Extracted from:

Pragmatic Guide to JavaScript

Pragmatic Guide to
JavaScript

Christophe Porteneuve

Edited by David McClintock

PRAGMATIC GUIDE

Introduction

If you've been paying even minimal attention to JavaScript these past few years, you've heard this before: it's the Next Big Language. Once the province of half-baked implementations and useless scrolling messages, it has become a world-class, dynamic, object-oriented language with super-fast implementations on the client and server sides.

On the one hand, JavaScript's designers are endowing it with a new healthy dose of power, through the EcmaScript 5 (ES5) specification. On the other hand, kick-ass engines (such as V8, JavaScriptCore, SpiderMonkey, Rhino, and Carakan) and emergent standards and technologies (with CommonJS[1] and Node[2] in the lead) make it usable both in browsers and as stand-alone, powerful architectures on the server. Even the upcoming Internet Explorer 9 is upping its JavaScript game with the promise of huge speed boosts.

Not only is JavaScript a powerful, dynamic language, but it now has a rich ecosystem of professional-grade development tools, infrastructures, frameworks, and tool kits. It is versatile, fast, and very well suited to a wide range of programming tasks, especially when it comes to web-based applications and services.

It's time to dive in!

What's This Book About, and Who Is It For?

This book is not really intended to teach you "JavaScript the language." For one thing, the language itself is not very complicated, so if you have prior experience in programming any reasonably common language—even if it's just the basics (variables, loops, and so on)—you'll get your bearings easily enough. You don't need to actually know some JavaScript already (although it could help), and you certainly don't need to be any sort of programming guru.

Actually, if you're looking for the nitty-gritty and the hardcore technical details of JavaScript, you'll be better off reading a dedicated resource, such as

1. http://commonjs.org/
2. http://nodejs.org/

the "JavaScript core skills" section of Opera's excellent Web Standards Curriculum.[3] Should you ever need even more intricate, implementation-level details, you could then head to either the official specs of the language or one of the massive "bible" books such as David Flanagan's *JavaScript: The Definitive Guide* [Fla06].

This book aims to provide you with quick yet qualitative solutions to common client-side JavaScript-based tasks, from low-level stuff (such as getting a reference to a DOM element) to intricate features (such as Ajax-based autocompletion). This means we'll tackle JavaScript, CSS, the DOM, Ajax, JSON, and more. We won't go deep into the server side; this book is mostly on the client side of things (most often the browser). You'll encounter a couple of tiny PHP scripts along the way, for illustration purposes, but you could write your server side any way you like—including in JavaScript, for instance, with Node!

It's not just for copying and pasting, either. The text for each task takes care to highlight the key concepts, the potential gotchas, and the technical tricks you should take away from the task. Ultimately, you should step away from this book as a better JavaScript programmer.

This Book and JavaScript Libraries

Let's speak plainly here. If you're doing any sort of nontrivial JavaScript programming and you're not relying heavily on good, established frameworks for it, You're Doing It Wrong. On the browser side, effectively pulling off web page scripting is a challenge. You face obstacles from all sides: DOM inconsistencies, faulty language implementations, CSS quirks, weird Ajax bugs, and more. On the server side, once you have a runtime ready, you still face the enormous task of putting together the basic bricks of an application server such as a datastore, a network stack, a module system, and so on.

Fortunately, great people already solved these challenges for you. There's a wealth of options, too, especially on the client side; take a look at Appendix C, on page 118, for details on the main JavaScript frameworks.

Because any competent and *pragmatic* JavaScript developer will rely on one or more good frameworks, this book takes care to illustrate all the major client-side frameworks in what I think of as "basic" tasks. I selected Prototype, jQuery, MooTools, YUI, Dojo, and ExtJS, which should cover most of the "developer mind share" in this business.

---

3. http://www.opera.com/company/education/curriculum/

For "nonbasic" tasks, I went mostly with my personal favorite, Prototype,[4] except for one task (the lightbox one), where the solution I deem superior ends up being a jQuery plug-in. But really, once you master the basic tasks, you can rewrite or adapt my solutions using your framework of choice. And indeed, to facilitate this, we're putting the entire codebase for this book up in a public GitHub repository.[5] This way, creating a variant favoring another framework (say, jQuery) is as easy as clicking GitHub's Fork button, and finding such derived versions of the codebase becomes a snap.

