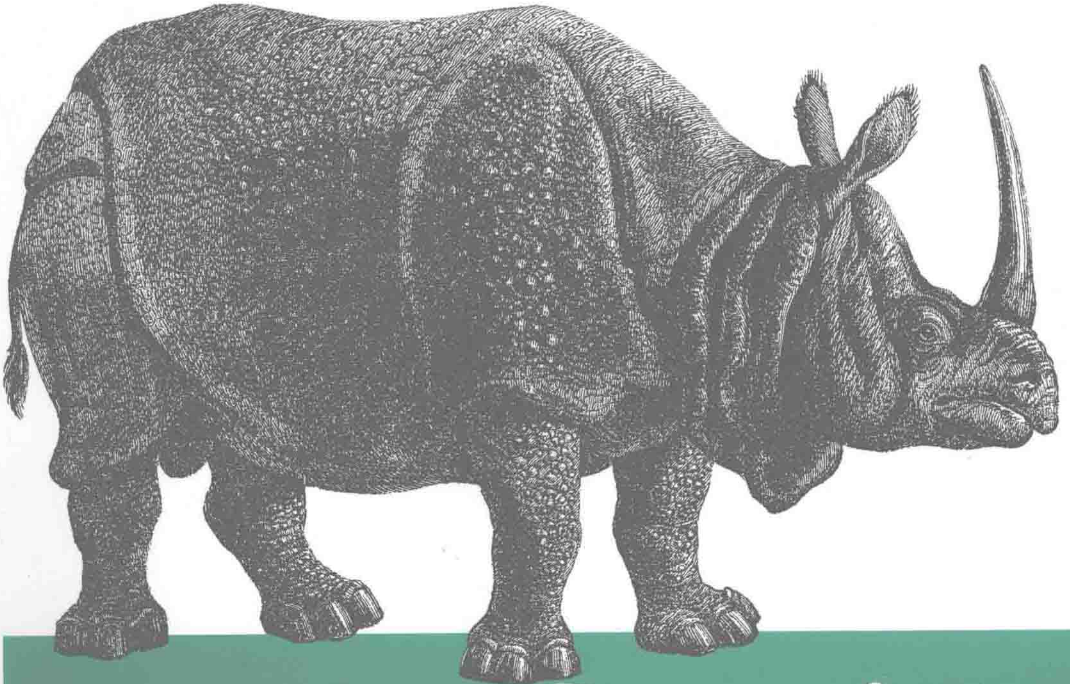
JavaScript权威指南（影印版）

第六版
上册



JavaScript

The Definitive Guide

O'REILLY®
東南大學出版社

David Flanagan 著

（第6版）

JavaScript权威指南（影印版）

JavaScript: The Definitive Guide

上 册

David Flanagan

O'REILLY®

Beijing · Cambridge · Farnham · Köln · Sebastopol · Tokyo

O'Reilly Media, Inc.授权东南大学出版社出版

东南大学出版社

图书在版编目（CIP）数据

JavaScript 权威指南：第 6 版：英文/（美）弗拉纳根（Flanagan, D.）著．—影印本．—南京：东南大学出版社，2011.10

书名原文：JavaScript:The Definitive Guide, Sixth Edition

ISBN 978-7-5641-2940-8

I. ①J… II. ①弗… III. ①JAVA 语言－程序设计－英文 IV. ①TP312

中国版本图书馆 CIP 数据核字（2011）第 173868 号

江苏省版权局著作权合同登记
图字：10-2010-447 号

JavaScript 权威指南 第六版

出版发行：东南大学出版社
地　　址：南京四牌楼 2 号　　邮编：210096
出 版 人：江建中
网　　址：http://www.seupress.com
电子邮件：press@seupress.com
印　　刷：扬中市印刷有限公司
开　　本：787 毫米 × 960 毫米　16 开本
印　　张：69.5
字　　数：1361 千字
版　　次：2011 年 10 月第 1 版
印　　次：2011 年 10 月第 1 次印刷
书　　号：ISBN 978-7-5641-2940-8
定　　价：128.00 元（上下册）

本社图书若有印装质量问题，请直接与读者服务部联系。电话（传真）：025-83792328

JavaScript权威指南（影印版）

JavaScript: The Definitive Guide

上 册

This book is dedicated to all who teach peace
and resist violence.

Preface

This book covers the JavaScript language and the JavaScript APIs implemented by web browsers. I wrote it for readers with at least some prior programming experience who want to learn JavaScript and also for programmers who already use JavaScript but want to take their understanding to a new level and really master the language and the web platform. My goal with this book is to document the JavaScript language and platform comprehensively and definitively. As a result, this is a large and detailed book. My hope, however, is that it will reward careful study, and that the time you spend reading it will be easily recouped in the form of higher programming productivity.

