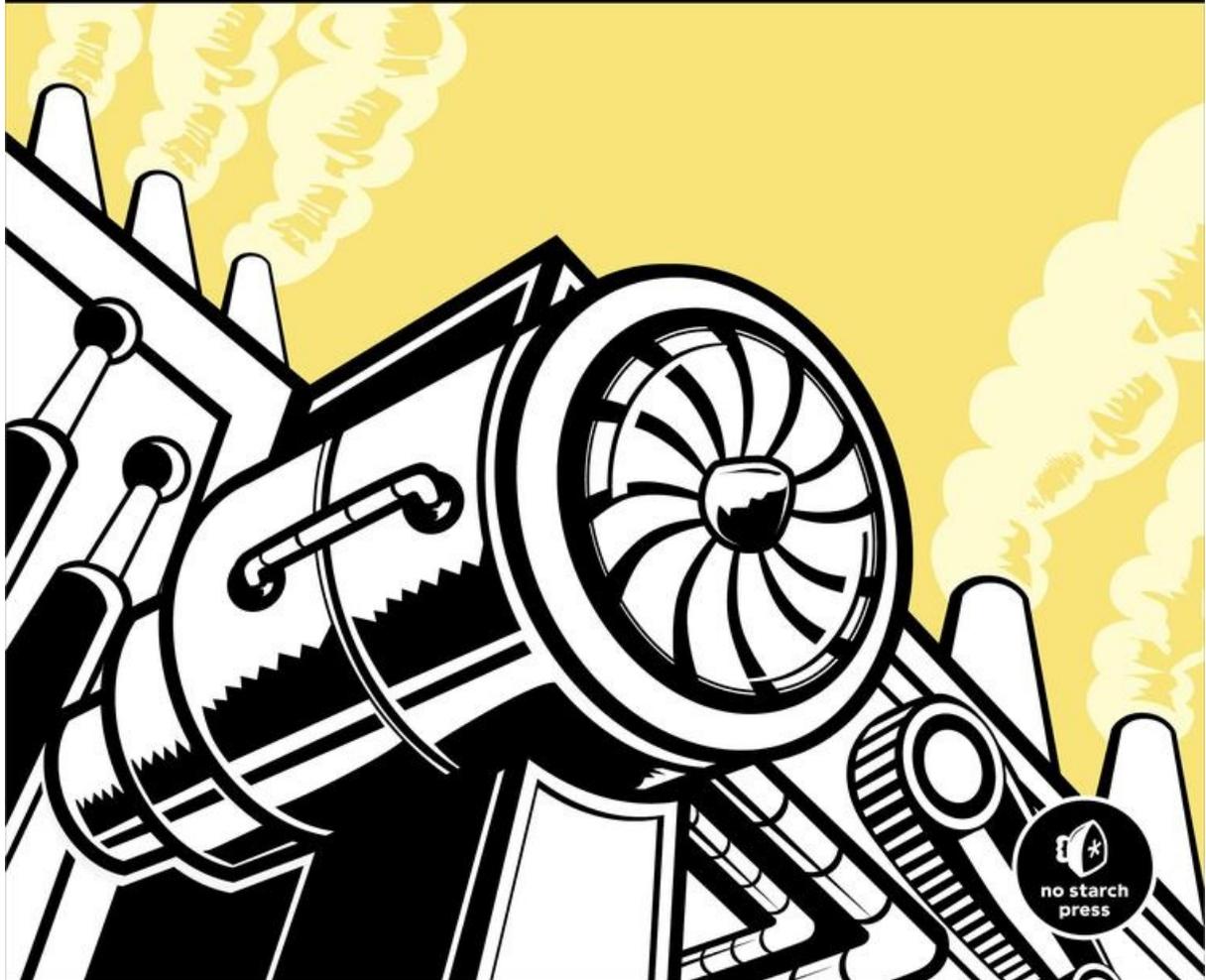


# THE PRINCIPLES OF OBJECT-ORIENTED JAVASCRIPT

NICHOLAS C. ZAKAS



# **The Principles of Object-Oriented Javascript**

**Nicholas C. Zakas**

Published by No Starch Press

---

## About the Author

Nicholas C. Zakas is a software engineer at Box and is known for writing on and speaking about the latest in JavaScript best practices. He honed his experience during his five years at Yahoo!, where he was principal frontend engineer for the Yahoo! home page. He is the author of several books, including *Maintainable JavaScript* (O'Reilly Media, 2012) and *Professional JavaScript for Web Developers* (Wrox, 2012).

