
Applied jQuery

DEVELOP AND DESIGN



Jay Blanchard

Applied jQuery

DEVELOP AND DESIGN

Jay Blanchard



Peachpit
Press

Applied jQuery: Develop and Design

Jay Blanchard

Peachpit Press

1249 Eighth Street
Berkeley, CA 94710
510/524-2178
510/524-2221 (fax)

Find us on the Web at: www.peachpit.com
To report errors, please send a note to: errata@peachpit.com
Peachpit Press is a division of Pearson Education.
Copyright © 2012 by Jay Blanchard

Editor: Rebecca Gulick
Development and Copy Editor: Anne Marie Walker
Technical Reviewer: Jesse R. Castro
Production Coordinator: Myrna Vadic
Compositor: Danielle Foster
Proofreader: Patricia Pane
Indexer: Valerie Haynes-Perry
Cover design: Aren Straiger
Interior design: Mimi Heft

Notice of Rights

All rights reserved. No part of this book may be reproduced or transmitted in any form by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. For information on getting permission for reprints and excerpts, contact permissions@peachpit.com.

Notice of Liability

The information in this book is distributed on an "As Is" basis, without warranty. While every precaution has been taken in the preparation of the book, neither the author nor Peachpit Press shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the instructions contained in this book or by the computer software and hardware products described in it.

Trademarks

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Peachpit was aware of a trademark claim, the designations appear as requested by the owner of the trademark. All other product names and services identified throughout this book are used in editorial fashion only and for the benefit of such companies with no intention of infringement of the trademark. No such use, or the use of any trade name, is intended to convey endorsement or other affiliation with this book.

ISBN 13: 978-0-321-77256-5
ISBN 10: 0-321-77256-3

9 8 7 6 5 4 3 2 1

Printed and bound in the United States of America

*To Mom, who taught me there was magic in books,
and to Dad, who taught me there was magic in me.*

ACKNOWLEDGMENTS

Projects like this are not possible without the support and understanding of a lot of people, something I really didn't understand when first embarking on the journey to create a book. Saying "thank you" isn't nearly enough, but I hope that you all understand how much I appreciate you!

Even with the blender of life roaring around us, Connie Kay, Kaitlyn, Brittany, Zach, Karla, and Morgan provided more love and support than you can imagine. I love you all!

To Rebecca Gulick, thank you for believing in me and helping a dream to come true!

To Anne Marie Walker, enough cannot be said about your gentle firmness, guidance, and subtle humor. I am eternally grateful to you!

To Jesse Castro, thanks for keeping me on the straight and narrow. Your insight, technical abilities, and encouragement blow me away!

To Larry Ullman, thanks for being the Ford Prefect to my Arthur Dent and guiding me through the galaxy! I kept my towel on my desk the whole time!

To Francis Govers, the twists and turns in my life are made all the more bearable by knowing that you are just a phone call or an e-mail away. Best friends don't get any better!

To the folks who have made up the teams of developers that I have worked with day in and day out, thank you for making me a better programmer and a better person! Your willingness to look over my shoulder and teach me something new is treasured beyond measure.

To the jQuery community, you are an amazing group of people, and I am honored to share electrons with you!

