JAVASCRIPT®
& JQUERY

interactive front-end
web development

JON DUCKETT

JAVASCRIPT & JQUERY

Interactive Front-End Web Development

JON DUCKETT

Additional material by:
GILLES RUPPERT & JACK MOORE

WILEY

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at http://booksupport.wiley.com. For more information about Wiley products, visit www.wiley.com.

TABLE OF CONTENTS

CODE
DOWNLOAD

Try out & download the code in this book

www.javascriptbook.com

CREDITS

INTRODUCTION

This book explains how JavaScript can be used in browsers to make websites more interactive, interesting, and user-friendly. You will also learn about jQuery because it makes writing JavaScript a lot easier.

To get the most out of this book, you will need to know how to build web pages using HTML and CSS. Beyond that, no prior experience with programming is necessary. Learning to program with JavaScript involves:

1

Understanding some **basic programming concepts** and the terms that JavaScript programmers use to describe them.

2

Learning **the language** itself, and, like all languages, you need to know its vocabulary and how to structure your sentences.

3

Becoming familiar with **how it is applied** by looking at examples of how JavaScript is commonly used in websites today.

The only equipment you need to use this book are a computer with a modern web browser installed, and your favorite code editor, (e.g., Notepad, TextEdit, Sublime Text, or Coda).
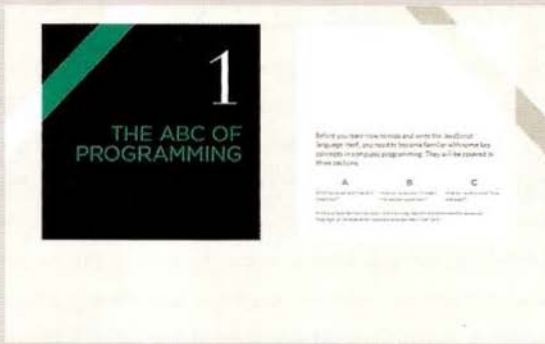
Introduction pages come at the beginning of each chapter. They introduce the key topics you will learn about.

Reference pages introduce key pieces of JavaScript. HTML code is shown in blue, CSS code in pink, and JavaScript in green.
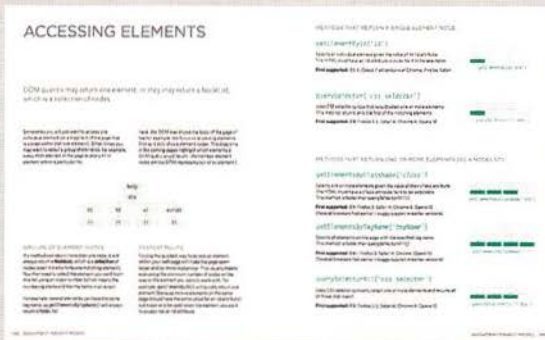
Background pages appear on white. They explain the context of the topics covered that are discussed in each chapter.
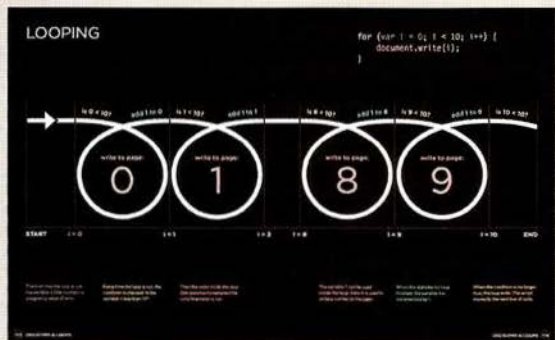
Diagram and infographics pages are shown on a dark background. They provide a simple, visual reference to topics discussed.

Example pages bring together the topics you have learned in that chapter and demonstrate how they can be applied.
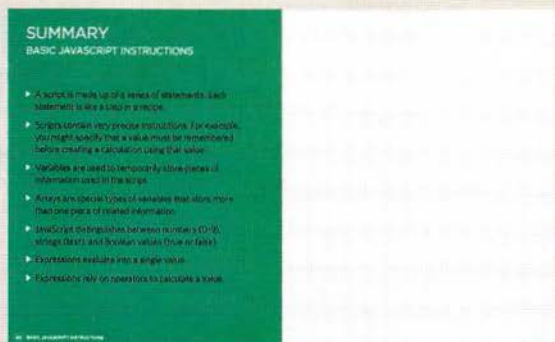
Summary pages come at the end of each chapter. They remind you of the key topics that were covered in each chapter.