## About the Technical Reviewer

Originally from the UK, Angus Croll is now part of Twitter's web framework team in San Francisco and is the co-author and principal maintainer of Twitter's open source Flight framework. He's obsessed with JavaScript and literature in equal measure and is a passionate advocate for the greater involvement of artists and creative thinkers in software development. Angus is a frequent speaker at conferences worldwide and is currently working on two books for No Starch Press. He can be reached on Twitter at [@angustweets](https://twitter.com/angustweets).

# Foreword

---

The name Nicholas Zakas is synonymous with JavaScript development itself. I could ramble on for pages with his professional accolades, but I am not going to do that. Nicholas is well-known as a highly skilled JavaScript developer and author, and he needs no introduction. However, I would like to offer some personal thoughts before praising the contents of this book.

My relationship with Nicholas comes from years of studying his books, reading his blog posts, watching him speak, and monitoring his Twitter updates as a JavaScript pupil. We first met in person when I asked him to speak at a jQuery conference several years ago. He treated the jQuery community to a high-quality talk, and since then, we have spoken publicly and privately over the Internet. In that time, I have come to admire him as more than just a leader and developer in the JavaScript community. His words are always gracious and thoughtful, his demeanor always kind.

His intent as a developer, speaker, and author is always to help, to educate, and to improve. When he speaks, you should listen, not just because he is a JavaScript expert, but because his character rises above his professional status.

This book's title and introduction make Nicholas's intentions clear: he has written it to help class-minded (that is, C++ or Java) programmers transition to a language without classes. In the book, he explains how encapsulation, aggregation, inheritance, and polymorphism can be accomplished when writing JavaScript. This is the ideal text to bring a knowledgeable programmer into the fold of object-oriented JavaScript development. If you are reading this book as a developer from another language, you are about to be treated to a concise and skillfully worded JavaScript book.

However, this book also stands to serve programmers coming from within the JavaScript fold. Many JavaScript developers have only an ECMAScript 3 (ES3) understanding of objects, and they are in need of a proper introduction to ECMAScript 5 (ES5) object features. This book can serve as that introduction, bridging a knowledge gap between ES3 objects and ES5 objects.

Now, you might be thinking, "Big deal. Several books have included chapters or notes on the additions to JavaScript found in ES5." Well, that is true. However, I believe this to be the only book written to date that focuses on the nature of objects by giving ES5 objects first-class citizenship in the entire narrative. This book brings a cohesive introduction to not only ES5 objects, but also the bits of ES3 that you need to grok while learning many of the new additions found in ES5.

As an author myself, I strongly believe this is the one book, given its focus on object-oriented principles and ES5 object updates, that needed to be written as we await ES6

updates to scripting environments.

Cody Lindley ([www.codylindley.com](http://www.codylindley.com))

Author of *JavaScript Enlightenment*, *DOM Enlightenment*, and *jQuery Enlightenment*

Boise, Idaho

December 16, 2013

# Acknowledgments

---

I'd like to thank Kate Matsudaira for convincing me that self-publishing an ebook was the best way to get this information out. Without her advice, I'd probably still be trying to figure out what I should do with the information contained in this book.

Thanks to Rob Friesel for once again providing excellent feedback on an early copy of this book, and Cody Lindley for his suggestions. Additional thanks to Angus Croll for his technical review of the finished version— his nitpicking made this book much better.

Thanks as well to Bill Pollock, whom I met at a conference and who started the ball rolling on publishing this book.

