**Innovation Fridays**

**CSS Cascade Layers**

This is still an experimental CSS feature that has recently reached some production browsers. Currently compatible in Chrome > v98 and FireFox > v96 while being actively worked on by the [CSS working group](https://www.w3.org/Style/CSS/)

**Goal**

Reduce the specificity complexity of CSS in a large code base that may contain any number of –

base stylesheets, style resets, styles from third party libraries, component specific styles, custom user authored styles

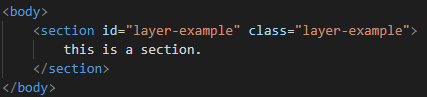
**Actions**

The @layer rule allows authors to explicitly layer their styles in the cascade, before specificity and order of appearance are considered.

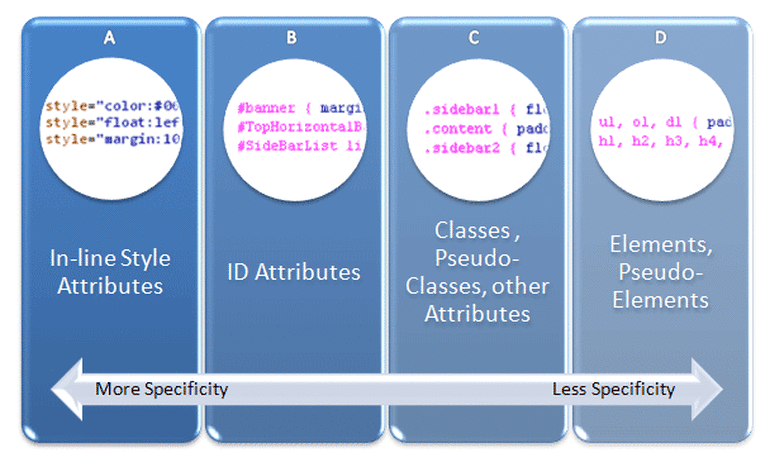
When using Cascade Layers, we don’t need to worry about the Selector Specificity of the CSS that is used in other Layers, nor about the order in which we load CSS into these Layers — **something that will come in very handy for larger teams or when loading in third-party CSS.**

EXAMPLE:

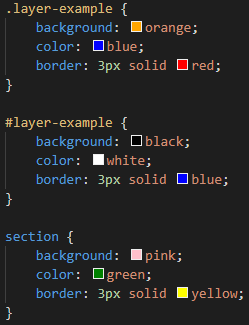
Consider the following markup:



Based on CSS specificity rules, any styles applied to the ID **#layer-example** would override styles on the class and the section element itself. We have no inline styles.



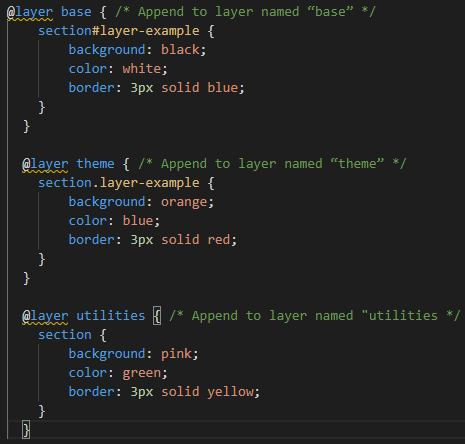
With the following stylesheet, only the styles for the ID **#layer-example** will be applied, regardless of order. Everything else in a lower specificity is overridden.



**Results**

Our section looks like this:

Now, Let’s move the styles into cascade layers:



The true power of Cascade Layers comes from its unique position in the Cascade:

before Selector Specificity and Order Of Appearance.

The section element itself now becomes the most powerful CSS selector, as it is in the layer defined last, **@layer utilities**.

**Results**

Despite the higher specificity of the ID and the class, our section now looks like this:



A “higher” layer can literally beat a traditionally stronger selector even with a weaker selector in the layer.

As mentioned above, this will be extremely useful for teams working within the same codebase. Team authored styles can be layered higher than resets, theming, components, etc. without having override problems due to unknown specificity.

**Next Steps**

The **@layer** [CSS](https://developer.mozilla.org/en-US/docs/Web/CSS) [at-rule](https://developer.mozilla.org/en-US/docs/Web/CSS/At-rule) is not currently supported by all modern browsers and can not yet be used in production. This feature should be kept in mind for a future enhancement to the organization of our CSS.

<https://developer.mozilla.org/en-US/docs/Web/CSS/@layer>