

John Dean, PhD

Associate Professor of Computer Science
Park University
Parkville, Missouri

WEB

PROGRAMMING

**with HTML5, CSS,
and JavaScript**



JONES & BARTLETT
LEARNING



World Headquarters

Jones & Bartlett Learning
5 Wall Street
Burlington, MA 01803
978-443-5000
info@jblearning.com
www.jblearning.com

Jones & Bartlett Learning books and products are available through most bookstores and online booksellers. To contact Jones & Bartlett Learning directly, call 800-832-0034, fax 978-443-8000, or visit our website, www.jblearning.com.

Substantial discounts on bulk quantities of Jones & Bartlett Learning publications are available to corporations, professional associations, and other qualified organizations. For details and specific discount information, contact the special sales department at Jones & Bartlett Learning via the above contact information or send an email to specialsales@jblearning.com.

Copyright © 2019 by Jones & Bartlett Learning, LLC, an Ascend Learning Company

All rights reserved. No part of the material protected by this copyright may be reproduced or utilized in any form, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the copyright owner.

The content, statements, views, and opinions herein are the sole expression of the respective authors and not that of Jones & Bartlett Learning, LLC. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not constitute or imply its endorsement or recommendation by Jones & Bartlett Learning, LLC and such reference shall not be used for advertising or product endorsement purposes. All trademarks displayed are the trademarks of the parties noted herein. *Web Programming with HTML5, CSS, and JavaScript* is an independent publication and has not been authorized, sponsored, or otherwise approved by the owners of the trademarks or service marks referenced in this product.

There may be images in this book that feature models; these models do not necessarily endorse, represent, or participate in the activities represented in the images. Any screenshots in this product are for educational and instructive purposes only. Any individuals and scenarios featured in the case studies throughout this product may be real or fictitious, but are used for instructional purposes only.

09369-8

Production Credits

VP, Product Management: David D. Cella
Director of Product Management: Matthew Kane
Product Manager: Laura Pagluica
Product Assistant: Mary Menzemer
Production Editor: Vanessa Richards
Director of Marketing: Andrea DeFranzo
Product Fulfillment Manager: Wendy Kilborn
Composition: codeMantra U.S. LLC

Cover Design: Kristin E. Parker
Rights & Media Specialist: Thais Miller
Media Development Editor: Shannon Sheehan
Cover Image (Title Page, Chapter Opener): Blue abstract:
© StationaryTraveller/Getty Images; Web: © Supanat Chantira/EyeEm/
Getty Images
Printing and Binding: LSC Communications
Cover Printing: LSC Communications

Library of Congress Cataloging-in-Publication Data

Names: Dean, John, 1962- author.

Title: Web programming with HTML5, CSS, and JavaScript / John Dean, PhD,
associate professor of computer science, Park University-Parkville Campus,
Department of Computer Science and Information Systems.

Description: Burlington, Massachusetts : Jones & Bartlett Learning, 2017. |
Includes bibliographical references and index.

Identifiers: LCCN 2017023256 | ISBN 9781284091793 (pbk.)

Subjects: LCSH: Internet programming. | HTML (Document markup language) |
Cascading style sheets. | JavaScript (Computer program language)

Classification: LCC QA76.625 .D43 2017 | DDC 006.7/4--dc23

LC record available at <https://lccn.loc.gov/2017023256>

6048

Printed in the United States of America

22 21 20 19 18 10 9 8 7 6 5 4 3 2 1



DEDICATION

To my father, Ray Dean.