HOW JAVASCRIPT MAKES WEB PAGES MORE INTERACTIVE

JavaScript allows you to make web pages more interactive by accessing and modifying the content and markup used in a web page while it is being viewed in the browser.

1
ACCESS CONTENT

You can use JavaScript to select any element, attribute, or text from an HTML page. For example:

- Select the text inside all of the <h1> elements on a page
- Select any elements that have a class attribute with a value of note
- Find out what was entered into a text input whose id attribute has a value of email

2
MODIFY CONTENT

You can use JavaScript to add elements, attributes, and text to the page, or remove them. For example:

- Add a paragraph of text after the first <h1> element
- Change the value of class attributes to trigger new CSS rules for those elements
- Change the size or position of an <img> element

3
PROGRAM RULES

You can specify a set of steps for the browser to follow (like a recipe), which allows it to access or change the content of a page. For example:

- A gallery script could check which image a user clicked on and display a larger version of that image.
- A mortgage calculator could collect values from a form, perform a calculation, and display repayments.
- An animation could check the dimensions of the browser window and move an image to the bottom of the viewable area (also known as the viewport).

JavaScript encompasses many of the traditional rules of programming.

It can make the web page feel interactive by responding to what the user does.

4
REACT TO EVENTS

You can specify that a script should run when a specific event has occurred. For example, it could be run when:

- A button is pressed
- A link is clicked (or tapped) on
- A cursor hovers over an element
- Information is added to a form
- An interval of time has passed
- A web page has finished loading

EXAMPLES OF JAVASCRIPT IN THE BROWSER

Being able to change the content of an HTML page while it is loaded in the browser is very powerful. The examples below rely on the ability to:

Access the content of the page
Modify the content of the page
Program rules or instructions the browser can follow
React to events triggered by the user or browser

SLIDESHOWS
Shown in Chapter 11

Slideshows can display a number of different images (or other HTML content) within the same space on a given page. They can play automatically as a sequence, or users can click through the slides manually. They allow more content to be displayed within a limited amount of space.

React: Script triggered when the page loads
Access: Get each slide from the slideshow
Modify: Only show the first slide (hide others)
Program: Set a timer: when to show next slide
Modify: Change which slide is shown
React: When user clicks button for different slide
Program: Determine which slide to show
Modify: Show the requested slide

FORMS
Shown in Chapter 13

Validating forms (checking whether they have been filled in correctly) is important when information is supplied by users. JavaScript lets you alert the user if mistakes have been made. It can also perform sophisticated calculations based on any data entered and reveal the results to the user.

React: User presses the submit button when they have entered their name
Access: Get value from form field
Program: Check that the name is long enough
Modify: Show a warning message if the name is not long enough

The examples on these two pages give you a taste of what JavaScript can do within a web page, and of the techniques you will be learning throughout this book.

In the coming chapters, you will learn how and when to access or modify content, add programming rules, and react to events.

RELOAD PART OF PAGE
Shown in Chapter 8

You might not want to force visitors to reload the content of an entire web page, particularly if you only need to refresh a small portion of a page. Just reloading a section of the page can make a site feel like it is faster to load and more like an application.

React: Script triggered when user clicks on link
Access: The link that they clicked on
Program: Load the new content that was requested from that link
Access: Find the element to replace in the page
Modify: Replace that content with the new content

FILTERING DATA
Shown in Chapter 12

If you have a lot of information to display on a page, you can help users find information they need by providing filters. Here, buttons are generated using data in the attributes of the HTML <img> elements. When the user clicks on one of the buttons, they are only shown the images with that keyword.

React: Script triggered when page loads
Program: Collect keywords from images
Program: Turn the keywords into buttons the user can click on
React: User clicks on one of the buttons
Program: Find the relevant subset of images that should be shown
Modify: Show the subset of images that use that tag

THE STRUCTURE OF THIS BOOK

In order to teach you JavaScript, this book is divided into two sections:

CORE CONCEPTS

The first nine chapters introduce you to the basics of programming and the JavaScript language. Along the way you will learn how it is used to create more engaging, interactive, and usable websites.

