized by category.

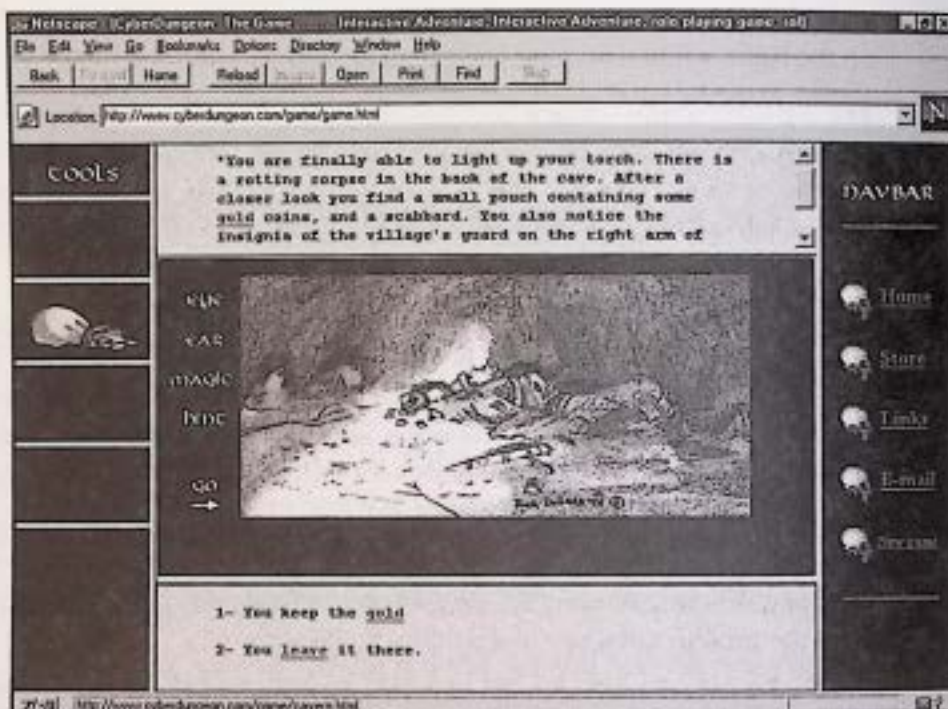
## The CyberDungeon

But frames aren't just for business documents. Take a look at Figure 10.2, which depicts the online CyberDungeon adventure game. I doubt that you will ever find 10 frames used as gracefully as they are on this site. (Usually it's bad practice to use more than 4–6 frames at a time.) This artfully done Web site anticipates the recent mantra of both Microsoft and Netscape, who are now encouraging developers to use HTML to create graphical user interfaces (or GUIs) for application programming.

The CyberDungeon site uses a set of frames down the left side of the screen to hold graphical icons of objects you (the resident adventurer) pick up in your explorations. The top frame of the center set of three displays the text description of your current location, while the larger frame below it shows a picture of the same scene. The bottom frame gives you choices to make along the way.

**FIG. 10.2**

The CyberDungeon game lives at <http://www.cyberdungeon.com> on the Web. It uses a set of 10 frames to create a familiar and friendly adventure game interface.



Finally, the tall right frame keeps the navigational menu for the CyberDungeon site handy as you play.

This site provides a wonderful example of how a well-designed HTML document using frames can replicate applications that previously had to be written in high-level languages like C or C++.

## The Mondrian Machine

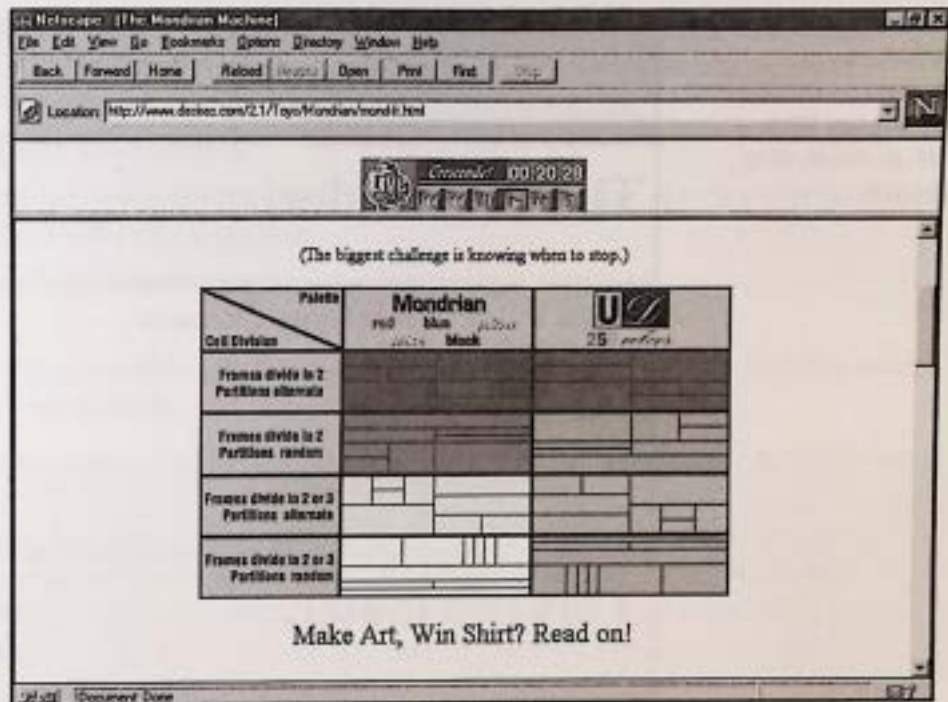
A final example is a wonderful lampoon of the too-ambitious use of frames. The Mondrian Machine site takes the overuse of frames—normally an ugly and heinous practice—and turns it into an art form (see Figure 10.3).

Clicking a selection in the table shown in Figure 10.3 brings up one of several different Mondrian Machines. Each, in its own unique way, creates an HTML document composed of a wild collection of frames which rapidly take over the entire screen. Each frame has no content except a background color, so the end effect is that of a painting by Mondrian, who became famous for dividing his canvases into so many colored rectangles. The effect, though humorous and somehow compelling, shows just how much trouble you can get into if you overdo the use of frames on your own pages.

That warning having been sounded, you'll now see how frames are created.

**FIG. 10.3**

The Mondrian Machine (<http://www.desires.com/2.1/Toys/Mondrian>) serves as both a fun Web toy and a graphic illustration of what can happen if you go too crazy with frames!



