Aaron Vegh

Web Development with the Mac®

www.wileydevreference.com

Web Development with the Mac®

Web Development with the Mac®

Aaron Vegh

Wiley Publishing, Inc.

Web Development with the Mac®

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

Published by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-0-470-53399-4

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Library of Congress Control Number: 2010921249

For Richard Moll, my high school teacher
who introduced me to the Mac.

For my dad, who gave me my first Mac: the Classic.

For Chris Pierson, who made me an expert.

And for those nameless armies of Apple engineers
that continue to prove that the Mac matters.

About the Author

Aaron Vegh is the principal owner of Innoveghtive Inc., a web development agency based near Toronto, Canada. Although Innoveghtive has been around for only the past three years, Aaron has been exchanging websites for money since the late 1990s. During his earlier career, he worked for a variety of companies, including investment banks, magazine publishers, and multinational technology corporations.

An avowed autodidact, Aaron taught himself every skill in this book, turning away from his English degree from McMaster University and his master's of publishing degree from Simon Fraser University.

Today, Aaron is a regular presenter at his local Macintosh User Group and an ardent attendee at local Mac developer events. He blogs at `aaron.vegh.ca` and tweets prolifically at `twitter.com/aaronvegh`.

Credits

Acquisitions Editor
Aaron Black

Project Editor
Christopher Stolle

Technical Editor
Dennis Cohen

Copy Editor
Marylouise Wiack

Editorial Director
Robyn Siesky

Business Manager
Amy Knies

Senior Marketing Manager
Sandy Smith

Vice President and Executive Group Publisher
Richard Swadley

Vice President and Executive Publisher
Barry Pruett

Project Coordinator
Kristie Rees

Graphics and Production Specialist
Andrea Hornberger

Media Development Project Manager
Laura Moss

Media Development Assistant Project Manager
Jenny Swisher

Media Development Associate Producer
Josh Frank or Shawn Patrick

Proofreading
Laura Bowman

Indexing
BIM Indexing & Proofreading Services

Contents

Acknowledgments

As you can imagine, a book like this doesn't spring from the loins of a sea goddess; it comes from the persistent labors of colossal machinery, and I'm indebted to each human piece of it.

For graciously putting me in touch with the good folks at Wiley, I'm thankful to Andy Finnell (on Twitter as @macgeek02). Without that initial contact, none of this would be happening.

My acquisitions editor, Aaron Black, was a pleasure to deal with and a true asset to the Wiley editorial team. He was an encouraging figure during the tough moments, and I felt he had my back in selling the idea for this book.

My project editor, Christopher Stolle, was my day-to-day contact while the book was being written. One couldn't hope for a more professional or qualified midwife to see this book to reality.

At home, writing a book of this magnitude took many, many hours away from my family. It's a measure of our strength that they were fully supportive during those long periods of required solitude, and I'm very grateful for the chance. It's makeup time now.

Introduction

The great Carl Sagan once said, "If you wish to make an apple pie from scratch, you must first invent the universe." It feels like that on the web too; before you can build a website, you have to learn a whole galaxy of technologies.

Back when the web was new, it was enough to learn a smattering of HTML. Not anymore: HTML has evolved, and it's just a starting point. Nowadays, if you want to mount a credible website, it needs to be beautiful, functional, and useful.

To achieve this, you need an understanding of techniques that cross traditional disciplinary boundaries. For many companies, getting a professional website means hiring a high-priced agency that's staffed with project managers, programmers, and designers.

Or does it? Can one person running his or her own agency do it all? Can he or she be the one who understands the mechanics of web serving, the details of front-end web development, the touchy-feely world of design, and the logic of application development?

My friend, I'm here to give you a most emphatic "yes."

Who Is This Book For?

This book is useful to anyone who wants to learn about web development. If you work for someone else, you'll find something here to enhance your existing skills. If you run a small company and you're looking to build your own site, this will give you the start-to-finish techniques for getting your site online.

But this book will mostly benefit the independent developer: that lone wolf, that single-minded individual, that seeker of personal fortune. You want to develop your business, and you want to find meaningful clients with substantive websites. You want to be their alpha and omega.

To be sure, being the jack-of-all-trades is not a status gained overnight. You'll build your skills gradually, adding pieces over time. But with focus, dedication, and stubbornness, you'll wake up one morning with this realization: "Yeah, I'm the complete package."

Why the Mac?

