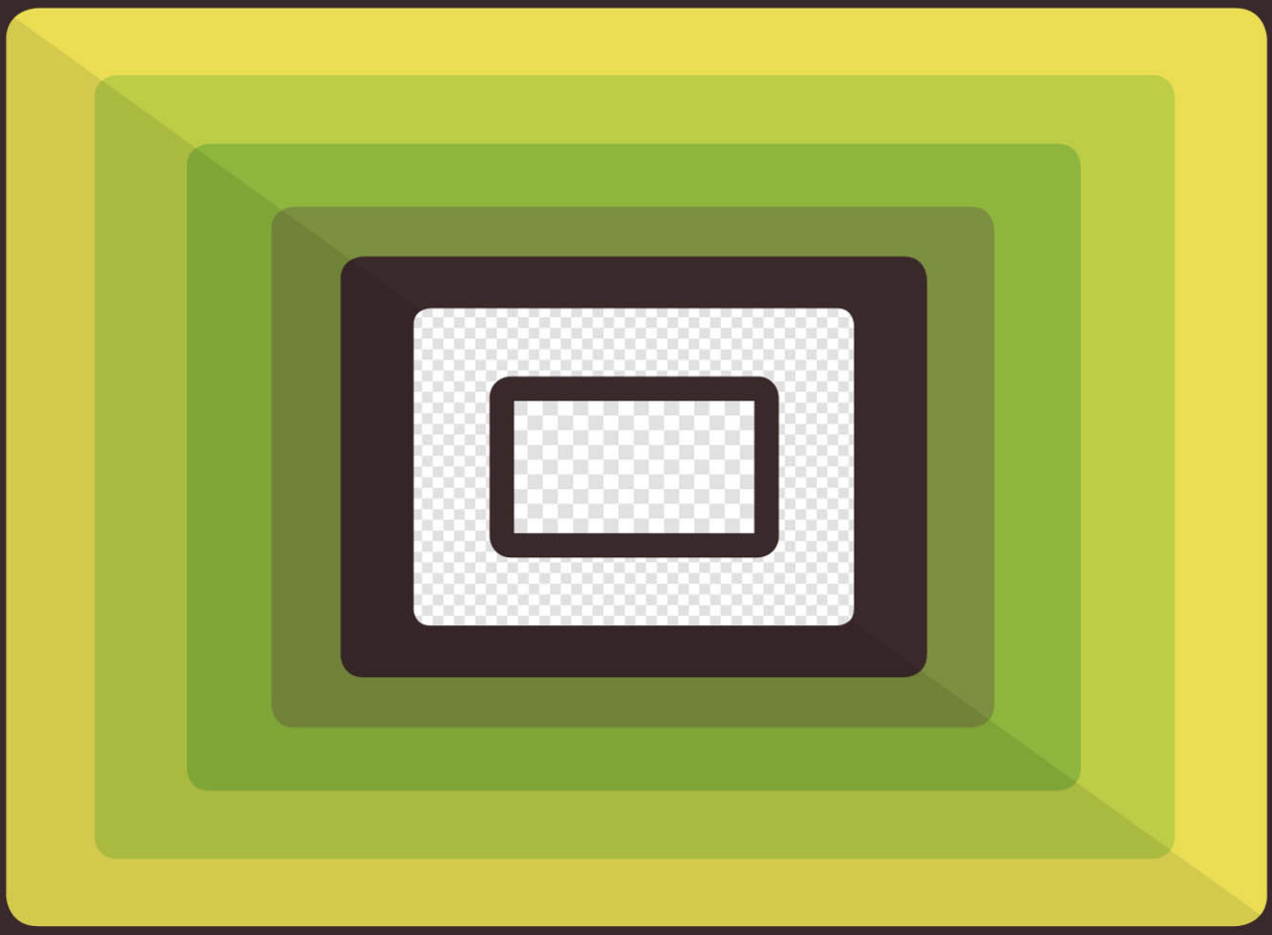
DESIGNING WEB & MOBILE GRAPHICS

Christopher Schmitt

FUNDAMENTAL CONCEPTS FOR WEB AND INTERACTIVE PROJECTS



DESIGNING WEB & MOBILE GRAPHICS

FUNDAMENTAL CONCEPTS FOR WEB AND INTERACTIVE PROJECTS

Christopher Schmitt

New Riders
VOICES THAT MATTER™

DESIGNING WEB AND MOBILE GRAPHICS:

Fundamental concepts for web and interactive projects

Christopher Schmitt

New Riders
www.newriders.com

New Riders is an imprint of Peachpit, a division of Pearson Education.

To report errors, please send a note to errata@peachpit.com

Senior Acquisitions Editor: Michael J. Nolan
Associate Development Editor: Margaret Anderson/Stellarvisions
Version Wrangler: Rose Weisburd
Production Editor: Becky Chapman Winter
Copyeditor: Gretchen Dykstra
Indexer: James Minkin
Proofreader: Jan Seymour
Cover and Interior Designer: Charlene Charles-Will
Compositor: Kim Scott/Bumpy Design
Illustrator: Richard Sheppard

Page 105, Figure 5.13: Engraving by G.J. Stodart after a photo by Fergus of Greenack, before 1890; Figure 5.15: Autochrome photograph by Louis Lumiere, 1910. Page 128, Figure 6.3: Photo courtesy of Flickr user FaceMePLS, under a Creative Commons - Attribution license.

ISBN 13: 978-0-321-85854-2
ISBN 10: 0-321-85854-9

9 8 7 6 5 4 3 2 1

Printed and bound in the United States of America

For Nick, Elisabeth, Matt, Mary Rose, Michael,
Ryan, Megan, Meredith, Gianna

Acknowledgments

A book such as this is blessed to have been touched by many skilled and talented folks.

Thanks to Molly Holzschlag and Estelle Weyl for some last minute edits, and to Christina Ramey for figure and example development.

New Riders is an integral part of the web design community, having published groundbreaking books and continuing to push the industry today. Thanks to Michael Nolan for giving me this opportunity to be part of their legacy.

Special thanks to Rose Weisburd, who remained polite and cool as the whooshing sound of deadlines grew louder and louder. I have profound appreciation for the dedication and resourcefulness of the crew at Peachpit: Becky Winter, Tracey Croom, Margaret Anderson, Gretchen Dykstra, Jan Seymour, Kim Scott, James Minkin, Claudia Smelser, Richard Sheppard. You could not find a finer team anywhere.

From speaking at conferences and helping students to writing books, web education is more than a second job to me. So, a very special thanks to Ari Stiles for understanding how much I love what I do and realizing it's an integral part of who I am. I love her for that.

CONTENTS

INTRODUCTION

In the beginning, print designers created web designs in Photoshop and exported them as one enormous image, declaring that a web site. It wasn't pretty.

Designers started to change their ways after realizing that text and multiple images can not only make designs with HTML and CSS, but make *great* designs. But a part of that process often meant designs would be at least 960 pixels wide or some fixed width.

With the increased adoption of mobile devices like smartphones and tablets that can present a rich web experience in portrait or landscape mode, the desktop browser window width is no longer the standard for which to design. This new mobile component to web design has led us to re-examine our best practices and adopt new techniques.

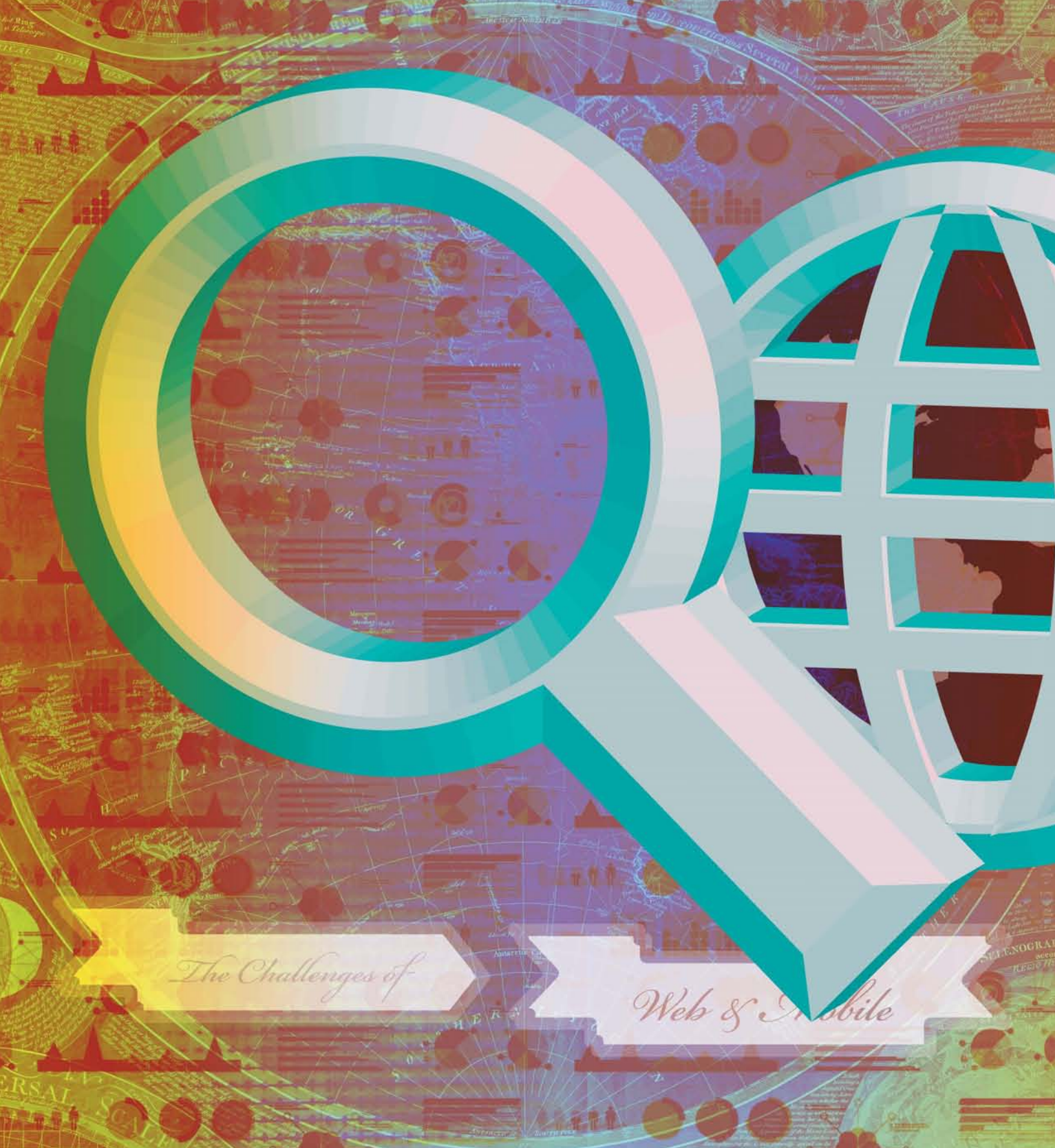
Designing Web and Mobile Graphics is intended to give the beginner or intermediate web designer a look into how to create and build up visuals that meet current web and mobile standards.

