

FOUNDATIONS OF WEB DESIGN

INTRODUCTION TO HTML & CSS



THOMAS MICHAUD



FOUNDATIONS OF WEB DESIGN: INTRODUCTION TO HTML AND CSS

Thomas Michaud

FOUNDATIONS OF WEB DESIGN: INTRODUCTION TO HTML AND CSS

Thomas Michaud

New Riders

www.newriders.com

To report errors, please send a note to errata@peachpit.com

New Riders is an imprint of Peachpit, a division of Pearson Education.

Copyright © 2014 by Thomas Michaud

Project Editor: Michael J. Nolan

Production Editor: Katerina Malone

Development Editor: Margaret Anderson/Stellarvisions

Technical Editor: Chris Mills

Copyeditor: Jennifer Needham

Proofreader: Patricia Pane

Indexer: Rebecca Plunkett

Cover & Interior Designer: Jonathon Wolfer

ISBN 13: 978-0-321-91893-2

ISBN 10: 0-321-91893-2

9 8 7 6 5 4 3 2 1

Printed and bound in the United States of America

NOTICE OF RIGHTS

All rights reserved. No part of this book may be reproduced or transmitted in any form by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. For information on getting permission for reprints and excerpts, contact permissions@peachpit.com.

NOTICE OF LIABILITY

The information in this book is distributed on an “As Is” basis without warranty. While every precaution has been taken in the preparation of the book, neither the author nor Peachpit shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the instructions contained in this book or by the computer software and hardware products described in it.

TRADEMARKS

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Peachpit was aware of a trademark claim, the designations appear as requested by the owner of the trademark. All other product names and services identified throughout this book are used in editorial fashion only and for the benefit of such companies with no intention of infringement of the trademark. No such use, or the use of any trade name, is intended to convey endorsement or other affiliation with this book.

To my grandmother, Nonnie; you pushed me to never stop learning.

And to my wife, Erica; you always sustain me.

*I hear and I forget. I see
and I remember. I do and
I understand.*

~ Confucius

ACKNOWLEDGMENTS

I don't know if I can thank enough people (or thank them enough) for all the support they've provided during the writing of this book. If I leave someone out, just let me know on Twitter @coldcoffee!

First off, I give thanks to my Creator, who has sustained me throughout the writing of this book—all things are truly possible.

Thank you to my amazing, beautiful, and patient wife, Erica, and wonderful kids, Dylan and Natalie, who have been a tremendous support and were extremely accommodating while I worked long hours writing this book. I'm going to be ever so grateful to be a part of the family again. I also give thanks to my mother-in-law, Mary, who helped watch the kids when my wife needed a break—you are truly a gift!

I'd like to thank my team at Peachpit Press: Michael Nolan, who took a chance on an unknown teacher; and Margaret Anderson, an amazing developmental editor, who helped to make my words clearer than I could have done on my own, kept me on task, and rapped my knuckles when necessary. I owe a great debt to Chris Mills, my tech editor, who has always been one of my web heroes, for all his suggestions and corrections. Thank you to Katerina Malone, Jennifer Needham, and Patricia J. Pane, who caught all big, small, and in-between mistakes, and Rebecca Plunkett for indexing the book.

Thanks also to my friends—Marc & Sharon, Kai & Kristi, Rob & Sara, Rudy & Stefanie, Tracy & Lori, Brad & Lori, and many others—who constantly gave me support and encouragement.

Thank you to my students—all of you whom I've taught over the past six years—who were the inspiration behind this book. Special thanks go to Jonathon Wolfer, my longtime student, who designed my book.

To iconmonstr (<http://iconmonstr.com/>) for many of the icons at the start of each chapter and within Chapter 13.

To the brilliant and funny Dr. Leslie Jensen-Inman, who saw something in me I didn't and recommended me to Michael Nolan: I'm so thankful for our email conversation, which began two years ago, about elevating web design in higher education—and for (most of all) your friendship.

Heartfelt thanks to my mom and dad for all those years of support and love ... I sure wish dad could have lived to see his youngest getting a book published.

CONTENTS

WELCOME	XI
Who This Book Is Forxi
Objectivesxi
Description	xii
Conventions Used	xii
Code Examples	xii
Tips & Notes	xiii
Files	xiii
Requirements	xiv
Text Editor	xiv
Image Editor	xiv
Browser	xiv
Web Hosting & Domain Registration	xv
Book + Website = Enhanced Learning	xv
Chapter Downloads	xv
Forums and Resources	xv
Video Demonstrations	xv
Structure	xvi
Teachers	xvi
Let's Get Started	xvii

PART 01: THE BASICS

CHAPTER 01 HTML FOUNDATIONS	3
Getting Started	4
Creating an HTML Document	4
Naming Files & Folders	5
HTML Structure	6
DOCTYPE	6
HTML ELEMENT	6
HEAD ELEMENT	7
BODY ELEMENT	7
Head Elements	8

Title Element	8
Meta Element	8
Language Attribute	10
Body Elements	10
Structuring Content	10
Wrapping Things Up	13

CHAPTER 02 TEXT ELEMENTS

Hands On	16
Outline	16
Text Headers	16
Main Topic	17
Sub-Topics	17
Cite Attribute	21
Creating Lists	22
Description Lists	24
Additional Text Elements	25
Adding Comments	28
Helpful Hook Element	28
Further Exploration	29
Wrapping Things Up	29

CHAPTER 03 LINKS & OBJECTS

Links	32
Objects: Images	37
Objects: Video	40
Wrapping Things Up	43

CHAPTER 04 TABLES

Table Element	46
Table Headings	46
Table Rows	47
Table Data Cell	48

Table Border	48
Table Caption	50
Table Header	50
Table Body	51
Table Footer	51
Spanning Columns	52
Wrapping Things Up	53

CHAPTER 05 CSS: INTRODUCTION	55
The Power of CSS	56
Selectors	57
Multiple Declarations	58
Adding CSS to HTML	60
Embedding CSS	60
Inline CSS	60
Link to an External CSS Document	61
Challenge: Move CSS out of HTML	62
Import a CSS Document	64
Selector Types	65
Universal Selector	65
Element Selector	66
Selector Grouping	66
Combinators	67
Pseudo-classes	72
:link pseudo-class	72
:visited pseudo-class	74
:hover pseudo-class	74
Challenge	75
:active pseudo-class	76
:focus pseudo-class	77
Wrapping Things Up	77

PART 02: WORKING TOGETHER 79

CHAPTER 06 HTML ATTRIBUTES	81
HTML: Class Attribute	82
Multiple Values	84
CSS: Class Selector	84
HTML: ID Attribute	86
CSS: ID Selector	88
Specificity Power of ID Selectors	89
HTML: Role Attribute	90
ARIA Landmarks	90
ARIA Landmark: banner	91
ARIA Landmark: navigation	91
ARIA Landmark: main	92
ARIA Landmark: complementary	92
ARIA Landmark: contentinfo	92
CSS: Attribute Selectors	94
Wrapping Things Up	95

CHAPTER 07 CSS: STYLING TEXT	97
Font Properties	98
Font Family	98
Font Size	102
Font Style	104
Font Variant	105
Font Weight	105
Font Property Shorthand	106
Text Properties	106
Text Align	106
Text Decoration	108
Text Indent	109

Text Transform 109

Spacing Properties 110

 line-height. 110

 Fixed vs. Relative Leading 111

 Challenge 111

Website Resources on Font Styles 111

Color 111

 Color Property 112

 Name Value Challenge 112

 Hex Value Challenge 114

Wrapping Things Up 115

CHAPTER 08 THE BOX MODEL 117

Box Dimensions. 118

 Working with the Box 120

 Content. 120

 Border 121

 Padding 124

 Margin 125

Background Color 126

Width, Height, and Overflow. 128

 Box Sizing 130

 Box-sizing Box Sizing 132

Browser Default Margin & Padding 134

 Resetting the Default 134

Wrapping Things Up 135

PART 03: LAYOUT & INTERACTIVITY 137

CHAPTER 09 LAYOUT PROPERTIES 139

Box Element Review 140

 Block Elements 140

 Inline Elements 140

Float. 141

 Float: none 142

 Float: left 142

Float: right 143

Float: inherit 144

Clear Floats 145

Fixing the Collapse. 146

Overflow 148

Position 151

 Position: static 151

 Position: relative. 151

 Position: absolute. 152

 Position: fixed. 153

Z-index. 154

Display 159

 Display: block. 160

 Display: inline 161

 Display: inline-block. 162

 Other Display Values 163

Wrapping Things Up 163

CHAPTER 10 PAGE LAYOUT 165

Fixed-Width Layouts 166

 Why Use a Fixed Design? 167

 Why Avoid a Fixed Design? 167

 Breaking Down the Structure. 169

 Coding CSS Fixed Rules 170

 Challenge 174

Fluid Layouts 175

 Fixed Grid to Fluid Grid 176

Responsive Layouts. 180

 Challenge 180

 Mobile First Would Be Best. 181

 Fixed to Responsive. 182

 @media. 182

 Convert Pixels to Ems 188

 Challenge 189

 Viewport 190

Wrapping Things Up 191

CHAPTER 11	NAVIGATION	193
	The HTML of Navigation	195
	Unordered Lists	195
	Ordered Lists	196
	Description Lists	196
	No Lists	197
	Types of Navigation	198
	Vertical Navigation	198
	Horizontal Navigation	206
	Additional Navigation Patterns	212
	Breadcrumbs	212
	Drop Down Menus	212
	Sitemaps	213
	Search Boxes	213
	Wrapping Things Up	213

