Ajax Bible

Steven Holzner, PhD

Wiley Publishing, Inc.

Ajax Bible

Ajax Bible

Steven Holzner, PhD

Wiley Publishing, Inc.

Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published simultaneously in Canada

ISBN: 978-0-470-10263-3

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Library of Congress Control Number: 2007920013

About the Author

Steven Holzner is the award-winning author of 102 computer books. He's written many bestsellers, including a number of well-received books on Ajax, such as *Ajax For Dummies* and *Ajax Visual Blueprint*. He is also a former contributing editor at *PC Magazine*, and he's been on the faculty of both MIT and Cornell University. His books have sold several million copies, and been translated into more than 20 languages around the world.

To Nancy, of course.

Credits

Product Development Supervisor
Courtney Allen

Project Editor
Katharine Dvorak

Technical Editor
Steve Wright

Copy Editor
Scott Tullis

Editorial Manager
Robyn Siesky

Business Manager
Amy Knies

Vice President and Executive Group Publisher
Richard Swadley

Vice President and Executive Publisher
Bob Ipsen

Vice President and Publisher
Barry Pruett

Project Coordinator
Patrick Redmond

Graphics and Production Specialists
Carrie A. Foster
Jennifer Mayberry
Barbara Moore
Amanda Spagnuolo

Quality Control Technicians
Laura Albert
Christine Pingleton

Proofreading and Indexing
Linda Seifert
Broccoli Information Management

Preface xiii
Acknowledgments xvii

Part I: Fundamental Ajax 1

Chapter 1: Essential Ajax 3

What Does "Ajax" Mean? 6
What Can You Do with Ajax? 8
Create Ajax live searches 8
Create an Ajax-enabled calculator 9
Talk with Ajax chat applications 10
Crunch numbers with spreadsheets 12
Browse Amazon 12
Get the answer with Ajax autocomplete 13
Log in with Ajax 15
Download images 17
Drag and drop with Ajax 18
Play games with Ajax 21
Modify Web pages on the fly 24
Add interactivity to maps 29
E-mail the Ajax way 30
Summary 32

Chapter 2: Know Your JavaScript 33

What Is JavaScript's Place in Ajax? 33
A Brief History of JavaScript 36
Getting Started with JavaScript 37
Fixing errors 39
Commenting your code 42
Putting your code in external script files 43
Responding to browser events 44
Creating JavaScript Functions 47
Passing arguments to functions 52
Returning values from functions 57
Storing Your Data in Variables 60

Operating on Your Data with Operators 62
Making Decisions with the If Statement 65
Executing Different Code in Different Browsers 68
Working with Loops 73
Looping with the for loop 74
Looping with the while loop 75
Connecting JavaScript to HTML Buttons 79
Summary 81

Chapter 3: Creating Ajax Applications 83

Writing Ajax 83
Setting up the application 86
Writing the JavaScript 87
Creating an XMLHttpRequest object 87
Opening the XMLHttpRequest object 92
Handling data downloads 95
Getting your data 98
Downloading the data 100
More ways to create XMLHttpRequest objects 101
Interacting with Server-Side Code 103
Passing Data to Server-Side Scripts 106
Passing data to the server with GET 106
Passing data to the server with POST 110
Using Ajax with XML 115
Creating the XML 117
Downloading the XML 118
Summary 125

Chapter 4: Serious Ajax Programming. 127

Working with Multiple Concurrent XMLHttpRequest Requests 128
Using multiple XMLHttpRequest objects 130
Storing XMLHttpRequest objects in an array 134
Using inner functions 138
Handling JavaScript Sent from the Server 145
Returning JavaScript 145
Returning JavaScript objects 149
Connecting to Google Suggest 151
Calling Another Domain in Ajax 159
A Login Example 160
Working with Ajax and Head Requests 163
Getting all head data 163
Getting specific head data 166
Eliminate Caching 167
Summary 168

