**Specify Everything**

A lot of cross-browser issues amount to this: you didn't specify something, and different browsers make different assumptions. Therefore:

Declare a valid doctype

Your doctype tells the browser what rules you'll be using in your code. If you don't specify, the browser has to guess, and different browsers will guess differently.

In my experience, a "strict" doctype makes IE behave better (enables things like CSS :hover selectors on divs in IE7).

Use Web standards

Avoid browser-specific markup, or only use it when its failure in other browsers won't be significant to the site experience.

Validate your HTML and CSS

You don't have to get everything perfect, but validation is good feedback. As [Jeff said](http://www.codinghorror.com/blog/archives/001234.html):

Knowing the rules and boundaries helps you define what you're doing, and gives you legitimate ammunition for agreeing or disagreeing. You can make an informed choice, instead of a random "I just do this and it works" one.

Imagine you opened a paragraph tag and never closed it. If you then open a list tag, did you mean it to be inside the paragraph or not? Validating will help you catch that, close the tag, and eliminate ambiguity.

Consider a CSS Reset

Different browsers assume different baseline CSS rules. You can help them all to act the same by explicitly ironing out the differences up front. Eric Meyer, who wrote [CSS: The Definitive Guide](http://meyerweb.com/eric/books/css-tdg/), uses[this reset](http://meyerweb.com/eric/thoughts/2007/05/01/reset-reloaded/). Another popular choice is [YUI Reset CSS](http://69.147.76.210/yui/reset/).

**Use a Javascript library for DOM interactions**

Whenever your Javascript needs to work with elements on your page, it's best to use a library like jQuery, Prototype, or MooTools. These libraries are used by many thousands of developers, and they take most of the inconsistencies between browsers' interpretation of Javascript, deal with those internally, and give you a consistent set of commands that just work. Trying to find and work around all these inconsistencies yourself is a waste of time and likely to create bugs.

**Test in multiple browsers, deal with IE last**

Test in multiple browsers as you go. Generally, you'll find that non-IE browsers behave similarly and IE is a special case - especially if you follow the advice above. When necessary, you can add IE hacks in a separate stylesheet and only load it for IE users.

[Quirksmode.com](http://www.quirksmode.org/) is a good place for hunting down random browser differences.

[Browsershots.org](http://browsershots.org/) can help show how your page will be displayed in an assortment of browsers and operating systems.

**Fail Gracefully**

No site will look perfect in every browser that exists. If a user doesn't have Flash, or Javascript, or advanced CSS, etc, you want your site to be usable anyway. Design with that in mind:

Check the bare HTML

Try loading your site with bare HTML - no styles, no scripts. Are menu options available? Does primary content precede secondary content? Is the site usable, even if ugly?

Consider test-driven progressive enhancement

Described in [this article](http://www.alistapart.com/articles/testdriven/), this technique uses javascript to check if a browser has a given capability, such as support for a given CSS property, before using it on the page. It is unlike browser sniffing because it tests for features rather than a specific browser.

**The most important CSS principles and tips**:

### Understand the CSS Box Model

### Understand the Difference Between Block and Inline

### Understand Floating and Clearing

### Use Internet Explorer First (Or Don’t)

### Understand Internet Explorer’s Most Common Problems

### Some Things Will Never Look the Same