

Head First JavaScript Programming

深入浅出JavaScript编程(影印版)



Watch out for
common JavaScript
traps and pitfalls

Avoid
embarrassing
typing conversion
mistakes



Bend your mind
around 120 puzzles
& exercises



A learner's guide to
JavaScript program-
ming

Launch your
programming
career in
one chapter



Learn why everything
your friends know about
functions & objects are
probably wrong



深入浅出JavaScript编程 (影印版)

Head First JavaScript Programming

Wouldn't it be dreamy if there was a JavaScript book that was more fun than going to the dentist and more revealing than an IRS form? It's probably just a fantasy...



Eric Freeman &
Elisabeth Robson 著

O'REILLY®

Beijing • Cambridge • 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

深入浅出 JavaScript 编程 :英文/(美)弗里曼(Freeman, E.),
(美)罗布森 (Robson, E.)著. —影印本. —南京: 东南大学
出版社, 2014.9

书名原文: Head First JavaScript Programming

ISBN 978-7-5641-5001-3

I. ①深… II. ①弗… ②罗… III. ①JAVA 语言—程
序设计—英文 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2014) 第 115565 号

江苏省版权局著作权合同登记

图字: 10-2013-362 号

©2014 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2014. Authorized reprint of the original English edition, 2013 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 2014。

英文影印版由东南大学出版社出版 2014。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复制。

深入浅出 JavaScript 编程 (影印版)

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: 常州市武进第三印刷有限公司

开 本: 787 毫米 × 980 毫米 12 开本

印 张: 58.5

字 数: 979 千字

版 次: 2014 年 9 月第 1 版

印 次: 2014 年 9 月第 1 次印刷

书 号: ISBN 978-7-5641-5001-3

定 价: 118.00 元

本社图书若有印装质量问题, 请直接与营销部联系。电话 (传真): 025-83791830

Praise for *Head First JavaScript Programming*

“Warning: Do not read *Head First JavaScript Programming* unless you want to learn the fundamentals of programming with JavaScript in an entertaining and meaningful fashion. There may be an additional side effect that you may actually recall more about JavaScript than after reading typical technical books.”

— **Jesse Palmer, Senior Software Developer, Gannett Digital**

“If every elementary and middle school student studied Elisabeth and Eric’s *Head First HTML and CSS*, and if *Head First JavaScript Programming* and *Head First HTML5 Programming* were part of the high school math and science curriculum, then our country would never lose its competitive edge.”

— **Michael Murphy, senior systems consultant, The History Tree**

“The *Head First* series utilizes elements of modern learning theory, including constructivism, to bring readers up to speed quickly. The authors have proven with this book that expert-level content can be taught quickly and efficiently. Make no mistake here, this is a serious JavaScript book, and yet, fun reading!”

— **Frank Moore, Web designer and developer**

“Looking for a book that will keep you interested (and laughing) but teach you some serious programming skills? *Head First JavaScript Programming* is it!”

— **Tim Williams, software entrepreneur**

“Add this book to your library regardless of your programming skill level!”

— **Chris Fuselier, engineering consultant**

“Robson and Freeman have done it again! Using the same fun and information-packed style as their previous books in the *Head First* series, *Head First JavaScript Programming* leads you through entertaining and useful projects that, chapter-by-chapter, allow programmers—even nonspecialists like myself—to develop a solid foundation in modern JavaScript programming that we can use to solve real problems.”

— **Russell Alleen-Willems, digital archeologist, DiachronicDesign.com**

“Freeman and Robson continue to use innovative teaching methods for communicating complex concepts to basic principles.”

— **Mark Arana, Strategy & Innovation, The Walt Disney Studios**

Praise for other books by Eric T. Freeman and Elisabeth Robson

“Just the right tone for the geeked-out, casual-cool guru coder in all of us. The right reference for practical development strategies—gets my brain going without having to slog through a bunch of tired, stale professor-speak.”

— **Travis Kalanick, CEO Uber**

“This book’s admirable clarity, humor and substantial doses of clever make it the sort of book that helps even non-programmers think well about problem-solving.”