Part II: Ajax in Depth 169

Chapter 5: Introducing Ajax Frameworks 171

Using the Ajax Utility Library 171
Get text from the server with getText 175
Get XML from the server with getXml 180
Post data to the server and get text back 187
Post data to the server and get XML back 193
Using libXmlRequest to Download XML 203
Using AJAXLib to Download XML 207
Summary 209

Chapter 6: More Advanced Ajax Frameworks. 211

Using the Majax Framework 211
Using the Sack Framework 216
Using the XHConn Framework 221
Using the uniAjax Framework 226
Using the AjaxGear Framework 229
Using the AjaxRequest Framework 234
Using the Http Framework to Avoid Caching 239
Using the Sarissa Framework to Handle XML 242
Using the Interactive Website Framework to Handle XML 247
Summary 250

Chapter 7: Using Server-Side Ajax Frameworks. 251

Working with PHP-Based Frameworks 251
Using Sajax and PHP 251
Using Xajax and PHP 260
Using LibAjax and PHP 265
Working with Java-Based Frameworks 271
Using Direct Web Remoting and Java 271
Using Ajax Tags and Java 276
Using SWATO with Java 279
Working with Ajax and Ruby on Rails 281
Downloading text with Ajax 282
Accessing downloaded data in code 288
Summary 294

Part III: Ajax and the DOM, XML, CSS, and Dynamic HTML 295

Chapter 8: The DOM and Event Handling 297

Introducing the DOM 297
Appending Elements Using the DOM 299

Replacing Elements Using the DOM 305
Creating an Ajax-Enabled Shopping Cart 310
Creating a cross-browser mouse event object 315
Handling mouse down events 320
Handling mouse move events 323
Handling mouse up events 325
Downloading Images with Ajax 328
Handling Ajax Timeouts 333
Summary 337

Chapter 9: XML and Ajax 339

Creating XML 339
Handling XML with JavaScript 344
Retrieving Data from an XML Document 349
Handling White Space in the Mozilla, Netscape, and Firefox Web Browsers 359
Handling White Space in a Cross-Browser Way 361
Accessing XML Data Directly 365
Accessing XML Attribute Values 371
Validating Your XML 375
Summary 378

Chapter 10: Cascading Style Sheets and Ajax 379

Ajax-Enabled Menus 379
Creating the styles 382
Working with the mouse 388
Downloading the menu items 395
Showing a menu 396
Handling menu item clicks 398
Hiding a menu 400
Getting Text Noticed in Ajax Applications 401
Scrolling Text 405
Styling Text Using CSS 407
Styling Colors and Backgrounds Using CSS 411
Styling Locations in Web Pages 416
Summary 418

Chapter 11: Dynamic HTML and Ajax 419

Creating Mouseovers 419
Using Dynamic Styles 423
Using document.write 427
Using Dynamic HTML Methods to Update Part of a Page 434
Using Dynamic HTML Properties to Update Part of a Page 437
Using Text Ranges to Update Part of a Page 443
Using createElement to Create New Elements on the Fly 447
Creating Dynamic Tables 452

Catching the User's Attention with Animation 457
Summary 464

Part IV: Advanced Ajax 465

Chapter 12: Introducing Ajax and PHP 467

Getting Started with PHP 467
Sending text back to the browser 469
Sending XML back to the browser 471
Commenting Your PHP 474
Working with Variables 475
Interpolating Variables in Strings 485
Working with Arrays 487
Creating arrays 488
Copying arrays 491
Handling Your Data with Operators 492
Making Choices with the if Statement 494
Using else statements 497
Using elseif statements 498
Using Switch Statements 499
Using loop Statements 501
Looping with the for loop 501
Looping with the while loop 503
Looping with the do...while loop 505
Looping with the foreach loop 506
Summary 508

