sitepoint premium

Handout
A Beginners Guide to Modernizr
Lesson 2 - Step 3
Ryan Seddon

The Modernizr CSS Classes

The Modernizr CSS classes aren't necessary in all situations, and sometimes it doesn't make sense to use them. They're simply providing a series of fallbacks, right? And the CSS decoration is probably more appropriate in certain situations.

Example of Modernizr's feature and no-feature classes.

So let's take a look at an example of using Modernizr's feature and no-feature classes and where it doesn't really make sense.

```css
.generatedcontent.no-flexbox .flex:after {
  /* '/2718' is the unicode for the cross symbol */
  content: '/2718';
}
.generatedcontent li:after {
  /* '/2713' is the unicode for the tick symbol */
  content: '\2713';
}
```

You can see I use the generated content after pseudoclass that allows you to add content via CSS. In my case I'm adding a cross or a t based on support. Modernizr has a feature detect for this, and you can see that I'm using the `generatedcontent` class here. Under this scenario, if the browser didn't support them, the fact that the text is green or red based on support is already a pretty good indication and a good enough fallback for all the browsers. So it probably isn't necessary to use that class.

In situations like this, it's recommended that you don't complicate the CSS with Modernizr feature test classes. By doing that you create complexity with your selectors where it isn't necessary, and you don't really need to do that for each new tag.

```css
.no-flexbox .flex:after {
  /* '/2718' is the unicode for the cross symbol */
  content: '/2718';
}
.flex:after {
  /* '/2713' is the unicode for the tick symbol */
  content: '\2713';
}
```