I've been using the Mac OS since 1987. That's so long ago that Microsoft was actually making the best software for the platform in the form of Microsoft Word. So, you might imagine that I'm a little biased in favor of the platform. Of course, having spent a substantial amount of time in Windows and Linux, I like to think that my bias is justified.

The advent of Mac OS X back in 2001 paved the way for the Mac as the ultimate web development platform. With its foundations as a Unix operating system, the Mac provides the same technologies that power the Internet; setting up the same environment as the one that will run your production server is a huge benefit.

But the Mac is also a very successful commercial operating system, and it therefore offers a number of terrific applications that make developing websites more enjoyable.

This book is an almost-600-page love song to the Mac OS; there's only one platform that gives you everything you need to do what's in these pages — and that's a Mac.

Tools You Need

Before you get started with this book, you should have a few tools that will make your life easier. The following list offers the best applications as well as some free ones:

- **Text editors.** The Mac doesn't include a built-in text editor. The TextEdit application may sound like it would meet your needs, but by default, this program writes to Rich Text Format (RTF). You need a text editor to write your code, regardless of the language. There are three great options for the Mac:
  - **TextMate.** To my mind, TextMate is the finest text editor on any platform. It's incredibly fast, has great syntax highlighting, and offers a massive set of add-on features. It just feels right to me. TextMate is available from `http://macromates.com` for about $60.
  - **TextWrangler.** From the makers of BBEdit, the longest-living text editor on the Mac, comes the free TextWrangler. It has all but a few of the high-end features of its commercial big brother. Find it at `http://barebones.com`.
  - **Coda.** From Panic Software, Coda is a combination text editor and file transfer solution. It's intended for web developers who have to manage websites, and I find it indispensable for ongoing maintenance. It's available at `http://panic.com` for $99.
- **File transfer programs.** Anyone who works with the web needs a way to transfer files to remote servers. A file transfer application is the way to go, and there are a couple of good options:
  - **Transmit.** Also from Panic, Transmit is the gold standard of FTP applications, and it includes support for secure transfers, WebDAV, and even Amazon's S3 service. It's a $30 download from `http://panic.com`.
  - **Cyberduck.** This is an open-source application that attempts to replicate the features of Transmit, although the interface isn't as polished. Still, you can't beat the price. You can get it from `http://cyberduck.ch`.

How This Book Is Organized

This book has four major parts to describe the different areas that you, as a web developer, must become comfortable with:

- **Part I** describes what I like to think of as the infrastructure technologies that make the web work. You'll learn about DNS, domain names, and hosting. You'll then move into the Unix operating system, and you'll learn how to use Linux to set up a complete hosting environment.
- **Part II** is all about the front-end technologies; you'll learn about HTML, CSS, and JavaScript. These are the tools that will give you the expertise to build static websites.
- **Part III** is where you'll learn about web design — from the basic concepts to the techniques that will drive your website. I also include search engine optimization (SEO) here to help people find your site. You'll also learn how to use Photoshop to build professional-quality site designs.
- **Part IV** is the big payoff; you'll leverage your expertise in static web development and add the dynamic component. You'll learn about databases, PHP, Ruby, and the Rails framework to build large-scale web applications.

Finally, I have included an appendix that provides advice and tips if you're involved in the business end of this field, another appendix with further resources, and a glossary.

This book is a very practical, hands-on guide to web technology. Because it covers a lot of ground, it requires close attention; this is not light reading on your back porch! Instead, I heartily encourage you to sit down at your Mac with this book at your side. Work through the many examples herein, and make sure that you're able to replicate my results. And even more important, continually demonstrate your understanding of the material by playing with your new knowledge; go beyond the examples and try new things for yourself. Be patient and be persistent, and this book will reward you.

Throughout this book, you'll also find icons with more information about a given topic:

- **Caution.** This describes something you should avoid doing or the consequences you might suffer if you attempt to do something that's opposite of what's suggested you do.
- **Cross-Ref.** This tells you where you can find related information to the techniques being described. In some cases, these are reminders of topics that have already been discussed.
- **Note.** This offers additional — albeit parenthetical — information that I have to share on a particular topic. Notes may not be required reading but can contribute to your overall appreciation for the material.
- **Tip.** This presents another way to do something or a shortcut for doing a particular task.

Being a self-employed web developer is a challenge, but with the right advice — and the right attitude — it can be an exciting and lucrative career.