Chapter 13: PHP: Functions and HTML Controls. 509

Working with Functions 509
Passing data to functions 512
Setting up default arguments 514
Creating variable-length argument lists 515
Returning values from functions 518
Returning multiple values from a function 519
Handling HTML Controls 522
Working with text fields 524
Working with checkboxes 527
Working with radio buttons 531
Working with list boxes 533
Working with password controls 538
Working with Image Maps 540
Uploading Files 544
Summary 550

Chapter 14: Handling User Input in PHP 551

Displaying All the Data in a Form....552
Looking at Server Variables 556
Getting User Input in Arrays 560
Wrapping PHP Applications in a Single Page 563
Using Hidden Controls to Check Whether a Page Has Been Seen Before 567
Validating User Input, Numbers, and Text 568
Validating user input 568
Validating numbers....577
Validating text....582
Handling HTML Sent by the User 586
Summary 589

Chapter 15: Ajax and Security 591

Protecting Against Malicious Users....591
JavaScript insertion....592
Looking at the Security Issues with Proxy Servers 600
Handling Security in Ajax Applications 601
Easy password protection 601
Server-side user authentication 605
Protecting Passwords Using Public and Private Keys....619
Summary 630

Chapter 16: Filters, MVC, and Ajax 631

Restricting Access with Filters 631
Logging User Access with Filters 642
Restricting Access Based on Time of Day 654
Model-View-Controller Applications 660
The controller....660
The view....664
The model 665
Installing the MVC Application 666
Accessing MVC with Ajax....666
Summary 668

Index 669

This book is all about making your Web applications look and feel like desktop applications, which is the whole idea behind Ajax. Web applications are becoming more and more popular these days, but there's still that problem with Submit buttons: when you click one, the whole page flickers and refreshes, and you have to wait until the page downloads.

That's exactly where Ajax comes in. With Ajax, you can communicate with the browser behind the scenes, get the data you need, and display it in a Web page. There's no page refresh, no waiting required, no flickering in the browser.

That's cool, because it means your Web applications start to look and feel like desktop applications. As your users' connections to the Internet get faster and faster, soon there will be just about no way to tell a Web application apart from a desktop application.

Ajax is the future of Web programming. With Ajax, applications in a Web browser can work just like those installed on the user's computer. It's no surprise that Ajax is the hottest Web programming topic in years.

About This Book

Everything Ajax is in this book. Part I starts with a guided tour of how Ajax is used today. Along the way you're going to see some very cool applications, as well as some games.

Ajax is based on JavaScript, and there's a section in this part on how to work with JavaScript. If you don't know JavaScript, you're going to need to pick it up, and you can do that here. If you do know JavaScript, you can skip this part of the book and go on to the more advanced topics such as working with Ajax and PHP and security.

Part II then charges into Ajax programming, showing you how to create Ajax-enabled applications from scratch. There are a few chapters on the basics of Ajax, and some on the more advanced, potent aspects. You're also going to see how to save yourself the need for nearly any programming at all when you learn how to work with Ajax frameworks. These frameworks, most of which are available for free, do the Ajax programming for you, letting you create full Ajax applications in a snap.

Ajax involves more than just JavaScript, however. Part III presents the full story on using XML, cascading style sheets, and server-side programming, including a chapter on each of these topics.

Part IV includes chapters on security, as well as chapters on other advanced topics, such as using Web-server filters with Ajax.

All this and more is coming up in this book. In other words, you're going to get the full Ajax story in this book, from soup to nuts. Ajax is going to become a rich toolset for you, ready to be put to work.

Conventions Used in This Book

Some books have many conventions that you need to know before you can even start. Not this one. All you need to know is that when new lines of code are introduced, they appear in bold, like this:

```
function getDataReturnText(url, callback)
{
  var XMLHttpRequestObject = false;

  if (window.XMLHttpRequest) {
    XMLHttpRequestObject = new XMLHttpRequest();
  } else if (window.ActiveXObject) {
    XMLHttpRequestObject = new
     ActiveXObject("Microsoft.XMLHTTP");
  }
        .
        .
        .
}
```

Note also that code that's been omitted has been indicated with three vertical dots.