## Frame Creation Basics

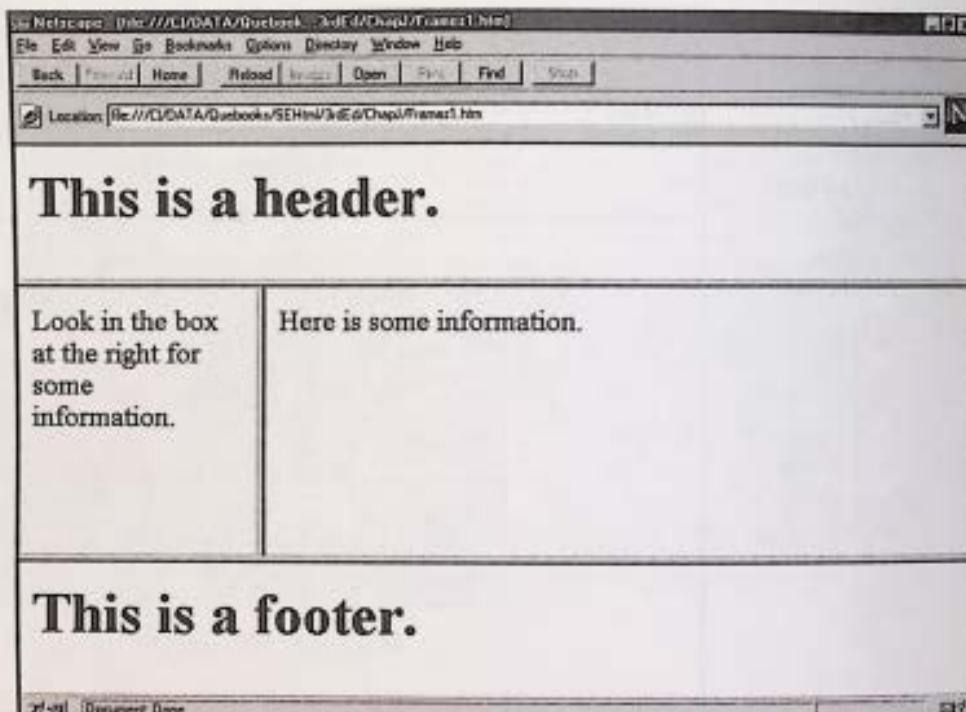
Diving in head first, take a look at an entire block of HTML that creates a frame document of medium complexity:

```
<HTML>
<HEAD>
</HEAD>
<FRAMESET ROWS="25%,50%,25%">
  <FRAME SRC="header.htm">
  <FRAMESET COLS="25%,75%">
    <FRAME SRC="label.htm">
    <FRAME SRC="info.htm">
  </FRAMESET>
  <FRAME SRC="footer.htm">
</FRAMESET>
<NOFRAMES>
Your browser cannot display frames.
</NOFRAMES>
</HTML>
```

This example, `Frames1.htm`, produces the frames page shown in Figure 10.4. As you can see, this HTML code produces four frames. The top frame spans the page and includes a header. There are two central frames, one for a label on the left, which takes up 25 percent of the screen width, and one for information on the right, which takes up the remaining space. Another frame fills the entire width of the bottom of the screen and contains a footer.

**FIG. 10.4**

This is the frame document produced by the preceding HTML code, as displayed by Netscape.



Though you won't get into the details for a couple more pages, it's important to note that this document calls four other HTML documents—header.htm, label.htm, info.htm, and footer.htm—containing the actual information that is displayed in each of the individual frames.

## The **FRAMESET** Container

Frames are contained in a structure called a **FRAMESET**, which takes the place of the **BODY** container on a frames-formatted page. An HTML document which contains a **FRAMESET** definition has no **BODY** section in its HTML code, and a page with a **BODY** section cannot use the **FRAMESET** tag.

### CAUTION

If you define a **BODY** section for a page that you compose with **FRAMESET** and **FRAME** commands, the frame structure will be completely ignored by browser programs, and none of the content contained in the frames will be displayed. Instead, you will only see the content contained in the **BODY** section.

Because there is no **BODY** container, **FRAMESET** pages can't have background images and background colors associated with them. (Remember, these are defined by the **BACKGROUND** and **BGCOLOR** attributes of the **BODY** tag, respectively.) However, the HTML files which contain the content for the individual frames can use background colors and images, since they do use the **BODY** tag.

Make sure you don't accidentally use **BODY** and **FRAMESET** within the same HTML document.

The `<FRAMESET></FRAMESET>` tag pair surrounds each block of frame definitions. Within the FRAMESET container you can only have FRAME tags or nested FRAMESET containers.

## Defining Rows and Columns

The FRAMESET tag has two major attributes: ROWS and COLS (columns). Here's a fully decked-out (but empty) generic FRAMESET container:

```
<FRAMESET ROWS="value_list" COLS="value_list">
</FRAMESET>
```

You can define any reasonable number of ROWS or COLS, or both, but you have to define something for at least one of them.

### CAUTION

If you don't define more than one row or column, browser programs will ignore your frames completely. Your screen will be left totally blank. In other words, you can't have a FRAMESET of just one row and one column, which would just be a single window, anyway. If you've defined at least two of either ROWS or COLS, however, you can safely omit the other attribute, and a value of 100 percent will be assumed for it.

The "value\_list" in your generic FRAMESET line is a comma-separated list of values which can be expressed as pixels, percentages, or relative scale values. The number of rows or columns is set by the number of values in their respective value lists. For example, the following:

```
<FRAMESET ROWS="100,240,140">
```

defines a frame set with three rows. These values are in absolute number of pixels. In other words, the first row is 100 pixels high, the second 240 pixels high, and the last 140 pixels high.

Setting row and column height by absolute number of pixels is bad practice, however. It doesn't allow for the fact that browsers run on all kinds of systems on all sizes of screens. While you might want to define absolute pixel values for a few limited uses—such as displaying a small image of known dimensions—it is usually better practice to define your rows and columns using percentages or relative values like the following:

```
<FRAMESET ROWS="25%,50%,25%">
```

This example would create three frames arranged as rows, the top row taking up 25 percent of the available screen height, the middle row 50 percent, and the bottom row 25 percent.

### TIP

Don't worry about having to do the math just right—if the percentages you give for the ROWS or COLS attributes don't add up to 100 percent, they will be scaled up or down proportionally to equal 100 percent.

Proportional values look like the following:

```
<FRAMESET COLS="*, 2*, 3*">
```

The asterisk is used to define a proportional division of space. Each asterisk represents one piece of the overall pie. You get the denominator of the fraction by adding up all the asterisk values (if there is no number specified, 1 is assumed). In this example, with an overall pie that has six slices, the first column would get 1/6 of the total width of the window, the second column would get 2/6 (or 1/3), and the final column would get 3/6 (or 1/2).

Remember that bare numeric values assign an absolute number of pixels to a row or column, values with a percent sign assign a percentage of the total width (for COLS) or height (for ROWS) of the display window, and values with an asterisk assign a proportional amount of the remaining space.

Here's an example using all three in a single definition:

```
<FRAMESET COLS="100, 25%, *, 2*">
```

This example assigns the first column an absolute width of 100 pixels. The second column gets 25 percent of the width of the entire display window, whatever that is. The third column gets 1/3 of what's left, and the final column gets the other 2/3.

So what are the space-allocation priorities? Absolute pixel values are always assigned space first, in order from left to right. These are followed by percentage values of the total space. Finally, proportional values are divided up based on what space is left.

### CAUTION

Remember, if you do use absolute pixel values in a COLS or ROWS definition, keep them small so you are sure they'll fit in any browser window, and balance them with at least one percentage or relative definition to fill the remainder of the space gracefully.

If you use a FRAMESET with both COLS and ROWS attributes, it will create a grid of frames. Here's an example:

```
<FRAMESET ROWS="*, 2*, *" COLS="2*, *">
```

This line of HTML creates a frame grid with three rows and two columns. The first and last rows each take up 1/4 of the screen height, and the middle row takes up half. The first column is 2/3 as wide as the screen, and the second is 1/3 the width.

<FRAMESET></FRAMESET> sections can be nested inside one another, as shown in your initial example. But don't get ahead of yourself. You need to look at the FRAME tag first.

## The FRAME Tag

The FRAME tag defines a single frame. It must sit inside a FRAMESET container. For example,

```
<FRAMESET ROWS="*, 2*">  
<FRAME>
```

```
<FRAME>  
</FRAMESET>
```

Note that the FRAME tag is not a container so, unlike FRAMESET, it has no matching end tag. An entire FRAME definition takes place within a single line of HTML code.