# Introduction

---

Most developers associate object-oriented programming with languages that are typically taught in school, like C++ and Java, which base object-oriented programming around classes. Before you can do anything in these languages, you need to create a class, even if you're just writing a simple command-line program.

Common design patterns in the industry reinforce class-based concepts as well. But JavaScript doesn't use classes, and this is part of the reason people get confused when they try learning it after C++ or Java.

Object-oriented languages have several characteristics:

**Encapsulation.** Data can be grouped together with functionality that operates on that data. This, quite simply, is the definition of an object.

**Aggregation.** One object can reference another object.

**Inheritance.** A newly created object has the same characteristics as another object without explicitly duplicating its functionality.

**Polymorphism.** One interface may be implemented by multiple objects.

JavaScript has all these characteristics, though because the language has no concept of classes, some aren't implemented in quite the way you might expect. At first glance, a JavaScript program might even look like a procedural program you would write in C. If you can write a function and pass it some variables, you have a working script that seemingly has no objects. A closer look at the language, however, reveals the existence of objects through the use of dot notation.

Many object-oriented languages use dot notation to access properties and methods on objects, and JavaScript is syntactically the same. But in JavaScript, you never need to write a class definition, import a package, or include a header file. You just start coding with the data types that you want, and you can group those together in any number of ways. You could certainly write JavaScript in a procedural way, but its true power emerges when you take advantage of its object-oriented nature. That's what this book is about.

Make no mistake: A lot of the concepts you may have learned in more traditional object-oriented programming languages don't necessarily apply to JavaScript. While that often confuses beginners, as you read, you'll quickly find that JavaScript's weakly typed nature allows you to write less code to accomplish the same tasks as other languages. You can just start coding without planning the classes that you need ahead of time. Need an object with specific fields? Just create an ad hoc object wherever you want. Did you forget to add a method to that object? No problem—just add it later.

Inside these pages, you'll learn the unique way that JavaScript approaches object-oriented programming. Leave behind the notions of classes and class-based inheritance and learn about prototype-based inheritance and constructor functions that behave similarly. You'll learn how to create objects, define your own types, use inheritance, and otherwise manipulate objects to get the most out of them. In short, you'll learn everything you need to know to understand and write JavaScript professionally. Enjoy!

## Who This Book Is For

This book is intended as a guide for those who already understand object-oriented programming but want to know exactly how the concept works in JavaScript. Familiarity with Java, C#, or object-oriented programming in other languages is a strong indicator that this book is for you. In particular, this book is aimed at three groups of readers:

- Developers who are familiar with object-oriented programming concepts and want to apply them to JavaScript
- Web application and Node.js developers trying to structure their code more effectively
- Novice JavaScript developers trying to gain a deeper understanding of the language

This book is not for beginners who have never written JavaScript. You will need a good understanding of how to write and execute JavaScript code to follow along.

## Overview

**Chapter 1** introduces the two different value types in JavaScript: primitive and reference. You'll learn what distinguishes them from each other and how understanding their differences is important to an overall understanding of JavaScript.

**Chapter 2** explains the ins and outs of functions in JavaScript. First-class functions are what makes JavaScript such an interesting language.

**Chapter 3** details the makeup of objects in JavaScript. JavaScript objects behave differently than objects in other languages, so a deep understanding of how objects work is vital to mastering the language.

**Chapter 4** expands on the previous discussion of functions by looking more specifically at constructors. All constructors are functions, but they are used a little bit differently. This chapter explores the differences while also talking about creating your own custom types.

**Chapter 5** explains how inheritance is accomplished in JavaScript. Though there are no classes in JavaScript, that doesn't mean inheritance isn't possible. In this chapter, you'll learn about prototypal inheritance and how it differs from class-based inheritance.

**Chapter 6** walks through common object patterns. There are many different ways to build and compose objects in JavaScript, and this chapter introduces you to the most popular patterns for doing so.

## Help and Support

If you have questions, comments, or other feedback about this book, please visit the mailing list at <http://groups.google.com/group/zakasbooks>.