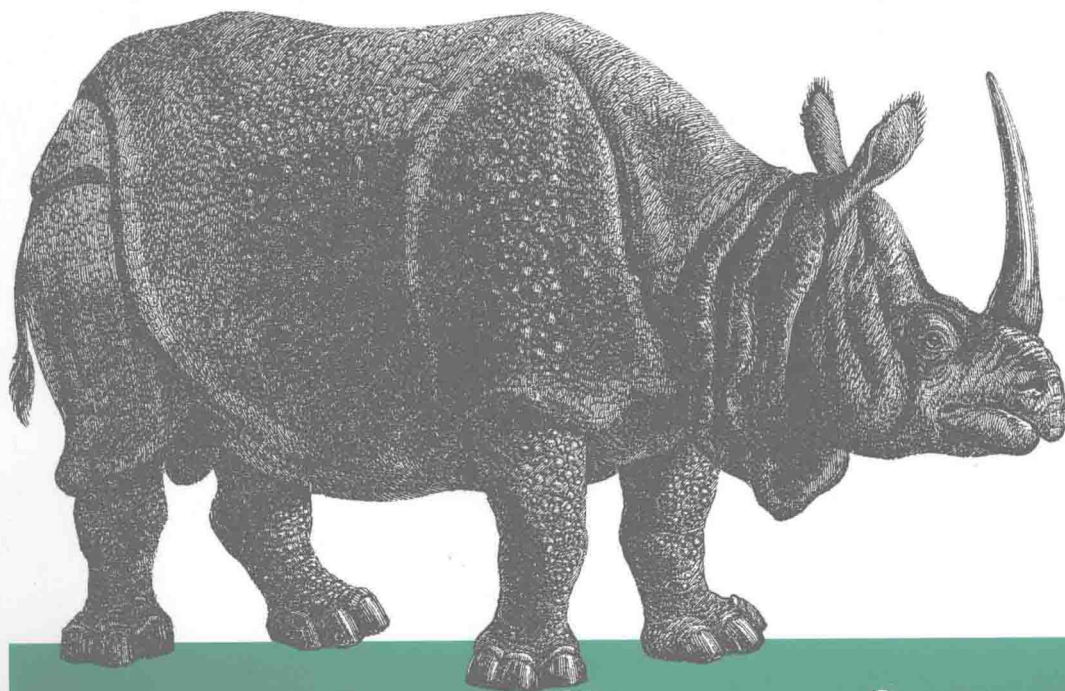


JavaScript权威指南 (影印版)

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



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

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

©2011 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2011. Authorized reprint of the original English edition, 2011 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2011。

英文影印版由东南大学出版社出版 2011。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

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

Preface	xiii
1. Introduction to JavaScript	1
1.1 Core JavaScript	4
1.2 Client-Side JavaScript	8
<hr/>	
Part I. Core JavaScript	
2. Lexical Structure	21
2.1 Character Set	21
2.2 Comments	23
2.3 Literals	23
2.4 Identifiers and Reserved Words	23
2.5 Optional Semicolons	25
3. Types, Values, and Variables	29
3.1 Numbers	31
3.2 Text	36
3.3 Boolean Values	40
3.4 null and undefined	41
3.5 The Global Object	42
3.6 Wrapper Objects	43
3.7 Immutable Primitive Values and Mutable Object References	44
3.8 Type Conversions	45
3.9 Variable Declaration	52
3.10 Variable Scope	53
4. Expressions and Operators	57
4.1 Primary Expressions	57
4.2 Object and Array Initializers	58
4.3 Function Definition Expressions	59

4.4	Property Access Expressions	60
4.5	Invocation Expressions	61
4.6	Object Creation Expressions	61
4.7	Operator Overview	62
4.8	Arithmetic Expressions	66
4.9	Relational Expressions	71
4.10	Logical Expressions	75
4.11	Assignment Expressions	77
4.12	Evaluation Expressions	79
4.13	Miscellaneous Operators	82
5.	Statements	87
5.1	Expression Statements	88
5.2	Compound and Empty Statements	88
5.3	Declaration Statements	89
5.4	Conditionals	92
5.5	Loops	97
5.6	Jumps	102
5.7	Miscellaneous Statements	108
5.8	Summary of JavaScript Statements	112
6.	Objects	115
6.1	Creating Objects	116
6.2	Querying and Setting Properties	120
6.3	Deleting Properties	124
6.4	Testing Properties	125
6.5	Enumerating Properties	126
6.6	Property Getters and Setters	128
6.7	Property Attributes	131
6.8	Object Attributes	135
6.9	Serializing Objects	138
6.10	Object Methods	138
7.	Arrays	141
7.1	Creating Arrays	141
7.2	Reading and Writing Array Elements	142
7.3	Sparse Arrays	144
7.4	Array Length	144
7.5	Adding and Deleting Array Elements	145
7.6	Iterating Arrays	146
7.7	Multidimensional Arrays	148
7.8	Array Methods	148
7.9	ECMAScript 5 Array Methods	153
7.10	Array Type	157