CHAPTER 12	FORMS	215
	How Do Forms Work?	217
	Form Elements	218
	The form Element	218
	Data Input	220
	Textarea	224
	Select Option	225
	Labeling Forms	226
	Fieldset	227
	A Contact Form	228
	Form	228
	Thank You	228
	Automated Response	229
	Processing Script	229
	Help Online	229
	Wrapping Things Up	229

PART 04: NEXT STEPS **231**

CHAPTER 13	WORKFLOW	233
	Asking Questions	237
	Content	238
	Sitemap	238
	Categorize	239
	Textual Content	239
	Visual Language	240
	OS Screenshot	240
	Sketching	242
	Wireframe	243
	Coding	244
	Prototypes	244
	Wrapping Things Up	245

CHAPTER 14	LOOKING AHEAD	247
	HTML5	250
	Resources: Books	251
	Resources: Online	251
	CSS3	252
	Resources: Books	253
	Resources: Online	253
	JavaScript	254
	Tabbed Boxes	254
	Lightbox Gallery	255
	JavaScript Libraries	256
	Resources: Books	256
	Resources: Online	257
	Wrapping Things Up	257

*I am always doing that
which I can not do, in
order that I may learn
how to do it.*

~ Pablo Picasso

WELCOME TO FOUNDATIONS OF WEB DESIGN: HTML & CSS

Since Sir Tim Berners-Lee, a research scientist, proposed and developed an internet-based hypertext system back in 1989, and then, in the early 1990s, developed the first HTML documents, HTML has been the backbone for creating websites. While style sheets have been a part of markup languages since the 1980s, CSS (Cascading Style Sheets), which was created to separate presentation (design) from content (markup documents), was first adopted in the mid-1990s; since then, CSS has become the standard styling language for the Web. Today, HTML and CSS need to be a part of every hobby and professional web designer's toolkit.

WHO THIS BOOK IS FOR

This book is written primarily for two types of readers:

- Absolute beginners at hand-coding HTML and CSS
- Those who have used a drag-and-drop website-builder application, but may have little to no idea what all those letters (`p`, `q`, `b`, etc.), numbers (`h1`, `h2`, etc.), and words (`strong`, `span`, `div`, etc.) inside the angled brackets (`< >`) really are

You may wish to learn HTML and CSS to help build a personal website (professional or for family) or to look to take a step toward working in the field of web design. Either way, I believe you'll find this book, and the accompanying resources, to be helpful in your journey.

OBJECTIVES

The objective of *Foundations of Web Design: HTML & CSS* is to provide a fundamental knowledge of HTML (**H**ypertext **M**arkup **L**anguage) and CSS (**C**ascading **S**tyle **S**heets). Learning is accomplished through hands-on coding demonstrations—in the book and in online video tutorials—and challenges. Additionally, you can download assignments in which you'll be asked to develop a basic website based upon chapter material, a one-page promotional site for a midterm project, and a personal website—professional, family, or client—for your final project.



ONLINE ASSIGNMENT REVIEWS

If you are a self-learner—and not in a classroom directed by an instructor—you will be able to submit your work online for questions and reviews.

DESCRIPTION

The different parts of this book work to build your knowledge and skill in a slightly different manner than other books of this type.

PART 1: BASICS

In Chapters 1-5, you'll learn the basics of creating and coding HTML and CSS documents. While it may seem very elementary, it lays an important foundation for the subsequent chapters.

PART 2: WORKING TOGETHER

In Chapters 6-8, you will start to see how HTML and CSS work together in developing more complex web pages through the use of attributes, selectors, typography, and the all-important box model. This is either where it “clicks” for many students or where they become “lost.” Make sure you take your time and work through each demonstration (multiple times, if necessary).

PART 3: LAYOUT AND INTERACTIVITY

Chapters 9-12 will teach you how to create layouts that are responsive for different devices, how to develop navigation elements that help users find information on your website, and how to design and use forms.

PART 4: NEXT STEP

Chapters 13 and 14 talk about the tasks that are involved in developing a website from beginning to end, and about additional skills a web designer needs—giving you a foretaste of what you can do next and pointing toward where you can learn those skills.

CONVENTIONS USED

Code examples, notes, and asides will have different typographical styles from the normal body text.

CODE EXAMPLES

HTML code that is being used as a current example is a bright blue color:

```
<h1>This is a header</h1>
```

HTML code that has been previously shown but is part of a new example is shown in a muted blue color:

```
<h1>This is a header</h1>
<p>This is a paragraph</p>
```

CSS code that is being used as a current example is a bright magenta color:

```
h1 {color: orange;}
```

CSS code that has been previously shown but is part of a new example is shown in a muted magenta color:

```
h1 {
  color: orange;
  font-size: 2em;
}
```

Sometimes lines of code are intended to be written on a single line, but the book requires a line wrap. An arrow shows where a line break occurs for print formatting purposes only and should be ignored.

```
<blockquote cite="http://alistapart.
com/article/uncle-sam-wants-you-to-
optimize-your-content-for-mobile">
```

TIPS & NOTES

»»» *TIP: TIPS & NOTES PROVIDE REMINDERS ABOUT THE CURRENT TOPIC OR STEP.*



SIDEBARS

Sidebars help you learn more about a topic through related information.

FILES

When downloading files, you'll find they are all compressed in the .zip format and will need to be uncompressed (or extracted) prior to use.

»»» *NEED HELP EXTRACTING? VIDEO*

TUTORIALS ARE AVAILABLE ON THE WEBSITE IF YOU DO NOT KNOW HOW TO UNCOMPRESS A ZIP FILE.

When referencing folder and document names for demonstration, the names will be in bold lettering:

foldername

document.html

document.css

If you see a forward slash (/) between two names

fowd_ch02_folder/01-book.html

it's telling you that the document (**01-book.html**) is located in a specific folder (**fowd_ch02_folder**).

REQUIREMENTS

In order to work through the assignments and projects in this book, you do not need expensive software or hardware.

TEXT EDITOR

You will need a code text editor—not a text editor that came with your computer—to write HTML and CSS. An excellent, and free, editor I would recommend is Komodo Edit (<http://activestate.com/komodo-edit>) for Windows, Mac, and Linux. Other free alternatives include Notepad++ for Windows, TextWrangler for Mac, and Bluefish for Linux.

»»» *NOTE: THE WEBSITE HAS LINKS AND TUTORIALS TO HELP YOU INSTALL AND SET UP KOMODO EDIT.*

IMAGE EDITOR

If you have Photoshop or Photoshop Elements, those are fine but a bit much for what you need at this point. Pixlr (<http://pixlr.com/editor/>) is a great free application that works directly in your browser. Additional image editors are included on the website.

BROWSER

It would be best to download one or two modern—standards-compliant—browsers such as Chrome, Firefox, or Opera (**Figure 1**). All of these can be installed on Linux, Mac, and Windows systems. The screenshots you'll see in this book were primarily taken while using Opera on Mac, unless otherwise noted.



FIGURE 1 Chrome, Firefox, and Opera are open-source browsers that keep current with many web standards and are available on multiple operating systems.

»»» *NOTE: IT'S BEYOND THE SCOPE OF THIS BOOK TO HAVE MAC USERS SET UP A WINDOWS OPERATING SYSTEM TO RUN INTERNET EXPLORER. WINDOWS USERS ALSO CANNOT INSTALL THE LATEST VERSION OF SAFARI. THERE'S MORE ON MULTI-BROWSER TESTING IN CHAPTER 13 ON THE WEBSITE.*



INTERNET EXPLORER

If you're a Windows user, you might be accustomed to using Internet Explorer (referred to throughout this book as IE). It's fine to continue using it, but the examples in this book will primarily work for version 8 (IE8) and above. Microsoft has stopped supporting IE6, and IE7 lags behind web standards common in most browsers. Check your version of IE by selecting About Internet Explorer under the Help tab (**Figure 2**).



SAFARI

If you're a Mac user, Safari is pre-installed on your computer and has kept up good support of web standards. However, make sure your version of Safari is as up to date as possible. At minimum, you should have Safari 5, but it would be best to have Safari 6. To check your version of Safari, simply select About Safari under the Safari menu (**Figure 3**).

WEB HOSTING & DOMAIN REGISTRATION

In the final set of assignments and projects, you will need to host your website online. Instructions on how to register a domain name, set up a free hosting account, and uploading files are supplied on the corresponding website.