CONTENTS

| | | |
|------------------|--|------------|
| | Introduction | viii |
| | Welcome to jQuery | xi |
| CHAPTER 1 | INTRODUCING JQUERY | XIV |
| | Making jQuery Work | 2 |
| | Working with the DOM | 6 |
| | Learning a Few jQuery Tips | 9 |
| | <i>Selecting Elements Specifically</i> | 9 |
| | <i>Making Quick Work of DOM Traversal</i> | 10 |
| | <i>Troubleshooting with Firebug</i> | 10 |
| | <i>Packing Up Your Code</i> | 11 |
| | <i>Using Return False</i> | 15 |
| | <i>Fiddling with jQuery Code</i> | 16 |
| | Combining jQuery with Other Code | 18 |
| | <i>Starting with HTML</i> | 18 |
| | <i>Styling Web Pages with CSS</i> | 18 |
| | <i>Using PHP and MySQL</i> | 18 |
| | Progressive Enhancement | 19 |
| | Planning Design and Interaction | 23 |
| | Wrapping Up | 23 |
| CHAPTER 2 | WORKING WITH EVENTS | 24 |
| | Using the Photographer's Exchange Web site | 26 |
| | Making Navigation Graceful | 27 |
| | <i>Creating and Calling Modal Windows</i> | 27 |
| | Binding Events to Other Elements | 34 |
| | <i>Building an Image Carousel</i> | 34 |
| | <i>Creating Sprite-based Navigation</i> | 50 |
| | Wrapping Up | 57 |
| CHAPTER 3 | MAKING FORMS POP | 58 |
| | Leveraging Form Events | 60 |
| | <i>Focusing on a Form Input</i> | 60 |
| | <i>Validating Email Addresses</i> | 62 |
| | <i>Making Sure an Input Is Complete</i> | 66 |
| | Tackling Uploads | 69 |
| | <i>Performing Client-side Validation</i> | 69 |

| | | |
|------------------|--|------------|
| | <i>Developing Server-side Validation</i> | 72 |
| | <i>Uploading Files</i> | 74 |
| | <i>Wrapping Up</i> | 89 |
| CHAPTER 4 | BEING EFFECTIVE WITH AJAX | 90 |
| | <i>Using AJAX for Validation</i> | 92 |
| | <i>Building the PHP Registration and Validation File</i> | 92 |
| | <i>Setting Up the jQuery Validation and Registration Functions</i> | 100 |
| | <i>Logging in the User</i> | 105 |
| | <i>Using AJAX to Update Content</i> | 108 |
| | <i>Getting Content Based on the Current User</i> | 108 |
| | <i>Loading Content Based on Request</i> | 110 |
| | <i>Loading Scripts Dynamically</i> | 112 |
| | <i>Using jQuery's AJAX Extras</i> | 116 |
| | <i>Using JSON</i> | 126 |
| | <i>Securing AJAX Requests</i> | 134 |
| | <i>Preventing Form Submission</i> | 135 |
| | <i>Using Cookies to Identify Users</i> | 139 |
| | <i>Cleansing User-supplied Data</i> | 141 |
| | <i>Transmitting Data Securely</i> | 144 |
| | <i>Wrapping Up</i> | 145 |
| CHAPTER 5 | APPLYING JQUERY WIDGETS | 146 |
| | <i>Using the jQuery UI Widgets</i> | 148 |
| | <i>Customizing the jQuery UI</i> | 148 |
| | <i>Including jQuery UI Widgets</i> | 152 |
| | <i>Using jQuery Plugins</i> | 171 |
| | <i>Beefing Up Your Apps with Plugins</i> | 172 |
| | <i>Pumping Up Your Sites</i> | 188 |
| | <i>Rolling Your Own Plugins</i> | 200 |
| | <i>Wrapping Up</i> | 203 |
| CHAPTER 6 | CREATING APPLICATION INTERFACES | 204 |
| | <i>Establishing the Foundation</i> | 206 |
| | <i>Creating the HTML</i> | 207 |
| | <i>Applying the CSS</i> | 209 |
| | <i>Making the Interface Resizable</i> | 214 |

| | |
|--|-----|
| Improving the Application Interface | 217 |
| <i>Creating Better Sprites</i> | 217 |
| <i>Loading Content with AJAX</i> | 226 |
| <i>Configuring Additional Enhancements</i> | 235 |
| Wrapping up | 247 |
| | |
| Index | 248 |

INTRODUCTION

As Web designers, you are painstakingly compelled to grab Web surfers' attention as quickly as possible and then keep them on your site to absorb the content. In addition to the product, service, or information that you are providing, the site must be visually attractive and offer stimulating (and valuable) interaction. The jQuery library is the main ingredient for providing the icing on your Web-site cake. If applied well, the effects of jQuery will convince visitors and application users to click around and sample all of your content.