You should have as many FRAME tags as there are spaces defined for them in the FRAMESET definition. In this example, the FRAMESET established two rows, so you needed two FRAME tags. However, this example is very, very boring, since neither of your frames has anything in it! (Frames like these are displayed as blank space.)

The FRAME tag has six associated attributes: SRC, NAME, MARGINWIDTH, MARGINHEIGHT, SCROLLING, and NORESIZE. Here's a complete generic FRAME:

```
<FRAME SRC="url" NAME="window_name" SCROLLING=YES|NO|AUTO MARGINWIDTH="value"  
MARGINHEIGHT="value" NORESIZE>
```

Fortunately, frames hardly ever actually use all of these options.

## Going to the Source

The most important FRAME attribute is SRC (source). You can (and quite often do) have a complete FRAME definition using nothing but the SRC attribute. For example,

```
<FRAME SRC="url">
```

SRC defines the URL of the content of your frame. This is usually an HTML file on the same system (paths are relative to the page containing the FRAMESET), so it usually looks something like the following:

```
<FRAME SRC="sample.htm">
```

Note that any HTML file called by the SRC attribute in a FRAME definition must be a complete HTML document, not a fragment. This means it must have HTML, HEAD, and BODY containers, and so on. For example, the file called by the SRC attribute in this example, sample.htm, might look like the following:

```
<HTML>  
<HEAD>  
<TITLE>  
</TITLE>  
</HEAD>  
<BODY>  
This is some Sample Text.  
</BODY>  
</HTML>
```

Of course, SRC can point to any valid URL. If, for example, you wanted your frame to display a GIF image that was located somewhere in Timbuktu, your FRAME might look like the following:

```
<FRAME SRC="http://www.timbuktu.com/budda.gif">
```

If you specify an URL the browser can't find, space will be allocated for the frame, but it won't be displayed, and you will get a nasty error message from your browser. Note that the effect is

quite different than simply specifying a FRAME with no SRC at all. <FRAME> will always be created, but left blank; <FRAME SRC="unknown URL"> will not be created at all—the space will be allocated and left completely empty. The former will fill with background color, while the latter will remain the browser's border color.

### CAUTION

Plain text, headers, graphics, and other elements cannot be used directly in a FRAME document. All of the content must come from the URL defined by the SRC attribute of the FRAME tags. If any other content appears on a FRAMESET page, it will be displayed and the entire set of frames will be ignored.

## Providing Alternate Content

"All of this is well and good," you say, "and I really, really want to use frames in my HTML documents. But I can't help feeling guilty about all those users who don't have frames-capable browsers. They won't be able to see my beautiful pages!" Don't worry. Here's where you can provide for them, too.

The <NOFRAMES></NOFRAMES> container is what saves you. By defining a NOFRAMES section and marking it up with normal HTML tags, you can provide alternate information for those without forms-capable browsers. This is how it works:

```
<NOFRAMES>
All your alternate HTML goes here.
</NOFRAMES>
```

You can safely think of this as an alternative to the BODY structure of a normal HTML page. Whatever you place between the <NOFRAMES> and </NOFRAMES> tags will appear on browsers without frames capability. Browsers with frames will throw away everything between these two tags.

**NOTE** If you want to include background graphics or images, you can add the BODY tag to your alternate content. For example,

```
<NOFRAMES>
<BODY BGCOLOR="red" BACKGROUND="bgimage.gif">
content...
</BODY>
</NOFRAMES>
```

As long as the BODY container is kept within the bounds of the NOFRAMES container, your document will work just fine. But there's no need to use the BODY tag within the NOFRAMES container unless you want to take advantage of its special attributes. ■

## A Few Simple Frame Examples

Frames are very flexible, which means they can get complicated quickly. Now that you understand the basics, take a look at a few frame examples so you can get your bearings.

### A Simple Page with Two Frames

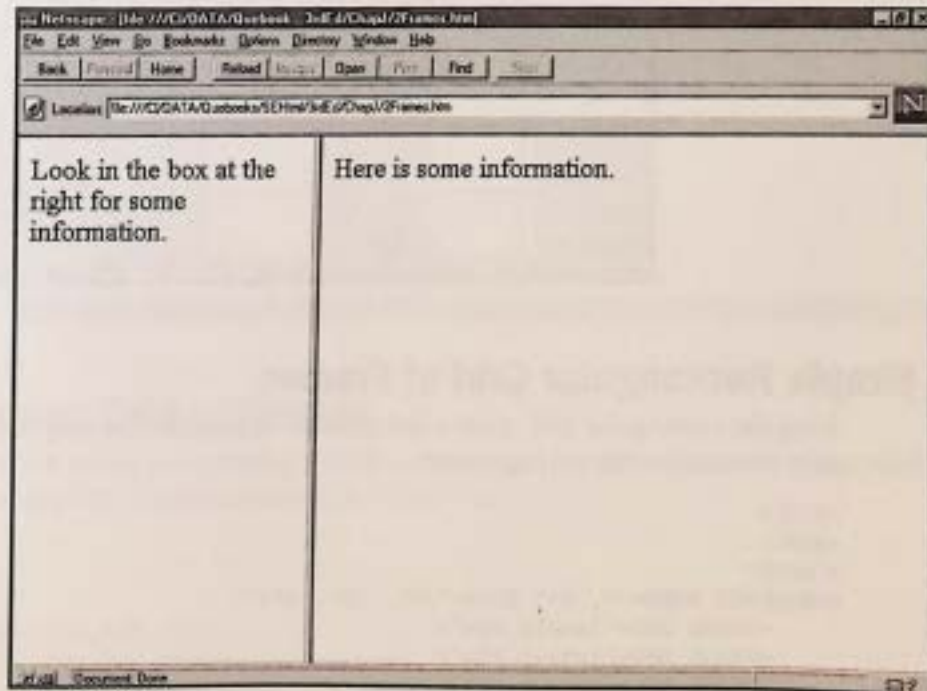
The simplest possible frame setup is one with two frames, like the following:

```
<HTML>
<HEAD>
</HEAD>
<FRAMESET COLS="*, 2*">
  <FRAME SRC="label.htm">
  <FRAME SRC="info.htm">
</FRAMESET>
</HTML>
```

This HTML code (2Frames.htm) defines a page with two frames, organized as two columns. The first column takes up 1/3 the width of the screen and contains the HTML document label.htm, and the second takes up the other 2/3 and contains the document info.htm. Figure 10.5 shows how Netscape Navigator displays this page.

**FIG. 10.5**

Netscape displays the simple two-column FRAMESET defined by the HTML code above.



