

Introduction to animating CSS properties

So far, you've seen how an element's visibility can be animated. But visibility is not the only thing that you can animate. We are going to talk about CSS properties that are numeric in nature, such as `top`, `bottom`, `left`, `right`, `width`, `height`, and so on. Also we will see how to animate properties with alphanumeric values. But first, let's take a look at jQuery's `animate` method.

Introduction to jQuery animate:

In its basic usage, you can pass in two arguments to the `animate` method. The first is an object containing key-value pairs representing the CSS properties and their values, which should be animated. The second argument is the duration within which these values should be reached.

Working with jQuery animate demo

Open the `animationBasics_Begin.htm` file before proceeding.

We have an image of a car, which we want to drive forward to the end of the road when the user presses the "Drive" button. Alternatively, pressing the "Push" button should push the car by a few units at a time. Finally, pressing the "Reverse" should back up the car to the starting point.

The first thing we'll do is drive the car forward to the end of the road marked by the dashed line:

```
$('#driveCar').click(function() {  
  $('#car').animate({  
    left: '580'  
  },1000);  
});
```

We animate the `left` property of the car image element all the way from its current position which is zero pixels to 580 pixels, a value that I've calculated as the end of the visible road. Animation's duration is set to 1000 milliseconds (1 second).

Now let's code the "Reverse" button:

```
$('#reverseCar').click(function() {  
  $('#car').animate({  
    left: '0'  
  },3000);  
});
```

The "Push" button:

```
$('#pushCar').click(function() {  
  $('#car').animate({  
    left: '+=50'  
  },400);  
});
```

Instead of typing in a fixed value for the `left` property we use `+=50`. This will move the car 50 pixels at a time incrementally.

Using jQuery animate with options

The `animate` method also accepts a more advanced set of arguments. By providing an object as the second argument, we can access more properties and features such as the `step` and the `complete` function. Let's see how we can use these options to animate CSS3 `transform` property which is non-numeric.

Open *animatingTransform_Begin.htm* from the accompanying files. In this example, we have a `div` with an another `div` that has red background; inside there is an `h1` tag. We also have a button that will trigger the animation.

Now examine CSS inside the *animatingTransform.css* file. The primary `div` has its `perspective` attribute set to `200`. The `perspective` attribute is necessary for treating this `div` as a **viewport** so that the elements inside it appear to transform in 3D.

Code the click handler for the button:

```
$('#play').on('click', function() {
    $('.panel').animate({
        'text-indent':1
    },{
        duration:4000,
        step:function(now) {
            $(this).css('transform','scale3d(' + now + ',' + now + ',' + now + ') rotateY(' + (now * 360) + 'deg)');
        }
    });
});
```

Since we are trying to animate a property which has an alphanumeric value, it is not directly supported by the `animate` method. So we animate any CSS property on a range of zero to one first. Ideally that shouldn't visibly effect the `div`. Inside the `step` function we will do the 3D transform. The `now` argument contains the current animation step value (from zero to one). With `scale3d` we animate the scale of the panel in the X, Y, and Z axis from 0 to 100 percent. With `rotateY` the panel is being rotated in the Y axis, as it is scaled up from 0 to 100 percent.

Refresh page in the browser and observe the result.

The animation works, but it only works once - if you press the "Play" button again, nothing happens. The reason is that the `text-indent` property is animated from zero to one, and animation won't happen for a property that already has the desired value. So let's append a `complete` event handler function which sets the value of `text-indent` back to zero when the animation ends:

```
$('#play').on('click', function() {
    $('.panel').animate({
        'text-indent':1
    },{
        duration:4000,
        step:function(now) {
            $(this).css('transform','scale3d(' + now + ',' + now + ',' + now + ') rotateY(' + (now * 360) + 'deg)');
        },
        complete:function() {
            $(this).css('text-indent', 0);
        }
    });
});
```