The trick is learning how to combine jQuery with other markup and languages effectively. You must gain knowledge in a wide range of disciplines, like HTML (HyperText Markup Language) and CSS (Cascading Style Sheets), to know how to properly mix in the right amount of jQuery. The goal of this book is to give you the knowledge to bring the HTML, CSS, and jQuery ingredients together to create compelling interactivity to your Web sites and applications.

Throughout the book, I'll also show you ways to use PHP, a popular server-side scripting language, and MySQL, a relational database product, to enhance your overall development and supercharge your applications. Both technologies translate easily to other Web development languages.

WHAT IS JQUERY?

Announced in 2006 by its creator, John Resig, jQuery quickly gained popularity and support as a new way to use JavaScript to interact with HTML and CSS. jQuery's simple selectors mimicked CSS selectors, making the library familiar and easy to learn for designers and developers alike. The jQuery library erased the worry that Web developers had suffered through when trying to create interactive sites across a wide range of browsers by handling most browser compatibility issues behind the scenes.

Topping off those two features is the shortened syntax used by jQuery. The following example shows how you would select an element based on its `id` attribute using jQuery:

```
$('#foo');
```

The jQuery selector is much shorter as opposed to the same example in old-school JavaScript:

```
document.getElementById( 'foo' );
```

It's no wonder that the Web-development community embraced jQuery's "write less, do more" mantra. Couple the simplicity of jQuery with its ability to support complex animations and achieve stupendous effects, and you get a JavaScript library that is flexible and capable of empowering you to provide your Web site visitors with an outstanding interactive experience.

WHO THIS BOOK IS FOR

This book is aimed at beginning to intermediate Web developers, but it doesn't matter where you are in your journey as a designer or developer. You should find examples in this book that will help you to bring your Web pages and applications to life with jQuery. It helps if you have a basic knowledge of HTML, CSS, JavaScript, and jQuery, but it is not necessary because the examples are fully baked and ready to go.

WHAT I USED

As of this writing, jQuery 1.5 had been released and is used for all of the examples in the book. You can download it at www.jquery.com. It is also available on the book's Web site at www.appliedjquery.com.

HTML, CSS, and JavaScript files are all plain-text files that you can create and edit in any plain-text editor.

Examples were all tested in Firefox 3 and Internet Explorer 8, with an occasional peek in Safari and Google Chrome.

WHERE TO FIND THE CODE

All of the code examples for the book are available from the *Applied jQuery* Web site at www.appliedjquery.com/downloads. There you can download a Zip file containing all of the examples, graphics, and other collateral needed to follow along.

The examples are arranged by chapter within the Zip file and include all of the necessary jQuery files to make the examples work right out of the box.

However, even though all of the files are available for download, I encourage you to type out each example as you progress through the book. Taking a hands-on approach will help you to learn how all of the technologies fit together and will reinforce the concepts in your brain.

LET'S GET STARTED

It's time for you to jump right in and get started learning how to use jQuery. In the first chapter I'll give you some good rules and tools to get you headed in the right direction for sweetening your Web development efforts with jQuery.

i

WELCOME TO jQuery

WELCOME TO jQUERY

jQuery is one of the most popular JavaScript libraries in use today because it lets you build JavaScript Web pages and Web applications quickly and easily, accomplishing in a single line of code something that would have required dozens of lines of JavaScript code. Grab yourself a computer and the handful of tools outlined below, and then dig into the following six chapters.



jQUERY

jQuery, which is free to download and use, comes in the form of a single .js file that you link to from your Web page, and your code accesses the library by calling various jQuery functions. Go to jquery.com and download the jQuery library.



jQUERY UI

Next, you'll want to download the jQuery UI library from jqueryui.com. This will equip you with some core interaction plugins as well as many UI widgets that I'll discuss later in the book.



TEXT EDITOR

You'll be doing some scripting, so get yourself a good text editor. Windows users typically opt for Microsoft Notepad or Notepad++, while Mac users often rely on BBEdit from Bare Bones Software.



BROWSER

Chances are you've already got a standards-compliant browser installed. Popular options are the latest versions of Microsoft Internet Explorer, Mozilla Firefox, Apple Safari, Google Chrome, and Opera.