You can download the code used in this book at `www.wiley.com/go/ajaxbible`.

What You'll Need

To use this book profitably, you'll need to know some basic HTML — not much, just enough to write a decent Web page. If you need to pick up HTML, take a look at one of the many excellent tutorials on the Internet. The HTML used in this book isn't very advanced, and even if you're not familiar with it, you can probably pick it up just by reading this book.

You're also going to have to know JavaScript. That's not a problem, because all the JavaScript you'll need to know is specifically introduced in this book. However, if you feel you need more, take a look at the JavaScript tutorials online, or check out the *JavaScript Bible*.

Because Ajax involves communicating with the server, there will also be some PHP involved in this book, and in case you're not familiar with PHP, that's also not a problem because the book contains a couple of chapters to bring you up to speed on PHP.

You'll also need a browser, such as Microsoft Internet Explorer or Mozilla Firefox, to use this book. However, because browsers have become so plentiful that you can barely do anything on a computer without bumping into one, that shouldn't be an issue — just use the browser you're accustomed to using.

And that's it! You're ready to go. Turn to Chapter 1 to see Ajax at work.

The book you hold in your hands is the product of many people's work. I'd especially like to thank acquisitions editor, Courtney Allen; project editor, Katharine Dvorak; technical editor, Steve Wright; copy editor, Scott Tullis; and project coordinator, Patrick Redmond.

Part I

Fundamental Ajax

IN THIS PART

Chapter 1
Essential Ajax

Chapter 2
Know Your JavaScript

Chapter 3
Creating Ajax Applications

Chapter 4
Serious Ajax Programming

Chapter 1

Essential Ajax

Welcome to the *Ajax Bible*! This is the home of all things Ajax. Ajax is the new technology that's blazing through the Internet, igniting Web sites everywhere. The Ajax revolution has come, and it's changing the Internet. In fact, Ajax is the basis of what's being called Web 2.0, the next version of the World Wide Web.

So what's it all about? The central idea is making Web applications look and feel just like desktop applications. For example, take a look at Figure 1.1, where you see the familiar Google search page. Enter a term to search for, such as "Ajax," and click the Google Search button.

IN THIS CHAPTER

Introducing Ajax

Looking at Ajax in action

Conducting Ajax Live Searches

Using Ajax chat

Enabling Autocomplete

Dragging and dropping with Ajax

Using Ajax-enabled shopping carts

FIGURE 1.1

Google searches for matches to the term you enter.

The screen flashes as it's updated with new data, and the matches that Google found to your search term appear, as you see in Figure 1.2.

FIGURE 1.2

Google displays the matches it finds.

That works OK, but that's not the Ajax way of doing things. Using Ajax, you can work behind the scenes, connecting to the server to get data without causing a page refresh in the browser. For example, take a look at the Google Suggest page at `www.google.com/webhp?complete=1&hl=en`, which is shown in Figure 1.3.

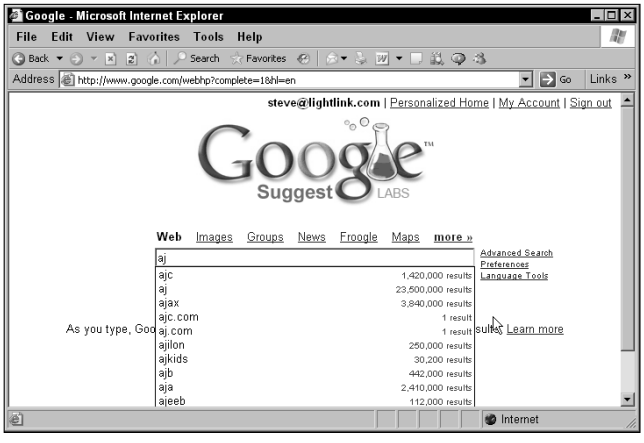
FIGURE 1.3

The Google Suggest page

As you type, the page in the browser actually connects to the Google server and looks up matches to the partial search term you entered. For example, type "aj," and you'll see a drop-down list box appear, as in Figure 1.4, with matches found by Google as you're typing.

