

## How to use this book *Intro*



---

### NOTE

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

---

# Who is this book for?

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

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

This book is for you.

## Who should probably back away from this book?

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

- ❶ Are you completely new to web development?
- ❷ 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.
- ❸ Are you a kick-butt web developer looking for a *reference book*?
- ❹ 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.

---

### NOTE

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

---



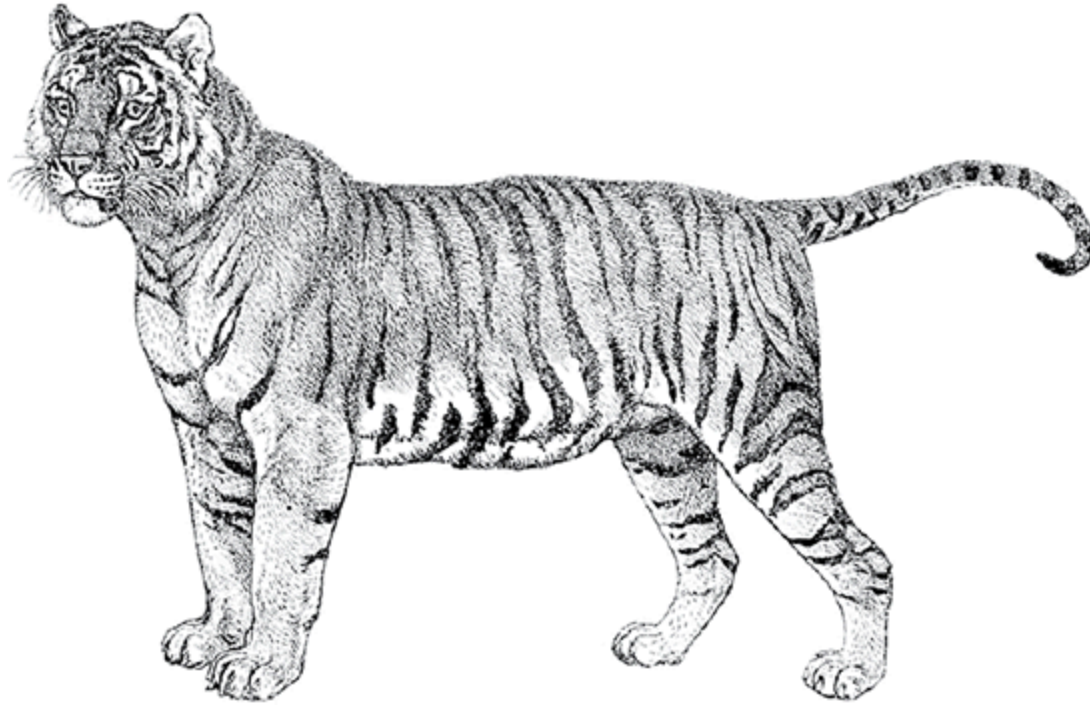
## We know what you're thinking

“How can *this* be a serious JavaScript book?”

“What’s with all the graphics?”

“Can I actually *learn* it this way?”

Your brain thinks  
*THIS* is important.



