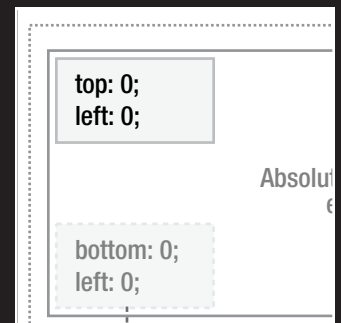
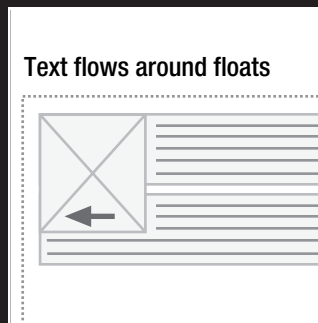
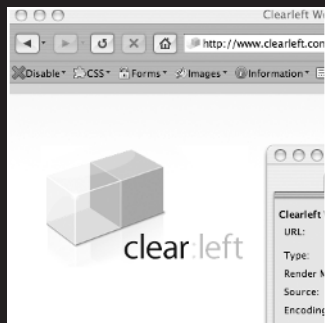


## 9 BUGS AND BUG FIXING



Compared to many programming languages, CSS is a relatively simple language to learn. The syntax is straightforward, and due to its presentational nature, there is no complicated logic to grapple with. The difficulties start when it comes time to test your code on different browsers. Browser bugs and inconsistent rendering are a major stumbling block for most CSS developers. Your designs look fine on one browser, but your layout inexplicably breaks on another.

The misconception that CSS is difficult comes not from the language itself, but the hoops you need to jump through to get your sites working in all the major browsers. Bugs are difficult to find information on, poorly documented, and often misunderstood. Hacks are seen by many as magic bullets—arcane sigils with exotic names that, when applied to your code, will magically fix your broken layouts. Hacks are definitely potent tools in your armory, but they need to be applied with care and generally as a last resort. A much more important skill is the ability to track, isolate, and identify bugs. Only once you know what a bug is can you look for ways to squash it.

In this chapter you will learn about

- How to track down CSS bugs
- The mysterious `hasLayout` property
- The most common browser bugs and their fixes

## Bug hunting

We all know that browsers are buggy, some of them more than others. When a CSS developer comes across a problem with their code, there is the immediate temptation to mark it as a browser bug and apply the appropriate hack. However, browser bugs aren't as common as everybody likes to think. The most common CSS problems arise not from the browser bugs, but from an incomplete understanding of the CSS specification.

Many developers are self-taught, and build up a mental model of how they believe things should behave. When something doesn't work the way they expect, the natural temptation is to blame the browsers and reach for a hack. To avoid these problems, it is always best to approach a CSS bug assuming that you have done something wrong. Only once you are sure that there are no errors on your part should you consider the problem to be the result of a browser bug.

## Common CSS problems

Some of the simplest CSS problems are caused by typos and syntactical errors in your code. One of the best ways to prevent these types of bugs is to run your code through the CSS validator (<http://jigsaw.w3.org/css-validator/>). This should pick up any grammatical errors, showing you the lines the errors are on and a brief description of each error (see Figure 9-1).