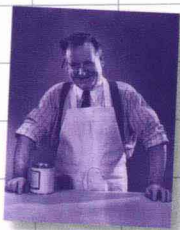


深入浅出移动互联网 (影印版)

Head First Mobile Web



Build
once, run
everywhere

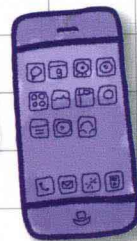


Be more supportive
(of your users)



Find your
way with
geolocation

Put your
pages on a
all-screen diet



Shape-shift your
sites with Responsive
Web Design

O'REILLY® 东南大學 出版社

Lyza Danger Gardner
& Jason Grigsby 著

深入浅出移动互联网 (影印版)

Head First Mobile Web

Wouldn't it be dreamy if there were a book to help me learn how to build mobile websites that was more fun than going to the dentist? It's probably nothing but a fantasy...



Lyza Danger Gardner
Jason Grigsby 著

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

深入浅出移动互联网：英文 / (美)加德纳(Gardner, L.D.),
(美)克莱格斯(Grigsby, J.)著 . —影印本 . —南京：东南
大学出版社，2012.6

书名原文：Head First Mobile Web

ISBN 978-7-5641-3414-3

I. ①深… II. ①加… ②克… III. ①移动通信—互
联网络—英文 IV. ① TN929.5

中国版本图书馆 CIP 数据核字 (2012) 第 065811 号

江苏省版权局著作权合同登记

图字：10-2012-3 号

©2011 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press,
2012. 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 2012。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。

深入浅出移动互联网（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：扬中市印刷有限公司

开 本：787 毫米 × 980 毫米 12 开本

印 张：39.5

字 数：661 千字

版 次：2012 年 6 月第 1 版

印 次：2012 年 6 月第 1 次印刷

书 号：ISBN 978-7-5641-3414-3

定 价：98.00 元（册）

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

Advance Praise for *Head First Mobile Web*

"If you have been considering buying a book about mobile development that is cross-browser and cross-vendor, you should stop right now and buy *Head First Mobile Web*. It's written by amazingly smart people [who] have great experience on mobile and don't stop at one platform, but work on all of them. Many developers spend days arguing [whether] they should go native or web. This book smoothly goes from introductory topics to advanced ones, giving you all the needed information to create exciting content for mobile."

— **Andrea Trasatti, leader of the DeviceAtlas project and cocreator of the WURFL repository of wireless device capability information**

"A pragmatic introduction to the chaotic world of mobile web development as it is today, with a glimpse of how we can and should approach it for tomorrow. *Head First Mobile Web* successfully presents practical techniques all readers can use immediately, while giving plenty of foundation and resources for more experienced developers to build upon."

— **Stephen Hay, web designer, developer, speaker, and cofounder of the Mobilism conference**

"Hands-on from the get-go, *Head First Mobile Web* provides an excellent introduction to the challenges and opportunities available when exploring the next chapter in web design."

— **Bryan and Stephanie Rieger, founders of yiibu.com**

Praise for other *Head First* books

“*Head First Object-Oriented Analysis and Design* is a refreshing look at subject of OOAD. What sets this book apart is its focus on learning. The authors have made the content of OOAD accessible [and] usable for the practitioner.”

— **Ivar Jacobson, Ivar Jacobson Consulting**

“I just finished reading HF OOA&D, and I loved it! The thing I liked most about this book was its focus on why we do OOA&D—to write great software!”

— **Kyle Brown, Distinguished Engineer, IBM**

“Hidden behind the funny pictures and crazy fonts is a serious, intelligent, extremely well-crafted presentation of OO analysis and design. As I read the book, I felt like I was looking over the shoulder of an expert designer who was explaining to me what issues were important at each step, and why.”

— **Edward Sciore, Associate Professor, Computer Science Department, Boston College**

“All in all, *Head First Software Development* is a great resource for anyone wanting to formalize their programming skills in a way that constantly engages the reader on many different levels.”