BRIEF TABLE OF CONTENTS

1	Introduction to Web Programming	1
2	Coding Standards, Block Elements, Text Elements, and Character References	33
3	Cascading Style Sheets	73
4	Organizing a Page's Content with Lists, Figures, and Various Organizational Elements	131
5	Tables and CSS Layout	167
6	Links and Images	217
7	Image Manipulations, Audio, and Video	269
8	Introduction to JavaScript: Functions, DOM, Forms, and Event Handlers	311
9	Additional JavaScript Basics: window Object, if Statement, Strings, Numbers, and Input Validation	351
10	Loops, Additional Controls, Manipulating CSS with JavaScript	425
11	Object-Oriented Programming and Arrays	499
12	Canvas	569
	Appendix A HTML5 and CSS Coding-Style Conventions	633
	Appendix B JavaScript Coding-Style Conventions	645
	Review Question Solutions	655

TABLE OF CONTENTS

Preface	xi
Acknowledgments	xviii
About the Author	xx

1 Introduction to Web Programming 1

1.1 Introduction	2
1.2 Creating a Website	3
1.3 Web Page Example	5
1.4 HTML Tags	7
1.5 Structural Elements	8
1.6 title Element	10
1.7 meta Element	11
1.8 HTML Attributes	11
1.9 body Elements: hr, p, br, div	13
1.10 Cascading Style Sheets Preview	15
1.11 History of HTML	16
1.12 HTML Governing Bodies	18
1.13 Differences Between Old HTML and HTML5	19
1.14 How to Check Your HTML Code	20
1.15 Case Study: History of Electric Power	23

2 Coding Standards, Block Elements, Text Elements, and Character References 33

2.1 Introduction	34
2.2 HTML Coding Conventions	35
2.3 Comments	36
2.4 HTML Elements Should Describe Web Page Content Accurately	37
2.5 Content Model Categories	38
2.6 Block Elements	40
2.7 blockquote Element	41
2.8 Whitespace Collapsing	44
2.9 pre Element	46
2.10 Phrasing Elements	46
2.11 Editing Elements	47
2.12 q and cite Elements	48

2.13	dfn, abbr, and time Elements	49
2.14	Code-Related Elements.	52
2.15	br and wbr Elements	55
2.16	sub, sup, s, mark, and small Elements	56
2.17	strong, em, b, u, and i Elements	58
2.18	span Element.	59
2.19	Character References.	60
2.20	Web Page with Character References and Phrasing Elements	63
2.21	Case Study: A Local Hydroelectric Power Plant.	66
3	Cascading Style Sheets (CSS)	73
3.1	Introduction	74
3.2	CSS Overview	74
3.3	CSS Rules	75
3.4	Example with Type Selectors and the Universal Selector	76
3.5	CSS Syntax and Style	77
3.6	Class Selectors	79
3.7	ID Selectors.	83
3.8	span and div Elements	84
3.9	Cascading	86
3.10	style Attribute, style Container	87
3.11	External CSS Files.	88
3.12	CSS Properties	91
3.13	Color Properties	92
3.14	RGB Values for Color	93
3.15	Opacity Values for Color	96
3.16	HSL and HSLA Values for Color	98
3.17	Font Properties.	100
3.18	line-height Property	107
3.19	Text Properties	108
3.20	Border Properties.	110
3.21	Element Box, padding Property, margin Property.	113
3.22	Case Study: Description of a Small City's Core Area.	118
4	Organizing a Page's Content with Lists, Figures, and Various Organizational Elements	131
4.1	Introduction	132
4.2	Unordered Lists	132
4.3	Descendant Selectors	137
4.4	Ordered Lists	138
4.5	Figures.	141
4.6	Organizational Elements.	145
4.7	section, article, and aside Elements.	147
4.8	nav and a Elements.	150