TROUBLESHOOTER

I rely heavily on the Firebug Web development tool for troubleshooting. Go to <http://getfirebug.com> and get a version that's specific to your browser. It's 100% free and open source, and you'll be grateful you've got it installed when something goes wrong.

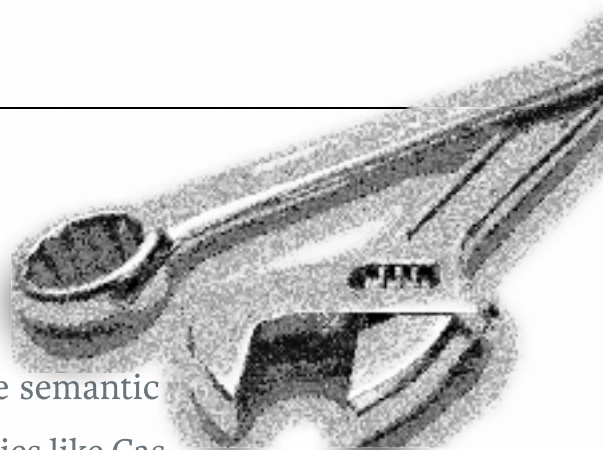


TESTING ENVIRONMENT

Rather than using an actual hosted Web site to test your jQuery creations, use a testing environment that's local on your own computer. I use XAMPP, which you can download from <http://apachefriends.org>.

1

INTRODUCING **jQUERY**



Rich, interactive Web sites that use semantic markup and unobtrusive technologies like Cascading Style Sheets (CSS) and JavaScript are becoming the de facto standard in Web development. Designers and developers are looking for new and better ways to bring their creations to life, and libraries like jQuery make this goal easily attainable.

To get started properly with jQuery, you need to equip yourself with the appropriate tools and concepts. So, I've gathered those tools for you and will help you to learn how to use them.

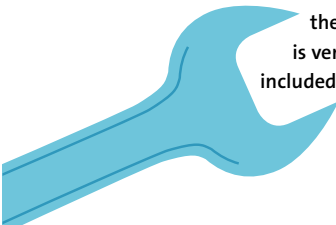
This chapter will give you a firm grasp of the basics of jQuery and the tools that will make working with jQuery straightforward. Also included are some tips for getting the most out of jQuery. But first things first; let's start with a "Hello World" example jQuery style.

MAKING JQUERY WORK

The strength of the jQuery library is its ability to interact with elements in your Web pages that you are already familiar with. Markup tags, class declarations, and attribute information in your Web pages can be easily connected to jQuery by using the simple concept of selectors.

A jQuery selector will wrap an element or set of elements into an object. Once you have created the jQuery object, you can effectively apply a multitude of jQuery methods to that object to create animations, send information to and from the server, or perform object manipulation.

No book on programming is worth its salt if it doesn't have a "Hello World!" example. To illustrate the power and flexibility of jQuery's selectors, let's create a "Hello World!" example.



NOTE: The Hello World code is the only code example not available in the download from the book's Web site. The reason is that I think it is very important that you type this one in yourself. Comments are also included in the example.

1. Start by establishing the basic markup for the HTML page:

```
<!DOCTYPE>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Hello World - jQuery Style</title>
```

2. Be sure to include the jQuery source file. Without this file none of the jQuery code will operate:

```
<script type="text/javascript"
  → src="jquery-1.5.min.js"></script>
```

3. Open a script tag to give the jQuery code a place to live within the page:

```
<script type="text/javascript">
```

-
- The jQuery functions that you are creating need to be available to run after the Web page has finished loading all of its elements. To accomplish this, you wrap the jQuery code in a *document ready* function. Just as it implies, the code wrapped in the function becomes available to run when the Web document is ready:

```
$(document).ready(function() {
```

- Create the first selector. This selector will get the markup element in the page having an id attribute equal to `first`. All id attributes are selected in jQuery (and CSS) by prepending the hash (#) sign to the information contained within the id attribute. You'll follow the selector with jQuery's `html` method by chaining the `html` method to the selector. This method will place the markup `<h1>Hello World!</h1>` into the selected element:

```
/* write 'Hello World!' to the first div */  
$('#first').html('<h1>Hello World!</h1>');
```

Chaining is the term used to describe applying one or more methods to jQuery objects. Chaining gives you a wide variety of possibilities to combine methods to create unique interactions for your Web-site visitors.