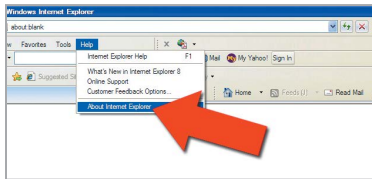


FIGURE 2
How to check your version of IE.

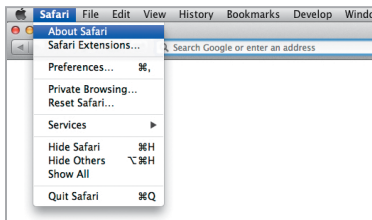


FIGURE 3
How to check your version of Safari.

BOOK + WEBSITE = ENHANCED LEARNING

What's different about this book?

While the book does contain a substantial amount of information and examples by itself, it's not intended to be a bulky, comprehensive resource on its own.

The corresponding website, found at <http://foundationsofwebdesign.com>, provides a great deal of additional material:

CHAPTER DOWNLOADS

At the beginning of a chapter that has files and documents to download, you'll see the following message:

CHAPTER CODE

»»» *THE CODE EXAMPLES FOR THIS CHAPTER CAN BE DOWNLOADED FROM THE WEBSITE ([HTTP://FOUNDATIONSOFWEBDESIGN.COM](http://foundationsofwebdesign.com)).*

FORUMS AND RESOURCES

Additionally, the website has a forum where you can answer questions, submit website assignments for review, download assignments, and find additional resources related to each chapter topic.

VIDEO DEMONSTRATIONS

Video tutorials are provided for each chapter to enhance and reinforce what is taught in the book. Why the combination of print and video? We learn best when we can combine the use of as many senses as possible, and, between the book and videos, I'm hoping (for most of you) to hit at least three (sight, hearing, and touch) of the five senses. The videos also supply a few advantages:

- You see me walk through each step of a demonstration.
- You work right along with the video.
- You can pause and replay if anything doesn't make sense.

STRUCTURE

How should you approach the material?

1. Read through the chapter and work through the demonstrations.
2. Watch the videos and work through the demonstrations.
3. Work on the assignment/projects (PDFs available on the site).

TEACHERS

If you're a teacher and wish to adopt this textbook for your class, I've provided the following materials to download from the website:

- Course syllabus
- Assignments
- Two projects
- Rubrics for grading

These documents can be modified to fit your class needs.



FLIPPED CLASSROOM

My classes use the “flipped classroom” model, which means students read and watch video demonstrations outside of class and work on assignments and projects during the class time. This allows the teacher to work as a guide and revisit concepts students don't understand. There is more information online, and I am happy to talk to you about this model if you're interested in finding out more.

LET'S GET STARTED

With the introduction behind you, it's time to jump in and get going on your first HTML document.

*If you hold a cat by the tail
you learn things you cannot
learn any other way.
~ Mark Twain*

This page intentionally left blank

This page intentionally left blank



TABLES

—

04

»»» CHAPTER CODE

THE CODE EXAMPLES FOR THIS CHAPTER CAN BE DOWNLOADED FROM THE WEBSITE ([HTTP://FOUNDATIONSOFWEBDESIGN.COM](http://foundationsofwebdesign.com)).

What are HTML tables used for? Think of spreadsheets or data you want to organize and present. Maybe you have statistics, a schedule, or a project you need to organize. Let's see how each element is used.

TABLE ELEMENT

The `table` element allows you to arrange and present data in rows and columns of cells. However, it can't do much without its supporting cast: the `tr` (table row), `td` (table data cell), and `th` (table header) elements. So, let's start out by using these basic elements with some common attributes.

STEP 4.1.1

Open the `table.html` document in the **fowd_ch04_folder** and enter a table element inside the `div` element.

```
<div class="container">
  <table>
  </table>
</div> <!-- end .container -->
```

TABLE HEADINGS

The `th` element holds the heading of each column. Let's say you're developing a reading list of your favorite books and you want columns for the authors, for the book titles, and for indicating if you've read each book.

STEP 4.1.2

Add the three `th` elements with the column topics within the `<table>` tags.

```
<table>
  <th>Author(s)</th>
  <th>Book Title</th>
  <th>Read?</th>
</table>
```

Save your document and view the result in the browser (**Figure 4.1**).

➤ YEARS PAST

Did you know that web designers used to design layouts using table elements! Well, it's not so crazy considering that tables were one way—before there was adequate CSS support—to make sure your layout looked the same from browser to browser. However, in modern web design, page layout is not the purpose of tables.

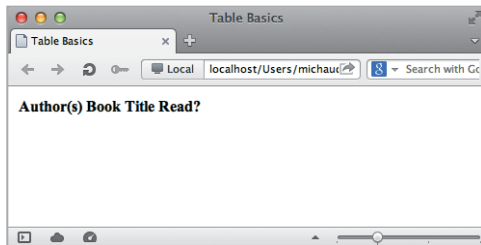


FIGURE 4.1
Table headers are bold by default.

TABLE ROWS

The `tr` element defines the rows of your table. While the three `th` elements already display as a “row,” the `tr` element gives structure and meaning to a row—and a table can have lots of rows.

```
<table>
<tr>
  <th>Author(s)</th>
  <th>Book Title</th>
  <th>Read?</th>
</tr>
</table>
```

The next row will add data about one of your favorite books. The data for each book will be contained in a row, between a set of `<tr>` `</tr>` tags. The first book is John Steinbeck's *The Grapes of Wrath*, which you will affirm you've read.

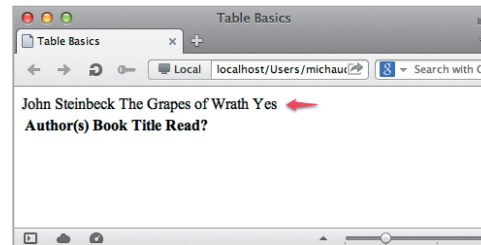


FIGURE 4.2
Unstructured table row.

```
<table>
<tr>
  <th>Author(s)</th>
  <th>Book Title</th>
  <th>Read?</th>
</tr>
<tr>
  John Steinbeck
  The Grapes of Wrath
  Yes
</tr>
</table>
```

When you view this site, notice that the data is above (not below) each heading and not within the borders of your table (**Figure 4.2**).

In order to get the data in the correct spot, you need to add another set of tags around each piece of data using the `td` element.

TABLE DATA CELL

The `td` element defines the cells of your table and contains your data.

STEP 4.1.3

In the example, the book's name, the author's name, and your "yes" or "no" (whether or not you've read the book) are your data. Each item of data needs to be within a `td` element.

```
<table>
<tr>
  <th>Author(s)</th>
  <th>Book Title</th>
  <th>Read?</th>
</tr>
<tr>
  <td>John Steinbeck</td>
  <td>The Grapes of Wrath</td>
  <td>Yes</td>
</tr>
</table>
```

Now each cell of data is below the header it matches (**Figure 4.3**).

TABLE BORDER

In order to see your table a little more easily, you're going to add a border around it. Although we've not discussed CSS (Cascading Style Sheets), this method is best for displaying borders.

STEP 4.1.4

Add styles, within the `<style></style>` tags located in the `head` element, to show the border for the `table`, `th`, and `td` elements within your HTML document (**Figure 4.4**).

```
<head>
  <meta charset="utf-8">
  <title>Table Demo</title>
  <style>
    table, th, td{
      border: 1px solid #333;
    }
  </style>
</head>
```

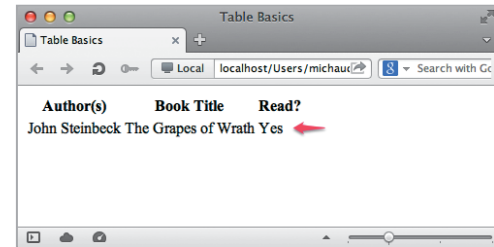


FIGURE 4.3
Structured table data.

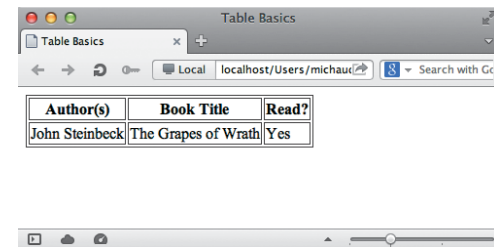


FIGURE 4.4
Your CSS styles create a border around your table and data cells.

➤ ADDING STYLES

While it's good practice to keep your content (HTML) separate from your design (CSS), the style element allows you to add CSS to your HTML document. It must be contained within the `head` element.

We'll get more in-depth into CSS in Chapter 5.

➤ BAD HTML

In the past, the border for a table and its data was displayed using the (now) deprecated `border` attribute.

```
<table border="1">
```

Deprecated HTML tags and attributes are those that have been (or will soon be) removed from the HTML specification. It is recommended that you not use these HTML tags because they may become obsolete.

For a list of deprecated HTML elements and attributes, view http://webdesign.about.com/od/htmltags/a/bltags_deprctag.htm.

ORDER HEADERS AND DATA

Each `td` has to be ordered according to the `th`. So your first `td` element should relate to the first `th` element, the second `td` element relates to the second `th` element, and so on (**Figure 4.5**).