— **Cory Doctorow, co-editor of Boing Boing, Science Fiction author**

“I feel like a thousand pounds of books have just been lifted off of my head.”

— **Ward Cunningham, inventor of the Wiki**

“One of the very few software books I’ve ever read that strikes me as indispensable. (I’d put maybe 10 books in this category, at the outside.)”

— **David Gelernter, Professor of Computer Science, Yale University**

“I laughed, I cried, it moved me.”

— **Daniel Steinberg, Editor-in-Chief, java.net**

“I can think of no better tour guides than Eric and Elisabeth.”

— **Miko Matsumura, VP of Marketing and Developer Relations at Hazelcast
Former Chief Java Evangelist, Sun Microsystems**

“I literally love this book. In fact, I kissed this book in front of my wife.”

— **Satish Kumar**

“The highly graphic and incremental approach precisely mimics the best way to learn this stuff...”

— **Danny Goodman, author of *Dynamic HTML: The Definitive Guide***

“Eric and Elisabeth clearly know their stuff. As the Internet becomes more complex, inspired construction of web pages becomes increasingly critical. Elegant design is at the core of every chapter here, each concept conveyed with equal doses of pragmatism and wit.”

— **Ken Goldstein, former CEO of Shop.com and author of
*This is Rage: A Novel of Silicon Valley and Other Madness***

Other O'Reilly books by Eric T. Freeman and Elisabeth Robson

Head First Design Patterns

Head First HTML and CSS

Head First HTML5 Programming

Other related books from O'Reilly

Head First HTML5 Programming

JavaScript: The Definitive Guide

JavaScript Enlightenment

Other books in O'Reilly's *Head First* series

Head First HTML and CSS

Head First HTML5 Programming

Head First Design Patterns

Head First Servlets and JSP

Head First SQL

Head First Software Development

Head First C#

Head First Java

Head First Object-Oriented Analysis and Design (OOA&D)

Head First Ajax

Head First Rails

Head First PHP & MySQL

Head First Web Design

Head First Networking

Head First iPhone and iPad Development

Head First jQuery

To JavaScript—you weren't born with a silver spoon in your mouth, but you've outclassed every language that's challenged you in the browser.

Authors of Head First JavaScript Programming



Eric Freeman

Eric is described by Head First series co-creator Kathy Sierra as “one of those rare individuals fluent in the language, practice, and culture of multiple domains from hipster hacker, corporate VP, engineer, think tank.”

Professionally, Eric recently ended nearly a decade as a media company executive—having held the position of CTO of Disney Online & Disney.com at The Walt Disney Company. Eric is now devoting his time to WickedlySmart, a startup he co-created with Elisabeth.

By training, Eric is a computer scientist, having studied with industry luminary David Gelernter during his Ph.D. work at Yale University. His dissertation is credited as the seminal work in alternatives to the desktop metaphor, and also as the first implementation of activity streams, a concept he and Dr. Gelernter developed.

In his spare time, Eric is deeply involved with music; you’ll find Eric’s latest project, a collaboration with ambient music pioneer Steve Roach, available on the iPhone app store under the name Immersion Station.

Eric lives with his wife and young daughter on Bainbridge Island. His daughter is a frequent visitor to Eric’s studio, where she loves to turn the knobs of his synths and audio effects.

Write to Eric at eric@wickedlysmart.com or visit his site at <http://ericfreeman.com>.



Elisabeth Robson

Elisabeth is a software engineer, writer, and trainer. She has been passionate about technology since her days as a student at Yale University, where she earned a Masters of Science in Computer Science and designed a concurrent, visual programming language and software architecture.

Elisabeth’s been involved with the Internet since the early days; she co-created the award-winning Web site, The Ada Project, one of the first Web sites designed to help women in computer science find career and mentorship information online.

She’s currently co-founder of WickedlySmart, an online education experience centered on web technologies, where she creates books, articles, videos and more. Previously, as Director of Special Projects at O’Reilly Media, Elisabeth produced in-person workshops and online courses on a variety of technical topics and developed her passion for creating learning experiences to help people understand technology. Prior to her work with O’Reilly, Elisabeth spent time spreading fairy dust at The Walt Disney Company, where she led research and development efforts in digital media.

