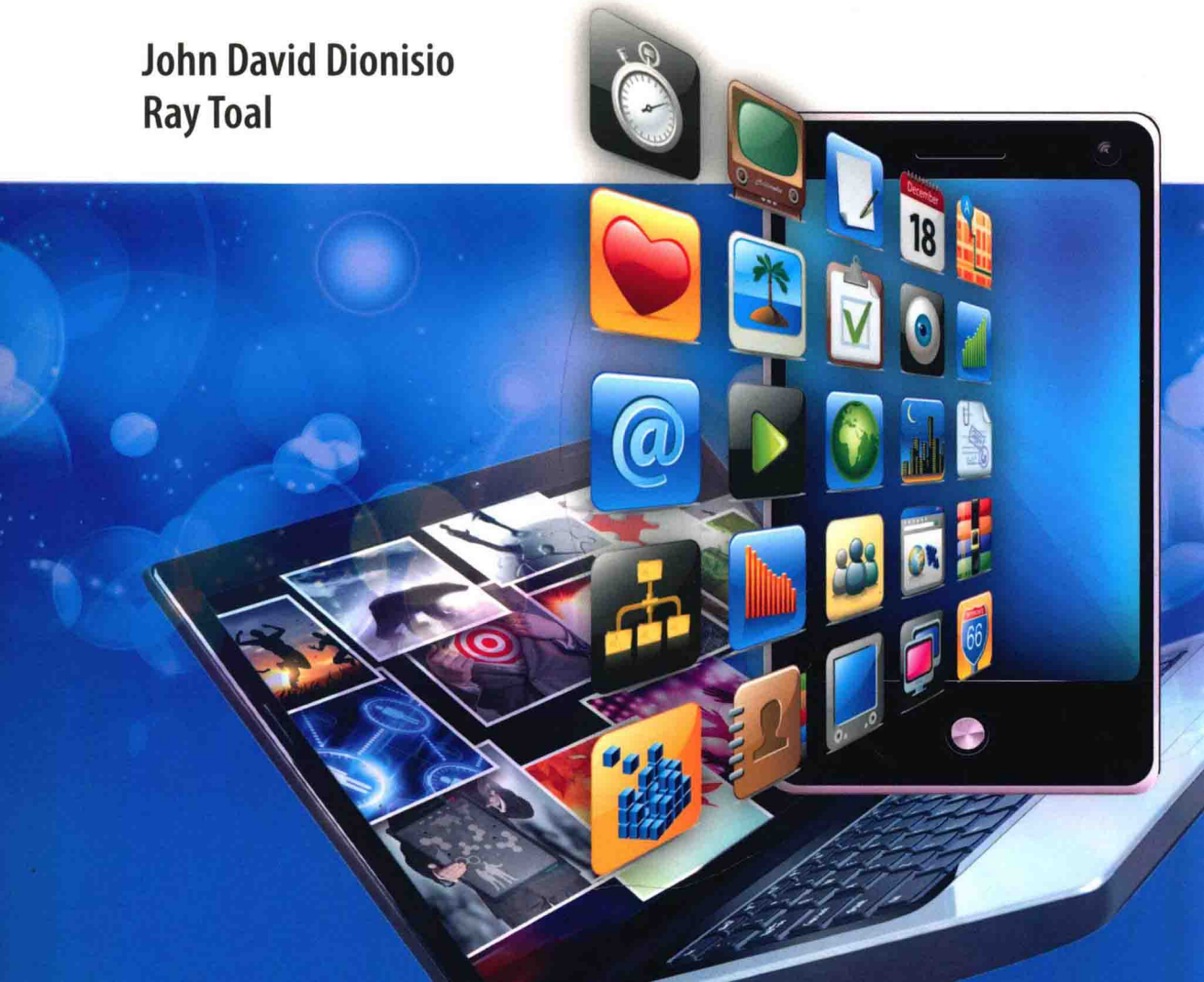
PROGRAMMING WITH

JavaScript

Algorithms and Applications for Desktop and Mobile Browsers

John David Dionisio
Ray Toal



PROGRAMMING WITH

JavaScript

Algorithms and Applications for Desktop and Mobile Browsers

John David Dionisio
Loyola Marymount University

Ray Toal
Loyola Marymount University

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.