FIGURE 1.4

Google Suggest looks for matches as you type.



Behind the scenes, using Ajax techniques, the Web page connects to Google Suggest and searches for matches to your search term as you're entering it. It then displays a drop-down list box of the matches it's found to your search term, letting you select from those matches — all without a page refresh. That's the crucial point: no page refresh was necessary. In the old days, when you wanted

to send data to the server, you had to click a button, such as the Google Search button. Then you had to wait as the screen flickered and was refreshed. Now, a Web page can send data to the server without creating a page refresh at all, as you see in this example, where your search term was sent to the server automatically and the server sent back data to be displayed in the drop-down list.

No longer do you need to perform a page refresh when you send data to the server, or when you receive data from the server. Instead, Web pages can now act much more like desktop applications, sending data to the server and receiving data back, all behind the scenes.

This conversion of Web applications, making them feel more like desktop applications, is what's meant by Web 2.0. How would you like it if your word processor flashed every time you typed a new character, and the entire document was displayed over again, with the cursor reset to the beginning of the document? Not a very attractive thought. Using Ajax, you can create online word processors that are practically indistinguishable from the desktop version — no flash, no flicker, no resetting the cursor location when you type. Just a smooth word-processing experience, just like the desktop version of the same application.

You can see why Ajax is causing a revolution in Web applications: now it's possible to create online applications that look and feel just like their desktop counterparts.

This chapter gets you started. You'll get an overview of the meaning of the term Ajax, and then a survey of how Ajax is used today. That survey is a very important part of this book because Ajax is turning up in more and more places — sometimes unexpectedly — and if you are familiar with the uses of Ajax, you'll know where you can use it in your own Web applications.

What Does "Ajax" Mean?

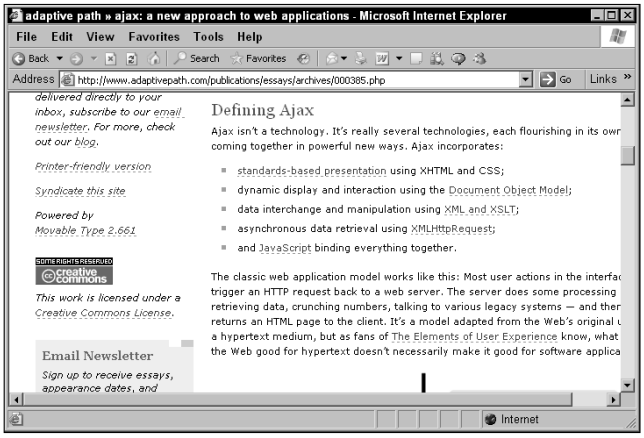
So where did the term "Ajax" come from, exactly? Take a look at Figure 1.5, which shows the very important first article written on Ajax, the article that coined the term and started everything. You can find that article at `www.adaptivepath.com/publications/essays/archives/000385.php`. This article is by Adaptive Path's Jesse James Garrett, who was the first to call this technology Ajax.

Here's how that article starts:

> "If anything about current interaction design can be called 'glamorous,' it's creating Web applications. After all, when was the last time you heard someone rave about the interaction design of a product that wasn't on the Web? (Okay, besides the iPod.) All the cool, innovative new projects are online.
>
> "Despite this, Web interaction designers can't help but feel a little envious of our colleagues who create desktop software. Desktop applications have a richness and responsiveness that has seemed out of reach on the Web. The same simplicity that enabled the Web's rapid proliferation also creates a gap between the experiences we can provide and the experiences users can get from a desktop application.
>
> "That gap is closing."

FIGURE 1.5

The original Ajax article



That gap is indeed closing, thanks to Ajax. So, according to the article, what does "Ajax" actually stand for? It stands for ***Asynchronous JavaScript and XML***. As you can begin to see from its name, and as you can read in the Jesse James Garrett article, Ajax is really a collection of technologies.