4.9	header and footer Elements	150
4.10	Child Selectors	152
4.11	CSS Inheritance	154
4.12	Case Study: Microgrid Possibilities in a Small City	155
5	Tables and CSS Layout	167
5.1	Introduction	168
5.2	Table Elements	170
5.3	Formatting a Data Table: Borders, Alignment, and Padding	172
5.4	CSS Structural Pseudo-Class Selectors	174
5.5	thead and tbody Elements	178
5.6	Cell Spanning	182
5.7	Web Accessibility	185
5.8	CSS display Property with Table Values	188
5.9	Absolute Positioning with CSS Position Properties	194
5.10	Relative Positioning	199
5.11	Case Study: A Downtown Store's Electrical Generation and Consumption	203
6	Links and Images	217
6.1	Introduction	218
6.2	a Element	218
6.3	Relative URLs	221
6.4	index.html File	224
6.5	Web Design	225
6.6	Navigation Within a Web Page	228
6.7	CSS for Links	235
6.8	a Element Additional Details	237
6.9	Bitmap Image Formats: GIF, JPEG, PNG	239
6.10	img Element	244
6.11	Vector Graphics	245
6.12	Responsive Images	250
6.13	Case Study: Local Energy and Home Page with Website Navigation	255
7	Image Manipulations, Audio, and Video	269
7.1	Introduction	270
7.2	Positioning Images	270
7.3	Shortcut Icon	272
7.4	iframe Element	274
7.5	CSS Image Sprites	279
7.6	Audio	282
7.7	Background Images	286
7.8	Web Fonts	290
7.9	Video	291
7.10	Centering Content Within the Viewport, Color Gradients	294

7.11	Case Study: Using an Image Map for a Small City's Core Area and Website Navigation with a Generic Home Page	297
8	Introduction to JavaScript: Functions, DOM, Forms, and Event Handlers	311
8.1	Introduction	312
8.2	History of JavaScript	313
8.3	Hello World Web Page	314
8.4	Buttons	315
8.5	Functions	316
8.6	Variables	318
8.7	Identifiers	319
8.8	Assignment Statements and Objects	319
8.9	Document Object Model	322
8.10	Forms and How They're Processed: Client-Side Versus Server-Side	323
8.11	form Element	326
8.12	Controls	327
8.13	Text Control	329
8.14	Email Address Generator Web Page	330
8.15	Accessing a Form's Control Values	332
8.16	reset and focus Methods	335
8.17	Comments and Coding Conventions	335
8.18	Event-Handler Attributes	338
8.19	onchange, onmouseover, onmouseout	339
8.20	Using noscript to Accommodate Disabled JavaScript	342
9	Additional JavaScript Basics: window Object, if Statement, Strings, Numbers, and Input Validation	351
9.1	Introduction	352
9.2	window Object	353
9.3	alert and confirm Methods	358
9.4	if Statement: if by itself	361
9.5	Game Night Web Page	363
9.6	prompt Method	365
9.7	Game Night Web Page Revisited	366
9.8	if Statement: else and else if Clauses	367
9.9	Strings	369
9.10	Word Ordering Web Page	375
9.11	More String Details	379
9.12	Arithmetic Operators	381
9.13	Math Object Methods	387
9.14	Parsing Numbers: parseInt, parseFloat	388
9.15	Water Balloons Web Page	389
9.16	Constraint Validation for Form Controls	396

9.17	Constraint Validation Using the Number Control's Attributes	397
9.18	Constraint Validation Using CSS Pseudo-Classes	400
9.19	Comparison Operators and Logical Operators.	401
9.20	JavaScript for the Improved Water Balloons Web Page.	406
9.21	Case Study: Dynamic Positioning and Collector Performance Web Page	410

10 Loops, Additional Controls, Manipulating CSS with JavaScript 425

10.1	Introduction	426
10.2	while Loop	426
10.3	External JavaScript Files	429
10.4	Compound Interest Web Page	430
10.5	do Loop.	437
10.6	Radio Buttons	442
10.7	Checkboxes	446
10.8	Job Skills Web Page.	449
10.9	for Loop.	452
10.10	fieldset and legend Elements	455
10.11	Manipulating CSS with JavaScript	456
10.12	Using z-index to Stack Elements on Top of Each Other	461
10.13	Textarea Controls	466
10.14	Dormitory Blog Web Page	469
10.15	Pull-Down Menus	476
10.16	List Boxes	479
10.17	Case Study: Collector Performance Details and Nonredundant Website Navigation	483