Production Credits
Publisher: Cathleen Sether
Senior Acquisitions Editor: Timothy Anderson
Managing Editor: Amy Bloom
Director of Production: Amy Rose
Marketing Manager: Lindsay White
V.P., Manufacturing and Inventory Control: Therese Connell
Associate Photo Researcher: Lauren Miller
Composition: Northeast Compositors, Inc.
Cover Design: Kristin E. Parker
Cover Image: Light: © yienkeat/ShutterStock, Inc.; Laptop: © Haywiremedia/ShutterStock, Inc.
Photo display: © James Thew/ShutterStock, Inc.; SmartPhone: © lassedesignen/Fotolia.com
App Icons: © abdulsatarid/ShutterStock, Inc.
Printing and Binding: Courier Kendallville
Cover Printing: Courier Kendallville

Library of Congress Cataloging-in-Publication Data
Dionisio, John David N., 1970-
Programming with JavaScript : algorithms and applications for desktop and mobile browsers / John David Dionisio, Ray Toal.
p. cm.
Includes bibliographical references and index.
ISBN-13: 978-0-7637-8060-9 (pbk.)
ISBN-10: 0-7637-8060-X (pbk.)
1. JavaScript (Computer program language) 2. Computer algorithms. 3. Application software—Development. I. Toal, Ray. II. Title.
QA76.73.J38D57 2013
005.3—dc23
2011018738

6048

Printed in the United States of America
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

Dedication

Love and thanks as always to Mei Lyn, Aidan, Anton, and Aila for their support.

JDND

Preface

What comes to mind when you hear the terms *programming* and *computer science*? Game-playing, socially challenged geeks? Computers? Those are certainly the popular images. In reality, however, *anyone* can program,[1] and computer science is about much more than computers. You are just as likely to see people programming phones, robots, navigation systems, and factory machinery as you are desktop computers.

Programming with JavaScript: Algorithms and Applications for Desktop and Mobile Browsers is an introduction to some of the main ideas and principles of computer science, with some forays into the related disciplines of software engineering and information technology. It aims to convey these principles *by encouraging you to develop fundamental skills in programming.* Computer science deals with many things—computation, algorithms, software systems, data organization, knowledge representation, language, intelligence, and learning—but it is programming experience that enables you to gain a better understanding of these topics, and the tools to explore them in depth.

Objectives

This book aims to:

- Introduce the field of computing by showing that it is a natural science, encompassing computer science, software engineering, computer engineering, information systems, and information technology.

[1] "Anyone can" means that great programmers can, and do, come from any background, not that programming can be learned without effort [Bra07].

- Dispel common myths about what computing is and show that computing provides a foundation for careers in many different areas, including medicine, law, business, finance, entertainment, the arts, education, economics, biology, nanotechnology, and gaming.
- Teach a respect for programming aesthetics, standards, style conventions, and judicious commenting early in the text, with the goal of preventing common bad habits from ever forming.
- Convey the power of JavaScript (as compared to other languages) by covering difficult material that is traditionally not taught to beginners. Some of this advanced material is isolated into sections marked with an asterisk (*) or is included in the appendices.
- Show that programming is not just about getting programs to work correctly, but is also about constructing programs that are readable, easily modifiable, and that run efficiently.
- Provide relevant case studies in distributed computing, touch-based user interfaces on phones and tablets, and graphics for both the student looking forward to employment and the professional programmer looking to keep current in modern software technology.