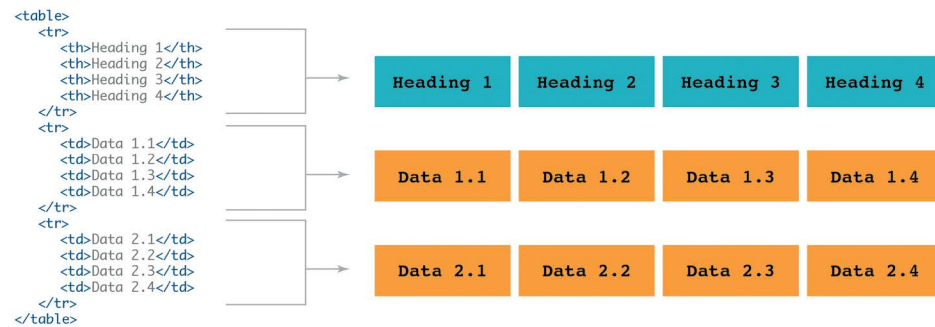


FIGURE 4.5 Illustration of an ordered layout for table headers and table data.

TABLE CAPTION

The `caption` element gives an overall title to the table. In this case, maybe you'd call it "My Reading List"—it's actually a list of books that I have either read or want to read.

STEP 4.1.5

The `caption` element for the table comes right after the opening `<table>` tag. Go ahead and add it to your code.

```
<table>
<caption>My Reading List</caption>
<tr>
  <th>Author(s)</th>
  <th>Book Title</th>
  <th>Read?</th>
</tr>
<tr>
  <td>John Steinbeck</td>
  <td>The Grapes of Wrath</td>
  <td>Yes</td>
</tr>
</table>
```

Notice the text for the caption is placed above the table headers, but outside of the border (**Figure 4.6**).

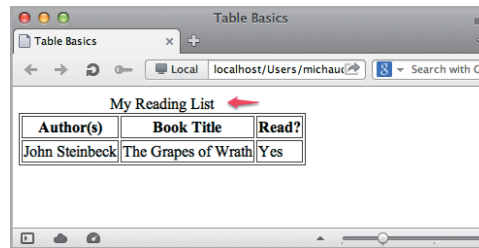


FIGURE 4.6

The caption is added to the top of the table, but not within the border.

TABLE HEADER

The `thead` element helps to group your table header (`th`) cells.

STEP 4.1.6

In this table, the header row is where the `th` elements are: "Book Title," "Author(s)," and "Read?" Simply add the opening `<thead>` tag before the opening `<tr>` tag and add the closing `</thead>` tag immediately after the closing `</tr>` tag for that row.

```
<table>
<caption>My Reading List</caption>
<thead>
<tr>
  <th>Author(s)</th>
  <th>Book Title</th>
  <th>Read?</th>
</tr>
</thead>
...
</table>
```

TABLE BODY

The `tbody` element groups the body—the set of rows where the content resides—of the table.

STEP 4.1.7

Place the `tbody` tags around the book list.

```
<table>
<caption>My Reading List</caption>
<thead>
<tr>
  <th>Author(s)</th>
  <th>Book Title</th>
  <th>Read?</th>
</tr>
</thead>
<tbody>
<tr>
  <td>John Steinbeck</td>
  <td>The Grapes of Wrath</td>
  <td>Yes</td>
</tr>
<!-- additional table rows and data
are stored in the tbody element -->
</tbody>
</table>
```

TABLE FOOTER

The `tfoot` element groups the summary (footer) area of the table. The summary of a table could show the copyright information for the data, give credit to who has created (or edited) the list, provide totals of column values, or indicate when the table was last updated.

STEP 4.1.8

After the closing of the `</tbody>` tag, add the `<tfoot></tfoot>` tags. Within the `tfoot` element, add a `tr` element and `td` element with the data Updated March 5, 2013.



MULTIPLE BODIES

It is possible to have more than one `tbody` inside a `table`. This is covered with the `rowspan` attribute in the bonus material for this chapter you can download on the website.

>>> NOTE: THE FOLLOWING EXAMPLE FOR PLACING `tfoot` APPLY ONLY TO THE MODERN HTML5 DOCUMENT. HOWEVER, THERE ARE STILL MANY HTML4 AND XHTML DOCUMENTS IN USE, SO IF THAT'S YOUR SITUATION, SEE THE ASIDE ON PAGE 52.

```
<table>
<caption>My Reading List</caption>
<thead>
  <tr>
    ...
  </tr>
</thead>
<tbody>
  <tr>
    ...
  </tr>
<!-- additional table rows and data are
stored in the tbody element -->
</tbody>
<tfoot>
  <tr>
    <td>Updated March 5, 2013</td>
  </tr>
</tfoot>
</table>
```

Now you'll view your document with the addition of your new elements (**Figure 4.7**).

The screenshot shows a web browser window titled "Table Basics" displaying a table with the following content:

Author(s)	Book Title	Read?
John Steinbeck	The Grapes of Wrath	Yes
Updated March 5, 2013		

FIGURE 4.7

The data in the `tfoot` now displays at the bottom of the table layout.



FOOTER PLACEMENT FOR HTML4 AND XHTML

If you're going to be developing (or updating) a website that has a DOCTYPE for HTML4 or XHTML Strict (plenty of websites still use them), you'll need to place the `tfoot` after the `thead` and before the `tbody` element.

```
<table>
  <caption></caption>
  <thead></thead>
  <tfoot></tfoot>
  <tbody></tbody>
</table>
```



MEANINGFUL MARKUP BUT NO STYLE

While the `thead`, `tbody`, and `tfoot` elements will not change the look of your table, they will add some meaning (semantics) and provide hooks to style your table. Although these elements are not required, I'd recommend their use, as there are additional benefits that go beyond the scope of this book.

SPANNING COLUMNS

The `colspan` attribute allows you to expand a data cell to span multiple columns in the table. So, in the example, you have the `thead` that consists of one row with three columns of data (**Figure 4.8**).

And, in the `tbody`, you currently have one row with three `td` elements (**Figure 4.9**).

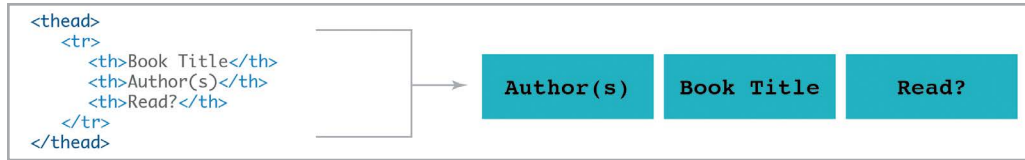
However, the `tfoot` element has one row (`tr` element) and only one `td` element (**Figure 4.10**).

STEP 4.1.9

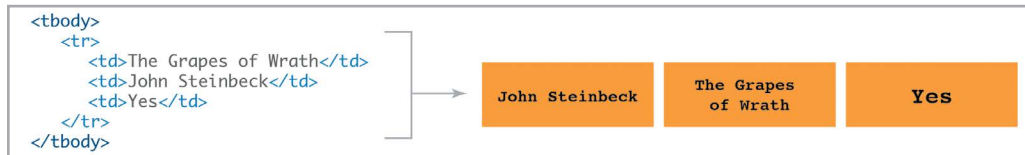
You can tell that `td` element to span multiple columns by adding the attribute `colspan` (short for "column span") and assigning a value (in this case, you want it to expand to three columns wide).

```
<tfoot>
<tr>
  <td colspan="3">Updated March 5, 2013</td>
</tr>
</tfoot>
```

The result gives you a `tfoot` element that spans all three columns (**Figure 4.11**).

**FIGURE 4.8**

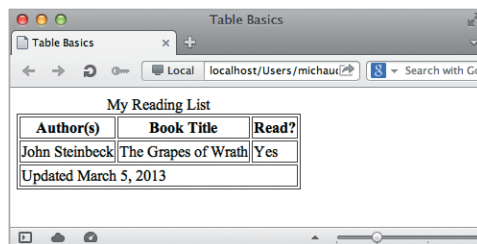
Your table heading has three columns.

**FIGURE 4.9**

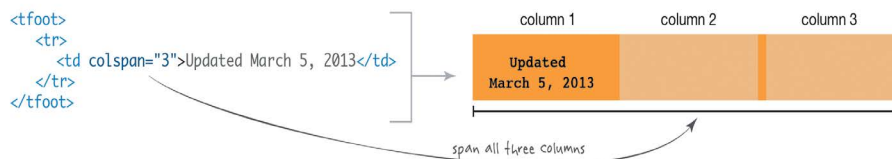
Your table body has three columns of data.

**FIGURE 4.10**

Your table footer has only one column of data.

**FIGURE 4.11**

Your footer now spans all three columns of your table.



WRAPPING THINGS UP

You now have the basic foundation of how to structure your data with the table element and the basic supporting cast. While they're not the easiest elements to mark up, tables can be extremely useful for box scores, budgets, etc. In the next chapter, we'll take a break from HTML markup and I'll introduce you to the design side of the Web with CSS—allowing your right brain to play a little.