11 Object-Oriented Programming and Arrays 499

11.1	Introduction	500
11.2	Object-Oriented Programming Overview	500
11.3	Classes, Constructors, Properties, new Operator, Methods.	502
11.4	Point Tracker Web Page	505
11.5	static Methods.	509
11.6	Event Handlers	513
11.7	Primitive Values Versus Objects	517
11.8	Using addEventListener to Add Event Listeners	518
11.9	Using Prototypes to Emulate a Class	520
11.10	Inheritance Between Classes.	524
11.11	Pet Registry Web Page	527
11.12	switch Statement.	536
11.13	Arrays.	538
11.14	Arrays of Objects	542
11.15	Book Club Web Page.	545
11.16	Case Study: Downtown Properties Data Processing.	554

12 Canvas	569
12.1 Introduction	570
12.2 Canvas Syntax Basics	571
12.3 Rectangles Web Page	572
12.4 Drawing Text with <code>fillText</code> and <code>strokeText</code>	577
12.5 Formatting Text	579
12.6 Drawing Arcs and Circles	582
12.7 Drawing Lines and Paths	585
12.8 Umbrella Web Page	590
12.9 Face Web Page	595
12.10 Using Canvas for Transformations	597
12.11 Moving Face Web Page	605
12.12 Case Study: Solar Shadowing Dynamics	610
Appendix A HTML5 and CSS Coding-Style Conventions	633
Appendix B JavaScript Coding-Style Conventions	645
Review Question Solutions	655
Index	671



PREFACE

Since HTML's introduction in 1993, web-programming technologies have been in flux, with web programmers using different versions of HTML for different browsers. The constant change made it difficult for authors to write quality textbooks about the subject. Consequently, most of the books were trade books, not textbooks. With HTML5's approval as a "stable recommendation" in 2014, web programmers and browsers appear to have embraced it fully. With the huge demand for web programmers in the workforce, there has been a significant demand for web-programming courses for quite a while. Now that web programming has coalesced around HTML5, there is a need for better textbooks about web programming.

Web programming is a large field, with different types of web programming implemented by different tools. All the tools work with the core language, HTML, so almost all the web-programming books describe HTML to some extent. This textbook covers HTML5, CSS, and JavaScript, all in depth. Those three technologies are known to be the pillars of client-side web programming. With client-side web programming, all web page calculations are performed on end users' computers (the client computers). There's also server-side web programming, which uses technologies such as ASP.NET, JSP, and PHP. With server-side web programming, most of the web page calculations are performed on the computers that host the web pages permanently (the server computers). Many books attempt to cover one of the server-side technologies; in doing so, they necessarily have to cover some HTML and CSS as well, because all web pages need those technologies to display results on the client computer. Typically, such books try to cram in too much for beginning web programmers to digest. Many books go to the opposite extreme and cover only HTML and CSS, leaving out JavaScript, which is essential for calculations. This book hits a sweet spot—covering all three client-side technologies in depth (HTML, CSS, and JavaScript), with no dependence on server-side technologies. After finishing this book, you should be prepared to write nicely formatted, interactive web pages that are able to perform calculations and show the results. And down the road if you decide that you want to write server-side web pages, your solid client-side foundation should properly prepare you to go forth and learn a server-side technology(ies).

Target Audience

Unlike many client-side web programming books, this book presents not only HTML and CSS, but also JavaScript, the document object model (DOM), and canvas. With that programming depth, the book works well for sophomore and junior Computer Science majors who have programming experience. With the first seven chapters devoted to HTML and CSS and a gentle introduction to programming concepts in the JavaScript chapters, the textbook can also work well for non-Computer Science majors with no prerequisite programming knowledge.

In addition to targeting students in a college setting, this textbook targets high school students with or without programming experience. Most likely, in a high school setting, this book's content would be covered in a year-long course. Finally, this book targets industry practitioners who want to learn client-side web programming. Industry practitioners should read the entire textbook at a pace determined on a case-by-case basis.