The "asynchronous" part means that the browser isn't going to wait for data to be returned from the server, but can handle that data as it's sent back, when it's sent back. In other words, data transfers take place behind the scenes, without making the browser pause and wait for something to happen. That's a crucial part of Ajax: You can handle data from the server when the server sends you that data. You don't have to put your whole application on hold until that data arrives. If you had to wait for that data, your application would be synchronous; and with slow Internet connections, that could be a problem.

The JavaScript part of the term Ajax is also very important because that's what makes Ajax happen in the browser. Ajax relies on JavaScript in the browser to connect to the server and to handle the data that the server sends back. All the Ajax applications you will develop in this book use JavaScript to connect to the server behind the scenes, uploading and downloading data. And when your data is downloaded, you can use JavaScript in the browser to handle that data, displaying it or crunching it as appropriate.

What about the XML part of the term Ajax? As you probably know, XML has become the lingua franca of the Web, providing a text-based way to send data back and forth across the Internet. The reason XML has become so popular is that it is indeed text-based, which means that you can sling XML around the Internet, because the Internet was designed to handle text-based documents (that is, HTML). For that reason, Ajax applications are often written to handle data sent back from the server using XML. In other words, when you contact the server, it'll send data back to you as an XML document.

In fact, XML is only one of the ways to handle data sent to you from the server. You can also send back plain text as well, and you're going to see both techniques extensively in this book.

Besides JavaScript and XML, Ajax also works with dynamic HTML and Cascading Style Sheets (CSS). Both of these technologies allow you to update the data displayed in a Web page, and, because you don't redraw the entire Web page with Ajax, but just a part of it, you rely on dynamic HTML and CSS quite a bit; both of them allow you to update specific parts of a Web page. You're going to see a lot more on dynamic HTML and CSS in this book because they allow you to refresh just part of a Web page, something that is central to Ajax-enabled applications.

The part of JavaScript that makes Ajax possible is the `XMLHttpRequest` object. This is a special object built into all modern browsers' version of JavaScript. As you're going to see, this is what makes it possible to connect to the server and handle data sent back from the server behind the scenes. It's not just JavaScript that makes Ajax tick, it's the `XMLHttpRequest` object inside JavaScript.

So there you have it; Ajax is a collection of technologies, not just a single technology. You use the `XMLHttpRequest` object built into JavaScript to connect to the server, and then handle the XML — or plain text — the server sends back using JavaScript. And you use dynamic HTML and CSS to display the results in the browser. It's lucky that all the parts of Ajax applications came together as they did — JavaScript, the `XMLHttpRequest` object, dynamic HTML, and CSS — because all together, they make it possible to make your online applications look like desktop applications.

Actually, the technology for Ajax has been around since 1998, and had already been used by a number of applications such as Microsoft's Outlook Web Access. But it didn't really catch on until early 2005 when some high-profile applications such as Google Suggest put it to work, and Jesse James Garrett wrote his article coining the term Ajax, which put everything under one roof.

Since that time, things have exploded as developers have realized that Web software can finally start acting and behaving like desktop software. So what can you do with Ajax? That's what the rest of this chapter is about.

What Can You Do with Ajax?

There's a great deal you can do with Ajax, and the following pages cover this treasure trove in some detail. Coming up is a good survey of the way Ajax is used today.