When not in front of her computer, you’ll find Elisabeth hiking, cycling or kayaking in the great outdoors, with her camera nearby, or cooking vegetarian meals.

You can send her email at beth@wickedlysmart.com or visit her blog at <http://elisabethrobson.com>.

Table of Contents (summary)

	Intro	xxv
1	A quick dip into JavaScript: <i>Getting your feet wet</i>	1
2	Writing real code: <i>Going further</i>	43
3	Introducing functions: <i>Getting functional</i>	79
4	Putting some order in your data: <i>Arrays</i>	125
5	Understanding objects: <i>A trip to Objectville</i>	173
6	Interacting with your web page: <i>Getting to know the DOM</i>	229
7	Types, equality, conversion, and all that jazz: <i>Serious types</i>	265
8	Bringing it all together: <i>Building an app</i>	317
9	Asynchronous coding: <i>Handling events</i>	381
10	First-class functions: <i>Liberated functions</i>	429
11	Anonymous functions, scope, and closures: <i>Serious functions</i>	475
12	Advanced object construction: <i>Creating objects</i>	521
13	Using prototypes: <i>Extra-strength objects</i>	563
	Appendix: The Top Ten Topics (we didn't cover): <i>Leftovers</i>	623

Table of Contents (the real thing)

Intro

Your brain on JavaScript. Here you are trying to *learn* something, while here your *brain* is doing you a favor by making sure the learning doesn't *stick*. Your brain's thinking, "Better leave room for more important things, like which wild animals to avoid and whether naked snowboarding is a bad idea." So how *do* you trick your brain into thinking that your life depends on knowing JavaScript programming?

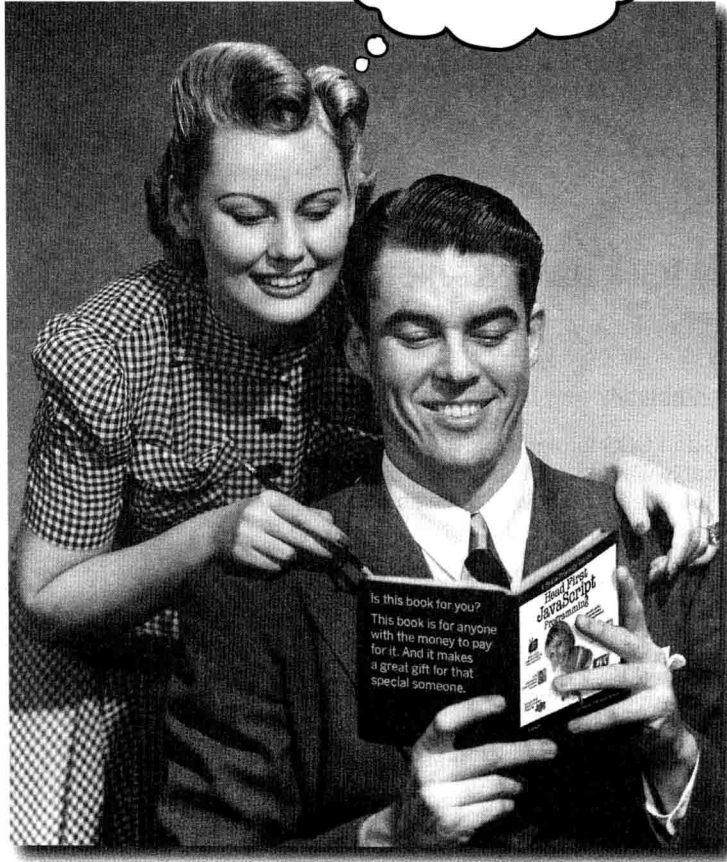


Who is this book for?	xxvi
We know what you're thinking.	xxvii
We think of a "Head First" reader as a learner.	xxviii
Metacognition: thinking about thinking	xxix
Here's what WE did:	xxx
Here's what YOU can do to bend your brain into submission	xxxi
Read Me	xxxii
Tech Reviewers	xxxv
Acknowledgments*	xxxvi

how to use this book

Intro

I can't believe they put *that* in a JavaScript book!