This book is divided into four parts. Part I covers the JavaScript language itself. Part II covers client-side JavaScript: the JavaScript APIs defined by HTML5 and related standards and implemented by web browsers. Part III is the reference section for the core language, and Part IV is the reference for client-side JavaScript. Chapter 1 includes an outline of the chapters in Parts I and II (see §1.1).

This sixth edition of the book covers both ECMAScript 5 (the latest version of the core language) and HTML5 (the latest version of the web platform). You'll find ECMAScript 5 material throughout Part I. The new material on HTML5 is mostly in the chapters at the end of Part II, but there is also some in other chapters as well. Completely new chapters in this edition include Chapter 11, *JavaScript Subsets and Extensions*; Chapter 12, *Server-Side JavaScript*; Chapter 19, *The jQuery Library*; and Chapter 22, *HTML5 APIs*.

Readers of previous editions may notice that I have completely rewritten many of the chapters in this book for the sixth edition. The core of Part I—the chapters covering objects, arrays, functions, and classes—is all new and brings the book in line with current programming styles and best practices. Similarly, key chapters of Part II, such as those covering documents and events, have been completely rewritten to bring them up-to-date.

A Note About Piracy

If you are reading a digital version of this book that you (or your employer) did not pay for (or borrow from someone who did) then you probably have an illegally pirated copy. Writing the sixth edition of this book was a full-time job, and it took more than a year. The only way I get paid for that time is when readers actually buy the book. And the only way I can afford to work on a seventh edition is if I get paid for the sixth.

I do not condone piracy, but if you have a pirated copy, go ahead and read a couple of chapters. I think that you'll find that this is a valuable source of information about JavaScript, better organized and of higher quality than what you can find freely (and legally) available on the Web. If you agree that this is a valuable source of information, then please pay for that value by purchasing a legal copy (either digital or print) of the book. On the other hand, if you find that this book is no more valuable than the free information on the web, then please discard your pirated copy and use those free information sources.

Conventions Used in This Book

I use the following typographical conventions in this book:

Italic
: Is used for emphasis and to indicate the first use of a term. *Italic* is also used for email addresses, URLs and file names.

`Constant width`
: Is used in all JavaScript code and CSS and HTML listings, and generally for anything that you would type literally when programming.

`Constant width italic`
: Is used for the names of function parameters, and generally as a placeholder to indicate an item that should be replaced with an actual value in your program.

Example Code

The examples in this book are available online. You can find them linked from the book's catalog page at the publisher's website:

http://oreilly.com/catalog/9780596805531/

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact O'Reilly for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books *does* require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation *does* require permission.

If you use the code from this book, I appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*JavaScript: The Definitive Guide*, by David Flanagan (O'Reilly). Copyright 2011 David Flanagan, 978-0-596-80552-4."

For more details on the O'Reilly code reuse policy, see *http://oreilly.com/pub/a/oreilly/ask_tim/2001/codepolicy.html*. If you feel your use of the examples falls outside of the permission given above, feel free to contact O'Reilly at *permissions@oreilly.com*.

Errata and How to Contact Us

The publisher maintains a public list of errors found in this book. You can view the list, and submit the errors you find, by visiting the book's web page:

> *http://oreilly.com/catalog/9780596805531*

To comment or ask technical questions about this book, send email to:

> *bookquestions@oreilly.com*

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our website at:

> *http://www.oreilly.com*

Find us on Facebook: *http://facebook.com/oreilly*

Follow us on Twitter: *http://twitter.com/oreillymedia*

Watch us on YouTube: *http://www.youtube.com/oreillymedia*

Acknowledgments

Many people have helped me with the creation of this book. I'd like to thank my editor, Mike Loukides, for trying to keep me on schedule and for his insightful comments. Thanks also to my technical reviewers: Zachary Kessin, who reviewed many of the chapters in Part I, and Raffaele Cecco, who reviewed Chapter 19 and the `<canvas>` material in Chapter 21. The production team at O'Reilly has done their usual fine job: Dan Fauxsmith managed the production process, Teresa Elsey was the production editor, Rob Romano drew the figures, and Ellen Troutman Zaig created the index.

In this era of effortless electronic communication, it is impossible to keep track of all those who influence and inform us. I'd like to thank everyone who has answered my questions on the es5, w3c, and whatwg mailing lists, and everyone who has shared their insightful ideas about JavaScript programming online. I'm sorry I can't list you all by name, but it is a pleasure to work within such a vibrant community of JavaScript programmers.