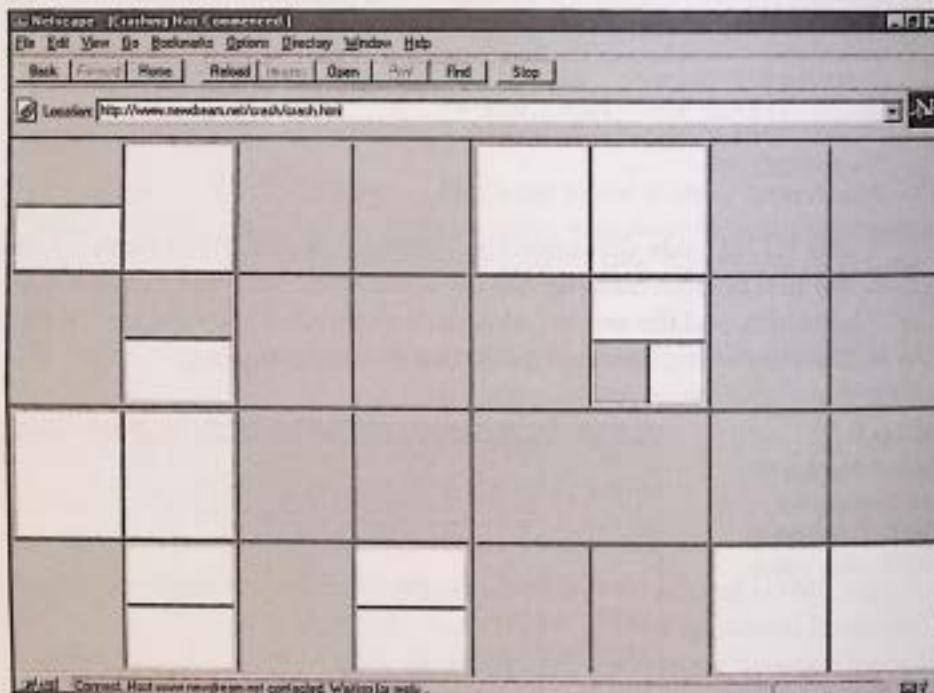
You could just as easily create 10 or more columns or use the same syntax, substituting the ROWS attribute to create two (or 10) rows. However, 10 columns or rows is way too many for any browser to handle gracefully. Your pages should probably never have more than three or four rows or columns.

**NOTE** If you want to display more information than three or four rows or columns, you should probably be using tables rather than frames. Remember, frames are most useful when you want to add an element of control in addition to formatting the display or if you need to update displayed data dynamically. Tables are best if all you want to do is format static data into rows and columns.

Too many frames can actually crash your browser. For a real-world example (if you don't mind your browser program crashing) check out <http://www.newdream.net/crash/> (see Figure 10.6).

**FIG. 10.6**

This Web site sets out to deliberately crash your browser by creating too many frames—and succeeds! Don't cause the same problem for your viewers.



## A Simple Rectangular Grid of Frames

A regular rectangular grid of rows and columns is just about as easy to implement as a rows-only or columns-only arrangement:

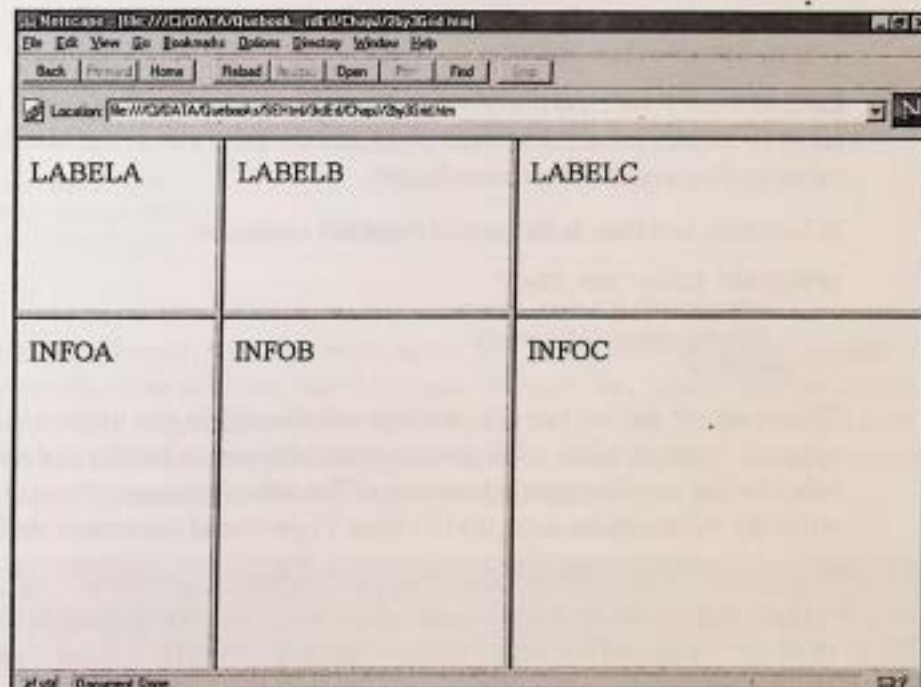
```
<HTML>
<HEAD>
</HEAD>
<FRAMESET ROWS="*, 2*" COLS="20%, 30%, 40%">
  <FRAME SRC="labela.htm">
  <FRAME SRC="labelb.htm">
  <FRAME SRC="labelc.htm">
  <FRAME SRC="infoa.htm">
  <FRAME SRC="infob.htm">
  <FRAME SRC="infoc.htm">
</FRAMESET>
</HTML>
```

This example (2by3Grid.htm) creates a grid with two rows and three columns (see Figure 10.7). Since you defined a set of six frames, you've provided six `FRAME` definitions. Note that they fill in by rows. That is, the first `FRAME` goes in the first defined column in the first row, the second frame follows across in the second column, and the third finishes out the last column in the first row. The last three frames then fill in the columns of the second row going across.

Also, note that the math didn't work out very well, since the percentage values in the `COLS` definition only add up to 90 percent. No problem, because the browser has adjusted all the columns proportionally to make up the difference.

**FIG. 10.7**

This two-by-three grid of frames was created by the preceding HTML example.



## Creating a Complex Grid of Frames

A bit tougher is the problem of creating a more complex grid of frames. For that, return to the example that opened this discussion (refer to Figure 10.4):

```
<HTML>
<HEAD>
</HEAD>
<FRAMESET ROWS="25%,50%,25%">
  <FRAME SRC="header.htm">
  <FRAMESET COLS="25%,75%">
    <FRAME SRC="label.htm">
    <FRAME SRC="info.htm">
  </FRAMESET>
</FRAMESET>
<FRAME SRC="footer.htm">
```

```

</FRAMESET>
<NOFRAMES>
Your browser cannot display frames.
</NOFRAMES>
</HTML>

```

This example (Frames1.htm) makes use of nested FRAMESET containers. The outside set creates three ROWS, with 25 percent, 50 percent, and 25 percent of the window height, respectively:

```
<FRAMESET ROWS="25%,50%,25%">
```

Within this definition, the first and last rows are simple frames:

```

<FRAME SRC="header.htm">
<FRAME SRC="footer.htm">

```

Each of these rows runs the entire width of the screen. The first row at the top of the screen takes up 25 percent of the screen height, and the third row at the bottom of the screen also takes up 25 percent of the screen height.

In between, however, is this nested FRAMESET container:

```

<FRAMESET COLS="25%,75%">
  <FRAME SRC="label.htm">
  <FRAME SRC="info.htm">
</FRAMESET>

```

This FRAMESET defines two columns that split the middle row of the screen. The row these two columns reside in takes up 50 percent of the total screen height, as defined in the middle row value for the outside FRAMESET container. The left column uses 25 percent of the screen width, while the right column occupies the other 75 percent of the screen width.

The frames for the columns are defined within the set of FRAMESET tags, which include the column definitions, while the FRAME definitions for the first and last rows are outside the nested FRAMESET command but within the exterior FRAMESET in their proper order.

This is not as confusing if you think of an entire nested FRAMESET block as a single FRAME tag. In this example, the outside FRAMESET block sets up a situation in which you have three rows. Each must be filled. In this case, they are filled by a FRAME, then a nested FRAMESET two columns wide, and then another FRAME.