— **Andy Hudson, Linux Format**

“If you’re a new software developer, *Head First Software Development* will get you started off on the right foot. And if you’re an experienced (read: long-time) developer, don’t be so quick to dismiss this....”

— **Thomas Duff, Duffbert’s Random Musings**

“There’s something in *Head First Java* for everyone. Visual learners, kinesthetic learners, everyone can learn from this book. Visual aids make things easier to remember, and the book is written in a very accessible style—very different from most Java manuals.... *Head First Java* is a valuable book. I can see the *Head First* books used in the classroom, whether in high schools or adult ed classes. And I will definitely be referring back to this book, and referring others to it as well.”

— **Warren Kelly, Blogcritics.org, March 2006**

“Rather than textbook-style learning, *Head First iPhone and iPad Development* brings a humorous, engaging, and even enjoyable approach to learning iOS development. With coverage of key technologies including core data, and even crucial aspects such as interface design, the content is aptly chosen and top-notch. Where else could you witness a fireside chat between a UIWebView and UITextField!”

— **Sean Murphy, iOS designer and developer**

Praise for other *Head First* books

“Another nice thing about *Head First Java*, Second Edition, is that it whets the appetite for more. With later coverage of more advanced topics such as Swing and RMI, you just can’t wait to dive into those APIs and code that flawless, 100,000-line program on java.net that will bring you fame and venture-capital fortune. There’s also a great deal of material, and even some best practices, on networking and threads—my own weak spot. In this case, I couldn’t help but crack up a little when the authors use a 1950s telephone operator—yeah, you got it, that lady with a beehive hairdo that manually hooks in patch lines—as an analogy for TCP/IP ports...you really should go to the bookstore and thumb through *Head First Java*, Second Edition. Even if you already know Java, you may pick up a thing or two. And if not, just thumbing through the pages is a great deal of fun.”

— **Robert Eckstein, Java.sun.com**

“Of course it’s not the range of material that makes *Head First Java* stand out, it’s the style and approach. This book is about as far removed from a computer science textbook or technical manual as you can get. The use of cartoons, quizzes, fridge magnets (yep, fridge magnets...). And, in place of the usual kind of reader exercises, you are asked to pretend to be the compiler and compile the code, or perhaps to piece some code together by filling in the blanks or...you get the picture.... The first edition of this book was one of our recommended titles for those new to Java and objects. This new edition doesn’t disappoint and rightfully steps into the shoes of its predecessor. If you are one of those people who falls asleep with a traditional computer book, then this one is likely to keep you awake and learning.”

— **TechBookReport.com**

“*Head First Web Design* is your ticket to mastering all of these complex topics, and understanding what’s really going on in the world of web design.... If you have not been baptized by fire in using something as involved as Dreamweaver, then this book will be a great way to learn good web design.”

— **Robert Pritchett, MacCompanion**

“Is it possible to learn real web design from a book format? *Head First Web Design* is the key to designing user-friendly sites, from customer requirements to hand-drawn storyboards to online sites that work well. What sets this apart from other ‘how to build a website’ books is that it uses the latest research in cognitive science and learning to provide a visual learning experience rich in images and designed for how the brain works and learns best. The result is a powerful tribute to web design basics that any general-interest computer library will find an important key to success.”

— **Diane C. Donovan, California Bookwatch: The Computer Shelf**

“I definitely recommend *Head First Web Design* to all of my fellow programmers who want to get a grip on the more artistic side of the business.”

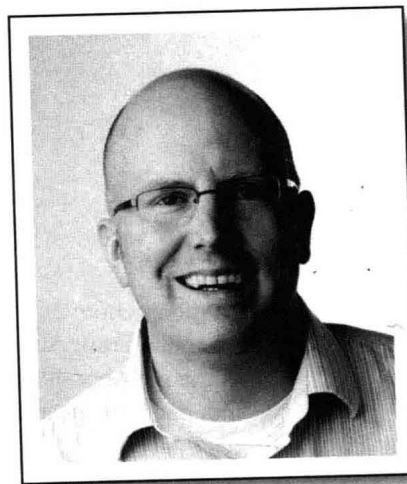
— **Claron Twitchell, UJUG**

To the phenomenal women in my family: my sister, Maggie; Momula, Fran; Aunt Catherine; stepmother, Christie; and above all, to the memory of my grandmother, Pearl, whose fierce and literate independence inspired generations.

