**CSS & jQuery**

At this point, you likely know that jQuery can link user events to dynamic effects on a web page. One of the most powerful toolsets in jQuery allows you to edit CSS property values. With these tools, you can change multiple visual aspects of an element at once.

In this lesson, you'll increase the breadth of effects you can produce on your web page by editing CSS property values and adding classes to elements.

You will also learn how to change CSS properties over time to create animations.

# .css()

To modify CSS properties of an element, jQuery provides a method called .css(). This method accepts an argument for a CSS property name, and a corresponding CSS value.

It's syntax looks like:

$('.example-class').css('color', '#FFFFFF');

Let's break the example above into two pieces:

* We call the .css() method on .example-class. The first argument is the CSS property we want to change. In this case, that's 'color'.
* The second argument specifies the new value for the given property in the first argument. In this case, the .css() method changes the color of .example-classelements to '#FFFFFF'.

Notice, both of the inputs to the .css() method are strings.

**CSS II**

In addition to changing one property at a time, the .css() method can accept many CSS propery/value pairs at once. You must pass the .css() method a JavaScript object with a list of key/value pairs like this:

{ color: '#FFFFFF', backgroundColor: '#000000', fontSize: '20px' }

There are a few important differences between the key/value pairs in the object above and the arguments from last exercise. Let's consider these differences below:

* The object is wrapped by an opening and closing curly brace: {}.
* Inside the object, there are three key/value pairs. The keys in the example object are color, backgroundColor, and fontSize.
* The values come after the colon : of each key. For instance, fontSize is a key, and its value is '20px'.
* When referencing CSS properties in an object, the property names are *camelCased*— they are modified to have no quotes or spaces, and to start each new word with a capital letter. Therefore, background-colorbecomes backgroundColor, and 'font-size'becomes fontSize.

To set multiple properties at once, you can pass the whole object into the .css() method as a single element.

$('.example-class').css({ color: '#FFFFFF', backgroundColor: '#000000', fontSize: '20px' })

In the example above, we use the .css()method to change the color, background color, and font size values of the .example-classelement.

# .animate()

The jQuery .animate() method enhances the visual possibilities by making CSS value changes over a period of time.

The first argument passed to .animate() is a JavaScript object of CSS property/value pairs that represent an element's end state. This is identical to the .css() method. For example, the following object could be passed to the .animate() method to change an element's height, width, and border thickness

{ height: '100px', width : '100px', borderWidth : '10px' }

Note that the names of CSS properties in the JavaScript object don't have spaces or dashes and are camelCased.

The second parameter of the .animate()method determines how long the animation takes to complete. It is optional; if you do not provide an argument, the default value is 400 milliseconds. You can use a number (in milliseconds) or the strings 'fast' or 'slow'. Below is an example of the .animate() method:

$('.example-class').animate({ height: '100px', width: '100px', borderWidth: '10px' }, 500);

In the example above, the height, width, and border width will change to 100px, 100px, and 10px respectively over a 500 millisecond period.

**.addClass()**

A JavaScript file can quickly get overloaded with styles if you regularly use the css method to modify element styles. It's a best practice to group all of the style code in a CSS file, and use jQuery to add and remove the classes from elements — this approach aligns to a design principle called *separation of concerns*.

Separation of concerns is a design principle stating that code should be separated based on its purpose in a program. In web development, that generally means the structure of a page is defined in an HTML document, styles are stored in a CSS file, and code that defines dynamic behavior is stored in a JavaScript file.

To keep CSS properties in a CSS file, jQuery can add a CSS class to an element with a method named addClass. It's syntax looks like this:

$('.example-class').addClass('active');

In the example above:

* .addClass() is called on the jquery .example-class selector.
* .addClass() adds the 'active' class to all .example-class elements.
* Notice that the argument passed to addClass does not have a period preceding it. This is because it expects a class, and therefore only needs the name of the class.

**.toggleClass()**

Similar to other effects methods, you can use a toggle method instead of chaining the .addClass() and .removeClass() methods together.

The .toggleClass() method adds a class if an element does not already have it, and removes it if an element does already have it. Its syntax looks like:

$('.example-class').toggleClass('active');

In the example above:

* .toggleClass() is called on .example-classelements.
* .toggleClass() will add the 'active' class to .example-class elements if they do not have it already.
* .toggleClass() will remove the 'active'class from .example-class elements if they do have it already.

**Review: Style Methods**

jQuery can be a wizard at managing your CSS, but as with any powerful magic, it must be handled with care and consideration or you'll end up with a giant mess!

There may be instances where modifying specific CSS properties with jQuery makes sense, but as a best practice, it's best to maintain an organized CSS file and use jQuery to manipulate clearly defined and well-named CSS classes.

In this lesson, you learned:

* The .css() method can change style properties of an element.
* The .css() method can accept multiple styles at once if you pass it a JavaScript object as its argument.
* The .animate() method can change specific style properties over a period of time.
* The .addClass() will add a CSS class to an element, and the .removeClass() method will remove a CSS class.
* The .toggleClass() method will toggle a class on or off an element.

Incredible work! In the next lesson, you'll learn about logical selectors. This will enable to you add a click event to one element, and make something happen elsewhere on the page.