By now you may be asking yourself, "I wonder if it is possible for a FRAME to use as its SRC a document that is, itself, a FRAMESET?" The answer is "Yes." In this case, you simply use the FRAME tag to point to an HTML document which is the FRAMESET you would have otherwise used in place of the FRAME.

Redefine the previous example (which used nested FRAMESETs) in terms of referenced FRAME documents instead. All you're doing is moving the nested FRAMESET to its own document. Here's the first (outside) file (Frames2.htm):

```

<HTML>
<HEAD>
</HEAD>
<FRAMESET ROWS="25%,50%,25%">
  <FRAME SRC="header.htm">
  <FRAME SRC="nested.htm">
  <FRAME SRC="footer.htm">
</FRAMESET>
<NOFRAMES>
Your browser cannot display frames.
</NOFRAMES>
</HTML>

```

And here's the second (inside) file ("Nested.htm").

```

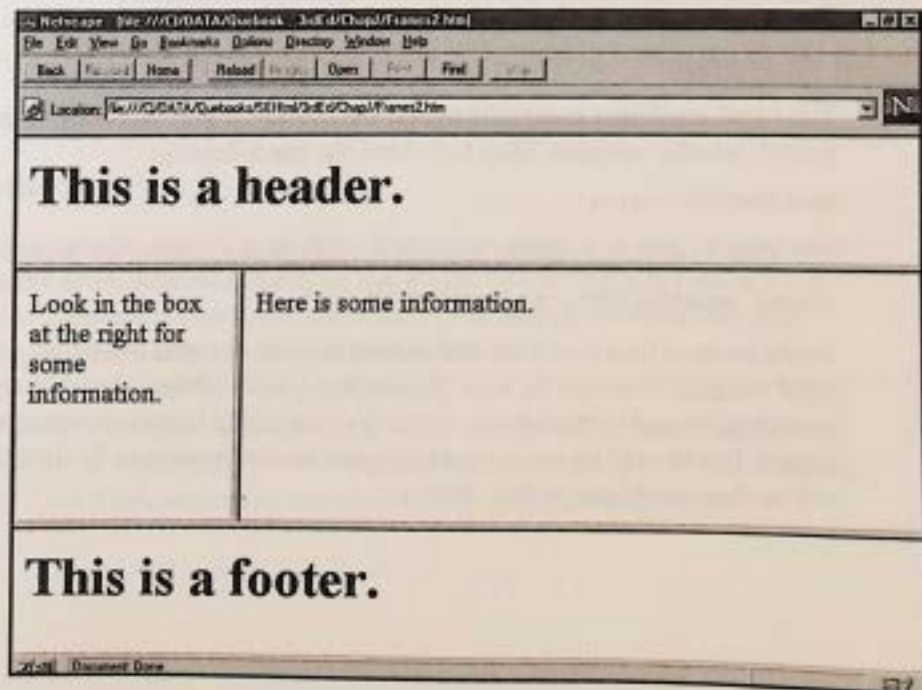
<HTML>
<HEAD>
</HEAD>
<FRAMESET COLS="25%,75%">
  <FRAME SRC="label.htm">
  <FRAME SRC="info.htm">
</FRAMESET>
</HTML>

```

In this case, the top and bottom rows behave as before. But the second row in the "outside" file is now just a simple FRAME definition like the others. However, the "inside" file that its SRC points to is frameset.htm, which you created with a FRAMESET all its own. When inserted into the original FRAMESET, it will behave just as if it appeared there verbatim. The resulting screen is identical to the original example (compare Figure 10.8 to Figure 10.4).

**FIG. 10.8**

FRAMESET containers can be nested or can call other documents containing their own FRAMESETS. The end result is the same.



**CAUTION**

Though it's possible to create nested FRAMESETs by using FRAME tags which call the same URL, it certainly isn't a good idea. This is called *infinite recursion*, which creates an infinite loop in a computer that consumes all memory and crashes the machine. Fortunately, frames-aware browsers check for this—if an SRC URL is the same as any of its ancestors it's ignored, just as if there were no SRC attribute at all.

By using nested FRAMESET containers in clever combinations, it is possible to create just about any grid of frames you can dream up. But remember that you're trying to create a friendly, useful interface, not show off how clever you can be with frames.

**NOTE** In Netscape Navigator 2.0, the toolbar's Back button didn't back you out of a frame, it backed you out of the whole FRAMESET to the previous page.

With versions 3.0 and above, choosing the Back button returns you to the previous state of the currently selected frame.

To navigate forward or backward within a frame, make sure you make the frame active first by clicking in it somewhere; then use the Forward or Back buttons or menu selections to navigate within that frame. ■

## Modifying a Frame's Look and Feel

Now that you understand how framesets are used to create various grids of frames, take a look at some of the attributes that modify how frames look and feel.

### Frame Margins

The FRAME attributes MARGINWIDTH and MARGINHEIGHT give you control over the width of the frame's interior margins. They both look like the following:

```
MARGINWIDTH="value"
```

The value is always a number and always represents an absolute value in pixels. For example,

```
<FRAME MARGINHEIGHT="5" MARGINWIDTH="7">
```

would create a frame with top and bottom interior margins five pixels wide, and left and right margins seven pixels wide. Remember, you're talking interior margins here, not borders. MARGINWIDTH and MARGINHEIGHT define a space within the frame within which content will not appear. Border widths are set automatically by the browser or by the BORDER attribute, which will be discussed later in this chapter.

## Frame Scrollbars

Your frames will automatically have scrollbars if the content you've specified for them is too big to fit the frame. Sometimes this ruins the aesthetics of your page, so you need a way to control them. That's what the `SCROLLING` attribute of the `FRAME` tag is for. Here's the format:

```
<FRAME SCROLLING="yes|no|auto">
```

There are three valid values for `SCROLLING`: `YES`, `NO`, and `AUTO`. `AUTO` is assumed if there is no `SCROLLING` attribute in your `FRAME` definition. `YES` forces the appearance of a scrollbar. `NO` keeps them away at all costs. For example, this `FRAME` definition turns on scrollbars:

```
<FRAME SCROLLING=YES>
```

## Frame Resizing

Frames are normally resizable by the user. But if you let the user drag your frames around, it can quickly muck up the look and feel of your beautifully designed frames. You will therefore almost always want to use the `NORESIZE` attribute of the `FRAME` tag to keep users from resizing your frames. Here's how:

```
<FRAME NORESIZE>
```

That's it. No values. Of course, when you set `NORESIZE` for one frame, none of the adjacent frames can be resized, either. Depending on your layout, using `NORESIZE` in a single frame will often be enough to keep users from resizing all the frames on the screen.

When you move over a resizable frame border with the mouse cursor, it will change to a double-arrow (see Figure 10.9), indicating that the frame can be resized. If you don't get the double-arrow, it means that resizing has been turned off with the `NORESIZE` attribute. To resize a resizable frame, grab the frame border by clicking and dragging it with your mouse to a new position.