Internet Infrastructure

In This Part

1 Bringing Your Business Online

Whether you want to build one site for yourself or five a month for your clients, you need to know how certain parts of the Internet fit together. In this chapter, I cover the Domain Name System, domain registrars, and hosting options for your website. By the time I'm done here, you should have a good understanding of how to get a website up and running; in fact, you'll do just that on your own Mac, which you'll configure as your primary development machine.

Domain Name System

Every Internet-connected device has an address, known as an *Internet Protocol address*, often just shortened to IP address. Like a telephone number, your computer's IP address has a fixed format: four groups of numbers in a range from 0 to 255, separated by periods.

Because every Internet-facing device must have a unique IP address, the limitations of this protocol become apparent: There are only 4.2 billion possible addresses. While that may sound like a lot, we're actually running out. Some estimates suggest that we'll use the last IP address in 2011. By then, we should have migrated to a new version of the protocol, known as IPv6.

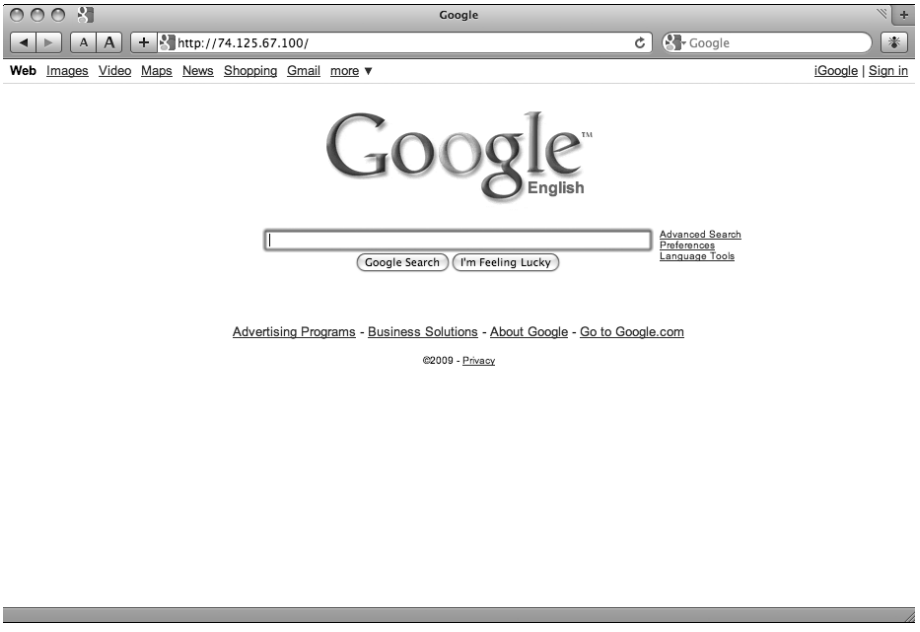
You can use an IP address to connect to any other computer on the Internet. For example, you can open a web browser and type the address **http://74.125.67.100/**, which gives you the Google home page, shown in Figure 1.1.

However, Google — or any other company, for that matter — doesn't advertise itself by its IP address because you're unlikely to remember that string of numbers. That's why we have the Domain Name System (DNS). Like a massive virtual phone directory, the DNS infrastructure provides hostname lookup services: You provide a domain name (like `google.com`), and the system returns an IP address. This happens behind the scenes in most any Internet application, such as web browsers and email clients.

1 In This Chapter

Figure 1.1

You can access the Google home page by using its IP address.



DNS is an amazingly powerful service and is one of the core technologies of the Internet. Like the Internet, DNS is highly *distributed* in nature. This means that there's not just one server that provides hostname lookup but many servers, with each one referring to another in a hierarchy. When a request is made for a website, for example, your computer queries your Internet Service Provider's (ISP's) DNS server for that name. If the DNS server has the record for that name, it provides the IP address for it. If no record is present, the server forwards the request to the ISP's DNS server and up the chain until an answer is found. A simplified diagram is shown in Figure 1.2.