Approach

Some web-programming books try to present exhaustive content without trying to explain it fully. That can lead to readers with poor understanding and weak retention of the content. In this book, we carefully consider the proper amount of content to present in each chapter, so that there is ample opportunity for explanations and learning. We try to engage readers by using a fun, conversational tone. Readers who are engaged are more likely to retain what they're reading. Readers who are having fun are more likely to dig deep and yearn to learn more.

Although we take the time to carefully explain HTML, CSS, and JavaScript syntax, we realize how important it is to develop the creativity and problem-solving skills necessary to become adept at web programming. With that in mind, we present the content in a manner that supports such development. Several of the leading web-programming textbooks embed much of their content within tutorials. Each tutorial provides step-by-step instructions that explain how to paste together code fragments to form a web page. That technique helps readers feel good about producing results, but when that's all there is, not much actual learning takes place. Because there's not much of a framework for organizing the content, that presentation strategy makes it difficult for readers to integrate new material into what they already know and to remember what they've learned. On the other hand, this book presents content with an optimal organization for learning. Within each chapter, we explain concepts, present problems, and solve those problems with short code fragments and also complete web pages.

At the end of each chapter, we present optional case study sections that build upon each other to create an increasingly sophisticated website. In the case studies, we provide guidance and a discussion of design decisions, rather than step-by-step instructions. This forces readers to apply what they've learned earlier in the chapter, and it gives readers a feel for the real-world design process. Perhaps most importantly, the case studies provide practice for readers in fostering their creativity and problem-solving skills.

Proper Flow

In conjunction with our careful consideration of what's covered in each chapter, we carefully consider the flow between content topics. We make sure that the order in which we present content and the transitions between concepts follow a natural progression that is conducive to learning and retention. For example, in the first three chapters, we present basic HTML concepts and then a thorough, but not exhaustive, explanation of CSS. This gives the reader an opportunity to appreciate the richness of HTML and CSS without feeling overwhelmed by details.

Other web programming textbooks present form controls (e.g., text boxes, pull-down menus, and buttons) without showing how to process the controls' input data. Without processing, the forms are impractical and readers are left to wonder, "What's the point?" The reason they don't show how to process the controls' input data is because such processing requires JavaScript, and they don't cover JavaScript until later or don't cover it at all. The solution is a no-brainer—cover JavaScript before form controls. That's what we do in this book, so when we introduce form controls, we're able to use JavaScript to process the input data. Processing the input data means we can calculate and display results, and that leads to more satisfied readers.

Real-World Context

More often than not, today's classroom students and industry practitioners prefer to learn with a hands-on, real-world approach. The following subsections illustrate that approach.

Complete Web Page Examples

Different books have different strategies for using code to illustrate new concepts. Most books provide short code fragments to illustrate new concepts, and that can be very helpful. But when books rely almost exclusively on code fragments and provide few complete web pages, readers tend to complain about their inability to run the code and their inability to get a feel for the big picture. In this book, we strike an appropriate balance between short code fragments and complete web pages so readers are able to grasp new concepts quickly (with short code fragments) and apply those concepts in the context of complete web pages.

Industry-Standard Web-Programming Software

The term "HTML5 standard" is a loose term in that it can refer to any of the different HTML5 versions. At the time of this book's writing, HTML 5.1 was the latest official "recommendation," so we present syntax and semantics from HTML 5.1.¹ For CSS and JavaScript, we present syntax and semantics for their latest versions as well.

In choosing to describe the latest versions of client-side web-programming software, we are cognizant of the risk of presenting content that industry hasn't caught up with yet. Thus, we present constructs only if they are supported by at least two of the three most popular web browsers—Google Chrome, Mozilla Firefox, and Microsoft Edge. But even more importantly, we present constructs only if they are part of the standards put forth by the World Wide Web Consortium (W3C), the Web Hypertext Application Technology Working Group (WHATWG), and Ecma International. The W3C and the WHATWG are the standards organizations for HTML5, CSS, and the DOM. Ecma International is the standards organization for JavaScript.