7.11	Array-Like Objects	158
7.12	Strings As Arrays	160
8.	Functions	163
8.1	Defining Functions	164
8.2	Invoking Functions	166
8.3	Function Arguments and Parameters	171
8.4	Functions As Values	176
8.5	Functions As Namespaces	178
8.6	Closures	180
8.7	Function Properties, Methods, and Constructor	186
8.8	Functional Programming	191
9.	Classes and Modules	199
9.1	Classes and Prototypes	200
9.2	Classes and Constructors	201
9.3	Java-Style Classes in JavaScript	205
9.4	Augmenting Classes	208
9.5	Classes and Types	209
9.6	Object-Oriented Techniques in JavaScript	215
9.7	Subclasses	228
9.8	Classes in ECMAScript 5	238
9.9	Modules	246
10.	Pattern Matching with Regular Expressions	251
10.1	Defining Regular Expressions	251
10.2	String Methods for Pattern Matching	259
10.3	The RegExp Object	261
11.	JavaScript Subsets and Extensions	265
11.1	JavaScript Subsets	266
11.2	Constants and Scoped Variables	269
11.3	Destructuring Assignment	271
11.4	Iteration	274
11.5	Shorthand Functions	282
11.6	Multiple Catch Clauses	283
11.7	E4X: ECMAScript for XML	283
12.	Server-Side JavaScript	289
12.1	Scripting Java with Rhino	289
12.2	Asynchronous I/O with Node	296

Part II. Client-Side JavaScript

13. JavaScript in Web Browsers	307
13.1 Client-Side JavaScript	307
13.2 Embedding JavaScript in HTML	311
13.3 Execution of JavaScript Programs	317
13.4 Compatibility and Interoperability	325
13.5 Accessibility	332
13.6 Security	332
13.7 Client-Side Frameworks	338
14. The Window Object	341
14.1 Timers	341
14.2 Browser Location and Navigation	343
14.3 Browsing History	345
14.4 Browser and Screen Information	346
14.5 Dialog Boxes	348
14.6 Error Handling	351
14.7 Document Elements As Window Properties	351
14.8 Multiple Windows and Frames	353
15. Scripting Documents	361
15.1 Overview of the DOM	361
15.2 Selecting Document Elements	364
15.3 Document Structure and Traversal	371
15.4 Attributes	375
15.5 Element Content	378
15.6 Creating, Inserting, and Deleting Nodes	382
15.7 Example: Generating a Table of Contents	387
15.8 Document and Element Geometry and Scrolling	389
15.9 HTML Forms	396
15.10 Other Document Features	405
16. Scripting CSS	413
16.1 Overview of CSS	414
16.2 Important CSS Properties	419
16.3 Scripting Inline Styles	431
16.4 Querying Computed Styles	435
16.5 Scripting CSS Classes	437
16.6 Scripting Stylesheets	440
17. Handling Events	445
17.1 Types of Events	447

17.2	Registering Event Handlers	456
17.3	Event Handler Invocation	460
17.4	Document Load Events	465
17.5	Mouse Events	466
17.6	Mousewheel Events	471
17.7	Drag and Drop Events	474
17.8	Text Events	481
17.9	Keyboard Events	484
18.	Scripted HTTP	491
18.1	Using XMLHttpRequest	494
18.2	HTTP by <script>: JSONP	513
18.3	Comet with Server-Sent Events	515
19.	The jQuery Library	523
19.1	jQuery Basics	524
19.2	jQuery Getters and Setters	530
19.3	Altering Document Structure	537
19.4	Handling Events with jQuery	540
19.5	Animated Effects	551
19.6	Ajax with jQuery	558
19.7	Utility Functions	571
19.8	jQuery Selectors and Selection Methods	574
19.9	Extending jQuery with Plug-ins	582
19.10	The jQuery UI Library	585
20.	Client-Side Storage	587
20.1	localStorage and sessionStorage	589
20.2	Cookies	593
20.3	IE userData Persistence	599
20.4	Application Storage and Offline Webapps	601
21.	Scripted Media and Graphics	613
21.1	Scripting Images	613
21.2	Scripting Audio and Video	615
21.3	SVG: Scalable Vector Graphics	622
21.4	Graphics in a <canvas>	630
22.	HTML5 APIs	667
22.1	Geolocation	668
22.2	History Management	671
22.3	Cross-Origin Messaging	676
22.4	Web Workers	680

22.5	Typed Arrays and ArrayBuffers	687
22.6	Blobs	691
22.7	The Filesystem API	700
22.8	Client-Side Databases	705
22.9	Web Sockets	712

Part III. Core JavaScript Reference

Core JavaScript Reference	719
---------------------------------	-----

Part IV. Client-Side JavaScript Reference

Client-Side JavaScript Reference	859
Index	1019

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”