
For the More Curious: Specificity! When Selectors Collide...

You have already seen how you can override styles. You included the link for `normalize.css` before the one for `styles.css`, for example. This made the browser use `normalize.css`'s styles as a baseline, with your styles taking precedence over the baseline styles.

This is the first basic concept of how the browser chooses which styles to apply to the elements on the page, known to front-end developers as *recency*: As the browser processes CSS rules, they can override rules that were processed earlier. You can control the order in which the browser processes CSS by changing the order of the `<link>` tags.

This is simple enough when the rules have the same selector (for example, if your CSS and `normalize.css` were to declare a different `margin` for the `body` element). In this case, the browser chooses the more recent declaration. But what about elements that are matched by more than one selector?

Say you had these two rules in your Ottergram CSS:

```
.thumbnail-item {
  background: blue;
  }

li {
  background: red;
}
```

Both of these match your `<li>` elements. What background color will your `<li>` elements have? Even though the `li { background: red; }` rule is more recent, `.thumbnail-item { background: blue; }` will be used. Why? Because it uses a class selector, which is more specific (i.e., assigned a higher specificity value) than the element selector.

Class selectors and attribute selectors have the same degree of specificity, and both have a higher specificity than element selectors. The highest degree of specificity goes to *ID selectors*, which you have not seen yet. If you give an element an `id` attribute, you can write an ID selector that is more specific than any other selector.

ID attributes look like other attributes. For example:

```
<li class="thumbnail-item" id="barry-otter">
```

To use the ID in a selector, you prefix it with `#`:

```
.thumbnail-item {
  background: blue;
}

#barry-otter {
  background: green;
}

li {
  background: red;
}
```

In this example, the `<li>` is matched by all three selectors, but it will have a green background because the ID selector has the highest specificity. The order of your rulesets makes no difference here, because each has a different specificity.

One note about using ID selectors: It is best to avoid them. ID values must be unique in the document, so you cannot use the `id="barry-otter"` attribute for any other element in your document. Even though ID selectors have the highest specificity, their associated styles cannot be reused, making them a maintenance "worst practice."

To learn more about specificity, go to the MDN page `developer.mozilla.org/en-US/docs/Web/CSS/Specificity`.

The Specificity Calculator at `specificity.keegan.st` is a great tool for comparing the specificty of different selectors. Check it out to get a more precise understanding of how specificity is computed.