¹ At the time of this book's writing, HTML 5.2 was a "working draft." It's a superset of HTML 5.1, and it doesn't appear to add all that many new constructs.

Industry-Standard Coding-Style Conventions

We follow Google's recommended coding-style conventions consistently throughout the book. In the book's appendices, we provide complete references for the book's HTML, CSS, and JavaScript coding-style conventions, which are based on Google's coding conventions. To supplement those references and to emphasize how important good style is, we include coding-style tips whenever we use new coding conventions for code that we're explaining.

Tutorials for Software Tools

We do not tie the textbook to any particular software tools. To develop and publish web pages, readers are free to use any tools they like. But to make learning easier, on the book's website, we provide tutorials for several popular tools: Visual Studio integrated development environment (IDE) for entering and testing web page code, WinSCP for uploading web pages to a web server, and Chrome's debugger for JavaScript debugging. Please visit go.jblearning.com/webprogramming to access these tutorials.

Homework Problems

At the end of each chapter, we provide three types of homework problems—review questions, exercises, and projects. Review questions serve as a review of what was covered earlier in the chapter. Most review questions use a short-answer format, but there are multiple-choice, true/false, and fill-in-the-blank questions as well. You can find solutions for the review questions at the end of the book.

Exercises are a bit more challenging than the review questions. They require readers to do more than just recall what they've read; they require readers to apply what they've learned. The exercises use short-answer, debugging, and write-a-code-fragment formats. Qualified instructors can access the exercise solutions by visiting go.jblearning.com/webprogramming.

Projects consist of problem descriptions whose solutions are complete web pages. Like the exercises, the projects require students to apply what they've learned in the chapter. But additionally, they require readers to employ design, creativity, and problem-solving skills in order to go from a description to a complete web page. Qualified instructors can access the project solutions by visiting go.jblearning.com/webprogramming.

Organization

There are three conceptual components of a web page: content, presentation, and behavior. We introduce content code, in the form of HTML elements, throughout the book, but the majority of such content code appears in the first 60% of the book. We introduce most of the presentation code, in the form of CSS, near the beginning of the book, but we sprinkle in additional CSS code as necessary later on. We introduce the behavior code, in the form of JavaScript, in the last 40% of the book. Throughout the entire book, while describing content, presentation, and behavior

details, we put those details into practice by building web pages that nurture the development of problem-solving skills and web-design skills.

We start with two chapters that are introductory in nature. They discuss general concepts involving the Web and basic HTML elements that enable readers to get a taste of what's possible. The book then devotes a whole chapter to CSS. Although the chapter describes CSS extensively, it does not attempt to cover CSS exhaustively. Later chapters introduce additional CSS details when appropriate. After the CSS chapter, the next four chapters present HTML element details and web-programming design principles. The last five chapters describe JavaScript basics, plus some advanced JavaScript constructs that unleash the power of dynamic HTML.

Chapters 1 and 2

In Chapters 1 and 2, we first explain basic concepts needed for building a website. For example, we describe text editors, web-hosting services, and browsers. We then narrow the focus and describe the basic components of a web page: HTML elements, tags, and attributes. We put those components into practice by examining a simple, but complete, web page. Chapter 1 concludes with a brief history of HTML.

Chapter 2 continues the theme of introducing material that is somewhat general in nature. For example, we describe the W3C, coding conventions, and comments. After that, the chapter provides details on quite a few HTML elements: block elements, editing elements, phrasing elements, and so on. The chapter concludes with a discussion of character references.

Chapter 3

In Chapter 3, we first present an overview of CSS concepts such as syntax, the different types of CSS rules, the different places where CSS rules can be applied, and how cascading determines the rules' priorities. The rest of the chapter provides details about many CSS properties and values.