- For this example, you'll create one additional method that connects, or *binds*, an event handler to a selector to create an action. The event handler will accept an action and perform additional jQuery functions to other selected elements. Start this portion of the example by binding jQuery's click handler to an element with an id of `link`:

```
/* a clickable 'Hello World!' example */  
$('#link').click(function() {
```

The `click` method exposes a handler function that allows you to build a string of actions that will be triggered by the `click` method.

- Set up a selector for the element with an id of `greeting` and apply the `html` method to it:

```
$('#greeting').html('<h1>Hello Again!</h1>');
```

-
8. Close out the jQuery code with the proper braces, brackets, and script tags:

```
});  
});  
</script>
```

Pay close attention to braces and brackets when you create jQuery code. It is critically important that each opening bracket or brace have a matching closing bracket or brace.

9. Finish up the head section of the markup and open the body of the Web page:

```
</head>  
<body>
```

10. Create an HTML div with an id of `first`. The initial jQuery selector that you created previously will interact with this element, adding the HTML markup that was specified between the div tags:

```
<div id="first"></div>
```

11. Create another HTML div with an id of `second`. You did not write any selectors for this element; it is just being used as a container for other elements:

```
<div id="second">
```

12. Create an anchor tag and give it an id of `link`. You wrote jQuery code earlier that will handle the link when it is clicked by a user:

```
<a href="#" id="link">Click Me!</a><br />
```

13. Create a span element with an id of `greeting`. When the link is clicked, the selector for `greeting` will apply the HTML markup you specified between the span tags:

```
<span id="greeting"></span>
```

14. Complete the page by closing out the markup tags properly:

```
</div>  
</body>  
</html>
```



FIGURE 1.1 The “Hello World!” message appears when the page loads, and the “Hello Again!” message appears when the link is clicked.

15. Save the file as *hello_world.html* and load it into your Web browser. If you have been diligent with your typing, you will be rewarded with a Web page identical to the one shown in **Figure 1.1**.

This example is just a small taste of how you can connect jQuery to elements in your Web pages to provide information and interactivity. The example also demonstrates how you can add elements to your Web pages using jQuery.

To work with jQuery effectively, you need to know how to work with all of the elements in a Web page and how they are assembled into a document.

A document?

That is absolutely correct: Web pages are documents that are intended for display in Web browsers. Because Web pages are documents, they follow some of the same rules that paper documents follow, and those rules are provided by the master document—the Document Object Model (DOM).