—Lyza

To my parents for buying that Commodore 64 so many years ago; to my lovely wife, Dana, without whose support and understanding this book wouldn't have happened; and to Katie and Danny—yes, Daddy can play now.

—Jason



← Jason

Lyza Danger Gardner (@lyzadanger) is a dev. She has built, broken, and hacked web things since 1996. Curiously, Lyza was actually born and raised in Portland, Oregon, the town where everyone wants to be but no one seems to be *from*.

Lyza started college early and cobbled together a motley education: a BA in Arts and Letters from Portland State University, followed by a master's program in computer science at the University of Birmingham (UK).

Lyza has written a lot of web applications (server-side devs, represent!), defeated wily content management systems, optimized mobile websites, pounded on various APIs, and worried a lot about databases. Fascinated by the way mobile technology has changed things, she now spends a lot of time thinking about the future of the Web, mobile and otherwise.

Since cofounding Cloud Four, a Portland-based mobile web agency, in 2007, Lyza has voyaged further into the deep, untrammled reaches of Device Land, exploring the foibles and chaos of mobile browsers and the mobile web. She has an odd set of anachronistic hobbies, and it has been said that she takes a fair number of photographs. She owns a four-letter .com domain. We'll bet you can guess what it is and go visit her there.

In 2000, **Jason Grigsby** got his first mobile phone. He became obsessed with how the world could be a better place if everyone had access to the world's information in their pockets. When his wife, Dana, met him, he had covered the walls of his apartment with crazy mobile dreams. To this day, he remains baffled that she married him.

Those mobile dreams hit the hard wall of reality—WAP was crap. So Jason went to work on the Web until 2007, when the iPhone made it clear the time was right. He joined forces with the three smartest people he knew and started Cloud Four.

Since cofounding Cloud Four, he has had the good fortune to work on many fantastic projects, including the Obama iPhone App. He is founder and president of Mobile Portland, a local nonprofit dedicated to promoting the mobile community in Portland, Oregon.

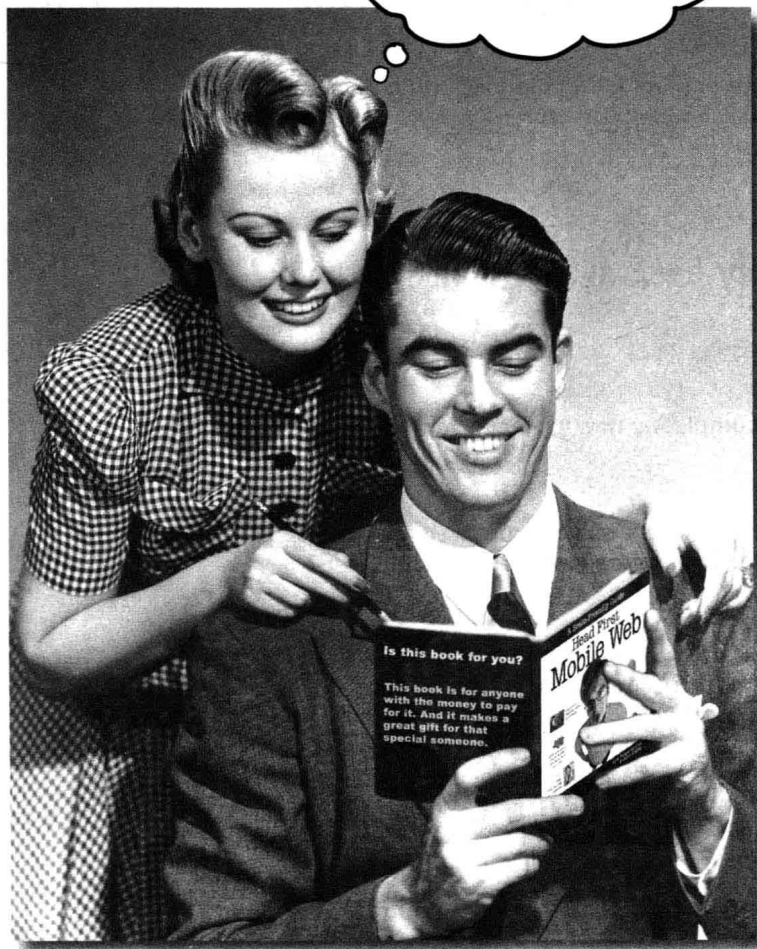
Jason is a sought-after speaker and consultant on mobile. If anything, he is more mobile obsessed now than he was in 2000 (sorry, sweetheart!).