DOWNLOAD REMINDER

»»» *THERE IS BONUS CONTENT TO DOWNLOAD FOR ADDITIONAL MARKUP FOR THE TABLE ELEMENT AND AN ASSIGNMENT TO TEST YOUR SKILLS!*

This page intentionally left blank

INDEX

Symbols

- + (adjacent sibling selectors), 68–69
- & (ampersand) character, 9, 11
- [] (attribute selectors), 94
- \ (backslash), 106
- . (class selectors), 70–71, 84
- : (colon), 72
- © (copyright character), 9
- { } (curly brackets)
 - adding fluid rules within, 184
 - placing selectors in, 57
- " " (double quotation marks), 99
- > (greater than) character
 - indicating child selector with, 68, 69
 - using as special character, 9
- # (ID selector), 71, 87, 88, 89
 - about, 88
 - class** vs., 89
 - power of, 87
- < (less than) character, 9
- ; (semicolons), 58
- ' ' (single quotation marks), 99
- ~ (tilde), 69
- * (universal selector), 65

A

- a** (anchor) tag
 - linking to sites with **href**, 32, 33, 43
 - turning text into links with, 22
 - using pseudo classes with, 72, 73, 74
 - wrapping text with, 32–34
- a** selector, 73, 76
- absolute paths, 34, 35–36
- absolute positioning, 152–153
- accessibility. *See also* ARIA landmarks
 - designing forms for, 226
 - loading images on mobile devices for, 187
 - resizing sites for mobile device, 191

- action** attribute, 218
- :active** pseudo-class
 - getting user feedback with, 208
 - uses for, 76
 - using for touch screen devices, 201, 208
- adjacent sibling (+) selectors, 68–69
- aligning text, 106–107
- a:link** pseudo-class, 73
- allowfullscreen** attribute, 41
- alt** attribute, 37, 39
- anchor tags. *See a* tag
- ARIA landmarks
 - banner** role value, 90, 91
 - complementary** role value, 92
 - contentinfo** role value, 92, 93
 - main** role value, 92
 - navigation** role value, 91, 195
 - role** attribute for, 90
- asymmetrical columns, 171–172
- @import** rule, 64
- @media** rules
 - adding media types with, 182–184
 - developing new, 86, 189
 - handling **max-width** with, 185
 - @media** queries for mobile devices, 189–191
 - using, 182
- attribute selectors, 94
- attributes
 - about HTML, 9, 82
 - alt**, 37, 39
 - avoiding cross-attribution, 84
 - case sensitivity of values, 85
 - class**, 82–84
 - deprecated, 49, 82
 - frameborder** and **allowfullscreen**, 41
 - height**, 39
 - id**, 83, 84, 86–87
 - multiple values for **class**, 84
 - role**, 90–93

- selecting, 84
- setting character, 9
- specificity of **ID** selectors for, 89
- src**, 37
- width**, 39, 40–41
- audience for websites, 167
- auto margins, 170
- auto widths for block-level elements, 187
- autoresponse folder, 228, 229

B

- Back button, 37
- background color
 - box model, 126–127
 - changing for primary and secondary content, 184
 - indicating link selection with, 201
- banner** role value, 90, 91
- b** tag, 26
- Berners-Lee, Sir Tim, 6, 36
- block-level elements
 - adding, 11
 - defined, 140, 159
 - floating, 141
 - horizontal navigation using, 161
 - illustrated, 140
 - inline-level elements within, 140
 - line breaks before and after, 19
 - setting auto widths for, 187
- block** value, 160, 161
- blockquote** element, 20–21
- body** element
 - adding and structuring, 10–12
 - adding block-level elements within, 11
 - function of, 7
 - positioning **fixed** elements to, 153
- bold text
 - CSS styling for, 105
 - HTML elements for, 26

- border-top** property, 108
 - borders
 - box model, 118, 119
 - coding shorthand for, 123, 124, 125
 - color of, 122, 123
 - defined, 121
 - setting style of, 121–122
 - table, 48–49
 - width of, 123
 - without values, 124
 - box-sizing** property, 132–133
 - boxes, 118–135. *See also* containing box; position
 - property
 - about, 118
 - adding content to, 120
 - adjusting left or right, 142–143
 - background colors for, 126–127
 - borders for, 118, 119, 121–124
 - child/parent elements in, 119, 120
 - collapsing containing block, 145
 - containing, 142
 - height adjustments for, 128–129, 133
 - hiding extra content, 150
 - illustrated, 118, 119
 - laying out inline, 162–163
 - margins for, 118, 119, 125
 - padding property for, 118, 119, 124–125
 - resetting browser defaults for, 134–135
 - role of **display** property in, 159–163
 - sample HTML code for, 120
 - scroll bar for, 149–150
 - sizing, 130–133
 - stacked box order, 142
 - tabbed JavaScript, 254
 - width of, 128–129, 132–133
 - br** element, 29
 - breadcrumbs, 34, 212
 - browsers. *See* web browsers
 - bulleted lists, 23
 - buttons
 - Back, 37
 - radio, 221, 227
 - submit, 223
- C**
- capitalizing text, 109
 - caption** element, 50
 - captions for tables, 50
 - Cascading Style Sheets. *See* CSS
 - case sensitivity
 - attribute value, 85
 - CSS properties for upper and lower case, 109–110
 - lowercase for document and folder names, 5
 - naming CSS documents, 62
 - recommendations for HTML, 6
 - cells
 - defining table, 48
 - expanding data to multiple, 52–53
 - centering text, 106
 - characters
 - encoding, 9
 - using special, 11
 - charset** attribute, setting, 9
 - checkboxes
 - labels for, 227
 - using on forms, 222
 - child** (>) selectors, 68, 69
 - child element
 - adding background color to, 127
 - in box model, 119, 120
 - inheriting **float** property, 144
 - setting width of, 128–129
 - Chrome, 167, 182
 - cite** tag, 22
 - class** (.) selectors, 70–71, 84, 89
 - class** attribute, 82–86
 - id** vs., 83, 87
 - selecting, 84
 - styling paragraphs with, 85–86
 - using for styling, 28
 - clearfix method, 146–148, 174
 - coding phase
 - about, 235, 236
 - prototyping and testing in, 244
 - collapsing
 - clearfix method for reversing, 146–148, 174
 - clearing overflow, 148–149, 173–174
 - containing element, 145
 - color
 - border, 122, 123
 - changing background, 184
 - designating box background, 126–127
 - hex values for, 113–114
 - modifying link, 73–74
 - naming CSS, 112
 - RGB, 114–115
 - styling with CSS, 111–115
 - transparent, 126
 - color** property, 111
 - colspan** attribute, 52–53
 - columns
 - converting multiple to single, 188–189
 - creating three, 174
 - floating nested elements in two, 171–172
 - using footers across, 52–53
 - combinators, 67–70
 - about, 67
 - adjacent sibling selectors, 68–69
 - child selectors, 68, 69
 - descendant selectors, 67–68, 69
 - general sibling selectors, 69
 - parent/child relationships in CSS, 67
 - comments, 28
 - complementary** role value, 92
 - conditional loading, 188
 - contact form, 228–229
 - contact-process.php** document, 228–229
 - containing box
 - absolute positioning within, 152–153
 - auto margins in parent, 170–171
 - balancing overflow layout in, 148–149
 - collapsing, 145

- defined, 142
- fixed positioning within, 153–154
- hiding extra content in, 150
- nesting floated elements within parent, 171–172
- relative positioning in, 152
- setting stacking order in, 154–158
- transforming to fluid widths, 176–180
- uncollapsing, 146–148, 173
- using static positioning in, 151, 158

content

- blockquotes within, 20–21
- citing source of, 21–22
- comments within, 28
- developing, 238–239
- emphasizing text in, 27
- hiding, 150, 187–188
- inline-level elements in, 19
- interpreting for visually impaired, 37, 39
- italicizing, 27
- keeping separate from design, 49
- line breaks in block-level, 19
- lists within, 22–23
- outlining, 16–19
- paragraph elements added to, 19–20
- scrolling, 149–150
- structuring text in, 10–12
- within box model, 118, 119, 120

contentinfo landmark value, 92, 93

copyrights

- adding in table footer, 51
- using special character for, 9

CSS (Cascading Style Sheets). *See also* CSS3; CSS rules; layout; styling links; and *specific CSS properties*

- adding to HTML, 60–61
- attribute selectors in, 94
- coding shorthand in, 123, 124, 125
- color styling with, 111–115
- creating CSS documents, 62
- .css files, 62
- declarations in, 57, 58, 60

- embedding in **head**, 60
- exercise in styling links, 75
- folders for, 62
- fonts in, 98–106
- hiding content on mobile devices, 187–188
- HTML hooks for, 28–29, 90
- ID** selector vs. HTML **id** attribute, 88
- importing CSS documents, 64
- increasing size of fonts with, 59
- inline, 60–61
- linking HTML to external CSS document, 60, 61, 63, 64
- moving out of HTML, 62, 63
- overriding previous display rules, 162
- parent/child relationships in, 67
- pseudo-classes in, 72–77
- samples of designs with, 56
- selector grouping, 66
- selectors in, 57–58, 65
- separating HTML from design, 49
- spacing text, 110–111
- styling blocks of text, 106–110
- table borders in, 48–49
- tagging semantic HTML vs., 25
- types of display properties in, 159
- versions of, 69
- viewing HTML element as box, 140
- working with box model, 118

CSS3

- additional screen properties in, 182
- further resources on, 252–253
- general sibling selector in, 69
- JavaScript with, 256
- jobs requiring, 248, 249

CSS documents

- adding media queries in, 182–183, 189–191
- creating and naming, 62
- importing, 64
- linking HTML to external, 60, 61, 63, 64
- moving CSS rules to, 62, 63
- writing CSS rules for, 84

CSS rules. *See also* **@media** rules

- about, 57
- adding to fixed-width website, 170–175
- browser support of, 134
- creating RGB, 115
- developing for fixed layouts, 168
- expanded declarations in, 60
- grouping CSS elements by, 66
- illustrated, 59, 60
- importing, 64
- media queries as, 189–191
- moving to CSS style sheet document, 62, 63
- multiple declarations in, 58–59
- navigation using, 199
- overriding, 86
- placing in external documents, 61
- using when link clicked, 76
- writing for hex values, 114
- writing to css documents, 84

CSS Zen Garden, 56–57

Custom size option (YouTube), 41

D

data input

- sending to server, 219
- structuring for forms, 220–223
- using drop-down menus, 225

dd element, 24

declarations

- compact or expanded formats for, 60
- CSS, 57
- !doctype html**, 6
- multiple, 58–59
- semicolons used in, 58

deprecated HTML, 49, 82

descendant selectors, 67, 69

description lists, 24–25, 196–197

design. *See* layout

designers. *See* web designers

designing phase, 235, 236

- developers, 249
- devices. *See* mobile devices
- directories, 5
- display** property, 159–163
 - about, 159
 - creating horizontal navigation with, 162
 - inline** value with, 161–162
 - overriding previous rules for, 162
 - purpose of, 140
 - rendering inline elements as block, 160–162
- div** element
 - adding, 11–12
 - defining sample website using, 169
 - displaying dimensions for, 181–182
 - layout of stacked boxes within, 141, 142
 - section** vs., 12–13
 - website structures using, 170
- dl** element, 24
- !doctype html** declaration, 6
- documents. *See also* CSS documents
 - adding titles for, 7
 - block-level elements for, 11
 - character attributes for, 9
 - contact-process.php**, 228–229
 - creating in HTML, 4–7
 - defining content language for, 10
 - headers for topics in, 17
 - index, 5, 32
 - linking to PDF, 34
 - naming, 5
 - saving, 7
 - starting with outline, 16, 17
 - sub-topics within, 17–18
- drop-down menus
 - navigating from, 212
 - using in forms, 225
- dt** element, 24
- E**
 - element selectors, 66
 - elements. *See also specific elements*
 - CSS parent/child relationships in, 67
 - empty, 37
 - em** element, 27, 111
 - em** value, 104, 188–190
 - embed** tag, 43
 - embedding
 - video in websites, 40
 - working with old embed code, 42–43
- F**
 - fieldset** element, 227
 - files
 - browser support for video, 40
 - .css, 62
 - naming, 5
 - selecting from forms, 223
 - size for links, 35
 - Firefox, 167, 182, 185
 - fixed leading, 111
 - fixed positioning, 153–154
 - fixed-width fonts, 100
 - fixed-width layout, 166–175
 - about, 166
 - changing to responsive design, 182–191
 - code for example, 169
 - converting to fluid layouts, 176–180
 - pros and cons of, 167
 - regions of, 169
 - setting up CSS rules for, 170–174
 - float** property, 141–150
 - adjusting box to left or right, 142–143
 - clearing floating elements, 145–146
 - creating horizontal navigation with, 207
 - floating nested elements in block, 145
 - inheritance of, 144–145, 184
 - nesting floated elements within parent container, 171–172
 - position** vs., 151
 - purpose of, 140, 141
 - removing from elements, 142
 - fluid layout, 175–180
 - converting fixed to, 176–180
 - handling gaps with overflow, 186–187
 - pros and cons of, 175–176
 - :focus** pseudo-class, 77
 - folders
 - autoresponse, 228, 229
 - CSS, 62
 - naming, 5
 - font-family** property
 - adding generic values for, 100, 101
 - basic rules for, 102
 - designating, 98–102
 - font stack for, 99
 - font** property, 106
 - font-size** property
 - changing leading with, 111
 - changing **px** to **em** value, 104
 - converting **px** value to percentage, 103
 - purpose of, 102
 - setting pixel size for, 102–103
 - font style** property, 104
 - font-variant** property, 105
 - fonts, 98–106. *See also specific CSS font properties*
 - bolding, 26, 106
 - combining CSS property values with **font**, 106
 - converting **px** value to percentage, 103
 - CSS increasing size of, 59
 - decreasing sizes for sub-level headings, 18
 - fixed- and variable-width, 100
 - fixed vs. relative leading for, 111
 - generic values for, 100, 101
 - italicized, 104
 - modifying properties of, 98
 - monospaced, 100–102
 - sans serif, 98–99
 - serif, 100
 - sizing, 102–104

- small caps for, 105
- styling, 104
- using font stack, 99

footers. *See also* **tfoot** element

- across spanned columns, 52–53
- adding copyrights in table, 51
- adding to fixed-width layout, 172–173
- clearing floating, 145–146
- placement in HTML4, 52

foreground color, 111

formatting hooks for HTML, 28–29

forms, 216–229

- automated responses to, 229
- contact, 228–229
- data input for, 220–223
- designing text areas of, 224
- establishing with **form**, 218
- file selection from, 223
- grouping sections of long, 227
- how they work, 217
- including checkboxes on, 222
- labeling, 226
- passwords for, 221
- radio buttons for, 221
- redirecting user to “Thank You” page, 228
- sending input data to server, 219
- structuring, 220
- submit buttons for, 223
- text input fields, 220
- uploading for testing, 229
- uses for, 216
- using drop-down menus, 225

Foundations of Web Design website, 13, 43, 65, 158

fragment identifiers, 37, 43

frameborder attribute, 41

Frost, Brad, 181, 191

G

Gardner, Liz, 189

general sibling (tilde) selectors, 69, 70

- browsers support for, 71

generic values for fonts, 100, 101

get values, 219

grid layouts, 168

H

h1 element, 11, 17, 18

head element

- applying CSS rules to, 58
- embedding CSS in, 60
- inserting, 7
- inserting **title** within, 8
- using sub-level headings, 18

headings

- adding headers, 11
- converting outline topics to, 16–19
- grouping table header cells, 50
- main topics as, 17
- number of levels of, 18
- table, 46–47
- using sub-topics as, 17–18

height

- adding video object, 40–41
- box model, 128–129, 133
- defining image, 39
- determining for box, 131

hex values, 113–114

hiding content

- in containing box, 150
- on mobile device layout, 187–188

horizontal navigation, 206–210

- converting vertical to, 207
- illustrated, 206
- resizing responsively, 209–210
- when to use, 210

:hover pseudo-class

- removing underlines from navigation menus, 202
- styling links with, 72, 74–75

hovering over hyperlink, 33, 72, 74–75

href attribute

- linking anchor tag to sites with, 32, 33, 43
- linking internal and external documents with, 63

HSL Color Picker, 113

html element, 6, 7

HTML (Hypertext Markup Language), 4–13. *See also*

- attributes; HTML5; inline elements
- adding CSS to, 60–61
- attribute selectors in, 94
- attributes in, 9, 82
- body elements for, 10–12
- box model in, 118
- class** attributes in, 82–84
- creating documents in, 4–7
- deprecated tags and attributes for, 49, 82
- description lists, 24–25, 196–197
- elements structuring documents, 6–7
- file and folder names in, 5
- head elements for, 8–10
- hooks for styling in, 28–29
- HTML4 footer placement, 52
- id** attributes in, 83, 84, 86–87
- keeping content separate from design in, 49
- linking to external CSS document, 60, 61
- moving CSS rules out of, 62, 63
- nav** element, 195
- ordered lists, 196
- preventing CSS from applying to, 182
- role** attributes with, 90
- section** vs. **div** in, 12–13
- structuring documents for, 6–7
- tables in, 46
- tagging styles vs. semantic, 25
- unordered lists in, 195
- using replaced elements, 140, 141
- versions of, 6

HTML4 footer placement, 52

HTML5

- attributes for forms in, 228
- main** tag in, 92
- nav** element in, 195
- preparing **doctype** for, 6
- resources on, 250–251
- role** attribute in, 90

section element in, 12–13
tfoot tags in, 51
 using **header** element in, 71
 video formats in, 40, 43
 hyperlinks. *See* links; styling links