WORKING WITH THE DOM

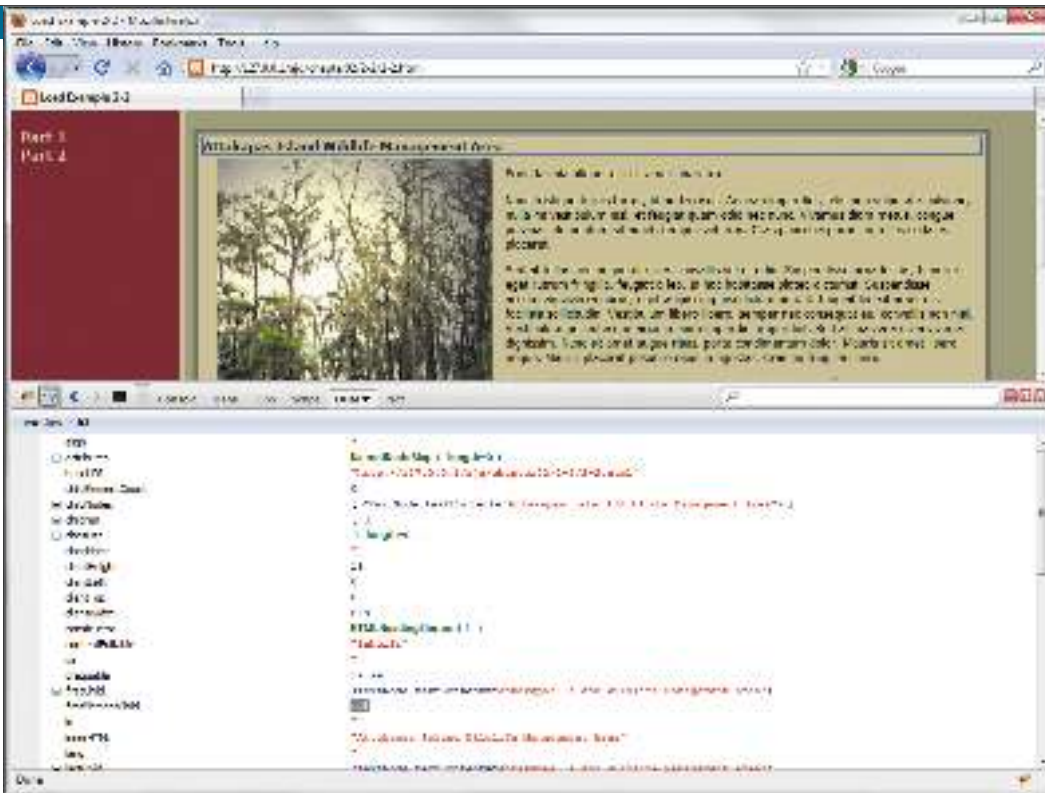


FIGURE 1.2 Examining the DOM using Firebug with Firefox.

At the heart of all of your Web pages is an API (Application Programming Interface) that describes everything on the page. It is the DOM. The DOM provides information for each element on the page, including styles associated with elements. The information in the DOM is stored in a tree-like structure.

Several DOM inspector applications are available either as stand-alone applications or as add-ins to many popular Web browsers. **Figure 1.2** shows the DOM inspector available with Firebug.

```
<div id="information">
  <ul> // child of div id="information"
    <li> // child of ul, sibling to the other li
      <a href="foo.html"> // child of nearest li
         // child of a href="foo.html"
      </a>
    </li>
    <li> // child of ul, sibling to the other li
      <a href="glorp.html"> // child of nearest li
         // child of a href="glorp.html"
      </a>
    </li>
  </ul>
</div>
```

FIGURE 1.3 An outline of the relationships between the elements in the list.

The DOM API is what allows languages like JavaScript and libraries like jQuery to interact with elements in your Web pages. You can use libraries like jQuery to virtually climb up and down the DOM tree to locate, add, remove, and modify elements. Because you'll be using jQuery to interact with the DOM, including adding and removing elements from it, you need to become familiar with how the DOM is constructed. You don't need to become an expert on the DOM, but you should know enough about the DOM to recognize what is going on when you manipulate it with jQuery.

Knowing the DOM becomes critically important when you start working with jQuery's parent and child type selectors. You must understand the relationship between the elements in the DOM so that you can effectively manipulate those elements. Consider the following HTML markup:

```
<div id="information">
  <ul>
    <li><a href="foo.html"></a></li>
    <li><a href="glorp.html"></a></li>
  </ul>
</div>
```

To know how to travel up and down the DOM tree, you must know what the relationships are between the elements. **Figure 1.3** shows how those relationships are defined.