Chapters 4 and 5

In these two chapters, we return to HTML elements, with an in-depth look at elements in charge of organizing a page's content (Chapter 4) and a comprehensive treatment of data tables and layout tables (Chapter 5). Chapter 4 focuses on lists, figures, and section-oriented elements, and it weaves in a few CSS details, such as descendent and child selectors, that were not part of Chapter 3's CSS coverage. Chapter 5 focuses on the `table` element and all of its associated elements: `caption`, `tr`, `td`, and so on. As an alternative to using the `table` element, Chapter 5 describes how to achieve table layout with CSS. Specifically, Chapter 5 describes CSS table values, absolute positioning, and relative positioning.

Chapters 6 and 7

In Chapters 6 and 7, we present what might be considered the seminal features of HTML—links and the ability to embed nontext objects (images, audio, and video) into a web page.

Chapter 6 describes how to implement a link and then discusses navigation techniques that rely on links. Chapter 6 then describes how to navigate to pictures, and presents various details about the `img` element, bitmap image formats, the SVG image format, and vector graphics.

Chapter 7 continues the discussion of images by describing how to manipulate them with positioning, image sprites, image maps, and so on. Chapter 7 then moves on to a discussion of other embedded objects—audio files and video files.

Chapters 8 and 9

In Chapters 8 and 9, we present the basics of JavaScript. In building a JavaScript web page, there's a lot to learn. Some books present a huge amount of syntax up front, leaving readers feeling overwhelmed and unable to create JavaScript web pages on their own. Other books introduce a little JavaScript syntax and illustrate what's going on with small, gimmicky web pages that do not comport with standard industry practices. This book strikes a balance between those two strategies. In Chapters 8 and 9, we present enough syntax to build standard-practice JavaScript web pages, but not so much syntax that readers feel overwhelmed. Here's a sample of the key JavaScript constructs introduced in Chapters 8 and 9: functions, variables, the Document Object Model (DOM), event handlers, `if` statement, strings, operators, and input validation.

In addition to introducing JavaScript constructs, we introduce HTML elements for forms, buttons, and text boxes. Those elements implement input/output functionality, and they enable the web pages to be more “real world.”

Chapters 10, 11, and 12

In Chapters 10 through 12, we introduce JavaScript constructs that are slightly more advanced. Chapter 10 presents `while` loops and `for` loops and puts them into practice by looping through form controls like radio buttons, checkboxes, and pull-down menu selections. These form controls could have been introduced earlier with the button and text box form controls, but they fit better in Chapter 10, when loops are introduced. Toward the end of Chapter 10, we describe how to manipulate CSS by using JavaScript and various properties in the DOM.

In Chapter 11, we introduce the concept of object-oriented programming (OOP) and describe how to work with objects in JavaScript. This includes how to implement classes with constructors, properties, and methods; how to instantiate objects; and how to implement inheritance between classes. In JavaScript, arrays are objects, so we wait until after the introduction of OOP to discuss arrays. In our description of arrays, we explain how to instantiate them, how to loop through an array's elements, and how to use an array's methods.

In Chapter 12, we introduce canvas, which is considered to be one of the most important new features of HTML5. We describe how to use the `canvas` element and JavaScript to draw rectangles, circles, arcs, lines, paths, and text, all within a graphical “canvas” drawing area. After discussing how to draw those graphics objects, the chapter describes how to translate, rotate, and scale them.

