

18: Selector order of precedence – Specificity

Selectors are not created equal

Not all selectors are created equal, as some have precedence (or higher *specificity*) over others. This means that if two rulesets with conflicting values apply to the same element, the one with the higher precedence wins. Of course, that sort of conflict can happen by mistake, but it's also used deliberately to narrow a selection, as it allows you to override a ruleset in a subset of cases.

Essentially, the pecking order is as follows:

- IDs
- Classes
- Other directly applied values, incl. element type
- Inherited values
- Universal selector

You can see that IDs trump classes, so if you give a paragraph both a class and an ID, then the values associated with the ID will overrule any that conflict with the class. And a rule that applies to an element of a certain class has precedence over a rule that applies to all elements of that type.

What you see here is greatly simplified, but follow the above for now and you can't go too far wrong. I suggest you refer to the wonderful Smashing magazine's article about CSS 'specificity' when you are ready to find out the full details.

Selectors of equal precedence

But what about selectors of equal precedence? Well, in this case the later entry 'wins'.

This is worth keeping in mind if you find yourself wondering why your CSS changes don't have any effect. When your CSS file becomes lengthy you may inadvertently enter a duplicate or conflicting ruleset. Changes you make to the earlier ruleset can be overruled by any conflicting rules in a later one.

Example

'All paragraph' styling

In the example below we have specified some styling for all paragraphs. There are settings for background colour and underlining (text-decoration). Remember that 'p' selects all paragraphs – it is an ['element type'](#) selector.

HTML

```
<p>
All paragraphs have a pink background.
</p>
```

CSS

```
p {
background-color: pink;
text-decoration: underline;
}
```

Result

All paragraphs have a pink background.

A class overrides the 'all paragraph' styling

But see what happens if we create a class, 'yellowbackground', that has a yellow background specified (background-color: yellow;). Classes have precedence over element selectors, so this overrides the pink specified in the above CSS. However, this is the only change, so the underlining is unaffected.

HTML

```
<p class="yellowbackground">
All paragraphs have a pink background.
</p>
```

CSS

```
p {
background-color: pink;
text-decoration: underline;
}

.yellowbackground {
background-color: yellow;
}
```

Result

All paragraphs have a yellow background.