Chapter 1 looks at some key concepts in computer programming, showing you how computers create models of the world using data, and how JavaScript is used to change the contents of an HTML page.

Chapters 2-4 cover the basics of the JavaScript language.

Chapter 5 explains how the Document Object Model (DOM) lets you access and change a document's contents while it is loaded into the browser.

Chapter 6 discusses how events can be used to trigger code.

Chapter 7 shows you how jQuery can make the process of writing scripts faster and easier.

Chapter 8 introduces you to Ajax, a set of techniques that allow you to just change part of a web page without reloading the entire page.

Chapter 9 covers Application Programming Interfaces (APIs), including new APIs that are part of HTML5 and those of sites like Google Maps.

PRACTICAL APPLICATIONS

By this point you will already have seen many examples of how JavaScript is used on popular websites. This section brings together all of the techniques you have learned so far, to give you practical demonstrations of how JavaScript is used by professional developers. Not only will you see a selection of in-depth examples, you will also learn more about the process of designing and writing scripts from scratch.

Chapter 10 deals with error-handling and debugging, and explains more about how JavaScript is processed.

Chapter 11 shows you techniques for creating content panels such as sliders, modal windows, tabbed panels, and accordions.

Chapter 12 demonstrates several techniques for filtering and sorting data. This includes filtering a gallery of images, and re-ordering the rows of a table by clicking on the column headings.

**Chapter 13** deals with form enhancements and how to validate form entries.

Unless you are already a confident programmer, you will probably find it helpful to read the book from start to finish the first time. However, once you have grasped the basics, we hope it will continue to be a helpful reference as you create your own scripts.

HTML & CSS:
A QUICK REFRESHER

Before looking at JavaScript, let's clarify some HTML & CSS terms.
Note how HTML attributes and CSS properties use name/value pairs.

HTML ELEMENTS

HTML elements are added to the content of a page to describe its structure. An element consists of the opening and closing tags, plus its content.

Tags usually come in pairs with an opening tag and a closing tag. There are a few empty elements with no content, (e.g., <img>). They have one self-closing tag.

Opening tags can carry attributes, which tell us more about that element. Attributes have a name and a value. The value is usually given in quotes.

```
            OPENING TAG                    CLOSING TAG

<p class="fruit">peach</p>

    ATTRIBUTE NAME  ATTRIBUTE VALUE
```

CSS RULES

CSS uses rules to indicate how the contents of one or more elements should be displayed in the browser. Each rule has a selector and a declaration block.

The CSS selector indicates which element(s) the rule applies to. The declaration block contains rules that indicate how those elements should appear.

Each declaration in the declaration block has a property (the aspect you want to control), and a value, which is the setting for that property.

```
    SELECTOR        DECLARATION BLOCK

.fruit {color: pink;}

            PROPERTY NAME  PROPERTY VALUE
```

BROWSER SUPPORT

Some early examples in this book do not work with Internet Explorer 8 and earlier (but alternative code samples that work in IE8 are available to download from `http://javascriptbook.com`). We explain techniques for dealing with older browsers in later chapters.

Each version of a web browser adds new features. Often these new features make tasks easier, or are considered better, than using older techniques.

But, website visitors do not always keep up with the latest browser releases, so website developers cannot always rely upon the latest technologies.

As you will see, there are many inconsistencies between browsers that affect JavaScript developers. jQuery will help you deal with cross-browser inconsistencies (it is one of the major reasons why jQuery rapidly gained popularity amongst web developers). But, before you learn jQuery, it helps to know what it is helping you to achieve.

To make JavaScript easier to learn, the first few chapters use some features of JavaScript that are not supported in IE8. But:

- You **will** learn how to deal with IE8 and older browsers in later chapters (because we know that many clients expect sites to work in IE8). It just requires knowledge of some extra code or requires you to be aware of some additional issues.
- Online, you will find alternatives available for each example that does not work in IE8. But please check the comments in those code samples to make sure you know about the about issues involved in using them.