You can find him blogging at <http://cloudfour.com>; on his personal site, <http://userfirstweb.com>; and on Twitter as @grigs. Please say hello!

how to use this book

Intro

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



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

Who is this book for?

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

- ① Do you have previous web design and development experience?
- ② Do you want to **learn, understand, remember,** and **apply** important mobile web concepts so that you can make your mobile web pages more interactive and exciting?
- ③ Do you prefer **stimulating dinner-party conversation** to **dry, dull, academic lectures**?

this book is for you.

It definitely helps if you've already got some scripting chops, too. We're not talking rocket science, but you shouldn't feel visceral panic if you see a JavaScript snippet.

Who should probably back away from this book?

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

- ① Are you **completely new** to web development?
- ② Are you already developing mobile web apps or sites and looking for a **reference book** on mobile web?
- ③ Are you **afraid to try something different**? Would you rather have a root canal than endure the suggestion that there might be more than one true way to build for the Web? Do you believe that a technical book can't be serious if there's a walrus-themed pub and an app called the Tartanator in it?

this book is not for you.

[Note from marketing: this book is for anyone with a credit card. Or cash. Cash is nice, too. — Ed]



We know what you're thinking

“How can *this* be a serious mobile web development 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.

So what does your brain do with all the routine, ordinary, normal things you encounter? Everything it *can* 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*.

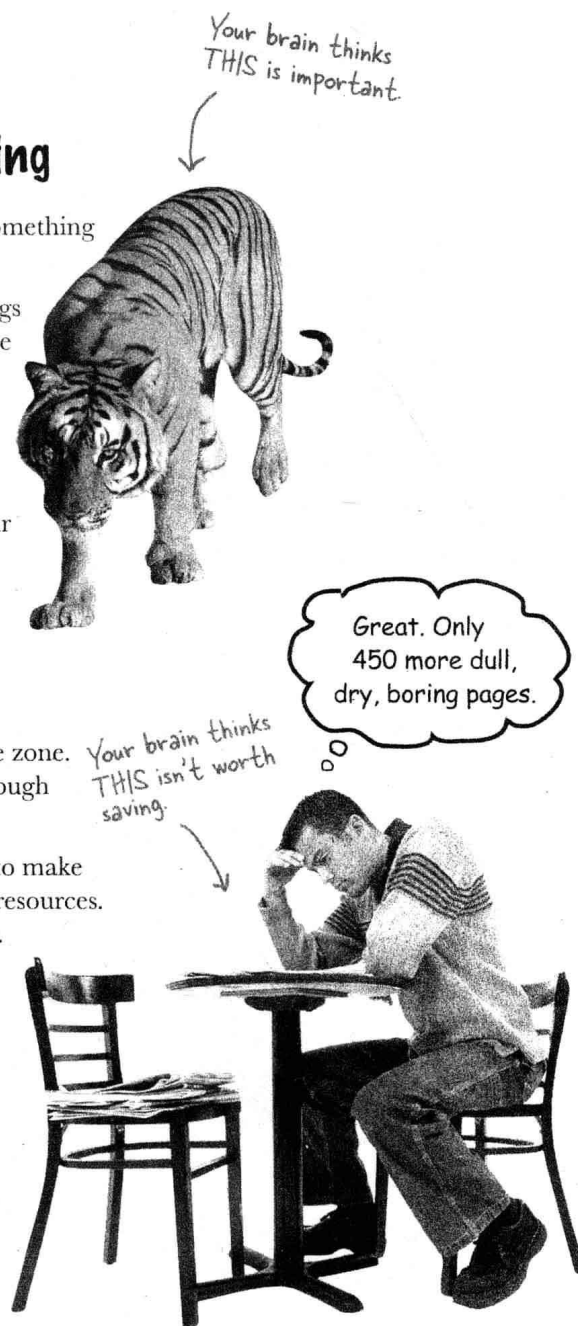
And that’s how your brain knows...

