sitepoint premium

Handout

A Beginners Guide to Modernizr
Lesson 3 - Step 5
Ryan Seddon

Introduction

At this step we're going to cover two similar methods: `Modernizr.prefixed` and `prefixedCSS`.

`Modernizr.prefixed` is a convenience method that takes the pain out of managing browsers that require a prefix for both CSS and DOM properties. `Modernizr.prefixedCSS` is only concerned about CSS properties, and allows you to specify them in a normal CSS style. For example, to set border radious, you'd use only `border-radius`. Modern browsers will have most of their API's unprefixed, but in reality our users will be on lots of different versions, and unfortunately prefixes are a necessary evil to deal with. Lucky for you, Modernizr's got your back.

Modernizr.prefixed

For testing a CSS property, you pass in the camel case version of the property. For example, `border-radius` will have to be written as `borderRadius`. If you pass an unsupported property, it will return `false`.

You may open Chrome's console and pass `marginStart` to `Modernizr.prefixed` - `WebkitMarginStart` will be returned as a result. You may not have come across margin start, and it's other property, `margin-end`. Essentially, they map to margin left and right. In right to left websites, start and end correspond to margin right and left. It gives you a way to cater for both; when the direction is changed, your URL just works.

When testing a DOM property such as `getUserMedia`, you need to pass the unprefixed name and a second argument that is the object Modernizr will test on to see if the current browser supports it. By default, prefix assumes we're testing for a property on a DOM element, and we'll do a check against that.

gUm Variable

Instead of just calling one of those prefixes directly, we'll actually store the return value to a variable. In our case we'll call it `gUM`, get user media, for short, so we don't have to worry about figuring out if the current browser has a prefixed version. We can just reference that variable which will call the function.

```
var gUM = Modernizr.prefixed('getUserMedia', navigator);
```

Don't forget that user media expects three arguments: an object (we want to specify which parts of the camera and audio we want to access in this case), success and error callback.

```
gUM({video: true}, function success() {}, function error() {});
```

Call it, and you'll see that it'll prompt us for access to the camera.

If you purely wish to just return the method name, you can pass a third argument to `Modernizr.prefixed`, and tell it not to return a function. Essentially, the third argument will be `false` to return only the string name if it's prefixed or not:

```
var gUM = Modernizr.prefixed('getUserMedia', navigator, false);
```

prefixedCSS

Let's take that margin example again, but use it with `Modernizr.prefixedCSS`. This time we'll write it as margin hyphen style like you would in CSS.

```
$('#prefixedCSS').on('click', function() {

  $('#prefixedCSSResult').text(
    Modernizr.prefixedCSS(
      $('#prefixedCSSValue').val()
    )
  );
```

As you can see, this returns a CSS prefixed property in a normal CSS convention: all lower case, separated by hyphens.