In this beautiful web, all these tasks can be done by one person. Each person like yourself, dear reader, has the power to be an independent content producer.

Designing Web & Mobile Graphics provides a foundation in HTML and CSS in the context of development. Building on quick successes from techniques and tips, we move from chapter to chapter to more advanced or unique web design solutions.

GET READY,... GO!

In web design, things are constantly changing. *Designing Web and Mobile Graphics* is the book intended to give you the foundation you need to work with images and much, much more.


The Challenges of
Web &

"I like the challenge of trying different things and wondering whether it's going to work or whether I'm going to fall flat on my face."

—*Johnny Depp*

Chapter **4**

CHALLENGES IN WEB DESIGN

Developing a new version of a browser once took a year; now, browser companies update their products every couple of months. The World Wide Web Consortium (W3C), the web standards body, frequently updates its recommendations on how browsers should work. Users are no longer chained to the desktop; they can access the Web on their smartphones, game consoles, kiosks, and so on. The websites we build need to adapt to meet the challenges of these different browsers and environments.

THE WEB ENVIRONMENT

Browsers have come a long way to produce a great base experience for visitors, but you might be inadvertently creating a situation where visitors see a different presentation than the one you think you're giving them.

Revealing Browser Issues

The following screenshots show how one page compares in different browsers: Internet Explorer 9 (**FIGURE 4.1**), Safari (**FIGURE 4.2**), Chrome (**FIGURE 4.3**), Firefox (**FIGURE 4.4**), Nexus 7 Chrome (**FIGURE 4.5**), Opera (**FIGURE 4.6**), and Mobile Safari iOS on an iPad (**FIGURE 4.7**).

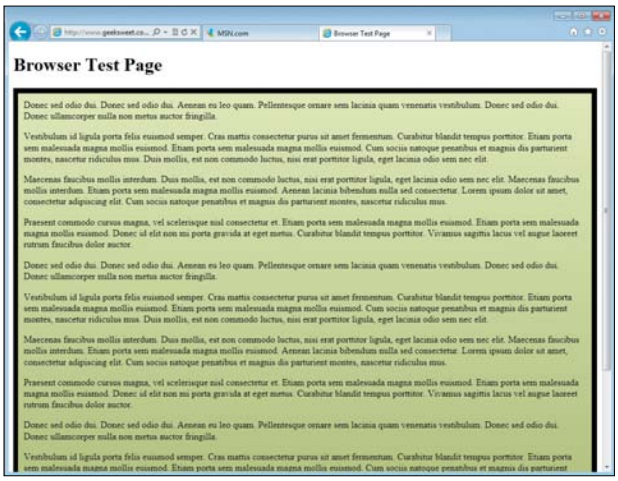


FIGURE 4.1 *A test page in Internet Explorer 9.*

FIGURE 4.3 *A test page in Chrome.*

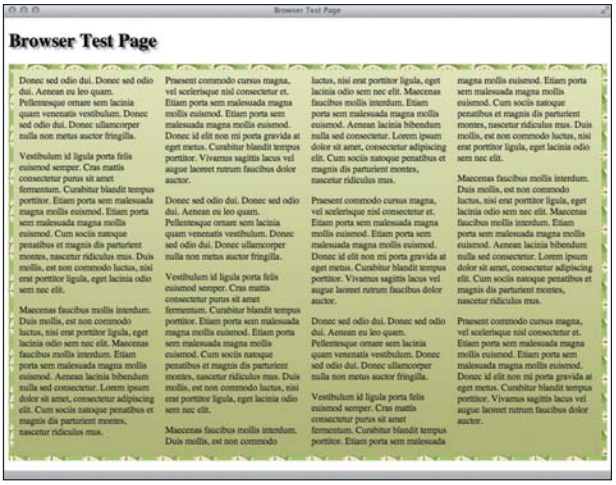


FIGURE 4.2 *A test page in Safari 6.*

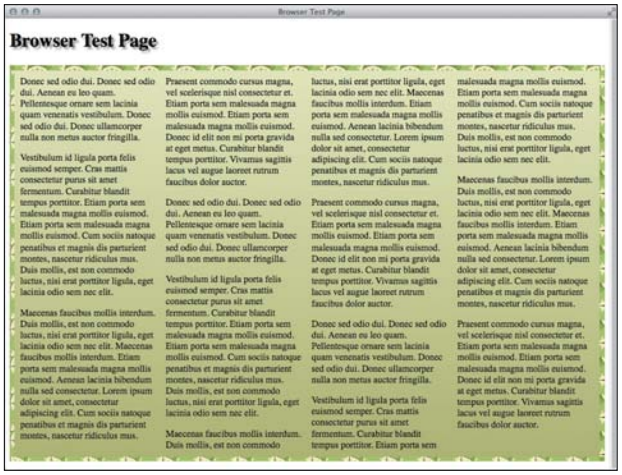


FIGURE 4.4 *A test page in Firefox.*

FIGURE 4.5 *A test page in Nexus 7 Chrome.*

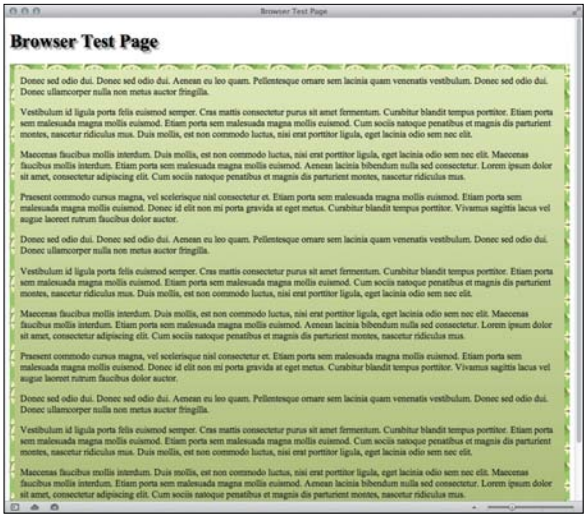


FIGURE 4.6 *A test page in Opera.*

FIGURE 4.7 *A test page in Mobile Safari on an iPad.*

TABLE 4.1 showcases which Cascading Style Sheets (CSS) features are supported by various browsers. A "Y" means the browser supports the CSS feature natively. An "N" means the browser does not support it. A "P" means the browser does support the feature, but it needs a custom CSS prefix in order for the feature to display. The custom prefixes will be explained in more detail in Vendor Prefixes later in this chapter.

TABLE 4.1 *CSS Feature Support by Browser*

Browser	Columns	Border Image	Gradients	Text Shadows
Internet Explorer 9	N	N	N	Y
Safari 6	P	Y	P	Y
Chrome	P	Y	P	Y
Firefox	P	Y	P	Y
Nexus 7 Chrome	P	P	P	Y
Opera	Y	P	P	Y
Mobile Safari iOS on iPad	P	P	P	Y

Using a Test Page

Now let's break down the test page to show which features are in it and how they are implemented. The test page covers some basic and advanced CSS functionality:

- **CSS3 multi-column** lets you set text in columns. In web design, you don't see multiple text columns unless the designer manually adjusts the number of words for each column or uses a JavaScript patch.

```
div {
 column-count: 4;
}
```

- **CSS3 border-image** lets you wrap an image around an HTML element. As the HTML stretches as text is added or removed, for example, the image stretches and adapts.

```
div {
 column-count: 4;
 border-image: url(border-img.png) 10px;
}
```

- **CSS3 gradients** sets color transitions in the background of elements.

```
div {
 background: linear-gradient(to bottom,
 rgba(30,87,153,1) 0%,
 rgba(41,137,216,1) 50%,
 rgba(32,124,202,1) 51%,
 rgba(125,185,232,1) 100%);
}
```

- **CSS3 text-shadows** lets you put one or more shadows on text.

```
div h1 {
text-shadow: 0 1px 1px #bbb,
  0 2px 0 #999,
  0 3px 0 #888,
  0 4px 0 #777,
  0 5px 0 #666,
  0 6px 0 #555,
  0 7px 0 #444,
  0 8px 0 #333,
  0 9px 7px #302314;
}
```

Color

We perceive the color around us thanks to our eyes, which are electromagnetic spectrum detectors. Colors make up only a small portion of this spectrum, which encompasses x-rays, gamma rays, microwaves, radio waves, all the colors we see, and much more.

A computer screen is made of tiny dots, or pixels, arranged in a grid. These pixels change color depending on what the computer instructs the monitor to display (**FIGURE 4.8**).

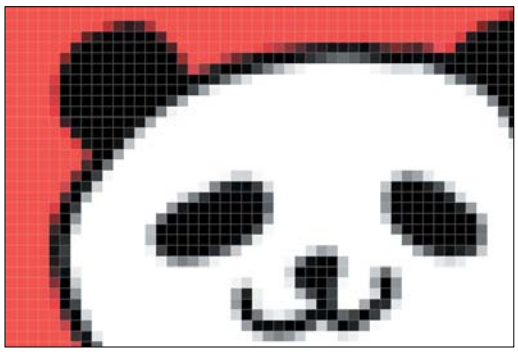


FIGURE 4.8 *Zooming in closely on a raster image shows the blocks of color arranged in a grid like the grid of pixels that makes up the screen.*

! STICK WITH RGB

When you see an option for CYMK in a digital imaging tool like InDesign or Photoshop, it's only there to help give a representation of what the colors look like when *printed*—not how they should appear on screen. Be sure to stick with RGB color mode when creating images or mocking up layouts.

The screens in our desktop PCs and mobile devices don't show colors the same way. No universal calibration system for on-screen color currently exists. Color on computer monitors can vary due to the display brand, the video card brand, the screen's age, the operating system, the amount of ambient light, colors appearing next to each other, and the age and condition of the viewer's eyes. A further translation happens in printing: some colors that are visible on a screen, where they're made with light, can't be printed with any kind of ink or toner (bright, pure blue is the classic example).

No matter how carefully you choose the colors on your screen, they'll never be absolutely accurate since there is no single standard for displaying them.

Screens and Pixels

Units of measurement are expected to be a constant. The problem with constants is that they change.

Take the humble meter, for example. How we determine the starting and ending point of the meter has altered over human history. In the 17th century, a meter was proposed to be part of the distance of the equator to the North Pole. One ten-millionth of the distance to be exact.

In 1875, a meter was then defined as the length of a platinum-iridium bar created by the International Bureau of Weights and Measures near Paris (see http://museum.nist.gov/object.asp?ObjID=37). Numerous bars of the same length were made and distributed around the world.

Some two hundred years later, the meter transformed again and is no longer tied to a physical object. A meter is now the distance *light* travels in a vacuum over 1/299,792,458th of a second. What will it be in the future? And don't ask me what happens to the length of a meter if you happen to be near a black hole, where light can't escape.

Defining a Pixel

The pixel has similar identity issues. What defines a pixel is a matter of when you ask the question. If you are a web designer working in the 1990s, the pixel would be about 1/96th of an inch and it would be fairly constant across operating systems and monitors. *Now* when you ask about the size of a pixel, the answer will depend on which kind of pixel you mean.

These days, people aren't "surfing the World Wide Web" with just the desktop anymore. The web can be accessed on screens that are 3.5 inches diagonal held close to the face, and on 60-inch TVs from across the room. To provide roughly the same experience on this wide range of devices, on-screen elements need to end up looking about the same whether they're being seen from 10 inches or 10 feet.

Let's say we set a head's font size to 24px. If we then look at that head on a retina display and a regular laptop, there should be no difference in size (**FIGURE 4.9**).

The W3C has recommended a standard visual angle pixel size that hardware and software manufacturers can refer to while developing their products. What this means is that web developers can use the CSS pixel as their unit of measurement, and let the browser and OS take care of mapping it to the device pixel, whatever its physical size may be. (There's just one hitch: so far, a perfect solution for scaling up photographic images hasn't been found, as we'll see in Chapter 13, "Images for Responsive Web Design.") The CSS pixel is an absolute length unit that is anchored to the reference pixel, which is an angular measurement (**TABLE 4.2**).

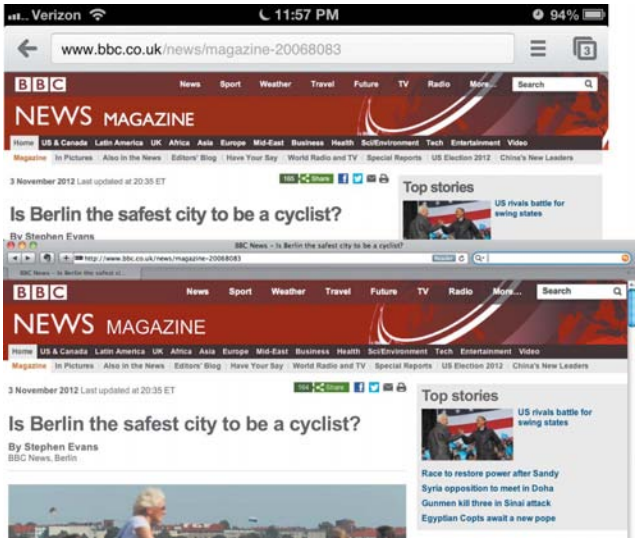


FIGURE 4.9 *The same headline seen in Chrome on an iPhone 5 with a 4-inch, 1136 × 640-pixel screen and in Safari on a MacBook Pro with a 13-inch, 1280 × 800-pixel screen.*

Accessibility

While monitors continue to improve in terms of color clarity, generating millions of colors as faithfully as possible, there are segments of the population who won't be able to see them. Seven percent of men cannot distinguish between red and green colors. Can you see the colors in **FIGURE 4.10**?

These deutan color vision deficiencies, along with others, must be taken into consideration when designing with color.

EYESIGHT STATISTICS According to the World Health Organization, 285 million people worldwide are visually impaired. That includes 39 million blind and 246 million low-vision people.

TABLE 4.2 *The CSS Pixel and the Reference Pixel*

	Type	Derived from	Used by
reference pixel	visual angle	physics of light	hardware+OS makers
CSS pixel	fixed+anchored	reference pixel	designers/developers

FIGURE 4.10 *A "normal" image compared to an image seen as a red-green color blind person sees it.*

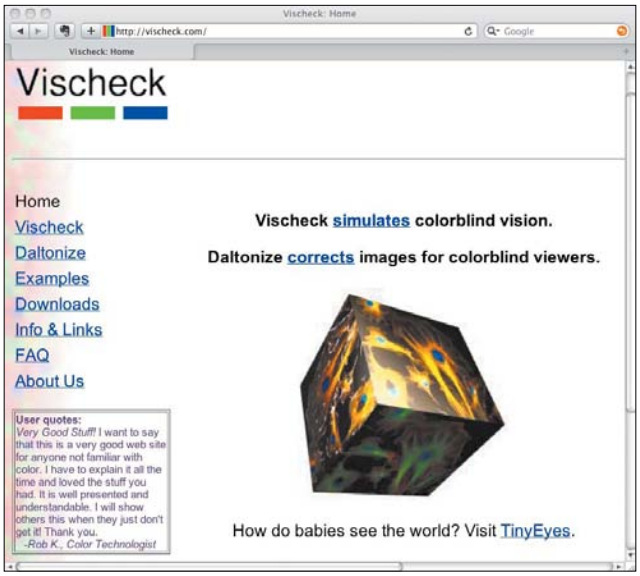


FIGURE 4.11 *The Vischeck page.*

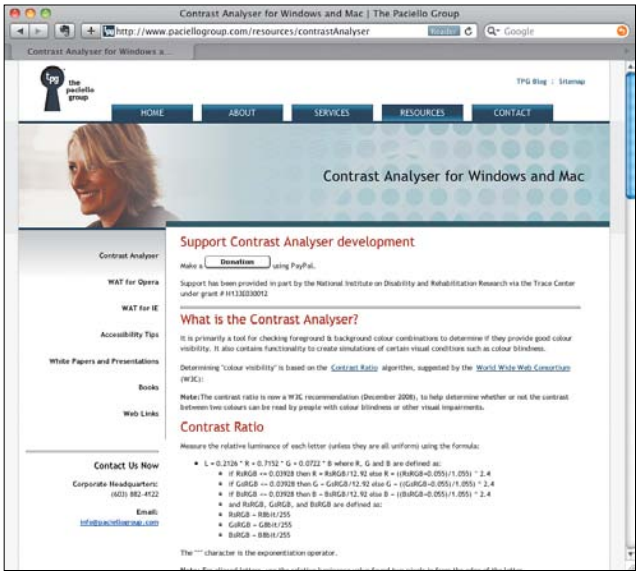


FIGURE 4.12 *The Contrast Analyser.*

Testing for Color Blindness

To check your design for color blindness issues, there are a couple of online tools.

- **Vischeck** (http://vischeck.com/) provides examples of color blindness and converts existing websites to showcase how they appear to people with different types of color blindness (**FIGURE 4.11**).
- **Contrast Analyser** (http://www.paciellogroup.com/resources/contrastAnalyser) uses the W3C's contrast ratio algorithm to determine whether colors have enough visibility or contrast and shows how colors look to people with different types of color blindness (**FIGURE 4.12**).

Color Vision Is Only One Part of Accessibility

Color blindness is just one accessibility issue. Designing for accessibility in general is another of the challenges of web design:

- People with mobility issues, such as those with carpal tunnel syndrome, may prefer to navigate via keyboard (http://webaim.org/techniques/keyboard/).
- People with reduced dexterity appreciate clickable areas that aren't too tiny, and forms that don't time out before they can finish filling them in (http://otal.umd.edu/UUPractice/mobility/).
- People with hearing impairments rely on captions or transcripts (http://webaim.org/articles/auditory/).
- People who are susceptible to photosensitive epileptic seizures want to avoid blinking elements (http://www.w3.org/TR/understanding-WCAG20/seizure-does-not-violate.html).
- People with learning disabilities that affect reading, such as dyslexia, can be helped by good typography, user-selectable type and background colors, and having content available in audio format (http://www.bdadyslexia.org.uk/about-dyslexia/further-information/dyslexia-style-guide.html); audio content also helps people with low vision.

- People with cognitive disabilities affecting memory and attention do better with simple navigation and no distracting animation (http://www.ncdae.org/resources/articles/cognitive/).
- Best practices include the use of **alt** attributes, unambiguous copy, **title** attributes for links, proper HTML structure, and graceful degradation (http://webaim.org/techniques/alttext/).

DETERMINING *YOUR* CROSS-BROWSER GOALS

To determine how to handle different browser issues, you need to understand your visitors. To do that, you need to look at your site statistics.

Analyzing Your Site Traffic

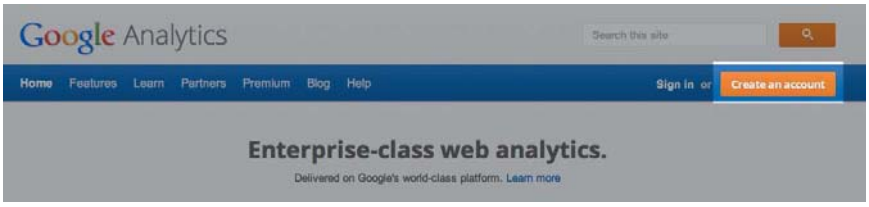
One of the most popular methods for tracking site statistics is Google Analytics (GA), which collects information about site visitors' behavior.

Setting Up Google Analytics

If you already have a Google account (like one for Gmail), use it to set up a Google Analytics account.

TO SET UP GOOGLE ANALYTICS, FOLLOW THESE 6 STEPS:

1. If you already have a Google account, go to Google Analytics at http://www.google.com/analytics/ and click the Create an account button, and then enter your Email and Password and click the Sign In button.



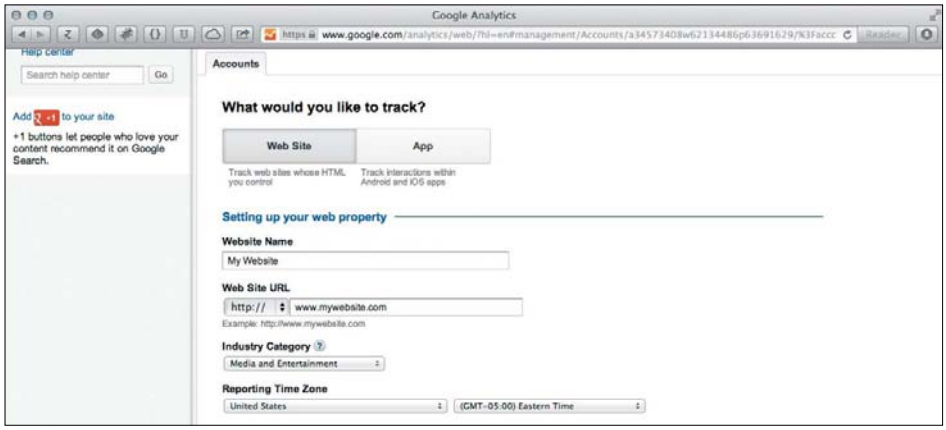
GOOGLE ANALYTICS IS FREE

While there are two available versions, free and premium, most of the time the free version is all you need.

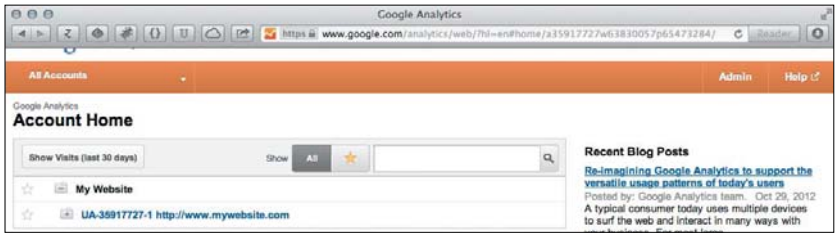
SIGNING UP FOR A GOOGLE ACCOUNT

Don't have a Google account? It's free and opens up a lot of other free resources, like email, web alerts, online word processing and spreadsheets, and more. Sign up at https://accounts.google.com/NewAccount.

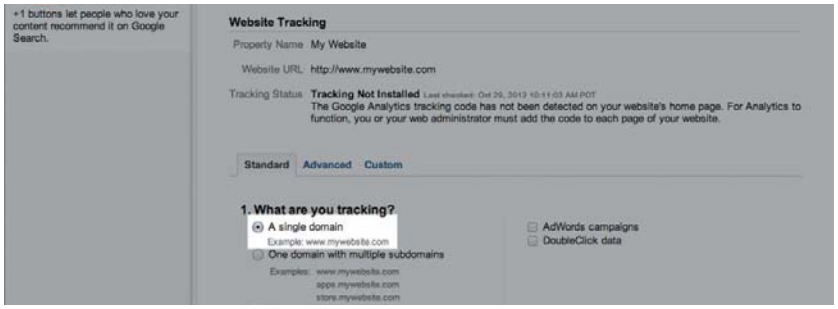
2. Fill out the new account creation form with details about your account name, site URL, industry category, and time zone. Also, review the Terms of Service for the account.



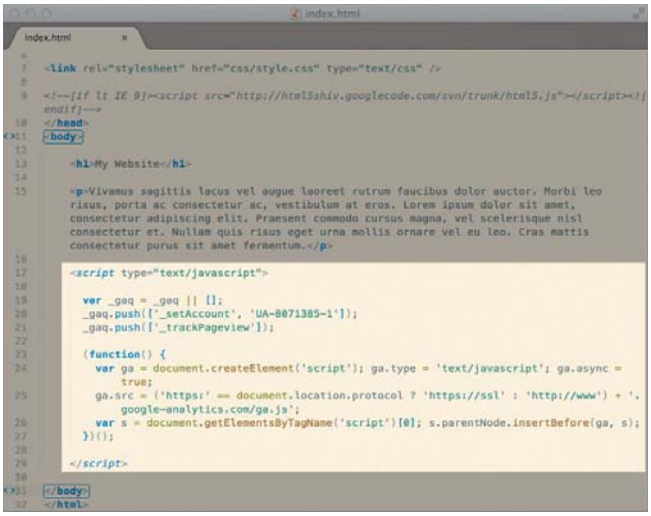
3. At this main page of Google Analytics, you see a listing of the site you've entered. Click Tracking Code from the navigation menu.



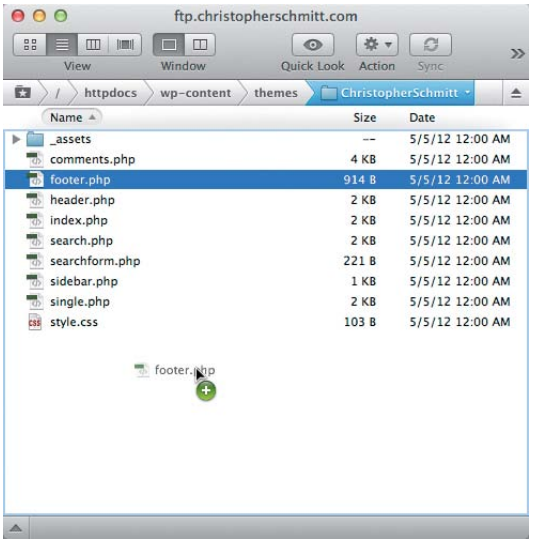
4. Select the domain name you're using. If you're hosting your own domain name, select "A single domain."



5. Copy and paste the code from Google's form field in every page of your site above the closing **body** tag.



6. Upload the new pages to the server.



Once the code has been added to your site, the next step is to wait. It takes anywhere from four hours to a couple of days for Google to start publishing data to review.

! SERVING USERS, NOT OURSELVES

By placing the JavaScript code at the bottom of the document, the browser gets to it later on and focuses on the other elements of the page instead.

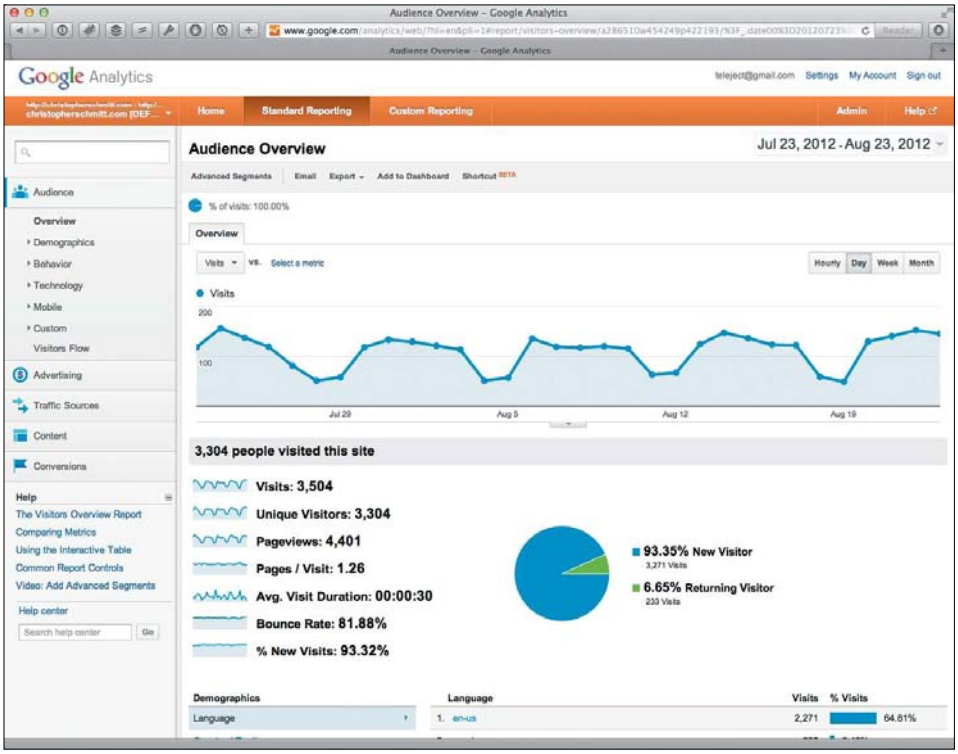
While instructions from Google Analytics state to place the code in the head element, this slows the rendering of a web page ever so slightly. Ensuring that your pages render as quickly as possible is more important than finding out how visitors are using the site. If pages render quickly, then visitors are more inclined to surf more of your site.

Finding Browser Statistics

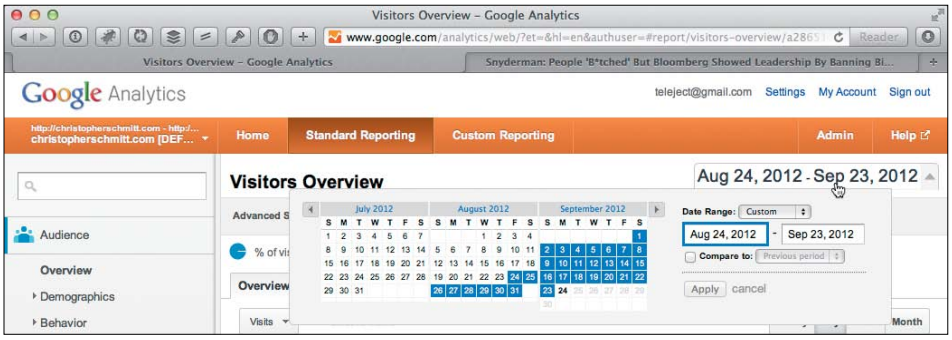
Finding browser statistics about your site's visitors is pretty straightforward. You can find information such as the types of browser your visitors use, and how often they use each browser both in number and percentage. You can even find out these details for specific browser versions.

TO FIND BROWSER INFORMATION ON YOUR SITE'S VISITORS, JUST FOLLOW THESE (5) STEPS:

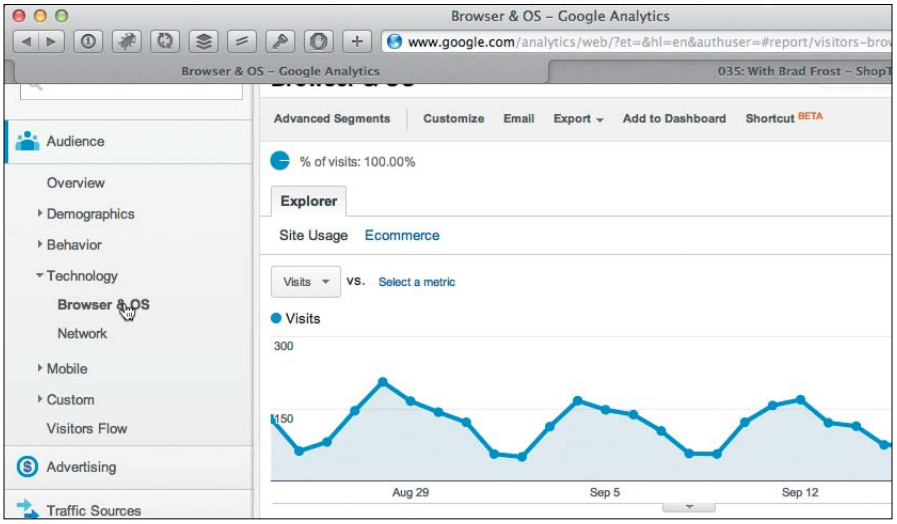
(1) After logging into Google Analytics, you see the Visitors Overview page.



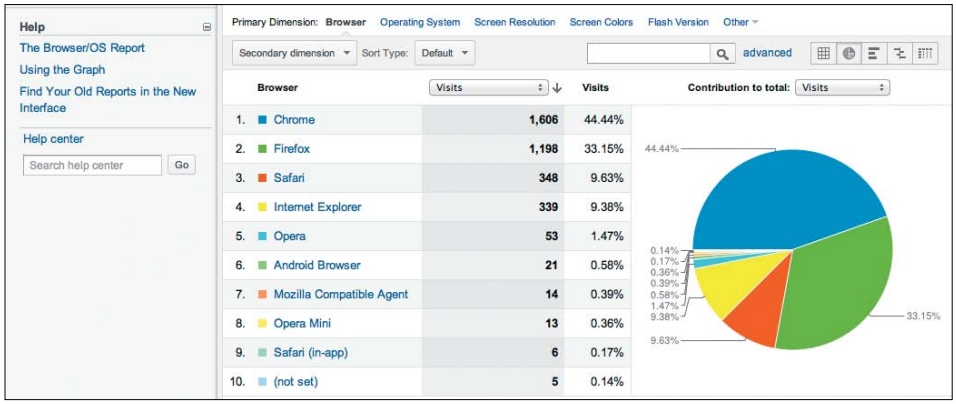
2. Click on the large date text in the upper right corner to select the date range you want to review.



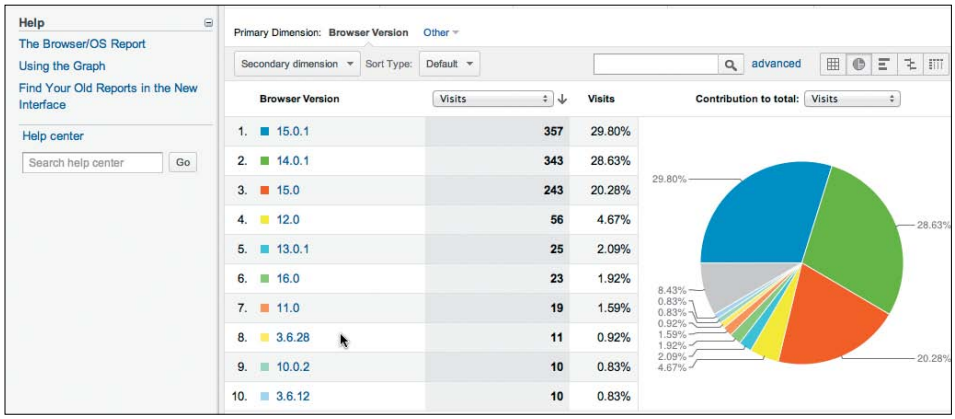
3. In the left column, select Technology > Browser & OS.



4 The different browsers are listed on the page and color-coded to a pie chart.



5 For a breakdown on the different versions of a browser visiting a site, click on the browser name. The example shown here is the breakdown of the versions of Firefox visiting the site.

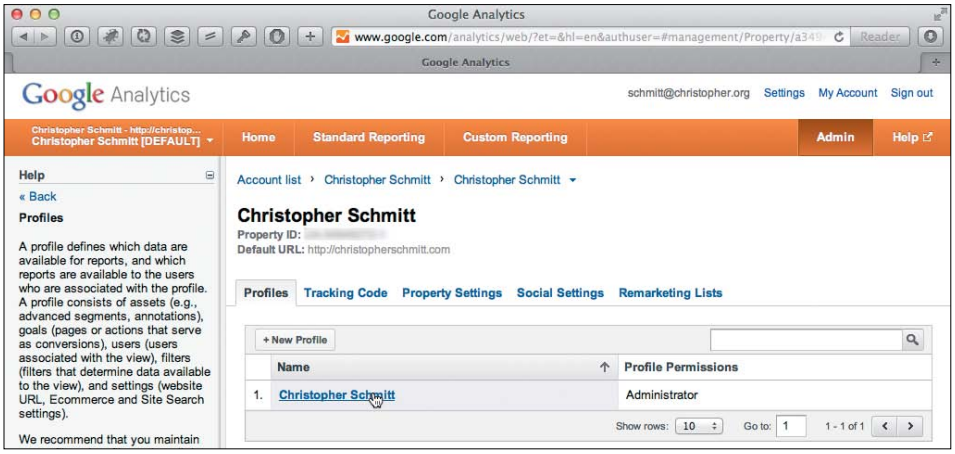


Updating Google Analytics Settings

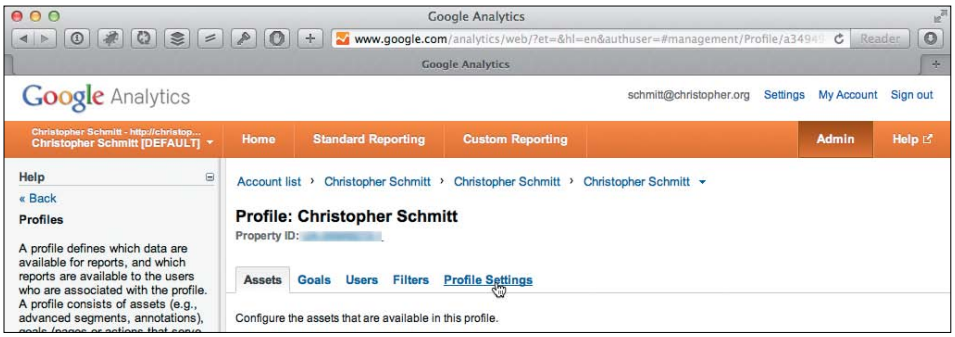
Going back to your profile to keep your site information and goals updated ensures that Google Analytics is gathering the right data to give you an accurate picture of your site usage.

TO UPDATE SITE INFORMATION AS IT CHANGES, FOLLOW THESE 4 STEPS:

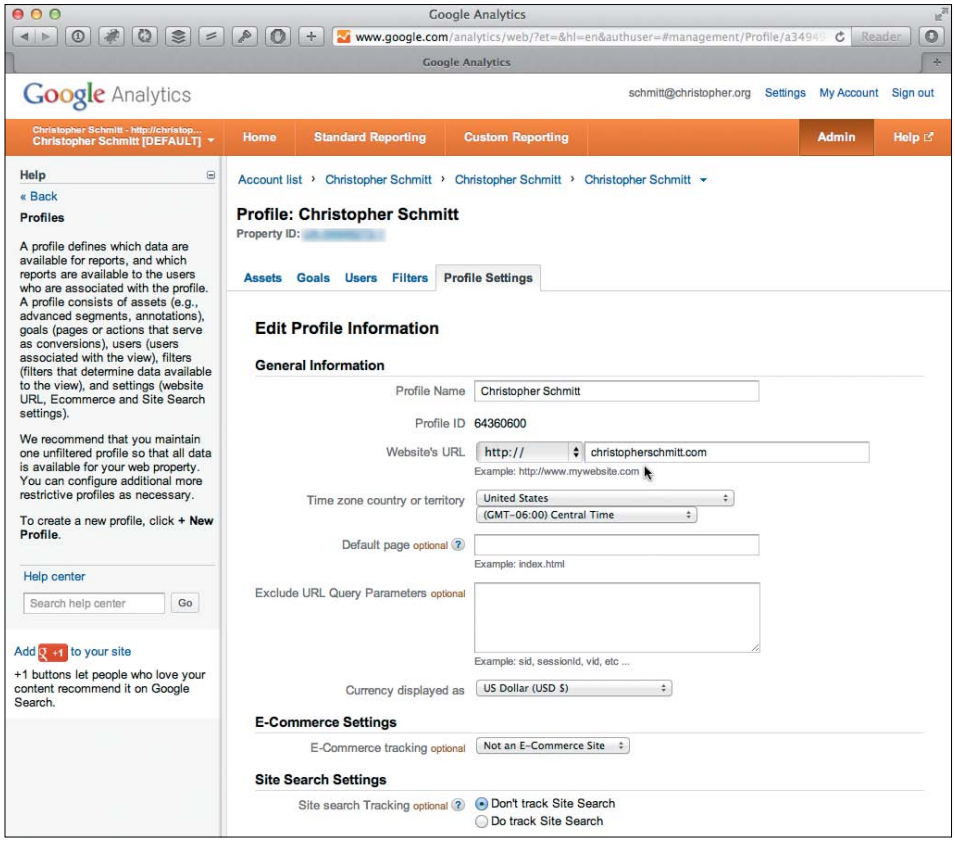
1. Click on the Profile name.



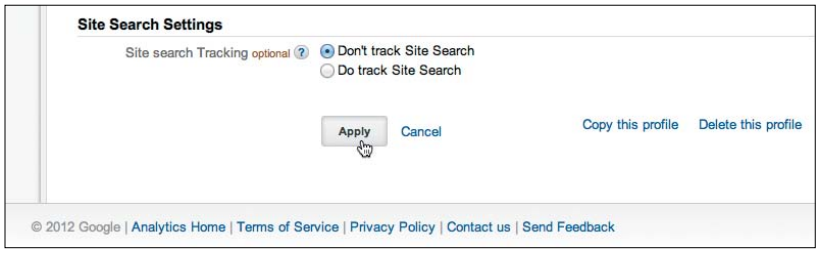
2. Select Profile Settings from the navigation submenu.



3 Make updates to the information.



4 Select Apply.



DEVELOPING YOUR SITE FOR DIFFERENT DEVICES

As we have found out already, browsers have varied support for CSS features. Another difference in browsers is in how they display HTML text and white space that doesn't have any CSS customization. The underlying browser styles can influence the CSS presentation layer, so we try to take the browser styles back to a basic and common foundation.

Resetting and Normalizing Browser Styles

Browsers have their own internal style sheets, which designers use to render basic default styles for everything from the distance between lines of text to link colors (**FIGURE 4.13**).

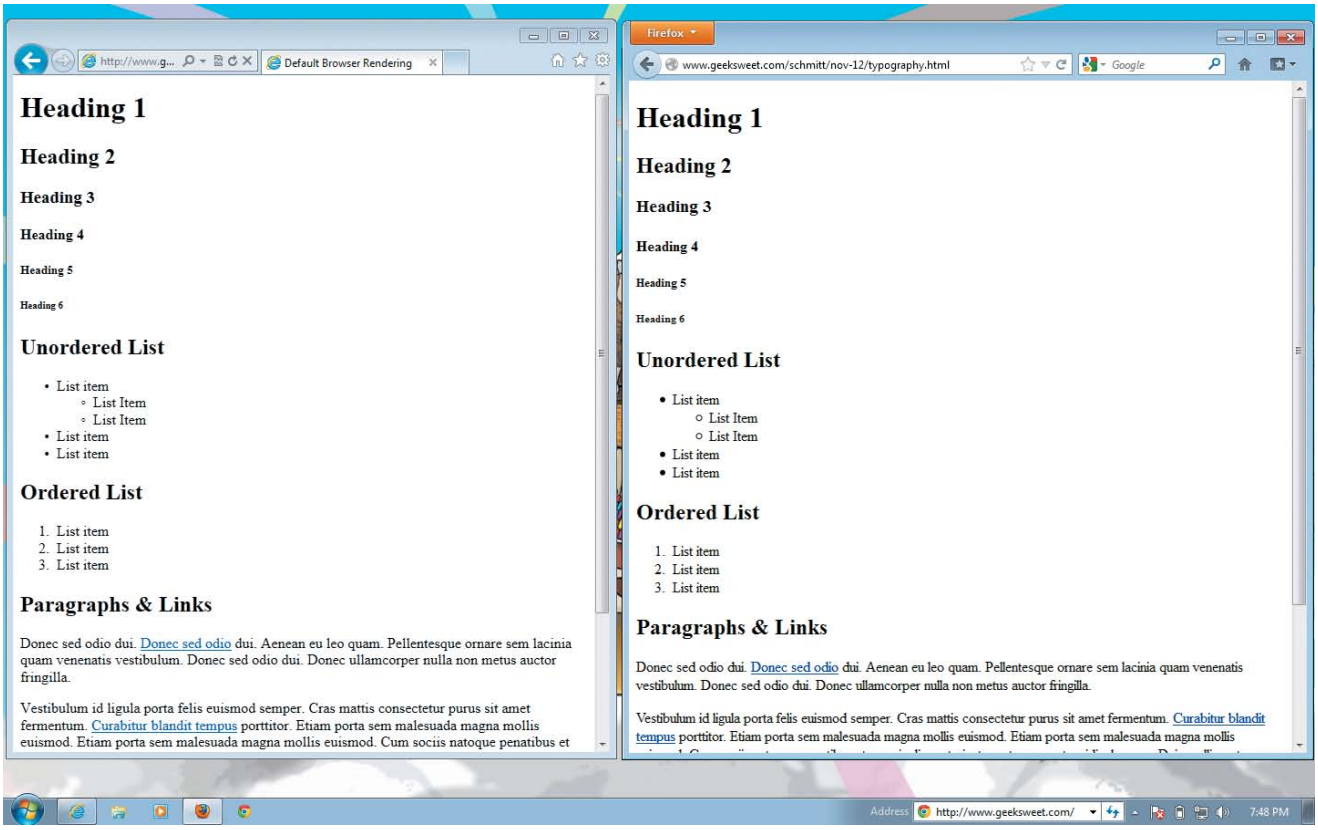


FIGURE 4.13 *Differences in Internet Explorer 9 vs. Firefox browser rendering on Windows.*

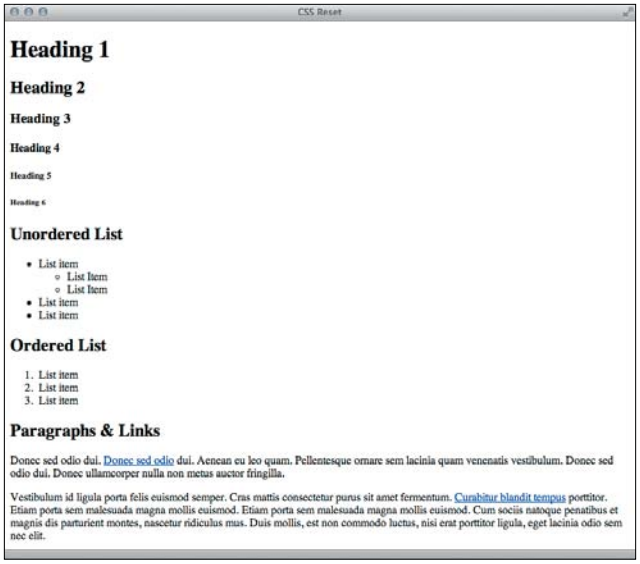


FIGURE 4.14 *Before the CSS Reset.*

Resetting Styles

One way to address browser inconsistencies is to remove or set all CSS properties to zero. This is done through a CSS style sheet known as CSS Reset, like the one premade from Yahoo! through its YUI Library. To set up a reset for a web page (**FIGURE 4.14**), place a link element at the top of any other reference to the style sheet (**FIGURE 4.15**):

```
<link rel="stylesheet" type="text/css"
href="http://yui.yahooapis.com/3.7.1/build/
cssreset/cssreset-min.css">
<link rel="stylesheet" type="text/css"
href="style.css">
```

Normalizing Styles

Using the CSS Reset approach means that all default settings are dialed back. What's left is an empty canvas with no hint of design or standards. That's where **`normalize.css`** steps in. Rather than removing everything, Normalize.css (http://necolas.github.com/normalize.css/) creates a cohesive standard for the default rendering of HTML elements (**FIGURE 4.16**).

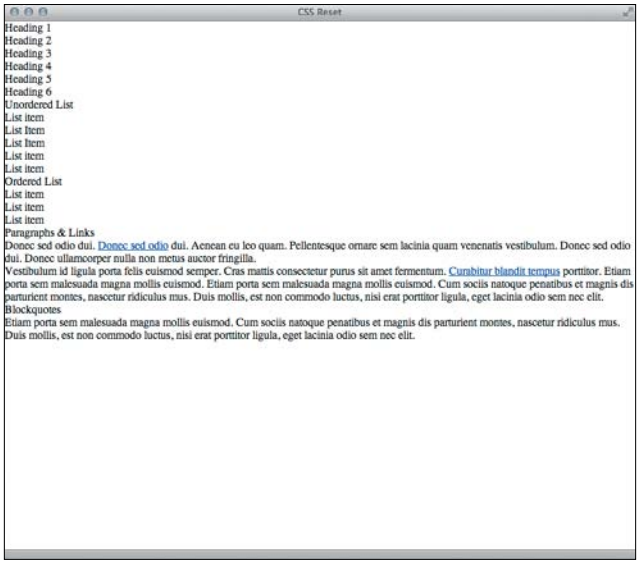


FIGURE 4.15 *After the CSS Reset.*

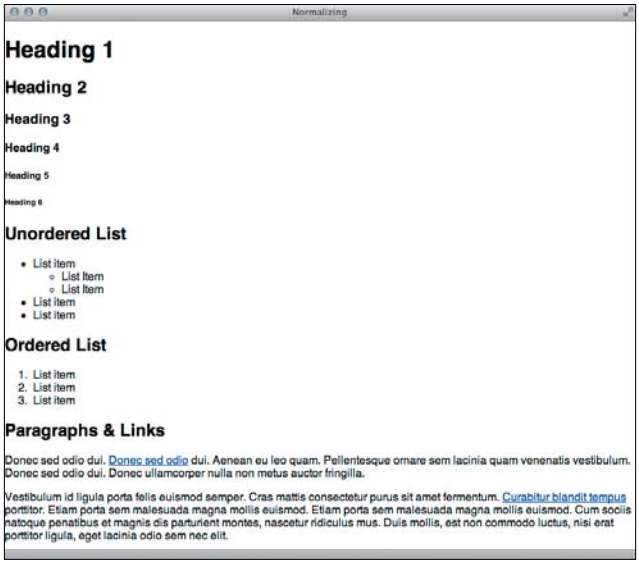


FIGURE 4.16 *After* **`normalize.css`**.

Normalizing style sheets gives you a solid base to build web pages without having to worry about little discrepancies in browser renderings.

Vendor Prefixes

Sometimes a new CSS feature brought into a browser is still in its beginning stages and needs more work. When this happens, the browser vendor usually provides what's called a **prefix CSS property**.

The Simple CSS Workaround

For example, a simple CSS gradient like the following code will be ignored in Safari 6:

```
div  {
 background-image: linear-gradient(#fff, #000);
}
```

To get Safari to work, we have to add a line of code that only Safari will recognize:

```
div  {
 background-image: -webkit-linear-gradient(#fff, #000);
 background-image: linear-gradient(#fff, #000);
}
```

The software that powers the rendering or display of a page in Safari is known as WebKit. Therefore, to specify a special feature like a gradient, we need to add "webkit" surrounded by hyphens. To support other browsers that have the same approach to new CSS features, we need to add the special vendor prefix for each of them (**TABLE 4.3**):

```
div  {
 background-image: -webkit-linear-gradient(#fff, #000);
 background-image: -moz-linear-gradient(#fff, #000);
 background-image: -ms-linear-gradient(#fff, #000);
 background-image: -o-linear-gradient(#fff, #000);
 background-image: linear-gradient(#fff, #000);
}
```

TABLE 4.3 *WebKit Vendor Prefixes*

Browser	Vendor Prefix
Safari	-webkit-
Firefox	-moz-
Internet Explorer	-ms-
Opera	-o-

WHY DO BROWSERS HAVE UNTESTED FEATURES?

Sometimes browser vendors want to gain an advantage over their competitors, so they put out a new feature that the others don't yet have. This scenario played out over and over again in what's been called the Browser Wars. Other times, browser vendors want to see if a new feature will be adopted by web developers, or they might feel that a feature is worthwhile to implement without consulting other interested parties.

The group that writes the CSS standards will not approve a CSS property that begins with a hyphen. This enables browser vendors to create vendor-prefixes that start with hyphens for the testing of features. For the specification, check out http://www.w3.org/TR/CSS21/syndata.html#vendor-keywords.

Automatic Vendor Prefixing

For one CSS feature in our text shadow example, five additional lines of code were needed to support browsers. If we need to add additional lines of code for each browser for each new CSS feature, the lines of code we generate would quickly get out of control. Web developer Lea Verou realized the vendor prefixes situation and developed a piece of JavaScript called -prefix-free (http://leaverou.github.com/prefixfree/), shown in **FIGURE 4.17**:

FIGURE 4.17 *The -prefix-free homepage.*

By downloading the file and placing the -prefix-free JavaScript file in the head element of your web page, the vendor-prefixes are automatically applied to the user's browsers *if* they need them:

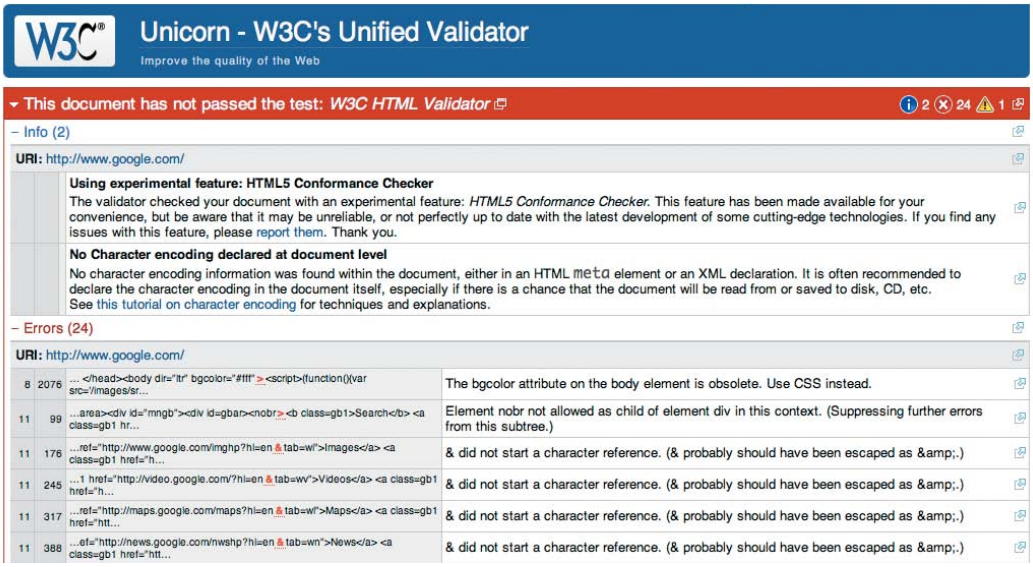
```
<head>
<meta charset="utf-8">
<title>My Web Page</title>
<link rel="stylesheet" href="style.css" />
<script src="prefixfree.js"></script>
</head>
```

That means you can focus on writing clean code and not worry about double-checking browser support features in the CSS.

Validation

Once you're satisfied with your design, you'll want to validate your HTML and CSS code to make sure it complies with all the requirements and rules that we've established thus far. A **validator** is a tool that checks the code for proper HTML and CSS syntax by making sure that all tags are closed and properly nested. If there are no syntax errors, then the page is said to *validate*.

FIGURE 4.18 *Validator reporting errors.*



More than likely your pages won't validate the first time you code them—it's surprisingly easy to forget a tag or a quote. Luckily the validators are specific as to what the error is and where it's located (**FIGURE 4.18**):

At first, validating your code may seem impossible, but it becomes easier as you get used to what clean, syntactical code looks like. The W3C has a markup validation service at http://validator.w3.org/unicorn/. Simply upload your document or provide its address, and the validator quickly gives you a report on both HTML and CSS errors. Working through the validator can be a little puzzling at times because of the error messages, but once you clean up any errors you get a clean bill of health (**FIGURE 4.19**).

Another valuable tool is HTML Tidy (http://infohound.net/tidy/). This tool actually fixes badly formed markup: it adds closing tags, changes mismatched tags, adds quotes to attribute values, and properly nests tags that are not nested (**FIGURE 4.20**). Standalone tidy applications are available for various platforms, and there are online versions as well. HTML Tidy also comes with many HTML and text editors.

BUILT-IN VALIDATORS WYSIWYG editors like Dreamweaver come with their own built-in validators.

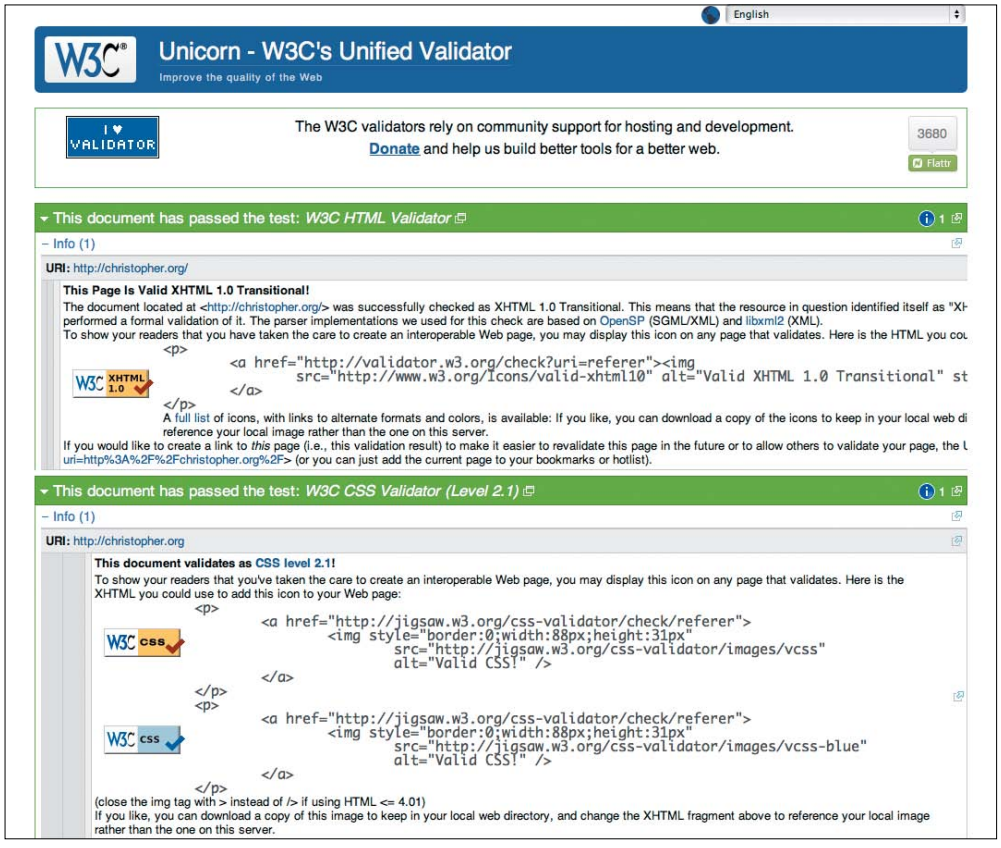


FIGURE 4.19 *Gaining the coveted seal of approval from a validator.*

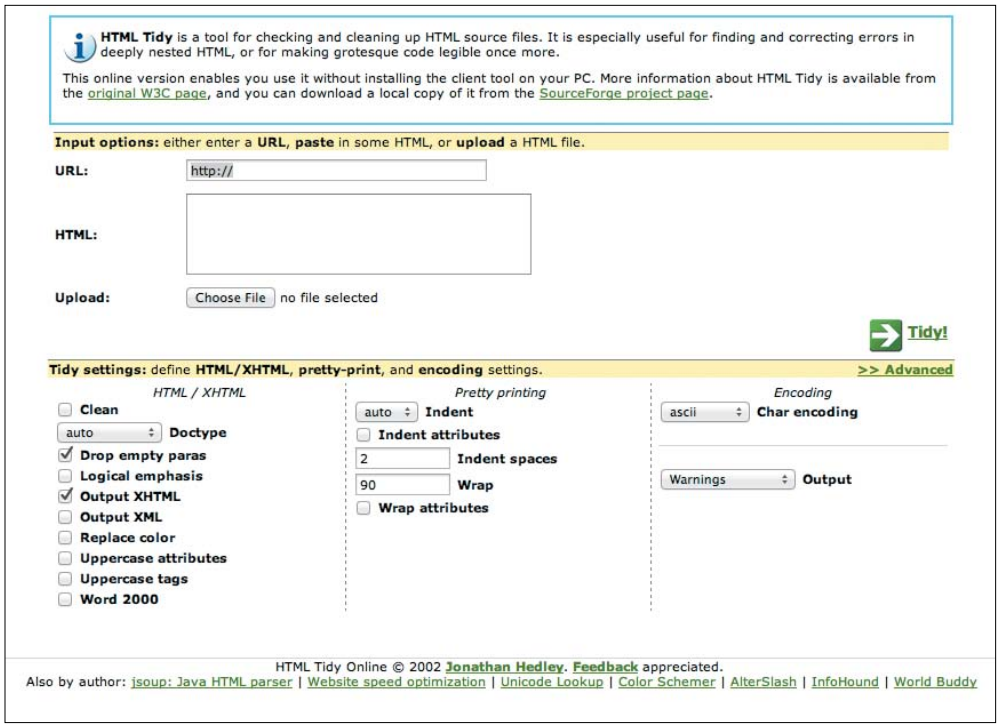


FIGURE 4.20 *HTML Tidy makes your code cleanly spaced and balanced.*

Testing

When designing for the web, it's important to test against as many browsers and devices as possible. It can be difficult to maintain a number of desktop browser installations—both new and older—and get your hands on a wide range of devices to create a full mobile testing suite. Here are some suggestions to make cross-device development a little less complex.

Software

Use services like BrowserStack (http://www.browserstack.com/) to quickly test different platforms, browsers, resolutions, and connection speeds (**FIGURE 4.21**).

If you own a Mac, buy a software package like Parallels (http://www.parallels.com/products/desktop/) or VMware Fusion (http://www.vmware.com/products/fusion/overview.html). You can download Oracle VirtualBox (https://www.virtualbox.org/) for free (**FIGURE 4.22**). These software packages allow for **virtualization**—running Windows and Windows-based browsers on a Mac. You'll still need to purchase a Windows OS license, but you won't have to buy a separate machine.

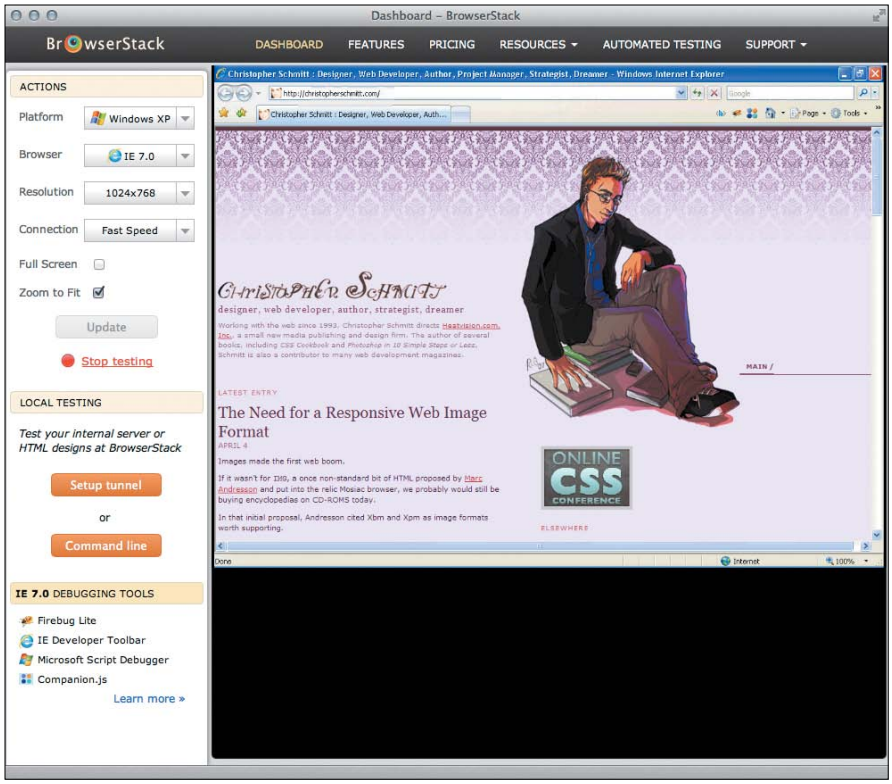


FIGURE 4.21 *Using the BrowserStack interface to check out a page design.*

FIGURE 4.22 *Oracle VirtualBox.*

Hardware

After creating a test page, upload it to the web and then take a breather. Go to your local computer store and check out your web page design in the store's display models. Check in on cell phone stores, too, to see how your page looks in their mobile devices.

If you can, buy a smartphone on eBay or ask friends and colleagues for their old phones and devices when they upgrade. You might be surprised how many people have old devices hanging around in a junk drawer. When *you* upgrade to a new device, keep your old one around for testing to create your own mini testing lab. Use services like Adobe Edge Inspect to test web pages on different devices. MobileTh.at is a new service, just being coded at the time of writing, that promises to simplify mobile testing.

IN CONCLUSION

Browsers have different levels of support for CSS features, as well as different interpretations of how to render HTML. By using tools like normalize.css and validating on multiple devices, we can address many of the challenges of web design. However, there's no substitute for knowing and learning about your audience and working to build them the best site possible. In the next chapter, we'll look at one of the most powerful parts of graphics and design: color.

This page intentionally left blank

INDEX

D

E

F

G

H

I

P

R

S

T

U

V

W

X

Y

Z