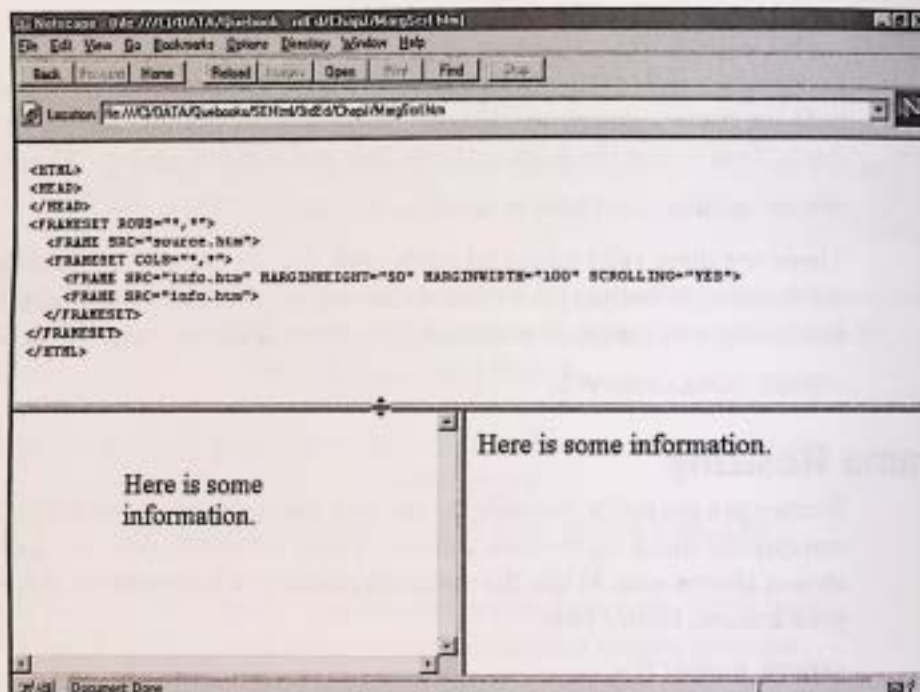
### TROUBLESHOOTING

**I've created a frame using the `NORESIZE` attribute. What do I do about users who are using small screens on which the entire contents of the frame may not fit?** Your best bet is to make sure the frame will hold all of its content at lower screen resolutions. That is, redesign the frame. Otherwise, consider reenabling, resizing, or adding scrollbars to the frame.

Figure 10.9 shows an example of a frames page where the lower-left frame has had its `MARGINHEIGHT` set to 50, `MARGINWIDTH` set to 100, and `SCROLLING` set to `YES`. The `NORESIZE` attribute has not been used, so you can see what the resizing cursor looks like.

**FIG. 10.9**

The HTML source for this FRAMESET is shown in the upper frame.



## Frame Borders

You use the `BORDER`, `FRAMEBORDER`, and `BORDERCOLOR` attributes to set the look and feel of the borders for your frameset. The `BORDER` attribute is used only with the `FRAMESET` tag, and sets the width of all the borders in the frameset. It is assigned a value in pixels, like the following:

```
<FRAMESET BORDER="5">
```

This example would set the width of the frame borders to five pixels. `BORDER` can be assigned a value of "0", in which case all the frames in your frameset will be borderless.

### TIP

The default value of `BORDER` is 5.

The `FRAMEBORDER` attribute can be used with either the `FRAMESET` or `FRAME` tag. It has two legitimate values, `YES` and `NO`. If `FRAMEBORDER=YES`, then frame borders are drawn with a 3-D look. If `FRAMEBORDER=NO`, frame borders are invisible, which really means that they are drawn in the default background color of the browser.

The default value of `FRAMEBORDER` is `YES`, which means that browser programs generally display 3-D frame borders.

The border for a frame will be invisible (not 3-D) only if `FRAMEBORDER="NO"` is set for all surrounding frames.

**TIP**

To create a page with entirely borderless frames, set `FRAMEBORDER="NO"` and `BORDER="0"` in the top `FRAMESET` definition.

The `BORDERCOLOR` attribute can be used with the `FRAMESET` tag or with the `FRAME` tag. `BORDERCOLOR` can be assigned a named color value or a hexadecimal RGB color value. Here's an example:

```
<FRAMESET BORDERCOLOR="red" ROWS="*,**">
  <FRAME SRC="info.htm" BORDERCOLOR="#FF00FF">
  <FRAME SRC="info.htm">
</FRAMESET>
```

In this example, the outer `FRAMESET` tag sets the `BORDERCOLOR` to "red", one of the named colors for most browsers. But the following `FRAME` tag sets `BORDERCOLOR` to the hexadecimal value `#FF00FF` (which happens to be purple). The lowest level definition takes precedence. Though the `FRAMESET BORDERCOLOR` is defined as "red", the border color of the first frame will instead be set to the hexadecimal RGB value `#FF00FF`. The adjacent frame, which has no `BORDERCOLOR` definition, will have a border of `#FF00FF` on the edge it shares with the other frame, but a color of "red" on borders it does not share with that frame.

**NOTE** RGB hexadecimal color values are precise but obscure. (Exactly what color is `#FA10D7`?) Named colors are easier to comprehend, but not all browsers support the same color names. In general, if you stick with the 16 common color names listed here, you'll be safe: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow. ■

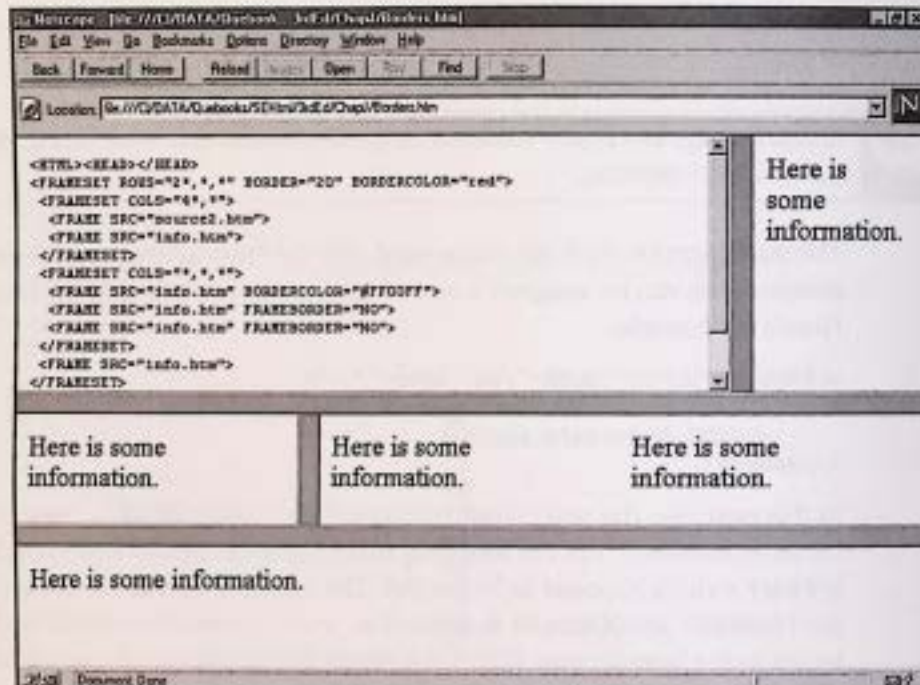
**TIP**

If two adjacent frames of the same priority attempt to define their own `BORDERCOLOR`, neither will take effect. They will revert to the `BORDERCOLOR` defined at the next higher `FRAMESET` level.

Figure 10.10 shows an example of using the `BORDER`, `FRAMEBORDER`, and `BORDERCOLOR` attributes to control the look and feel of your frame borders. Note that the only frame to maintain the `BORDERCOLOR` defined in the outside `FRAMESET` definition is the one in the upper right, the only frame which doesn't share a border with the left-most center frame, which redefines the `BORDERCOLOR`. Actually, the right-most center frame would have also had a red border on the left, but it has had its left border turned off by the `FRAMEBORDER="NO"` attribute it shares with the central frame. Note this complex interplay of attributes carefully. If you use them often, their interrelationships are sure to throw you for a loop more often than they make sense.