I

i element, 27
id attributes
 class vs., 83, 87
 ID selector vs., 88
 using, 84, 86–87
ID (#) selectors
 about, 71, 88
 class vs., 89
 power of, 87
iframe element, 40
 images
 defining width and height of, 39
 designing site's visual language, 240–243
 making fluid, 179–180
 preparing in image editors, 38
 resizing, 39
img element, 37, 142–143
@import rule, 64
 importing CSS documents, 64
 indenting text, 109
 index.html documents, 5, 32
 inheritance
 effect on **float** property, 144–145, 184
 order of precedence for pseudo-classes, 76
 removing inherited **float** values, 142
inline-block element, 162–163, 208
 inline elements
 about, 19
 adding clickable space with, 208
 b, 26
 benefits of CSS, 60–61
 contained within block-level elements, 140
 defined, 140, 159

effect of **block** value on, 160, 161
 hooks for styling in, 28–29
i, 27
 illustrated, 140
span, 28–29
strong, 26
 transforming to inline-block element, 162–163
 using in text, 19, 25
 using **inline** value with **display**, 161–162
 Internet Explorer
 image width in IE 7, 179
 supporting versions of, 167
 unable to use **inherit** property in, 144
italic fonts, 104

J

Jaffe, Jeffrey, 6
 JavaScript
 about, 254
 CSS3 and, 256
 libraries of, 256
 lightboxes, 255
 programming skills in, 249
 resources, 256–257
 responsive navigation using, 211
 tabbed content in, 254
 Jobs, Steve, 206
 jQuery, 256
 justifying text, 106, 107

K

keyboard navigation, 77

L

labels
 radio and checkbox, 227
 using for forms, 226
lang attribute, 10

language preferences, 10
 launch phase, 235, 236
 layout, 140–163, 166–191. *See also* fixed-width layout;
 fluid layout; responsive layout
 adjusting **position** property, 151–154
 controlling box-level properties in, 159–163
 designing site navigation, 194
 fixed-width, 166–175
 fluid, 175–180
 grids for designing, 168
 moving box to left or right, 142–143
 ordered header and table, 49
 requiring no instructions, 222
 responsive design for, 142, 166
 separating content from design, 49
 setting stacking order with **z-index**, 154–158
 sketching, 166, 174, 175, 242
 using **float** property, 141–150
 using tables for, 47
 web vs. print, 141
 leading for text, 111
 left alignment, 106, 107
legend element, 227
li tag, 22
 lightboxes, 255
 line breaks
 before and after block-level elements, 19
 br tag for, 29
line-height property, 106, 110
line-through value, 107, 108
:link pseudo-class, 72–73
 links, 32–37. *See also* navigation; styling links
 adding clickable space around horizontal navigation,
 208
 changing text into clickable, 22
 color of clicked, 34
 connecting to PDF documents, 34
 creating to style sheets, 7
 exercise in styling, 75
 highlighting current page, 203
 hovering mouse over, 33, 72, 74–75

- illustrated, 33
- including file size for, 35
- linking HTML to external CSS document, 60, 61, 63, 64
- making, 32–37
- removing underlines from, 108, 202
- resizing navigation, 209–210
- within pages using fragment identifiers, 37, 43

lists

- creating, 22–23
- description, 24–25, 196–197
- designing pages without, 197
- ordered, 196
- structuring form with, 220
- unordered, 62–63, 195

LoVe HAte mnemonic, 76, 77

lowercase text, 110

M

Mac computers

- best san serif fonts for, 99
- screenshot software for, 240, 241

main role value, 92

Marcotte, Ethan, 166, 180

margins

- illustrated, 118, 119
- resetting browser defaults for, 134–135
- setting box, 125
- using auto, 170

max-width property, 183, 185

@media rules

- adding media types with, 182–184
- developing new, 86, 189
- handling **max-width** with, 185
- @media** queries for mobile devices, 189–191
- using, 182

menus

- designing mobile device navigation, 211
- drop-down, 212
- highlighting current page on, 203

- moving vertical menu to right, 202
- removing underlines from, 202

metadata

- adding browsing information in **meta** tag, 8
- defining, 7
- placing in HTML documents, 9

method attribute, 219

min-width property, 183

mind mapping, 238

mnemonics

- LoVe HAte, 76, 77
- TRouBLE, 122

mobile devices

- accessibility aids for, 187
- accessing websites, 166, 167
- changing navigation for, 204–205
- coding for touch screens, 201, 208, 210
- creating design for multiple, 180–181
- designing navigation menus for, 211
- fixed designs on, 167
- hiding content on, 187–188
- including in **@media** rule, 182–184
- removing unnecessary elements for, 187
- resizing sites for, 190–191
- testing designs for, 181–182
- viewport scaling for, 190–192, 204–205

modal dialogues, 255

monospaced fonts, 100–102

mood boards, 240–241

multiple bodies within tables, 51

N

naming

- CSS color values, 112
- CSS documents, 62
- files and folders, 5

nav element, 91, 195

navigation, 194–213. See *also* horizontal navigation;

- scrolling; vertical navigation
- ARIA landmarks for, 90–93, 195

- breadcrumb, 34, 212
- changing for smaller viewports, 204–205
- deleting gaps at bottom of element, 186
- designing site, 194
- drop-down menus, 212
- further reading on responsive, 211
- highlighting current page link, 203
- horizontal, 161, 162, 206–210
- removing underlines from menus, 202
- responsive, 204–205, 209–210
- scroll bar in box, 149–150
- search boxes for, 213
- sitemaps, 213
- types of HTML lists for, 195–197
- unordered lists for, 195
- using `:focus`, 77
- vertical, 198–205

navigation role value, 91

nested elements

- clearing overflow in, 148–149
- creating fluid layout with, 176, 177–180
- floating all in containing block, 145
- preventing box collapse of, 146
- two-column effect for, 171–172

none value for **float** property, 142

numbered lists, 23

O

object element, 42, 43

objects

- creating video, 40–43
- using **img**, **src**, and **alt** tags for, 37

ol tag, 22–23

online job boards, 246

ordered lists, 196

organizational resources, 238

outlining content

- adding headers for topics, 17
- converting topics to headings, 16–19
- starting documents by, 16, 17
- sub-topics, 17–18

overflow
 clearing, 148–149, 173–174
 handling fluid layout gaps with, 186–187
 using **overflow** property, 129

overflow value, 107, 108

overriding
 previous display rules, 162
 rules with multiple class values, 86

P

p element
 adding to content, 19–20
 using, 11, 19

padding
 adding box, 124–125
 coding shorthand for, 125
 illustrated, 118, 119
 resetting defaults for, 134–135

pages. *See* websites

paragraphs
 about, 19
 adding, 11, 19–20
 applying sibling selectors to, 68–70
 inheriting **float** property of, 144–145

param element, 42, 43

parent element
 adding background color to, 126–127
 edge of in box model, 119, 120
 inheriting **float** property of, 144
 setting width of, 128–129

passwords, 221

paths
 absolute, 34, 35–36
 relative, 35–36

PDF documents, 34

percentages of pixels, 103, 176

performance with universal selectors, 65

phones. *See* mobile devices

PHP scripts, 217, 218, 228–229

pixels. *See* **px**

placeholders, 39

planning phase, 235, 236

position property, 151–154
 absolute positioning, 152–153
 applying relative positioning, 151–152
 combining stacking order with, 154–158
 fixed positioning, 153–154
 purpose of, 140, 151
 using static positioning, 151, 158

post values, using in forms, 219

printing, 141

projects. *See also* workflow
 organizing, 237
 phases of, 235–236

properties. *See also specific properties*
 defined, 57
 omitting values for font, 106

prototyping websites, 244

pseudo-classes, 72–77
:active, 76, 201, 208
a:link, 73
:focus, 77
:hover, 72, 74–75
:link, 72–73
 order of precedence and inheritance in, 76, 77
 uses for, 72
:visited, 72, 74

px (pixels)
 changing to **em** value, 104, 188–190
 converting to percentages, 103, 176
 designing fixed layouts in, 167, 168
 designing with different values of, 199
 image dimensions in, 39
 recalculating line-height values of, 111
 setting size of, 102–103

Q

quotations
blockquote element around, 20–21
 citing sources of, 21–22
q element for, 20

R

radio buttons
 labels for, 227
 using on forms, 221

regions of, 169

relative paths, 35–36

relative positioning, 151–152

replaced elements, 140, 141

resizing images, 39

resources
 browser testing tools, 244
 CSS3, 253
 HTML5, 250–251
 inspirational websites, 239
 JavaScript, 256–257
 organizational, 238
 responsive layouts, 211
 Responsive Resources websites, 191
 screenshot tools, 240, 241
 style tile, 241
 website sketch paper, 242
 wireframe applications, 243
 workflow, 236

responsive layout, 180–191
 about responsive design, 142, 166
 converting fixed to, 182–191
 developing responsive navigation, 204–205, 209–210
 further reading on, 211
 illustrated, 190, 191