Editors, reviewers, and contributors to previous editions of this book have included: Andrew Schulman, Angelo Sirigos, Aristotle Pagaltzis, Brendan Eich, Christian Heilmann, Dan Shafer, Dave C. Mitchell, Deb Cameron, Douglas Crockford, Dr. Tankred Hirschmann, Dylan Schiemann, Frank Willison, Geoff Stearns, Herman Venter, Jay Hodges, Jeff Yates, Joseph Kesselman, Ken Cooper, Larry Sullivan, Lynn Rollins, Neil Berkman, Nick Thompson, Norris Boyd, Paula Ferguson, Peter-Paul Koch, Philippe Le Hegaret, Richard Yaker, Sanders Kleinfeld, Scott Furman, Scott Issacs, Shon Katzenberger, Terry Allen, Todd Ditchendorf, Vidur Apparao, and Waldemar Horwat.

This edition of the book is substantially rewritten and kept me away from my family for many late nights. My love to them and my thanks for putting up with my absences.

— David Flanagan (davidflanagan.com), March 2011

Table of Contents

Part I. Core JavaScript

Part II. Client-Side JavaScript

Part III. Core JavaScript Reference

Part IV. Client-Side JavaScript Reference

CHAPTER 1

Introduction to JavaScript

JavaScript is the programming language of the Web. The overwhelming majority of modern websites use JavaScript, and all modern web browsers—on desktops, game consoles, tablets, and smart phones—include JavaScript interpreters, making JavaScript the most ubiquitous programming language in history. JavaScript is part of the triad of technologies that all Web developers must learn: HTML to specify the content of web pages, CSS to specify the presentation of web pages, and JavaScript to specify the behavior of web pages. This book will help you master the language.

If you are already familiar with other programming languages, it may help you to know that JavaScript is a high-level, dynamic, untyped interpreted programming language that is well-suited to object-oriented and functional programming styles. JavaScript derives its syntax from Java, its first-class functions from Scheme, and its prototype-based inheritance from Self. But you do not need to know any of those languages, or be familiar with those terms, to use this book and learn JavaScript.

The name "JavaScript" is actually somewhat misleading. Except for a superficial syntactic resemblance, JavaScript is completely different from the Java programming language. And JavaScript has long since outgrown its scripting-language roots to become a robust and efficient general-purpose language. The latest version of the language (see the sidebar) defines new features for serious large-scale software development.

JavaScript: Names and Versions

JavaScript was created at Netscape in the early days of the Web, and technically, "JavaScript" is a trademark licensed from Sun Microsystems (now Oracle) used to describe Netscape's (now Mozilla's) implementation of the language. Netscape submitted the language for standardization to ECMA—the European Computer Manufacturer's Association—and because of trademark issues, the standardized version of the language was stuck with the awkward name "ECMAScript." For the same trademark reasons, Microsoft's version of the language is formally known as "JScript." In practice, just about everyone calls the language JavaScript. This book uses the name "ECMAScript" only to refer to the language standard.

For the last decade, all web browsers have implemented version 3 of the ECMAScript standard and there has really been no need to think about version numbers: the language standard was stable and browser implementations of the language were, for the most part, interoperable. Recently, an important new version of the language has been defined as ECMAScript version 5 and, at the time of this writing, browsers are beginning to implement it. This book covers all the new features of ECMAScript 5 as well as all the long-standing features of ECMAScript 3. You'll sometimes see these language versions abbreviated as ES3 and ES5, just as you'll sometimes see the name JavaScript abbreviated as JS.

When we're speaking of the language itself, the only version numbers that are relevant are ECMAScript versions 3 or 5. (Version 4 of ECMAScript was under development for years, but proved to be too ambitious and was never released.) Sometimes, however, you'll also see a JavaScript version number, such as JavaScript 1.5 or JavaScript 1.8. These are Mozilla's version numbers: version 1.5 is basically ECMAScript 3, and later versions include nonstandard language extensions (see Chapter 11). Finally, there are also version numbers attached to particular JavaScript interpreters or "engines." Google calls its JavaScript interpreter V8, for example, and at the time of this writing the current version of the V8 engine is 3.0.

To be useful, every language must have a platform or standard library or API of functions for performing things like basic input and output. The core JavaScript language defines a minimal API for working with text, arrays, dates, and regular expressions but does not include any input or output functionality. Input and output (as well as more sophisticated features, such as networking, storage, and graphics) are the responsibility of the "host environment" within which JavaScript is embedded. Usually that host environment is a web browser (though we'll see two uses of JavaScript without a web browser in Chapter 12). Part I of this book covers the language itself and its minimal built-in API. Part II explains how JavaScript is used in web browsers and covers the sprawling browser-based APIs loosely known as "client-side JavaScript."

Part III is the reference section for the core API. You can read about the JavaScript array manipulation API by looking up "Array" in this part of the book, for example. Part IV is the reference section for client-side JavaScript. You might look up "Canvas"