Student Resources

At the book's website, go.jblearning.com/webprogramming, students can view and download these resources:

- ▶ Student-version lecture slides in PowerPoint format without hidden notes.
 - The student-version slides are identical to the teacher-version slides except that the hidden notes and hidden slides are omitted.
 - Omitting the hidden slides incentivizes students to attend class, where the teacher provides additional content.
- ▶ Source code and resource files for all the textbook example web pages
- ▶ Tutorials for web-developer software tools

Instructor Resources

At the book's website, go.jblearning.com/webprogramming, instructors can view and download these resources:

- ▶ Teacher-version lecture slides in PowerPoint format with hidden notes
 - Hidden notes provide comments that supplement the displayed text in the lecture slides.
 - For example, if the displayed text asks a question, the hidden notes provide the answer.
- ▶ Exercise solutions
- ▶ Project solutions
- ▶ Test bank questions
- ▶ Additional projects

ACKNOWLEDGMENTS

Writing this textbook has been a labor of love, albeit an arduous 3-years-plus labor of love. I am indebted to my team at Jones & Bartlett Learning for their generous commitment to this project. Without them, the content may have been awesome, but no one would have known about it because there would be no book. Our fearless leader, Product Manager Laura Pagluica, kept everyone on track, especially me. She provided sagacious advice in lots of different areas, and on occasion, she (rightfully) reigned in my (sometimes excessive) exuberance. Taylor Maurice gathered and compiled the early-chapter reviews, and Mary Menzemer did the same for the later-chapter reviews. Their work helped greatly in my effort to incorporate reviewer feedback into the book. I'd also like to thank Thais Miller and Vanessa Richards, who did a great job during the production phase.

I'd like to acknowledge the many teachers who provided feedback during the writing phase. Their comments showed attention to detail and tremendous insight, and they led to many improvements in wording and content. So thank you to:

Simon Baev

Georgia Southwestern State University

John Beatty

La Salle University

Bill Bennett

Mt. San Jacinto College

Noni McCullough Bohonak

University of South Carolina Lancaster

Dan Brandon

Christian Brothers University

Blase B. Cindric

University of Mount Union

Stephen Crandall

Notre Dame College

Julius Dichter

University of Bridgeport

Zarreen Farooqi

University of Akron

Kelly Furnas

Kansas State University

Steven Gutierrez

Lake Erie College

William J. Hitchcock

Loras College

Nanette Hogg

University of Nebraska at Kearney

Brian W. Horton

The University of Texas at Arlington

Zhenyu Huang

Central Michigan University

Deborah J. Hwang

University of Evansville

Clara James
Minneapolis Community & Technical College

Ravinder Kang
Highline Community College

Amitava Karmaker
University of Wisconsin–Stout

Gilliean Lee
Lander University

Haim Levkowitz
University of Massachusetts Lowell

Ted Markowitz
University of New Haven

François Neville
Bemidji State University

Ralph Phillips
Central Oregon Community College

Susan Reeder
Seattle University

Jamil Saquer
Missouri State University

Ethel Schuster
Northern Essex Community College

Derrf Seitz
Georgia Military College

Robert Sfarzo
Cuesta College

R. Duane Skaggs
Morehead State University

Bob Sweeney
University of South Alabama

Joo Tan
Kutztown University of Pennsylvania

Visa Thiangarajan
Magnolia Science Academy

Joe Triplett
Ohio University

Marilyn Turmelle
School for the Talented and Gifted

Michael Van Hilst
Nova Southeastern University

Christopher T. VanOosterhout
Muskegon Community College

Sam Vegter
Western Piedmont Community College

Jerry Westfall
Liberty University

My web-programming students have not been shy about making suggestions and finding errors. In particular, thank you Olivia Leung, Matthew Prybyszczuk, Jessica Detweiler, Alexis Fry, Chrisman Miller, and Yves Akanza.



ABOUT THE AUTHOR

John Dean is an Associate Professor and the Department Chair of the Department of Computer Science and Information Systems at Park University. He earned a PhD degree in computer science from Nova Southeastern University and an MS degree in computer science from the University of Kansas. Dean has worked in industry as a software engineer and project manager, specializing in Java and various web technologies—HTML, CSS, JavaScript, JavaServer Pages, and servlets. He has taught a full range of computer science courses, including client-side and server-side web-programming courses.