RGB color, 114–115

right alignment, 106, 107

role attribute, 90–93

rows of tables, 47, 51

Rubin, Dan, 222

S

- Safari, 167
- Salminen, Viljami, 236
- sans serif fonts, 98–99
- saving
 - documents, 7
 - documents before previewing, 33
- screen-reading software, 37, 39
- screenshots, 240–241
- scripts
 - about, 7
 - PHP, 217, 218, 228–229
- scrolling
 - adding scroll bar to box, 149–150
 - unable to scroll fixed elements, 154
- search boxes
 - examples of, 216
 - navigating with, 213
- search engines, 213
- section** element, 12–13
- selectors
 - adjacent sibling, 68–69
 - attribute, 94
 - child**, 68, 69
 - class**, 70–71, 89, 94
 - combinators, 67–70
 - descendant, 67, 69
 - element, 66
 - general sibling, 69, 70
 - grouping, 66
 - ID**, 71, 87, 88, 89
 - order of elements within, 88
 - placing in curly brackets, 57
 - pseudo-element, 77
 - types of, 65
 - universal, 65
 - using, 57–58
- semantic HTML, 25
- sequential navigation, 196
- serif fonts, 100
- server scripting languages, 218
- sitemaps, 238

- sizing
 - box model, 130–133
 - fonts, 102–104
 - images, 39
- sketching website ideas, 166, 174, 175, 242
- small caps, 105
- space-separated values, 122
- spacing between text lines, 110
- span** element, 28–29
- special characters
 - encoding for, 9
 - using in text, 11
- src** attribute, 37
- stacking order, 154–158
 - decreasing **z-index** value for, 158
 - default, 155, 156
 - illustrated, 142
 - increasing **z-index** value for, 157
 - normal, 155
 - using **z-index** for, 154–158
- static positioning, 151, 158
- strong** element, 26
- style sheets, 7
- style tiles, 241
- styling links
 - exercise in, 75
 - with hover effect, 72, 74–75
 - links with focus, 72, 77
 - unvisited links, 73
 - visited links, 72, 74
 - working with active link, 72, 76
- submit buttons, 223

T

- table of contents, 18
- tables, 46–53
 - body or rows within, 51
 - borders for, 48–49
 - captions for, 50
 - data cells for, 48
 - defining rows of, 47
 - footers for, 51–52
 - headings for, 46
 - ordered layouts for headers and data, 49
 - page layout using, 47
 - uses for, 46
 - using **table** element, 46
- table** element, 46
- tablets. *See also* mobile devices
 - resizing sites for, 190–191
 - sketching websites on, 242
 - touch screen, 201, 208, 210
- tags. *See also* **a** tag
 - adding attributes to, 82
 - deprecated, 49
 - role** attributes for, 90
- tbody** element
 - grouping rows in table body with, 51, 52
 - spanning columns with, 52–53
- td** element, 48, 49
- testing
 - mobile device design, 181–182
 - prototypes, 244–245
- text, 16–29, 98–115. *See also* fonts; lists
 - adding lines to, 107–109
 - aligning, 106–107
 - block quotes within, 20–21
 - bolding with **b** or **strong** elements, 26
 - citing source of quoted, 21–22
 - comments within, 28
 - creating website, 239
 - CSS property for bolding, 105
 - defining values for form entry, 220
 - description lists in, 24–25
 - designing form input areas, 224
 - emphasizing, 27
 - fixed vs. relative leading for, 111
 - font properties for, 98–106
 - hooks for HTML formatting, 28–29
 - indenting, 109
 - inline elements for, 19, 25

- italicizing, 27, 104
- line spacing for, 110
- lists in, 22–25
- outlining content, 16, 17
- small caps for, 105
- spacing, 110–111
- styling blocks of, 106–110
- tags for inserting, 11
- text properties, 106–110
- upper and lower case, 109–110
- using special characters in, 11
- text-align** property, 106–107
- text-decoration** property, 107
- text editors, 4–7
- text-indent** property, 109
- text-transform** property, 109
- tfoot** element
 - adding footers with, 51–52
 - including with spanned columns with, 52–53
- “Thank You” page, 228
- thead** element, 50, 52–53
- th** element
 - grouping with **thead**, 50
 - ordering **td** elements, 49
 - uses for, 46
- titles
 - adding for tables, 50
 - appearing on browser bar, 9
 - inserting, 8
 - using **title** attribute, 39
- tools
 - browser testing, 244
 - Firefox Responsive Design view, 185
 - screenshot software, 240, 241
 - Web Developer Toolbar, 181–182
- touch screens, 201, 208, 210
- transparent** color value, 126
- TRouBLE mnemonic, 122
- tr** element, 47
- type. *See* fonts

U

- UI (User Interface) jobs, 249
- uncollapsing containing box, 146–148, 173–174
- underline** value, 107, 108, 109
- universal (*) selector, 65
- unordered lists, 62–63, 195
- upload buttons, 223
- uppercasing text, 109
- URL paths, 35
- users
 - determining website's, 167
 - getting feedback on clicks of, 208
 - redirecting to “Thank You” page, 228
 - sending email to, 229
- UX (User Experience) jobs, 249

V

- values
 - avoiding cross-attribution of, 84
 - class** attributes with multiple, 84
 - creating borders without, 124
 - CSS, 57
 - generic font, 100, 101
 - hex, 113–114
 - multiple **class** attribute, 84
 - omitting font property, 106
 - overriding rules with multiple class, 86
 - space-separated, 122
- versions
 - CSS, 69, 252–253
 - HTML, 6
 - Internet Explorer, 144, 167, 179
- vertical navigation, 198–205
 - converting to horizontal, 207
 - highlighting current page on menu, 203
 - illustrated, 198
 - moving menu to right, 202
 - responsive, 204–205
 - when to use, 210

- video objects, 40–43
 - adding width and height for, 40–41
 - frameborder** and **allowfullscreen** attributes for, 41
- video tutorials, 43, 65, 158
- viewports
 - changing navigation layout for smaller, 204–205
 - scaling for mobile devices, 190–192
- :visited** pseudo-class, styling links with, 72, 74
- visual impairments. *See* accessibility
- visual language of website, 240–243

W

- W3C (World Wide Web Consortium), 6
- WAI-ARIA (Web Accessibility Initiative - Accessible Rich Internet Applications), 90
- Walton, Trent, 234, 236
- web browsers. *See also* fonts
 - adding metadata for, 8
 - appearance of title in, 9
 - caching pages in history, 73
 - clearfix method for older, 148
 - considering support for, 167
 - default line-height setting for, 110
 - defining paths to websites for, 35–36
 - fluid layouts in, 175–176
 - Internet Explorer support, 144, 167, 179
 - opening linked documents in, 33–34
 - position** and **z-index** default values for, 155
 - preventing CSS styles from applying to HTML, 182
 - pros and cons of fixed designs for, 167
 - rendering description lists, 25
 - resetting default margins and padding for, 134–135
 - Responsive Design view tools in Firefox, 185
 - returning to original website, 37
 - setting default **px** size, 102–103
 - support for general sibling selector, 71
 - testing tools for, 244
 - upload buttons for, 223
 - video support in, 40, 43
 - viewing box dimensions in, 181

web designers

- developers vs., 249
- finding jobs for, 248
- online job boards for, 238
- requirements for, 248–249
- UX and UI, 249
- workflow for, 168

Web Developer Toolbar, 181–182

websites. *See also* layout; navigation; web browsers

- accompanying book, 13
- adding CSS rules to fixed-width, 170–175
- audience for, 167
- Back button for, 37
- categorizing your, 239
- code for fixed layout, 169
- creating text for, 239
- defining content language for, 10
- designing navigation for, 194
- devices accessing, 166, 167
- highlighting current page link, 203
- interpreting content for visually impaired, 37, 39
- limited layout properties for, 141
- locating index documents on, 5, 32
- organizing visual language, 240–243
- prototyping, 244
- redirecting user to “Thank You” page, 228
- responsive design for, 142
- Responsive Resources, 191
- search boxes on, 216
- sending email from, 229
- setting paths to, 35–36
- sitemaps for, 238
- sketching ideas of, 166, 174, 175, 242
- storing pages in browser cache, 73
- structuring with **div** element, 170
- tutorials on book’s, 43, 65, 158
- validating code for, 93
- video objects for, 40–43
- when to link CSS documents to, 61
- wireframes of, 243

width

- assigning font, 100
- border, 123
- box model, 128–129, 132–133
- defining image, 39
- designing fluid rules for, 183–184
- increasing box’s, 172

width attribute, 39, 40–41

Windows computers

- best san serif fonts for, 99
- screenshot software for, 240, 241

wireframe mockups, 243

workflow, 234–245

- categorizing your website, 239
- coding phase, 235, 236, 244
- creating website text, 239
- designing phase, 235, 236
- developing content, 238–239
- launch phase, 235, 236
- linear vs. overlapping project phases, 235–236
- organizing visual language, 240–243
- planning phase, 235, 236
- questions organizing, 237
- responsive, 236
- sketching ideas, 242
- testing apps, 181–182, 244–245
- web designer’s, 168

World Wide Web Consortium (W3C), 6

X

XHTML4, 52

Y

YouTube, 40–43

Z**z-index** property, 154–158