In this section, we answer the burning question:
"So, why DID they put that in a JavaScript book?"

Who is this book for?

If you can answer “yes” to all of these:

- ① Do you have access to a computer with a **modern web browser** and a **text editor**?
- ② Do you want to **learn, understand** and **remember** how to **program with JavaScript** using the best techniques and the most recent standards?
- ③ Do you prefer **stimulating dinner party conversation** to **dry, dull, academic lectures**?



We consider an updated version of Safari, Chrome, Firefox or IE version 9 or newer to be modern.

this book is for you.

[Note from marketing: this book is for anyone with a credit card.]

Who should probably back away from this book?

If you can answer “yes” to any one of these:

- ① **Are you completely new to web development?**
Are HTML and CSS foreign concepts to you? If so, you'll probably want to start with *Head First HTML and CSS* to understand how to put web pages together before tackling JavaScript.
- ② **Are you a kick-butt web developer looking for a *reference book*?**
- ③ **Are you *afraid to try something different*?** Would you rather have a root canal than mix stripes with plaid? Do you believe that a technical book can't be serious if JavaScript objects are anthropomorphized?

this book is not for you.



We know what you're thinking.

"How can this be a serious book?"

"What's with all the graphics?"

"Can I actually learn it this way?"

And we know what your brain is thinking.

Your brain craves novelty. It's always searching, scanning, *waiting* for something unusual. It was built that way, and it helps you stay alive.

Today, you're less likely to be a tiger snack. But your brain's still looking. You just never know.

So what does your brain do with all the routine, ordinary, normal things you encounter? Everything it *can* to stop them from interfering with the brain's *real* job—recording things that *matter*. It doesn't bother saving the boring things; they never make it past the "this is obviously not important" filter.

How does your brain *know* what's important? Suppose you're out for a day hike and a tiger jumps in front of you. What happens inside your head and body?

Neurons fire. Emotions crank up. *Chemicals surge*.

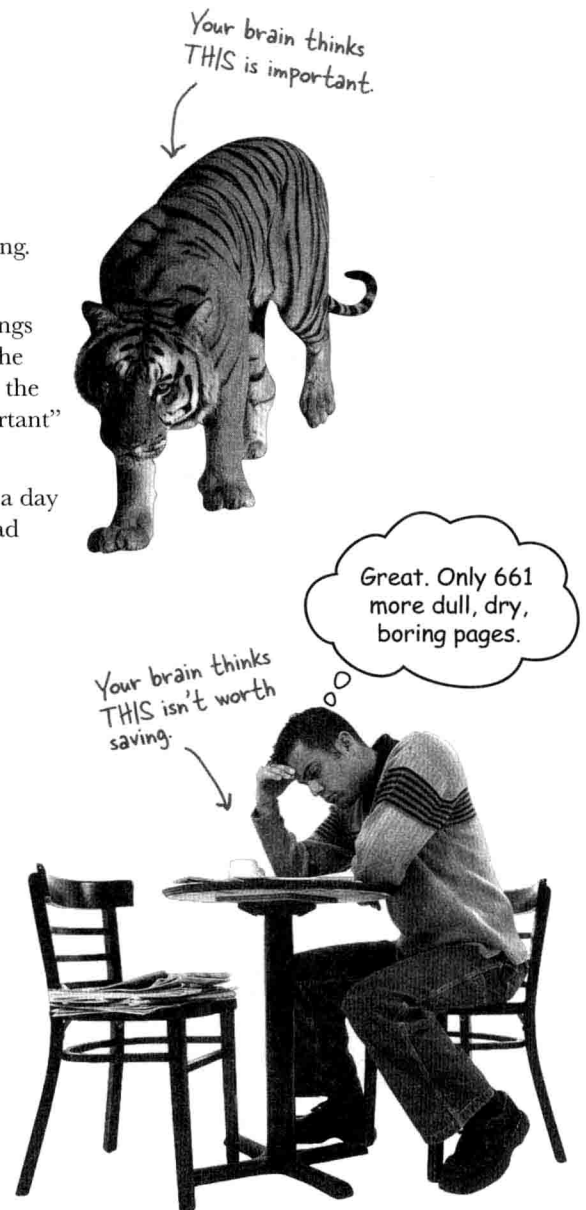
And that's how your brain knows...