LINE BREAKS AND COMMENTS

Because JavaScript allows you to continue code through line breaks, jQuery will, too. This means that you can spread chained jQuery methods over several lines. Spreading lengthy chains over multiple lines makes the jQuery methods visually easier to follow and troubleshoot. Quite often you'll see jQuery chains similar to the following example:

```
var nextImage = $('img[src="bar.jpg"]') // define the
→ starting point
  .closest('li') // travel up to the closest list item
  .next() // move to the next list item
  .find('img') // find the image tag in the next list item
  .attr('src'); // grab the source attribute of the found
→ image tag
```

I cannot stress enough the importance of commenting your code well. Although I won't be commenting a lot of the code in the book for space reasons, you can expect to see a lot of commentary within the code samples on the Web site. My personal style is to use the double slash at the end of a line when the comment is quick and use larger comment blocks (beginning with `/*` and ending with `*/`) when I need to be more descriptive.

```
// is a short comment
/* this comment may span multiple lines and can be very
→ descriptive */
```

Armed with this knowledge, you can traverse the DOM elements for this list. Given that you know the image source in the first list item, you can retrieve the source attribute from the second image in the list easily, like this:

```
var nextImage = $('img[src="bar.jpg"]') // define the starting point
  .closest('li') // travel up to the closest list item
  .next() // move to the next list item
  .find('img') // find the image tag in the next list item
  .attr('src'); // grab the source attribute of the found
→ image tag
```

The variable `nextImage` now contains the value `murkle.jpg`.

LEARNING A FEW JQUERY TIPS

As I use and continue to learn more about the jQuery library, I've accumulated some good rules of thumb, including being specific about jQuery selectors, caching selectors, and packing up code to make it more efficient. These and other tips provided here will make your code more effective, provide you with some good tools, and shorten your development time.

SELECTING ELEMENTS SPECIFICALLY

To find the elements that you want to act on, jQuery has to traverse the DOM tree. Depending on the length and complexity of a page, the DOM can be quite large. Using grossly formed selectors can slow performance and lead to frustration.

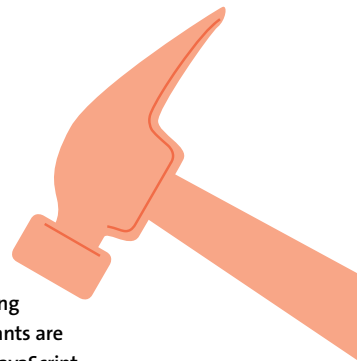
jQuery reads selectors right to left, so if you have a selector like this:

```
$("#div ul li a");
```

jQuery will gather all the anchors first, determine if they are within list items, and then find out if the list item is contained within an unordered list that is contained within a <div>.

Whew! It would be better to give one group of these items a class or an id attribute that will allow you to more directly identify one or more of the elements involved. For instance, the anchor tags in this group can be navigation elements and given a class of navigation (). That will allow you to shorten the selector to \$(".navigation"). As an added bonus, the element can be more easily referred to and styled in CSS!

TIP: Thanks go out to the very supportive jQuery community for the tips included in this section. You can learn a lot by participating in the jQuery forums at <http://forum.jquery.com>. Forum participants are always willing to lend a hand to help you solve your jQuery and JavaScript problems.



- [read *Job U: How to Find Wealth and Success by Developing the Skills Companies Actually Need for free*](#)
- [Doctor at Large \(Doctor Series, Book 3\) pdf](#)
- [read Lessons from the Hanoi Hilton: Six Characteristics of High Performance Teams online](#)
- [read online Betrayal \(Empty Coffin, Book 2\)](#)
- [download online Diez \(posibles\) razones para la tristeza del pensamiento pdf, azw \(kindle\), epub, doc, mobi](#)
- [download Normandy 1944: Allied landings and breakout \(Campaign, Volume 1\)](#)

- <http://cavalldecartro.highlandagency.es/library/OpenCL-Parallel-Programming-Development-Cookbook.pdf>
- <http://aseasonedman.com/ebooks/The-Adventures-of-Amir-Hamza--Lord-of-the-Auspicious-Planetary-Conjunction.pdf>
- <http://bestarthritiscare.com/library/Modern-Japan--A-Historical-Survey--5th-Edition-.pdf>
- <http://transtrade.cz/?ebooks/First-Rider-s-Call--Green-Rider--Book-2-.pdf>
- <http://chelseaprintandpublishing.com/?freebooks/What-Hearts.pdf>
- <http://www.1973vision.com/?library/Philosophy-and-the-Problems-of-Work--A-Reader.pdf>