**Don’t use IDs in selectors**

Absolutely the most wrong advice I have ever heard in my life with regard to how to author CSS. This rule comes from the author’s absolute raging hard-on for “Object Oriented CSS” (hard-on is a bad term given Nicole is a woman, and “Object Oriented CSS” is a bad term given the likely confusion with actual OO coding). OO-CSS, used the right way, is great. But to suggest that you never use ID’s is simply throwing the baby out along with the dirty bath water. ID’s are extremely useful and you absolutely should use them. They are the fastest way a browser can select a given element. They are useful for in-page anchoring and if they’re already there in the markup then use them as style hooks. They are also, oddly enough, perfectly correct to use as long as they’re only ever one instance of that ID on any given page.

Arguments about “selector specificity” are at best a stretch to justify a bad rule. Author your CSS properly and you won’t fall into any specificity traps. I can count on my hand the number of times that’s been a problem for me in 6 years of coding CSS and HTML for a living on various different types and scales of website.

The only time this rule could be called accurate is if you are using ID’s to style a specific chunk of HTML with intent to allow its use anywhere on the site. e.g., if you’re using it to style a HTML gallery widget that a user can include on any page via their CMS. They may specify more than one on a page. But, that’s the one and only time it’s a concern.

**Honorable Mention**  
It’s also worth mentioning, that I am not advocating getting rid of IDs from your markup alltogether. IDs can help speed up your javascript and they can relay meaning to your document.. all good things.

But in for CSS, I suggest not using id selectors. There are others that believe using id selectors do have purpose, however I dont find the good outweighs the bad. In the end, the choice is yours to make.

This wasn’t a hard answer:

1. The element is not re-usable on that page.
2. This is the begining of a downward sprial into specificity
3. Usually, IDs refer to something very specific, and abstracting would be tough
4. Any performance gains picked up by using id, is negated by adding any other selector to the left fo that id