This must be important! Don't forget it!

But imagine you're at home, or in a library. It's a safe, warm, tiger-free zone. You're studying. Getting ready for an exam. Or trying to learn some tough technical topic your boss thinks will take a week, ten days at the most.

Just one problem. Your brain's trying to do you a big favor. It's trying to make sure that this *obviously* non-important content doesn't clutter up scarce resources. Resources that are better spent storing the really *big* things. Like tigers. Like the danger of fire. Like how you should never again snowboard in shorts.

And there's no simple way to tell your brain, "Hey brain, thank you very much, but no matter how dull this book is, and how little I'm registering on the emotional Richter scale right now, I really *do* want you to keep this stuff around."

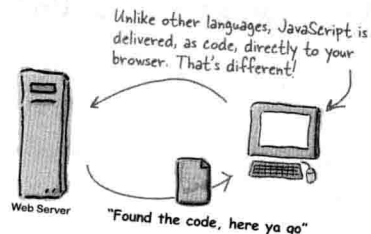


We think of a “Head First” reader as a learner.

So what does it take to *learn* something? First, you have to *get* it, then make sure you don't *forget* it. It's not about pushing facts into your head. Based on the latest research in cognitive science, neurobiology and educational psychology, *learning* takes a lot more than text on a page. We know what turns your brain on.

Some of the Head First learning principles:

Make it visual. Images are far more memorable than words alone, and make learning much more effective (up to 89% improvement in recall and transfer studies). It also makes things more understandable. **Put the words within or near the graphics** they relate to, rather than on the bottom or on another page, and learners will be up to twice as likely to solve problems related to the content.



I really think JavaScript should go in the <head> element.



Not so fast! There are performance and page loading implications!

Use a conversational and personalized style. In recent studies, students performed up to 40% better on post-learning tests if the content spoke directly to the reader, using a first-person, conversational style rather than taking a formal tone. Tell stories instead of lecturing. Use casual language. Don't take yourself too seriously. Which would you pay more attention to: a stimulating dinner party companion, or a lecture?

Get the learner to think more deeply. In other words, unless you actively flex your neurons, nothing much happens in your head. A reader has to be motivated, engaged, curious and inspired to solve problems, draw conclusions and generate new knowledge. And for that, you need challenges, exercises and thought-provoking questions, and activities that involve both sides of the brain and multiple senses.

Now that I have your attention, you should be more careful using global variables.



Get—and keep—the reader's attention. We've all had the “I really want to learn this but I can't stay awake past page one” experience. Your brain pays attention to things that are out of the ordinary, interesting, strange, eye-catching, unexpected. Learning a new, tough, technical topic doesn't have to be boring. Your brain will learn much more quickly if it's not.

Touch their emotions. We now know that your ability to remember something is largely dependent on its emotional content. You remember what you care about. You remember when you *feel* something. No, we're not talking heart-wrenching stories about a boy and his dog. We're talking emotions like surprise, curiosity, fun, “what the...?” , and the feeling of “I Rule!” that comes when you solve a puzzle, learn something everybody else thinks is hard, or realize you know something that “I'm more technical than thou” Bob from engineering *doesn't*.



Metacognition: thinking about thinking

If you really want to learn, and you want to learn more quickly and more deeply, pay attention to how you pay attention. Think about how you think. Learn how you learn.

Most of us did not take courses on metacognition or learning theory when we were growing up. We were *expected* to learn, but rarely *taught* how to learn.

But we assume that if you're holding this book, you really want to learn how to create JavaScript programs. And you probably don't want to spend a lot of time. And you want to *remember* what you read, and be able to apply it. And for that, you've got to *understand* it. To get the most from this book, or *any* book or learning experience, take responsibility for your brain. Your brain on *this* content.

The trick is to get your brain to see the new material you're learning as Really Important. Crucial to your well-being. As important as a tiger. Otherwise, you're in for a constant battle, with your brain doing its best to keep the new content from sticking.

So how **DO** you get your brain to think JavaScript is as important as a tiger?

There's the slow, tedious way, or the faster, more effective way. The slow way is about sheer repetition. You obviously know that you *are* able to learn and remember even the dullest of topics, if you keep pounding on the same thing. With enough repetition, your brain says, "This doesn't *feel* important to him, but he keeps looking at the same thing *over* and *over* and *over*, so I suppose it must be."

The faster way is to do **anything that increases brain activity**, especially different *types* of brain activity. The things on the previous page are a big part of the solution, and they're all things that have been proven to help your brain work in your favor. For example, studies show that putting words *within* the pictures they describe (as opposed to somewhere else in the page, like a caption or in the body text) causes your brain to try to make sense of how the words and picture relate, and this causes more neurons to fire. More neurons firing = more chances for your brain to *get* that this is something worth paying attention to, and possibly recording.

A conversational style helps because people tend to pay more attention when they perceive that they're in a conversation, since they're expected to follow along and hold up their end. The amazing thing is, your brain doesn't necessarily *care* that the "conversation" is between you and a book! On the other hand, if the writing style is formal and dry, your brain perceives it the same way you experience being lectured to while sitting in a roomful of passive attendees. No need to stay awake.

But pictures and conversational style are just the beginning.



Here's what WE did:

We used **pictures**, because your brain is tuned for visuals, not text. As far as your brain's concerned, a picture really *is* worth 1024 words. And when text and pictures work together, we embedded the text *in* the pictures because your brain works more effectively when the text is *within* the thing the text refers to, as opposed to in a caption or buried in the text somewhere.

We used **redundancy**, saying the same thing in *different* ways and with different media types, and *multiple senses*, to increase the chance that the content gets coded into more than one area of your brain.

We used concepts and pictures in **unexpected** ways because your brain is tuned for novelty, and we used pictures and ideas with at least *some emotional content*, because your brain is tuned to pay attention to the biochemistry of emotions. That which causes you to *feel* something is more likely to be remembered, even if that feeling is nothing more than a little **humor**, **surprise** or **interest**.

We used a personalized, **conversational style**, because your brain is tuned to pay more attention when it believes you're in a conversation than if it thinks you're passively listening to a presentation. Your brain does this even when you're *reading*.

We included more than 100 **activities**, because your brain is tuned to learn and remember more when you *do* things than when you *read* about things. And we made the exercises challenging-yet-do-able, because that's what most *people* prefer.

We used **multiple learning styles**, because *you* might prefer step-by-step procedures, while someone else wants to understand the big picture first, while someone else just wants to see a code example. But regardless of your own learning preference, *everyone* benefits from seeing the same content represented in multiple ways.

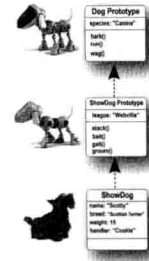
We include content for **both sides of your brain**, because the more of your brain you engage, the more likely you are to learn and remember, and the longer you can stay focused. Since working one side of the brain often means giving the other side a chance to rest, you can be more productive at learning for a longer period of time.

And we included **stories** and exercises that present **more than one point of view**, because your brain is tuned to learn more deeply when it's forced to make evaluations and judgements.

We included **challenges**, with exercises, and by asking **questions** that don't always have a straight answer, because your brain is tuned to learn and remember when it has to *work* at something. Think about it—you can't get your *body* in shape just by *watching* people at the gym. But we did our best to make sure that when you're working hard, it's on the *right* things. That **you're not spending one extra dendrite** processing a hard-to-understand example, or parsing difficult, jargon-laden or overly terse text.

We used **people**. In stories, examples, pictures, etc., because, well, *you're* a person. And your brain pays more attention to *people* than it does to *things*.

We used an **80/20** approach. We assume that if you're going to be a kick-butt JavaScript developer, this won't be your only book. So we don't talk about *everything*. Just the stuff you'll actually *need*.



Be the Browser



BULLET POINTS

Puzzles





Here's what YOU can do to bend your brain into submission

So, we did our part. The rest is up to you. These tips are a starting point; listen to your brain and figure out what works for you and what doesn't. Try new things.

Cut this out and stick it on your refrigerator.

1 Slow down. The more you understand, the less you have to memorize.

Don't just *read*. Stop and think. When the book asks you a question, don't just skip to the answer. Imagine that someone really *is* asking the question. The more deeply you force your brain to think, the better chance you have of learning and remembering.

2 Do the exercises. Write your own notes.

We put them in, but if we did them for you, that would be like having someone else do your workouts for you. And don't just *look* at the exercises. **Use a pencil.** There's plenty of evidence that physical activity *while* learning can increase the learning.

3 Read the "There are No Dumb Questions"

That means all of them. They're not optional sidebars—*they're part of the core content!* Don't skip them.

4 Make this the last thing you read before bed. Or at least the last *challenging* thing.

Part of the learning (especially the transfer to long-term memory) happens *after* you put the book down. Your brain needs time on its own, to do more processing. If you put in something new during that processing-time, some of what you just learned will be lost.

5 Drink water. Lots of it.

Your brain works best in a nice bath of fluid. Dehydration (which can happen before you ever feel thirsty) decreases cognitive function.

6 Talk about it. Out loud.

Speaking activates a different part of the brain. If you're trying to understand something, or increase your chance of remembering it later, say it out loud. Better still, try to explain it out loud to someone else. You'll learn more quickly, and you might uncover ideas you hadn't known were there when you were reading about it.

7 Listen to your brain.

Pay attention to whether your brain is getting overloaded. If you find yourself starting to skim the surface or forget what you just read, it's time for a break. Once you go past a certain point, you won't learn faster by trying to shove more in, and you might even hurt the process.

8 Feel something!

Your brain needs to know that this *matters*. Get involved with the stories. Make up your own captions for the photos. Groaning over a bad joke is *still* better than feeling nothing at all.

9 Create something!

Apply this to something new you're designing, or rework an older project. Just do *something* to get some experience beyond the exercises and activities in this book. All you need is a pencil and a problem to solve... a problem that might benefit from using JavaScript.

10 Get Sleep.

You've got to create a lot of new brain connections to learn to program. Sleep often; it helps.

Read Me

This is a learning experience, not a reference book. We deliberately stripped out everything that might get in the way of learning whatever it is we're working on at that point in the book. And the first time through, you need to begin at the beginning, because the book makes assumptions about what you've already seen and learned.

We teach the GOOD parts of JavaScript, and warn you about the BAD parts.

JavaScript is a programming language that didn't come up through the ivy leagues with plenty of time for academic peer review. JavaScript was thrown out into the world out of necessity and grew up in the early browser neighborhood. So, be warned: JavaScript has some great parts and some not so great parts. But, overall, JavaScript is brilliant, if you use it intelligently.

In this book, we teach you to use the great parts to best advantage, and we'll point out the bad parts, and advise you to drive around them.

We don't exhaustively cover every single aspect of the language.

There's a lot you can learn about JavaScript. This book is not a reference book; it's a learning book, so it doesn't cover everything there is to know about JavaScript. Our goal is to teach you the fundamentals of using JavaScript so that you can pick up any old reference book and do whatever you want with JavaScript.

This book does teach you JavaScript in the browser.

The browser is not only the most common environment that JavaScript runs in, it's also the most convenient (everyone has a computer with a text editor and a browser, and that's all you need to get started with JavaScript). Running JavaScript in the browser also means you get instant gratification: you can write code and all you have to do is reload your web page to see what it does.

This book advocates well-structured and readable code based on best practices.

You want to write code that you and other people can read and understand, code that will still work in next year's browsers. You want to write code in the most straight-forward way so you can get the job done and get on to better things. In this book we're going to teach you to write clear, well-organized code that anticipates change from the get-go. Code you can be proud of, code you'll want to frame and put on the wall (just take it down before you bring your date over).

We encourage you to use more than one browser with this book.

We teach you to write JavaScript that is based on standards, but you're still likely to encounter minor differences in the way web browsers interpret JavaScript. While we'll do our best to ensure all the code in the book works in all modern browsers, and even show you a couple