**FIG. 10.10**

The HTML source for this border-manipulating example is shown in the top frame.

**CAUTION**

Microsoft Internet Explorer supports one more attribute for `FRAMESET`, called `FRAMESPACING`. It is intended to define the space between frames and can assume pixel values. However, since Netscape Navigator and other browser programs don't support the `FRAMESPACING` attribute, it is strongly advised that you not use it unless you are creating content for an audience that uses Internet Explorer exclusively and you have a good reason for having to specify the spacing between frames.

## Targeted Hyperlinks

Though you've examined in depth how to create and nest framesets and how to control the look and feel of frames, you have yet to understand how to use frames to control navigation, which is their major application. To use frames to their full advantage, you need to know how to name and target frames.

### Naming and Targeting Frames

The `NAME` attribute assigns a name to a frame that can be used to link to the frame, usually from other frames in the same display. This example,

```
<FRAME SRC="info.htm" NAME="Joe">
```

creates a frame named "Joe", which can be referenced via a hyperlink like the following:

```
<A HREF="moreinfo.htm" TARGET="Joe">Click Here to Jump to Joe</A>
```

Note the `TARGET` attribute in the hypertext link that references the name of your frame. When selected, this hyperlink will replace the content of the named frame "Joe"—which was given the content file `info.htm` when it was created—with the content in the file `moreinfo.htm`. Note that, while a hyperlink without a named `TARGET` replaces the content in its own window or frame with the content named in the `HREF` attribute, a hyperlink with a named `TARGET` instead replaces the content in the targeted frame or window with the content named in the `HREF` attribute. This is the only trick you need to know for creating killer frames-based navigational systems. Of course, there are some fine points.

## Legitimate Names

If you don't create an explicit name for a frame, it will simply have no name, and you won't be able to use links in one frame to open documents or images in another. You'll want to name all frames whose content will be changed by clicking a link in a different frame.

All frame names must begin with an alphanumeric character. Don't use an underscore as the first character in a frame name. Other than that, you're pretty much on your own.

However, there are four reserved *implicit* names built into HTML, and all of them *do* start with an underscore. These are listed in Table 10.1 below. All other names starting with an underscore will be ignored.

**Table 10.1** Reserved Implicit Frame Names

| Name                 | Purpose                                                                                                                                                                                           |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>_blank</code>  | Load content directed to this name into a new, unnamed window. This name is used to completely wipe out the current frameset and start with a new, blank window.                                  |
| <code>_self</code>   | Load content directed to this name into the calling frame.                                                                                                                                        |
| <code>_parent</code> | Load content directed to this link to the calling frame's parent frameset window. If it has no parent frameset, this is the same as using the name <code>_self</code> .                           |
| <code>_top</code>    | Load content directed to this link to the top level frameset related to the calling frame. If the calling frame is already the top level, this is the same as using the name <code>_self</code> . |

Here are a few examples to help clarify how these reserved names work.

If a frame contains the following link:

```
<A HREF="stuff.htm" TARGET="_blank">
```

then clicking the link will launch a new, unnamed browser display window which will contain the content defined in `stuff.htm`. This can be a simple HTML document or an entirely new `FRAMESET` definition. Whichever, this call wipes the slate clean and starts completely over.

If a frame contains the following link:

```
<A HREF="stuff.htm" TARGET="_self">
```

then clicking the link will simply cause the frame which contains the link to clear, and its content will be replaced with whatever is in stuff.htm.

If a frame contains the following link:

```
<A HREF="stuff.htm" TARGET="_parent">
```

then the frameset which contains the frame that contains this link will be replaced by stuff.htm.

Finally, if a frame contains the following link:

```
<A HREF="stuff.htm" TARGET="_top">
```

then clicking the link replaces the entire browser window with the contents of stuff.htm.

**NOTE** Hyperlinks using the <A> tag aren't the only tags that can make use of the TARGET attribute. The AREA, FORM, and BASE tags also use the TARGET attribute and can be used effectively to extend and expand the utility of named frames.

Remember, too, that windows can also be named by using the TARGET attribute within the <A> tag; using named windows in conjunction with named frames adds a whole new dimension to HTML document navigation. ■

## Updating More Than One Frame at a Time

You've seen that you can click a link in one frame to change the content in another by naming the target frame by using the NAME attribute of the FRAME tag when creating the target frame, and then using the TARGET attribute of the <A> hyperlink tag when defining the link, as in this example:

```
<FRAME SRC="info.htm" NAME="Joe">
<A HREF="moreinfo.htm" TARGET="Joe">Click Here to Jump to Joe</A>
```

The first line of the preceding HTML is used in the frame definition document, and the second line is used in the document which links to the first.

You've also seen that you can use special implicit names to target some frames and framesets, depending on their relationship to the frame which contains the calling link. But what if you want to update more than one frame by clicking a single link? This is possible if you set up your document correctly. The key is to update a frameset, not a single frame. To do this, you need to create a subframeset that is contained in its own file, as was done earlier in the example shown back in Figure 10.8. If you recall, you began with an HTML document that included one frameset nested inside another:

```

<HTML>
<HEAD>
</HEAD>
<FRAMESET ROWS="25%,50%,25%">
  <FRAME SRC="header.htm">
  <FRAMESET COLS="25%,75%">
    <FRAME SRC="label.htm">
    <FRAME SRC="info.htm">
  </FRAMESET>
  <FRAME SRC="footer.htm">
</FRAMESET>
</HTML>

```

Then you took the nested frameset out and put it into its own file.

Here's the original file, with the nested frameset replaced by the `FRAME` definition pointing to the second file, nested.htm. Note that you'll name this frame this time using the `NAME` attribute, calling it "Inner":

```

<HTML>
<HEAD>
</HEAD>
<FRAMESET ROWS="25%,50%,25%">
  <FRAME SRC="header.htm">
  <FRAME SRC="nested.htm" NAME="Inner">
  <FRAME SRC="TestLink.htm">
</FRAMESET>
</HTML>

```

Also note the last `FRAME SRC` file has been renamed `TestLink.htm`—this file will contain the targeted link you want to test. Call this modified file `NameTest.htm`. Now here's the file (nested.htm) that the frame named "Inner" calls:

```

<HTML>
<HEAD>
</HEAD>
<FRAMESET COLS="25%,75%">
  <FRAME SRC="label.htm">
  <FRAME SRC="info.htm">
</FRAMESET>
</HTML>

