The Difference Between ID and Class

css-tricks.com/the-difference-between-id-and-class/

Published July 9, 2008 by Chris Coyier

ID's and Classes are "hooks"

We need ways to describe content in an HTML/XHTML document. The basic elements like <h1>, <p> and <ul> will often do the job, but our basic set of tags doesn't cover every possible type of page element or layout choice. For this we need ID's and Classes. For example <ul id="nav">, this will give us the chance to target this unordered list specifically, so that we may manipulate it uniquely to other unordered lists on our page. Or we might have a section on our page that has no relevant tag to signify it, for example a footer, where we might do something like this: <div id="footer">. Or perhaps we have boxes in our sidebar for keeping content over there separated in some way: <div class="sidebar-box">.

These ID's and Classes the "hooks" we need to build into markup to get our hands on them. CSS obviously needs these so that we may build selectors and do our styling, but other web languages like Javascript depend on them too. But what is the difference between them?

ID's are unique

- Each element can have only one ID
- Each page can have only one element with that ID

When I was first learning this stuff, I heard over and over that you should only use ID's once, but you can use classes over and over. It basically went in one ear and out the other because it sounded more like a good "rule of thumb" to me rather than something extremely important. If you are purely an HTML/CSS person, this attitude can persist because to you, they really don't seem to do anything different.

Here is one: your code will not pass validation if you use the same ID on more than one element. Validation should be important to all of us, so that alone is a big one. We'll go over more reasons for uniqueness as we go on.

Classes are NOT unique

- You can use the same class on multiple elements.
- You can use multiple classes on the same element.

Any styling information that needs to be applied to multiple objects on a page should be done with a class. Take for example a page with multiple "widgets":

```
<div class="widget"></div>
<div class="widget"></div>
<div class="widget"></div>
```

You can now use the class name "widget" as your hook to apply the same set of styling to each one of these. But what if you need one of them to be bigger than other other, but still share all the other attributes. Classes has you covered there, as you can apply more than one class:

```
<div class="widget"></div>
<div class="widget big"></div>
<div class="widget"></div>
```

No need to make a brand new class name here, just apply a new class right in the class attribute. These classes are space delimited and most browsers support any number of them (actually, it's more like thousands, but way more than you'll ever need).

There are no browser defaults for any ID or Class

Adding a class name or ID to an element does nothing to that element by default.

This is something that snagged me as a beginner. You are working on one site and figure out that applying a particular class name fixes a problem you are having. Then you jump over to another site with the same problem and try to fix it with that same class name thinking the class name itself has some magical property to it only to find out it didn't work.

Classes and ID's don't have any styling information to them all by themselves. They require CSS to target them and apply styling.

Barcodes and Serial Numbers

Maybe a good analogy here is bar codes and serial numbers. Take an iPod in a store. On the packaging will be a bar code. This tells the store what the product is, so when it is scanned, the system knows exactly what the product is and what it costs. It might even be able to know what color it is or where it was kept in the store. All iPod of this same type have the exact same bar code on them.

The iPod will also have a serial number on it which is absolutely unique to any other iPod (or any other device) in the world. The serial number doesn't know the price. It could, but for the store this wouldn't be a very efficient way to store and use that data. Much easier to use the barcode, so that for example, if the price changed, you could just change the price for that bar code and not every individual serial number in your system.

This is much like ID's and Classes. Information that is reuseable should be kept in a class and information that is totally unique should be kept in an ID.

ID's have special browser functionality

Classes have no special abilities in the browser, but ID's do have one very important trick up their sleeve. This is the "hash value" in the URL. If you have a URL like http://yourdomain.com#comments, the browser will attempt to locate the element with an ID of "comments" and will automatically scroll the page to show that element. It is important to note here that the browser will scroll whatever element it needs to in order to show that element, so if you did something special like a scrollable DIV area within your regular body, that div will be scrolled too.

This is an important reason right here why having ID's be absolutely unique is important. So your browser knows where to scroll!

Elements can have BOTH

There is nothing stopping you from having both an ID and a Class on a single element. In fact, it is often a very good idea. Take for example the default markup for a WordPress comment list item:

```
<li id="comment-27299" class="item">
```

It has a class applied to it that you may want for styling all comments on the page, but it also has a unique ID value (dynamically generated by WordPress, nicely enough). This ID value is useful for direct linking. Now I can link directly to a particular comment on a particular page easily.

CSS doesn't care

Regarding CSS, there is nothing you can do with an ID that you can't do with a Class and vise versa. I remember when I was first learning CSS and I was having a problem, sometimes I would try and troubleshoot by switching around these values. Nope. CSS doesn't care.

Javascript cares

JavaScript people are already probably more in tune with the differences between classes and ID's. JavaScript depends on there being only one page element with any particular, or else the commonly used getElementById function wouldn't be dependable. For those familiar with jQuery, you know how easy it is to add and remove classes to page elements. It is a native and built in function of jQuery. Notice how no such function exists for ID's. It is not the responsibility of JavaScript to manipulate these values, it would cause more problems than it would be worth.

If you don't need them, don't use them

As you can see, classes and ID's are very important and we rely on them every day to do the styling and page manipulation that we need to do. However, you should use them judiciously and semantically.

This means avoiding things like this:

```
<a href="http://css-tricks.com" class="link">CSS-Tricks.com</a>
```

We already know this element is a link, it's an anchor element! No particular need here to apply a class, as we can already apply styling via its tag.

Also avoid this:

```
<div id="right-col">
```

ID is appropriately used here as the page will likely only have a single right column, but the name is inappropriate. Try and describe the context of the element, not where it is or what it looks like. An ID here of "sidebar" would be more appropriate.

Microformats are just specific class names

Think microformats are over your head? They aren't! They are just regular markup that make use of standardized class names for the information they contain. Check out a standard vCard:

```
<div class="vcard">
 <a class="fn org url" href="http://www.commerce.net/">CommerceNet</a>
 <div class="adr">
 <span class="type">Work</span>:
 <div class="street-address">169 University Avenue</div>
 <span class="locality">Palo Alto</span>,
 <abbr class="region" title="California">CA</abbr>
 <span class="postal-code">94301</span>
 <div class="country-name">USA</div>
 </div>
 <div class="tel">
 <span class="type">Work</span> +1-650-289-4040
 </div>
 <div class="tel">
 <span class="type">Fax</span> +1-650-289-4041
 </div>
 <div>Email:
 <span class="email">info@commerce.net</span>
 </div>
</div>
```