Also note that all the code for this book, besides being available in a neatly packaged code archive on the book's website,[6] is available live for your testing and tweaking pleasure at http://demos.pocketjavascript.com/.

This Book at a Glance

This book is divided into theme-oriented parts, each with a number of tasks. It concludes with a few appendixes, some of which you may want to read *before* the main body of the book (especially the cheat sheet and the one about debugging JavaScript).

- Part 1 covers a few critical JavaScript code patterns that are too often ignored by JavaScript developers. They're just about the language, so they're framework-agnostic but indispensible for good coding on a daily basis. Be sure to start here!

- Part 2 is mostly about what I refer to as "basic" tasks, focusing on fundamental DOM and CSS manipulations, plus event handling and timers. Because of their "basic" status, I took care to list the relevant code for all major frameworks, so you can pick whatever suits you best. You should also check out Appendix C, on page 118, when reading this part so you get a good picture of the framework landscape and make informed decisions.

- Part 3 is all about the user interface, especially visual effects and neat UI ideas: good-looking tooltips, lightboxes, image preloading, infinite scrolling, and the like.

- Part 4 is complementary to Part 3, because it focuses on forms, a critical part of most web applications. Among other things, a number of tools are there to assist, simplify, and validate input.

- Part 5 is all about the client-server relationship, with topics such as cookies, JSON, and Ajax (same- and cross-domain).

---

4.   Full disclosure: I'm a member of Prototype Core.
5.   http://github.com/tdd/pragmatic-javascript
6.   http://pragprog.com/titles/pg_js

- Part 6, the final part, pushes this idea further by talking with third-party services, in the best mashup spirit. I chose three trendy topics here: playing with Twitter, Flickr, and geo-related APIs.

- Appendix A is my take on a JavaScript cheat sheet; I attempted to condense both the reference of the language and the important tips, leaving out a few language elements I felt were superfluous. I hope you find it useful.

- Appendix B is about debugging JavaScript; you owe it to yourself to know everything inside it, if only to spare you countless hours of hair-tearing, particularly when it comes to Internet Explorer.

- Appendix C tries to provide a useful description of the major frameworks I chose to include in this book. I did my best to provide an accurate depiction of all of them, presenting them in their best light and giving you a few tips about how best to choose a framework, on a case-by-case basis.

- Appendix D acts as a quick reference to the best helpful resources about JavaScript itself and the main frameworks; it sums up the relevant parts of Appendix C, plus a number of extra resources, mostly language-related. I put it at the end of the book so it's easier to locate.

How to Read This Book

In the Pragmatic Guide series, each chapter consists of two facing pages—one with text and one with code. If you're reading this book on paper, this flows naturally. But if you're reading an electronic edition of this book, you may want to set your reader to display two pages at once, in the side-by-side or "two-up" mode, provided your display is large enough. This will give you the best results.

The Pragmatic Bookshelf

The Pragmatic Bookshelf features books written by developers for developers. The titles continue the well-known Pragmatic Programmer style and continue to garner awards and rave reviews. As development gets more and more difficult, the Pragmatic Programmers will be there with more titles and products to help you stay on top of your game.

Visit Us Online

Pragmatic Guide to JavaScript
http://pragprog.com/titles/pg_js
Source code from this book, errata, and other resources. Come give us feedback, too!

Register for Updates
http://pragprog.com/updates
Be notified when updates and new books become available.

Join the Community
http://pragprog.com/community
Read our weblogs, join our online discussions, participate in our mailing list, interact with our wiki, and benefit from the experience of other Pragmatic Programmers.

New and Noteworthy
http://pragprog.com/news
Check out the latest pragmatic developments, new titles and other offerings.

Buy the Book

If you liked this eBook, perhaps you'd like to have a paper copy of the book. It's available for purchase at our store: pragprog.com/titles/pg_js.

Contact Us