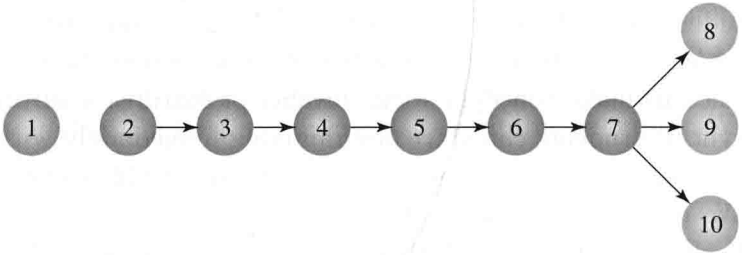
Organization

We've structured this text so that you can read it cover to cover if you like. It tells a story about computing, programming, and especially JavaScript, outlined as follows:

- Introduction to the field of computing (*Chapter 1*)
- Theory and practice of (JavaScript) programming (*Chapters 2–8*)
  - Getting started with programming (*Chapter 2*)
  - Data (*Chapter 3*)
  - Programming in the small I: Statements (*Chapter 4*)

  - Programming in the small II: Functions (*Chapter 5*)
  - Programming in the small III: Events (*Chapter 6*)
  - Programming in the large I: Software systems construction (*Chapter 7*)
  - Programming in the large II: Distributed computing (*Chapter 8*)
- Advanced topics (*Chapters 9–10*)

While you need not read the text exactly cover to cover, you may want to keep the chapter dependencies in mind, shown here:



Note that Chapter 1 stands alone: it's optional. Readers who want to jump right in to programming can start with Chapter 2.

Audience

This text is designed as a primary resource for a first-year college course in computer science or software engineering. No previous programming experience is assumed. However, advanced students and professional programmers new to JavaScript should also find the text useful, as we do not shy away from technical areas of the language perceived as difficult or "advanced." In fact, we believe that professional programmers can benefit a great deal from the numerous review questions and exercises spread throughout the text, as well as our coverage of modern topics in the JavaScript world, including ECMAScript 5, HTML 5, Ajax, jQuery, Graphics, and Animation.

JavaScript

A note to instructors: We enthusiastically adopt JavaScript as the language with which to train new computer scientists. JavaScript has not traditionally found much traction in introductory university-level computer science courses; this is probably due to various misunderstandings about the language [Cro01]. We argue, however, that JavaScript is an *ideal* language for such courses.

First, thanks to the ubiquity of web browsers, every student already has access to a JavaScript interpreter; no download or installation is required. Second, the language finds middle ground in the debate between professors who claim that beginning students should focus not on programming but on abstract algorithms given in pseudocode, and those who argue that students require hands-on programming experience to make concepts stick. JavaScript features a surprisingly clear and simple syntax; students can start programming immediately without fretting about classes, "public static" methods, the mysterious `void`, consoles, packages, and so on. We realize many schools have tried the simple-language approach in CS1 with ML, Scheme, Ruby, or Python, but with the rise of the Web as a platform for running applications (both on desktop and mobile devices), none of these languages can boast nearly the same degree of popularity as JavaScript.

Finally, as functional programming, long thought of as being of interest only to academic computer scientists, becomes more important in the new world of multicore processors and Big Data, JavaScript as a teaching language makes a great deal of sense. Functional programming in JavaScript tends to be fairly accessible to beginning students, perhaps more so than languages known for having "too many parentheses" or a reliance on special constructs like blocks, continuations, or generators.

Additional Resources

Visit go.jblearning.com/Dionisio for answers to end-of-chapter exercises, source code, PowerPoint Lecture Outlines, errata, and additional bonus material outside the scope of this text.

Acknowledgments

We'd like to express our thanks to Loren Abrams, Turn Media; B. J. Johnson, Claremont Graduate University; Philip Dorin, Loyola Marymount University; Daniel Bogaard, Rochester Institute of Technology; Michael Hennessy, University of Oregon; and Laurence Toal, Wellesley College, for their careful readings of early drafts and many constructive comments. Thanks also to Kira Toal and Masao Kitamura for providing several images, and to Jasmine Dahilig, Tyler Nichols, and Andrew Fornery for their assistance in preparing ancillary materials. We are also grateful for the excellent support from the staff at Jones & Bartlett Learning, including Tim Anderson, Senior Acquisitions Editor; Amy Bloom, Managing Editor; and Amy Rose, Production Director, without whose professionalism and hard work this book would not have been possible. We'd also like to thank Caskey Dickson and Technocage, Inc., for hosting our cross-site scripting examples. Without them, there would be no sites to cross!

Contents