Handout

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

Introduction

We've played around with classes, and we know that Modernizr has both a class for supported features and a class for when it's not supported, which is a feature name prefixed by the `no-`. A good question would be "when should I use which?". My short answer is "it depends".

So, let's dive into the two examples of using the positive case and the negative case and we'll show you which has it's advantages and disadvantages.

Positive Class

We'll start with the **positive class** in Modernizr. With this we write our CSS as if no browser supports any features and we only add it once Modernizr has added the class to say it is supported. We write for the lowest common denominator, and only enhance through Modernizr positive feature classes.

We have an unordered list of features, and we want to use the positive class to employ flexbox.

```
.flexbox ul {
  display: flex;
}
```

Next up, we'll set a minimum width of the list items.

```
.flexbox li {
  flex-basis: 150px;
}
```

`flex-basis` is essentially a minimum width for flexbox items.

The next step is to make the flexbox container wrap over multiple lines. Early versions of Firefox didn't support wrapping flexbox items. So Modernizr has another feature detect, that checks for support of that specific flexbox property:

```
.flexwrap ul {
  flex-wrap: wrap;
}
```

If I was to recommend which one to use of the two, I would go for this one. The reason being we can make assumptions in CSS that all our users are in pretty decent evergreen browsers. When they're not supported is when we can leverage the Modernizr classes to give those users an experience that's still useable but we can fold back and we can know what features they support .For example, if we use flexbox, we can fall back using the `no-flexbox` class andmake it float, for example.

Negative Class

Now let's take on a **negative case** here we can target browsers that don't support flexbox and instead we can remove the positive class name styling and make the assumption that most browsers will have support.

```css
.no-flexbox li {
  float: left;
  margin: 0 20px;
}

.no-flexbox .flex {
  color: red;
}
.no-flexbox .flex:after {
  /* '/2718' is the unicode for the cross symbol */
  content: '/2718';
}
.no-peerconnection .webrtc {
  color: red;
}
.no-peerconnection .webrtc:after {
  content: '\2718';
}
.no-webp .webp {
  color: red;
}
.no-webp .webp:after {
  content: '\2718';
}
.no-serviceworker .serviceworkers {
  color: red;
}
.no-serviceworker .serviceworkers:after {
  content: '\2718';
}
```

We've just learned two different ways of approaching how we can use Modernizr classes in CSS.