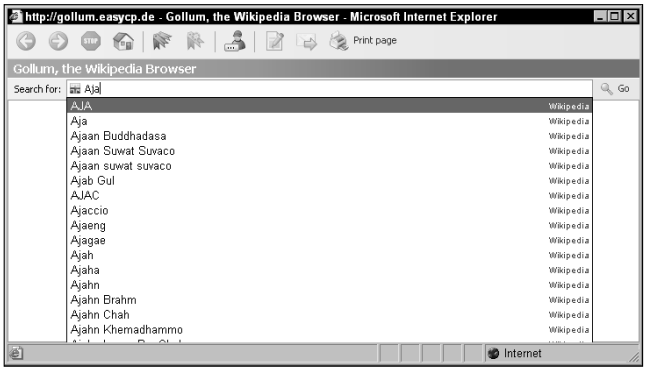
Create Ajax live searches

One of the most popular uses of Ajax is to create ***live searches***, and you've already seen an example with Google Suggest at the beginning of this chapter. With a live search, the user can enter a partial search term, and using Ajax, the Web application connects to the server and finds matches to that partial search term.

There are plenty of live searches besides Google Suggest available online. For example, take a look at Gollum at `http://gollum.easycp.de/en/`, which is a live search of Wikipedia, the online free encyclopedia at `www.wikipedia.org`. Gollum is shown in Figure 1.6.

FIGURE 1.6

Gollum performs live searches of Wikipedia.



Enter a partial search term in Gollum, such as "Aja" for Ajax, and you can see the results in Figure 1.6, where Gollum has connected to Wikipedia behind the scenes and found matches to your partial search term. Those matches are displayed, as is usual for a live search, in a drop-down list, and you can select the one that you're looking for. When you do, the matching Wikipedia article is opened in your browser.

Create an Ajax-enabled calculator

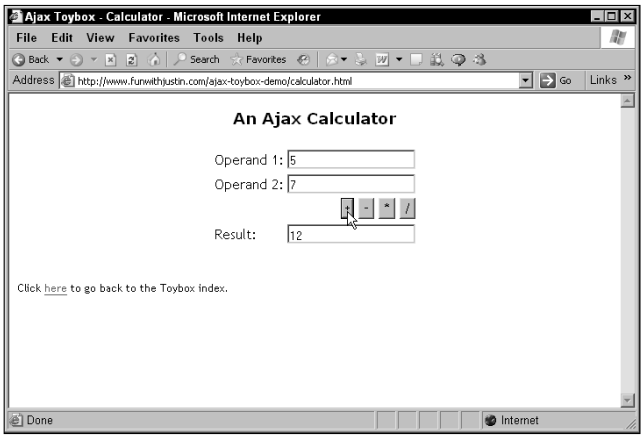
Any situation where you have to send data to the server and handle the data sent back to you behind the scenes is perfect for Ajax. So how about an Ajax-enabled calculator? You can find one at `www.funwithjustin.com/ajax-toybox-demo/calculator.html`, as shown in Figure 1.7.

To use the calculator, just enter two operands to work with, such as 5 and 7 in Figure 1.7, and click the operation you want to perform — addition, subtraction, multiplication, or division. Using Ajax, this Web page sends your operands to a program on the server that adds, subtracts, multiplies, or divides your numbers as appropriate and sends the results back.

The results then appear in the bottom text field, as you can see in Figure 1.7, where 5 and 7 are added. And it's all done without a page refresh — no browser flicker. This application, like other Ajax applications, looks just as if it's a desktop application.

FIGURE 1.7

An Ajax-enabled calculator



Talk with Ajax chat applications

Ajax is great anywhere intensive updating is required, such as chat applications, where any number of users can type and their text appears automatically to everyone currently logged in. Ajax is a good choice here because the text being displayed is always being updated, and having to watch it flicker as the whole page is updated would be very annoying.

Using Ajax, however, you can update text anywhere in a page easily, no page refresh required. Take a look, for example, at `www.phpfreechat.net/demo.en.php`, the PHP Free Chat page. This page connects to a PHP script on the server to support a chat application. When you first navigate to PHP Free Chat, it asks you to enter a username, as you see in Figure 1.8.

FIGURE 1.8

Signing in for PHP Free Chat

After you've signed in, you can type your text in the text field that appears at the bottom of Figure 1.9; when you press Enter, that text is sent, using Ajax, to the server, which adds that text to the text that others have typed, and the results appear in the chat box, as you can see in Figure 1.9.