## We know what your *brain* is thinking

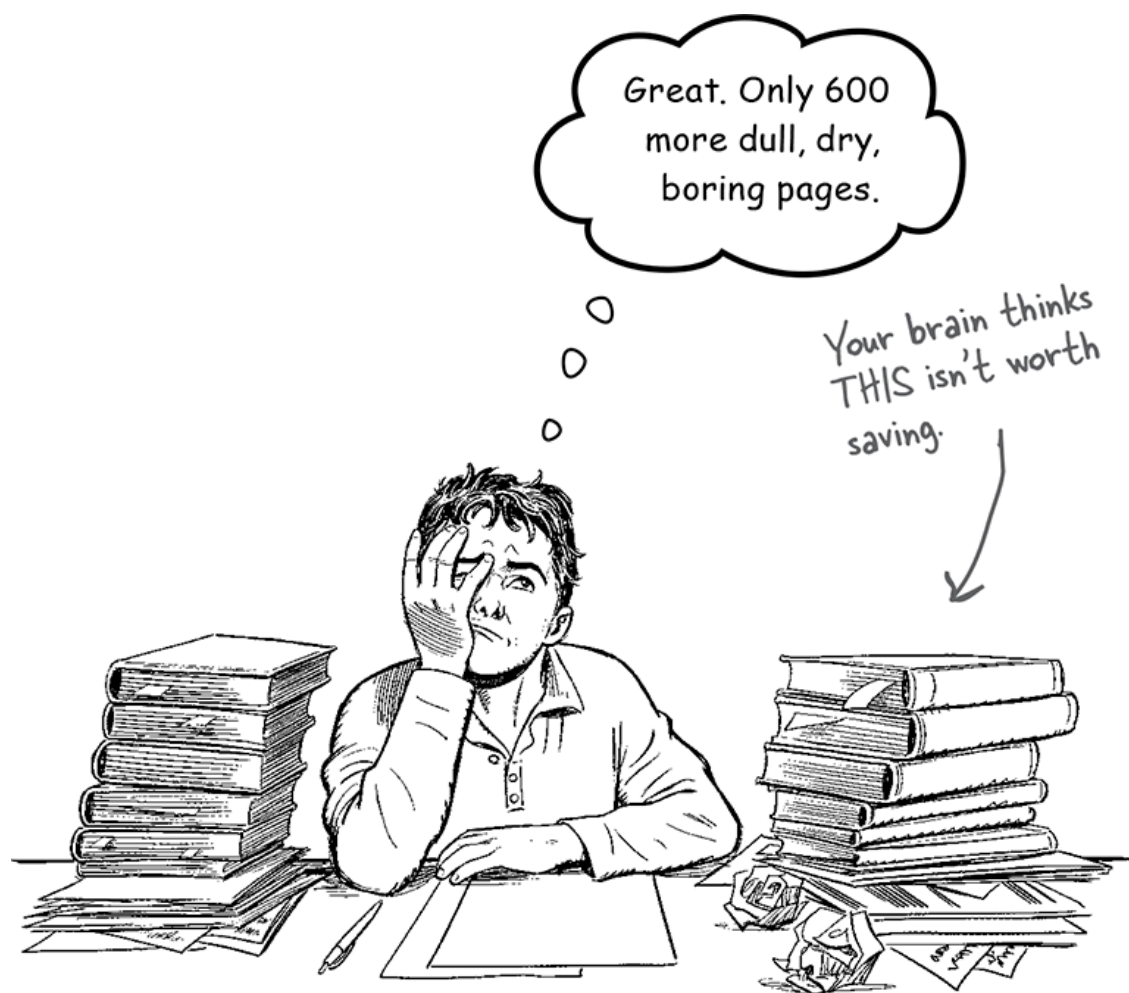
Your brain craves novelty. It's always searching, scanning, *waiting* for something unusual. It was built that way, and it helps you stay alive.

So what does your brain do with all the routine, ordinary, normal things you encounter? Everything it *can* 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, 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* 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 have posted those "party" photos on your Facebook page. 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 emo-

tional Richter scale right now, I really *do* want you to keep this stuff around.”

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). They also make 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 be able 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.** 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

I wonder how  
I can trick my brain  
into remembering  
this stuff...



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 how to design user-friendly websites. And you probably don't want to spend a lot of time on it. If you want to use what you read in this book, 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 treat JavaScript like it was 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, but they keep 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 on 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 **visuals**, because your brain is tuned for visuals, not text. As far as your brain's concerned, a visual really *is* worth a thousand words. And when text and visuals work together, we embedded the text *in* the visuals 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 a paragraph 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 visuals in **unexpected** ways because your brain is tuned for novelty, and we used visuals and ideas with at least *some emotional content*, because your brain is tuned to pay attention to the bio-chemistry 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 80 **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 we asked **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, visuals, etc., because, well, because *you're* a person. And your brain pays more attention to *people* than it does to *things*.



## 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.

---

### NOTE

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 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.

**6 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.

**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 Write a lot of code!**

There's only one way to learn to program: **writing a lot of code**. And that's what you're going to do throughout this book. Coding is a skill, and the only way to get good at it is to practice. We're going to give you a lot of practice: every chapter has exercises that pose a problem for you to solve. Don't just skip over them—a lot of the learning happens when you solve the exercises. We included a solution to each exercise—don't be afraid to **peek at the solution** if you get stuck! (It's easy to

get snagged on something small.) But try to solve the problem before you look at the solution. And definitely get it working before you move on to the next part of the book.

## 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 modern JavaScript.**

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. JavaScript has been through quite the journey to get to where it is today: a fast, brilliant, modern language that is the programming language of the web.

In this book, we explore modern JavaScript, covering the updated syntax and semantics that make coding more efficient and effective. We provide practical examples and best practices to help you master today's JavaScript and build dynamic, robust applications.

### **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 straightforward 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).

**Programming is serious business. You're going to have to work, sometimes hard.**

If you've already had some programming experience, then you know what we're talking about. If you're coming straight from *Head First HTML and CSS*, then you're going to find writing code is a little, well, different. Programming requires a different way of thinking. Programming is logical, at times very abstract, and requires you to think in an algorithmic way. But no worries; we're going to do all that in a brain-friendly way. Just take it a bit at a time, make sure you're well nourished, and get plenty of sleep.

That way, these new programming concepts will really sink in.

**The activities are NOT optional.**

The exercises and activities in this book 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. The crossword puzzles are the only

things you don't have to do, but they're good for giving your brain a chance to think about the words 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 examples are as lean as possible.**

Our readers tell us that it's frustrating to wade through 200 lines of an example looking for the 2 lines they need to understand. Most examples in this book are shown within the smallest possible context, so that the part you're trying to learn is clear and simple. Don't expect all of the examples to be robust, or even complete—they are written specifically for learning, and aren't always fully functional.

