

Preface

Before you begin, it is important to understand various styles employed in this book. Please do not skip this section, because it contains important information that will aid you as you read the book.

More code, less words

Please examine the code examples in detail. The text should be viewed as secondary to the code itself. It is my opinion that a code example is worth a thousand words. Do not worry if you're initially confused by explanations. Examine the code. Tinker with it. Reread the code comments. Repeat this process until the concept being explained becomes clear. I hope you achieve a level of expertise such that well-documented code is all you need to grok a programming concept.

Exhaustive code and repetition

You will probably curse me for repeating myself and for being so comprehensive with my code examples. And while I might deserve it, I prefer to err on the side of being exact, verbose, and repetitive, rather than make false assumptions authors often make about their reader. Yes, both can be annoying, depending upon what knowledge you bring to the subject, but they can also serve a useful purpose for those who want to learn a subject in detail.

Color-coding Conventions

In the JavaScript code examples (example shown below), orange is used to highlight code directly relevant to the concept being discussed. Any additional code used to support the orange colored code will be green. The color gray in the code examples is reserved for JavaScript comments (example shown below).

```
<!DOCTYPE html><html lang="en"><body><script>  
  
// this is a comment about a specific part of the code  
var foo = 'calling out this part of the code';
```

```
</script></body></html>
```

In addition to code examples being color-coded, the text in this book is colored so as to denote JavaScript words/keywords v.s. JavaScript code v.s. regular text. Below, I take an excerpt from the book to demonstrate this coloring semantic.

"Consider that the *cody* object created from the `Object()` constructor function (i.e `var cody = new Object()`) is not really different from a string object created via the `String()` constructor function. To drive this fact home, examine the code below:"

Notice the use of gray italic text for code references, orange text for JavaScript words/keywords, and regular black text for everything in-between.

jsFiddle, JS Bin, and Firebug lite-dev

The majority of code examples in this book are linked to a corresponding jsFiddle (<http://jsfiddle.net/>) page, where the code can be tweaked and executed online. The jsFiddle examples have been configured to use the Firebug lite-dev (<http://fbbug.googlecode.com/svn/lite/branches/firebug1.3/content/firebug-lite-dev.js>) plugin so that the log function (i.e. `console.log`) will work in most any modern browser regardless of if the browser has its own console. Before reading this book make sure you are comfortable with the usage and purpose (<http://stackoverflow.com/questions/4743730/what-is-console-log-and-how-do-i-use-it>) of `console.log`.

In situations where jsFiddle & Firebug lite-dev caused complications with the JavaScript code JS Bin (<http://jsbin.com/about>) & Firebug Lite-dev will be used. I've tried to avoid a dependency on a browser console by using Firebug lite-dev but with certain code examples the solution itself gets in the way of code execution. In these situations the console built into your web browser will have to be leveraged to output logs. If you are not using a browser with a built in JavaScript console I would suggest upgrading or switching browsers (<http://browsehappy.com/>).

When JS Bin is used, keep in mind that the code has to be executed manually (clicking 'Render') which differs from the page load execution done by jsFiddle.