

Selector specificity

So far, we've looked at the priorities given to various sources of style information and methods for attaching style to markup. Once the set of applicable style rules has been chosen, there may still be conflicts. For this reason, the cascade continues at the rule level.

In the following example, there are two rules that reference the strong element.

```
strong {color: red;}  
h1 strong {color: blue;}
```

The user agent assigns different levels of weight to the various selector types. The more specific the selector, the more weight it is given to override conflicting declarations. In the previous example, all the strong text in the document will render in red. However, if the strong text appears within a first-level heading, it will be blue instead, because an element in a particular context is more specific and carries more weight than the element alone.

The following is a list of selector types in order by weight from least to most. The selector types and terminology are explained in Chapter 17.

- Individual element and pseudoelement selectors (e.g., `p`, or `:first-letter`)
- Contextual selectors (e.g., `h1 strong`)
- Class selectors (e.g., `p.special`)
- ID selectors (e.g., `p#intro`)

Keep in mind that any rule marked `!important` will override conflicting rules regardless of specificity or order.

Rule order

Finally, once styles have been sorted by author, attachment method, and specificity, there may still be conflicts within a single style sheet source. When a style sheet contains several conflicting rules of identical weight, whichever one comes last has the most weight and overrides the others in the list. For instance, in the following example, all of the first-level headings in the document would be red, because the last rule wins.

```
h1 {color: green;}  
h1 {color: blue;}  
h1 {color: red;}
```

This “last-one-listed wins” scenario was mentioned earlier in relation to multiple link elements and `@import` commands. It also applies within a single declaration block. In the following example, the first declaration makes the border on all sides of a `div` gray using the shorthand `border-color` property. The second declaration conflicts with the first by specifying that the top border should be black. Because