We've placed all the example files on the web so you can download them. You'll find them at <http://wickedlysmart.com/hfjs>.

### **The Brain Power exercises don't usually have answers.**

For some of them, there is no right answer, and for others, part of the learning experience 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.

### **We often give you only the code, not the markup.**

After we get past the first chapter or two, we often give you just the JavaScript code and assume you'll wrap it in a nice HTML wrapper. Here's a simple HTML page you can use with most of the code in this book. If we want you to use other HTML, we'll tell you:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title>Your HTML Page</title>
```

```
    <script>
```

← Your JavaScript code will typically go here.

```
  </script>
```

```
</head>
```

```
<body>
```

← Any web page content will go here.

```
</body>
```

```
</html>
```



But don't worry; at the beginning of the book we'll take you through everything.

**Get the code examples, help, and discussion.**

You'll find everything you need for this book online at  
<http://wickedlysmart.com/hfjs>, including code sample files.

# Tech reviewers



Jeff Straw



Ismaël Martin "Bing" Demiddel



Frank D. Moore

These guys really rocked it; they were there for us throughout the review process and provided invaluable, detailed feedback on everything!



Alfred J. Speller



Bruce Forkush



Javier Ruedas

## *Thank you to our amazing review team*

This book has been more carefully reviewed than any of our previous books. Over 270 people participated in reading and critiquing the first edition of this book in real time as we wrote it. This worked better than we ever imagined and was instrumental in shaping every aspect of *Head First JavaScript Programming*. Our heartfelt thanks to everyone who participated; the book is so much better because of you.

For the first edition, the amazing technical reviewers pictured above provided feedback above and beyond, and each made significant contributions to the book. The following reviewers also made contributions across different aspects of the book: **Galina N. Orlova, J. Patrick Kelley, Claus-Peter Kahl, Rob Cleary, Rebeca Dunn-Krahn, Olaf Schoenrich, Jim Cupec, Matthew M. Hanrahan, Russell Alleen-Willems, Christine J. Wilson, Louis-Philippe Breton, Timo Glaser, Charmaine Gray, Lee Beckham, Michael Murphy, Dave Young, Don Smallidge, Alan Rusyak, Eric R. Liscinsky, Brent Fazekas, Sue Starr, Eric (Orange Pants) Johnson, Jesse Palmer, Manabu Kawakami, Alan McIvor, Alex Kelley,**

Yvonne Bichsel Truhon, Austin Throop, Tim Williams, J. Albert Bowden II, Rod Shelton, Nancy DeHaven Hall, Sue McGee, Francisco Debs, Miriam Berkland, Christine H Grecco, Elhadji Barry, Athanasios Valsamakis, Peter Casey, Dustin Wollam, and Robb Kerley.

For the second edition, we had a fabulous group of technical reviewers who provided valuable feedback as we updated this book to include new features added to the language since the first edition. Thank you to Adam Scott, Maya Shavin, Stoyan Stefanov, and Edward Wong for your feedback.

# Acknowledgments for the first edition

↙ Esteemed Reviewer,  
David Powers



↖ Don't let the smile fool  
you, this guy is hard-core  
(technically, of course).

We're also extremely grateful to our esteemed technical reviewer, **David Powers**. The truth is we don't write books without David anymore, he's just saved our butts too many times. It's getting a little like Elton and Bernie; we're starting to ask ourselves if we actually could write a book without him. David ~~helps us~~ forces us to make the book more sound and technically accurate, and his second career as a standup comic really



comes in handy when we're tuning the more humorous parts of the book. Thank you once again, David—you're the ultimate professional, and we sleep better at night knowing we've passed your technical muster.



Meghan Blanchette

A huge, massive thanks to our editor for the first edition, **Meghan Blanchette**, who cleared the path for this book, removed every obstacle to its completion, waited patiently, and sacrificed family time to get it done. She helped keep us sane during the book writing process, which is probably the best thing an editor can do for an author.





Mike Hendrickson

And another big shoutout to our Chief Editor Emeritus, **Mike Hendrickson**, who spearheaded this book from the very beginning. Thanks again, Mike; none of our books would have happened without you. You've been our champion for well over a decade and we love you for it!

## Acknowledgments for the second edition

### *Our editor:*

Our editor for the second edition, **Michele Cronin**, helped us through every stage of updating this book and put up with our questions, delays, and asks with the utmost grace. Once again, we were lucky to have an editor who helped keep us sane while being patient throughout the process.

Thank you so much, Michele!



↑ Michele Cronin

***The O'Reilly team:***

It takes a whole team of people to write a book, and we couldn't do it without everyone. Thank you especially to:

**Mary Tresseler, Melissa Duffield, David Michelson, Kristen Brown, Chris Faucher, Rachel Head, and Ellie Volkhausen.**

---

**NOTE**

The large number of acknowledgments is because we're testing the theory that everyone mentioned in a book acknowledgment will buy at least one copy, probably more, what with relatives and everything. If you'd like to be in the acknowledgments of our next book, and you have a large family, write to us.

---