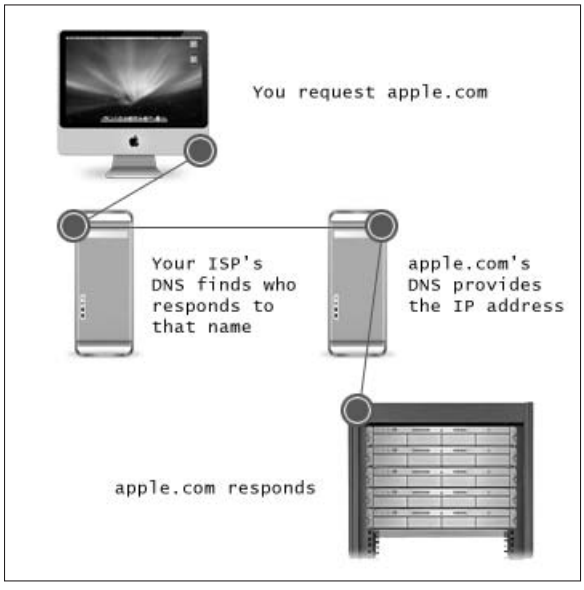
You can find out the IP address of any domain name by using the Terminal application on your Mac. If you haven't used it before, you'll find it in `/Applications/Utilities/`.

NOTE

Terminal provides access to the command-line interface for the Mac operating system. If you're not familiar with the conventions of typing commands and working with text-based feedback, Terminal may seem like a foreign land to you. Hang in there: I'll introduce the command line gradually throughout this book. With enough practice, you'll find yourself appreciating the advantages of the command line and know when it provides the best solution for your needs. For remote administration, Terminal is a necessity, so I'll spend as much time as possible using it for working with online resources.

Figure 1.2

A typical DNS request



With Terminal open, you see a prompt where you can type in commands. Type **dig google.com** and then press Return. You see the following:

```
achilles:~ aaronvegh$ dig google.com
; <<>> DiG 9.4.3-P1 <<>> google.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21768
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 4,
   ADDITIONAL: 4
;; QUESTION SECTION:
;google.com.                    IN     A
;; ANSWER SECTION:
google.com.             159    IN     A      74.125.67.100
google.com.             159    IN     A      74.125.45.100
google.com.             159    IN     A      209.85.171.100
;; AUTHORITY SECTION:
google.com.             345565 IN     NS     ns4.google.com.
google.com.             345565 IN     NS     ns1.google.com.
google.com.             345565 IN     NS     ns2.google.com.
google.com.             345565 IN     NS     ns3.google.com.
;; ADDITIONAL SECTION:
ns4.google.com.         345565 IN     A      216.239.38.10
```

```
ns1.google.com.        345565 IN     A      216.239.32.10
ns2.google.com.        345565 IN     A      216.239.34.10
ns3.google.com.        345565 IN     A      216.239.36.10
;; Query time: 775 msec
;; SERVER: 10.0.1.1#53(10.0.1.1)
;; WHEN: Sun May 31 10:05:01 2009
;; MSG SIZE  rcvd: 212
```

The command `dig` takes a single argument by default: the name of the domain for which you want information. You see a number of sections: a `QUESTION SECTION`, which says that you're looking for `A` records for `google.com`; an `ANSWER SECTION`, which tells you there are three main IP addresses (known here as `A` records) that respond to the domain; and an `AUTHORITY SECTION`, which shows you the DNS servers for the domain.

A DNS server manages different kinds of records. The `A` record provides the address for a web server; an `NS` record provides additional name server addresses; and an `MX` record provides *mail exchangers*, which are servers that respond to email requests.

You can use `dig` to query specific DNS records. If you want to know the mail servers for a domain, you can use a less verbose form of `dig` that provides exactly what you're looking for:

```
achilles:~ aaronvegh$ dig apple.com mx +short
20 mail-in3.apple.com.
20 eg-mail-in2.apple.com.
10 mail-in11.apple.com.
10 mail-in12.apple.com.
10 mail-in13.apple.com.
20 mail-in1.apple.com.
20 mail-in2.apple.com.
20 mail-in6.apple.com.
100 mail-in3.apple.com.
10 mail-in14.apple.com.
```

The short syntax leaves out the verbose formatting of the initial example, providing just the addresses. `MX` records also include a priority; the lower the number, the higher the priority. Given the importance of email, having more servers increases the redundancy, ensuring that messages are delivered even when some machines go offline.

The `dig` command is very useful if you're dealing with a domain that isn't responding. One of the first troubleshooting steps I do is to check if the DNS server has the right IP address for the `A` record: Copy the IP address that `dig` provides and then paste it into a browser.

The next thing you want to know about a domain name is the address of its DNS server. You can acquire that by using a tool called `whois`. This is a way to query the DNS system directly, pulling domain registry information for your domain. What you get back from this command varies depending on the registry — records for `.com` domain names provide voluminous contact information and legal text. But here's a brief domain example: