This chapter is a bit different than the previous because its sole purpose is to get you excited about the new features coming soon to the browsers you use every day. Some of these features are rolled into the current HTML5 specification; others remain outside the HTML5 domain, are in specific browsers only, or are set for HTML.next (http://w3.org/wiki/HTML/next).

In this chapter, I’ll cover some really amazing APIs and features such as Web Intents, WebSockets, WebGL, WebRTC, bleeding-edge CSS3, and others. I’ll also focus on the emerging browsers from the Google Chrome team, Mozilla, Apple, and Opera, as well as to discuss how they, as well as Microsoft, Adobe, W3C, and WHATWG, are making specific additions to the open web standard that will impact your campaigns moving forward.

As you’ve learned throughout the book, HTML5 is an evolving specification, but working group members currently are determining what should stay in HTML5 and what should be pushed off until HTML.next. It may seem odd that there are two different specifications of the HTML5 specification, one managed by the W3C, which aims to make HTML5 a snapshot specification, and one managed by the WHATWG, which aims to make the HTML standard an organic, ever-growing document that builds and iterates through the years of the open Web’s growth. However, it’s not terribly important to get wrapped up in which specification is which; just be sure to know when the browsers have implemented these new features and when user adoption will take place.

Finally, please take this chapter with a “grain of salt.” A lot of these features may not make it into the HTML5 specification and could be pushed off to the next iteration, or some could even be retired. However, be aware that working groups are committed to creating and enabling amazing things inside your browsers, and being on top of them allows you to better prepare for what your clients will eventually ask for. Whether it lands in HTML5, 6, 6.x, or something else entirely, shouldn’t matter much. If you can use a feature and, more importantly, use it where your target audience is, that’s the most important part to take away from this chapter. That said, let’s dig into the really amazing and bleeding-edge features of the modern Web!

Emerging Browsers

In this chapter, I will showcase what’s to come, in other words, the emerging and most forward-thinking stuff that’s being cooked up by some really smart people and coming to your browser. As I’ve stressed in previous chapters, you should be working with the latest version of your favorite browser, and to be quite honest, if you’ve made it this far in the book using IE 6 to IE 8, God bless you, and please send me your address so I can personally come to your house and install an update for you.

For this chapter specifically, you should be working with one of the browsers shown in Figure 12-1.
You can download these bleeding-edge browsers at one of the following locations: Firefox Aurora (http://aurora.mozilla.org), Chrome Canary (http://tools.google.com/dlpage/chromesxs), Opera Next (http://opera.com/browser/next), and Webkit Nightly (http://nightly.webkit.org). Keep in mind that these browsers are not 100 percent stable and should be used with caution in production environments because there could be bugs in the code base. It’s best to use these browsers only for testing experimental features.

Note At the time of this writing, Internet Explorer does not have a beta version of its latest browser (though IE 10 is set to be very bleeding edge).

Downloading one or all of these browsers gives you VIP access to all the beta features that these ultra-modern browsers grant access to.

New CSS Features

Now that the setup is out of the way, let’s head into the important topics of this chapter, starting with emerging CSS features. The following sections are geared to many of the new enhancements of the CSS specification.

CSS Regions

First up is the new CSS feature called regions. Adobe has submitted a draft to the W3C for this feature, which is effectively a new specification for free-flowing text from one region of content to another. This allows you to have freely moving text that is device and screen independent, which allows for a great addition for copy layout in responsive web and creative designs. As you know from your knowledge of synced ad units, you could in theory have free-flowing text from one ad into another on the same page. To use CSS regions, all you’ll need to do is include some empty div containers and some CSS declarations, as shown in Listing 12-1.

Listing 12-1. CSS Regions Example

```html
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<style type="text/css">
```