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, 10 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* nonimportant 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.

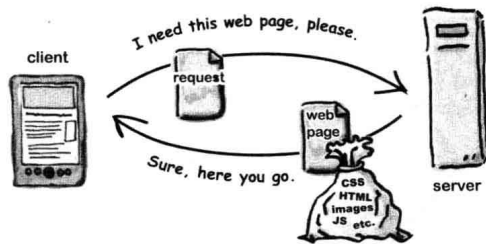
And there’s no simple way to tell your brain, “Hey brain, thank you very much, but no matter how dull this book is, and how little I’m registering on the emotional Richter scale right now, I really *do* want you to keep this stuff around.”



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

So what does it take to *learn* something? First, you have to *get* it, and 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.

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.



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* to learn.

But we assume that if you're holding this book, you really want to learn about mobile web development. And you probably don't want to spend a lot of time. And since you're going to build more sites and apps in the future, you need to *remember* what you read. 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.

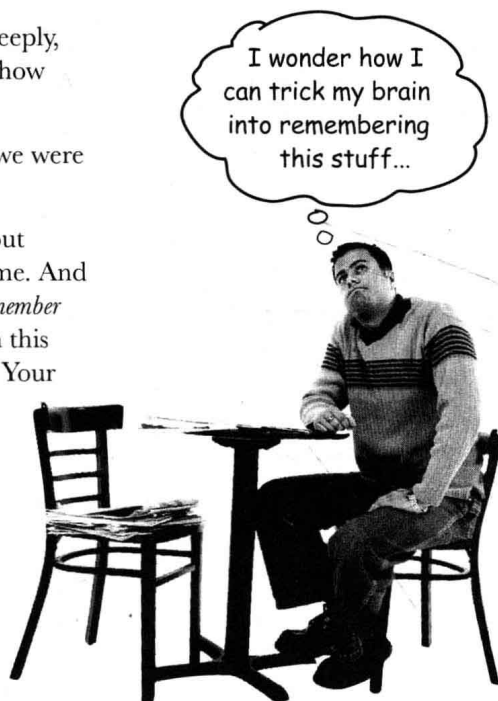
So just how do you get your brain to think that mobile web development is a hungry tiger?

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 the same thing into your brain. 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 makes 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 a thousand 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 loads of **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-doable, because that's what most people prefer.

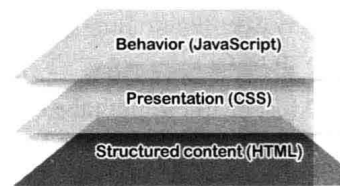
We used **multiple learning styles**, because *you* might prefer step-by-step procedures, while someone else wants to understand the big picture first, and someone else just wants to see an example. But regardless of your own learning preference, *everyone* benefits from seeing the same content represented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

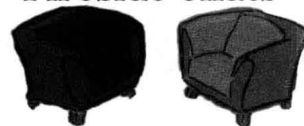
And we included **stories** and exercises that present **more than one point of view**, because your brain is tuned to learn more deeply when it's forced to make evaluations and judgments.

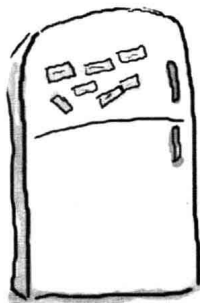
We included **challenges**, with exercises, and by asking **questions** that don't always have a straight answer, because your brain is tuned to learn and remember when it has to *work* at something. Think about it—you can't get your *body* in shape just by *watching* people at the gym. But we did our 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*.



Fireside Chats





Cut this out and stick it on your refrigerator.

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.

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 "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 your daily work; use what you are learning to make decisions on your projects. 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 the tools and techniques you're studying for the exam.

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 expect you to know HTML and CSS.

If you don't know HTML and CSS, pick up a copy of *Head First HTML with CSS & XHTML* before starting this book. We'll explain some of the more obscure CSS selectors or HTML elements, but don't expect to learn about that foundational stuff here.

We expect you to feel comfy around web scripting code.

We're not asking you to be a world-class JavaScript expert or to have done a graduate computer science project using PHP, but you'll see examples using both languages throughout the book. If the merest notion of a `for` loop makes you hyperventilate (or if you have no idea what we're talking about), you might consider tracking down a copy of *Head First PHP & MySQL* or *Head First JavaScript* and then heading on back here.

We expect you to know how to track things down.

We'll be blunt. The mobile web is an enormous topic, and mastering it involves expanding your *existing* web development skills. There are too many things to know about the Web for any one person to memorize, whether it's a detail of JavaScript syntax or the specifics of a browser's support for an HTML5 element attribute. Don't be too hard on yourself. Part of the toolset of a good web dev is keeping your Google chops sharp and knowing when and how to hit the Web to look up info about web topics. We bet you're good at that already.

We expect you to go beyond this book.

It's a big and beautiful mobile web world out there. We hope we can give you a shove to start you on your journey, but it's up to you to keep up your steam. Seek out the active mobile web community online, read blogs, join mailing lists that are up your alley, and attend related technical events in your area.

The activities are NOT optional.

The exercises and activities are not add-ons; they're part of the core content of the book. Some of them are to help with memory, some are for understanding, and some will help you apply what you've learned. ***Don't skip the exercises.*** They're good for giving your brain a chance to think about the ideas and terms you've been learning in a different context.

The redundancy is intentional and important.

One distinct difference in a Head First book is that we want you to *really* get it. And we want you to finish the book remembering what you've learned. Most reference books don't have retention and recall as a goal, but this book is about *learning*, so you'll see some of the same concepts come up more than once.

The Brain Power exercises don't have answers.

For some of them, there is no right answer, and for others, part of the learning experience of the exercise is for you to decide if and when your answers are right. In some of the Brain Power exercises, you will find hints to point you in the right direction.

Software requirements

As for developing any website, you need a text editor, a browser, a web server (it can be locally hosted on your personal computer), and the source code for the chapter examples.

The text editors we recommend for Windows are PSPad, TextPad, or EditPlus (but you can use Notepad if you have to). The text editors we recommend for Mac are TextWrangler (or its big brother, BBEdit) or TextMate. We also like Coda, a more web-focused tool.

If you're on a Linux system, you've got plenty of text editors built in, and we trust you don't need us to tell you about them.

If you are going to do web development, you need a web server. You'll need to go to Appendix ii, which details installing a web server with PHP. We recommend doing that now. No, seriously, head there now, follow the instructions, and come back to this page when you're done.

For Chapter 5, you'll need to install the WURFL (Wireless Universal Resource File) API and data. And for Chapter 8, you'll need the Android SDK and some related tools. You guessed it: there are appendixes for those tasks, too.

You'll also need a browser—no, strike that—**as many browsers as possible** for testing. And the more **mobile devices with browsers** you have on hand, the better (don't panic; there are many emulators you can use if you don't have hardware).

For developing and testing on the desktop, we highly recommend Google's Chrome browser, which has versions for Mac, Windows, and Linux. Learning how to use the JavaScript console in Google's Chrome Dev Tools is well worth the time. This is homework you need to do on your own.

Last of all, you'll need to get the code and resources for the examples in the chapters. It's all available at <http://hf-mw.com>.

↑
The hf-mw.com site has the starting point of code for all the chapters. Head on over there and get downloading.

The code and resources for the examples in the chapters are all available at <http://hf-mw.com>.