```

The file which occupies the bottom frame contains the file `TestLink.htm`, which is listed here:

```

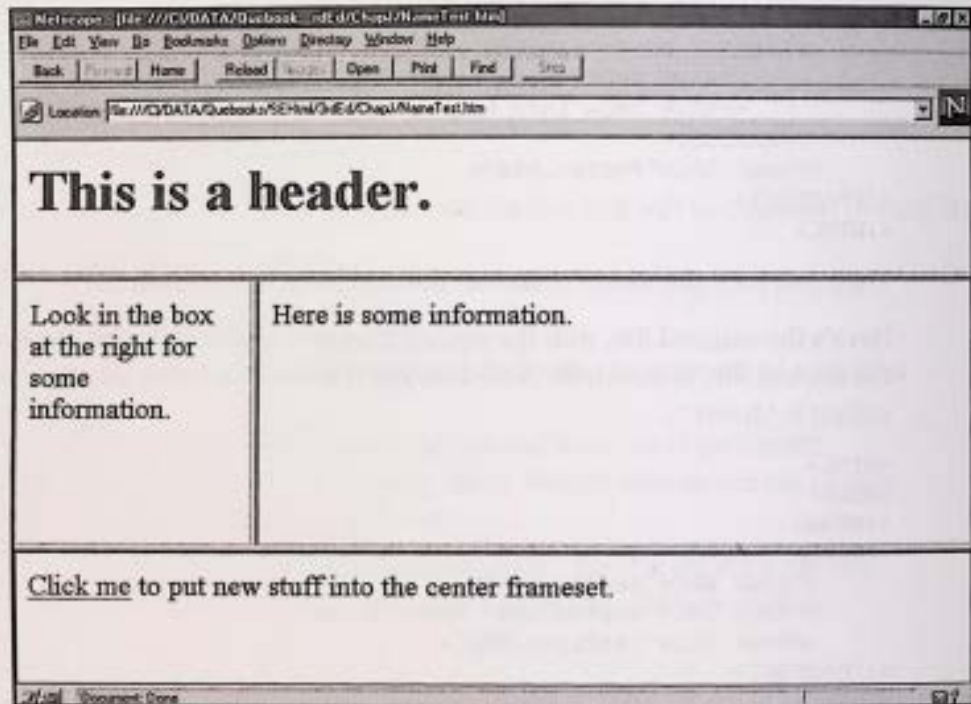
<HTML>
<HEAD>
<TITLE>
</TITLE>
</HEAD>
<BODY>
<A HREF="NewStuff.htm" TARGET="Inner">Click me</A>
to put new stuff into the upper center frameset.
</BODY>
</HTML>

```

The frameset created by these two files (and their associated content files) is shown in Figure 10.11.

**FIG. 10.11**

Here's your test page, all set to change two frames with one mouse click.



When the link `Click me` in the bottom window is clicked, it will replace the two frames in the center frameset (which was defined in the file `nested.htm`) with new information from the document `NewStuff.htm`. Say that the following is the content of `NewStuff.htm`:

```
<HTML>
<HEAD>
<TITLE>
</TITLE>
</HEAD>
<BODY>
Here is some NEW STUFF!
</BODY>
</HTML>
```

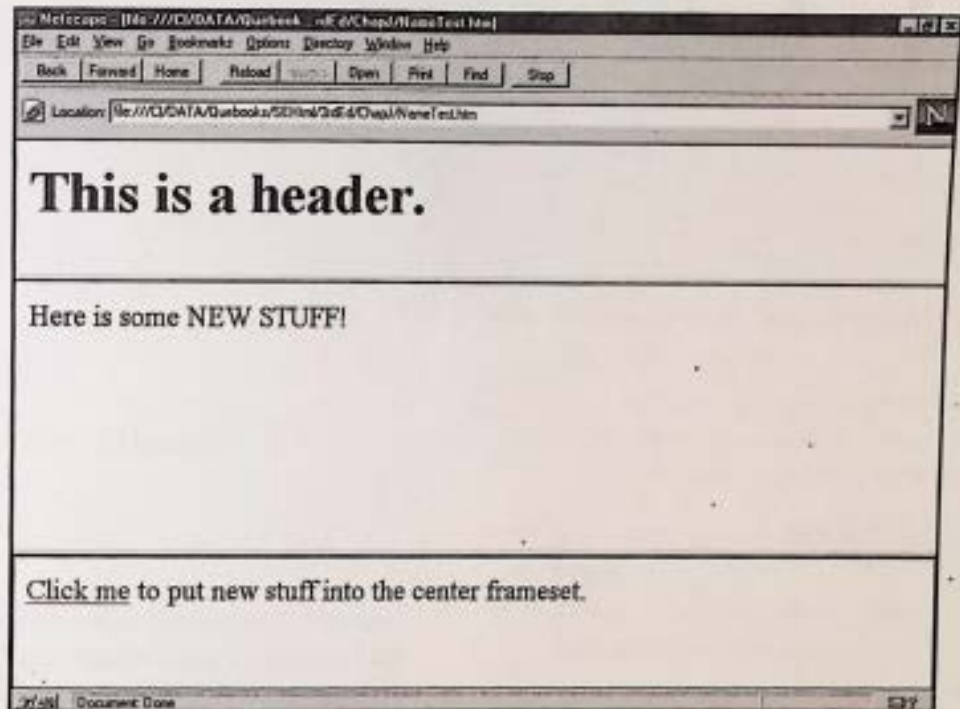
When you click the link in the bottom window, you now get the result shown in Figure 10.12.

With one click, you've replaced the two frames of the central frameset with a single frame containing new content. You could just as easily have replaced the two frames with two new frames—the same size or different sizes—or with three frames or with a whole new frameset. All you would have to do is define the new frameset in your new content file, `NewStuff.htm`.

If you're careful and think ahead when defining your framesets, you can easily create hyperlinks that can update almost any combination of frames with a single mouse click.

**FIG. 10.12**

Clicking the hyperlink in the bottom window has replaced the two-frame central frameset with the single frame of content from a different file.



## TROUBLESHOOTING

**I have two frames that don't have the same parent frameset. Can I update both of them with one hyperlink?** Yes. You will have to write a JavaScript or VBScript application to handle them. In fact, this is how Netscape Communications has done multiframe updates from single hyperlinks on its own Web site.

See Chapter 27, "JavaScript/JScript," for information on how to get started writing your own JavaScript applications. Also, see Chapter 29, "VBScript," to learn more about writing VBScripts.

# Special Edition USING HTML<sup>3.2</sup> Third Edition

*The most up-to-date, comprehensive reference on the HTML standard!*

Que's *Special Edition Using HTML 3.2, Third Edition* is the complete reference for all aspects of HTML 3.2 Web page creation. The expert authors give you an insider's view of new Web page technology, such as Java™, JavaScript™, and ActiveX™. You'll also find up-to-date coverage of HTML 3.2 features supported by Netscape™ and Microsoft Internet Explorer™.

- Learn all aspects of the tags supported in HTML 3.2
- Incorporate browser-specific content, using Netscape Navigator or Internet Explorer extended HTML tags
- Make your site browser independent so that any browser will display it correctly
- Discover the latest HTML technologies like Style Sheets and Dynamic HTML
- Learn how scripting can increase the interactivity of, or automate, your Web sites
- Make your site compatible with the new WebTV™ devices

## User Level

New Casual Accomplished Expert

Category: Internet-Web Publishing

Covers: Through Level 3.2 for Windows, Macintosh, UNIX, and Windows NT

**que**®

Visit us on the Internet at: <http://www.quecorp.com>



## CD-ROM includes:

- Internet Explorer 3.0
- Special Web hosting offers from WinNet and NetCom
- HTML Editors and Utilities: BBEdit Lite, HotDog Pro, HomeSite, Msp Thru and many others...
- Web Plug-ins: Audio, Graphics, Remote Access, Productivity, Video, VRML
- Web Services: Apollo, Microsoft Internet Information Server, WebQuest
- Java Books: Effects, Egon, Flash, Ewgie, Swami, and more
- Web Utilities, Helper Applications, and other useful tools: Create Search Engine, HyperMail, SMTPAN, ActiveX Control Pad, ListServ, and VB5 Control Creation Edition

- Also, even more books, 4 QueBooks in HTML format:



- ForeFront Web Printer (4-pack version): Use ForeFront Web Printer 4-pack to print samples from these books



If you are not completely satisfied, you can return this product for a full refund. Call 1-800-858-7674

\$49.99 USA / \$70.95 CAN / £46.99 Net UK (inc of VAT)



0 29236 10978 3

ISBN 0-7897-1097-8



9 780789 710970

9 4999

