COMP 1017

Day 13

Cascade Specificity Inheritance

CSS Battle Royale

When CSS rules collide,
who remains?

What’s the first rule of our CSS Battle Royale?

The cascade determines the winner.

This means that when there are two conflicting rules, the rule written last will be rendered last.

The last rule written is the last rule standing.

What’s the second rule of our CSS Battle Royale?

The specificity of a selector determines the winner.

Let’s take a quick look at how specificity is calculated.

Selector & Example	Calculated Weight
Element Selector `p, h1, h2 …`	1
Class Selector `body.container-960px`	10
Descendant Selector `ul li a` (Any anchor tag inside of a list item in an unordered list.)	2
Descendant Combinator `ul.my-things li` (Any list item inside of an unordered list with the class ‘my-things’.)	11
ID Selector `#jumbotron`	100
!important `p { color: red !important; }`	101

The specificity of a selector can override the order of the cascade.

Introducing … rule number three!

In CSS we have a special piece of syntax that we use to make sure a certain declaration will win over all others.

It looks like this:

```
p {color:red !important;}
```

!important is the heaviest hitter of them all. It’s even stronger than IDs.

In general, we do not recommend using `!important*`.

* or, to only use it *very sparingly*

But, if you must …

Always look for a way to use specificity before even considering `!important`.

Only use `!important` on page-specific CSS that overrides foreign CSS.

Never use `!important` on site-wide CSS.

Inheritance

This just might have something to do with parent-child elements …

Inheritance is one of the last pieces we need* to understand what style is applied to which element.

* well, for now …

Inheritance means that some property values applied to an element will also be applied to its children.

So, if you make the entire <body> use comic sans, then everything inside the <body> will look like poorly-kerned garbage.

For more on inheritance, check out this supplemental reading:

MDN: Cascade & Inheritance

For more on how to calculate specificity values, go see:

css-tricks.com/specifics-on-css-specificity/