jQuery Animations

Hello and welcome to the lesson on jQuery Animations. The animate method is one of the most powerful capabilities of jQuery. It allows you to create simple or complex animations and everything in between. It's probably only your experience and your imagination that will limit you and you can add all sorts of different parameters and functions to it, which makes it very, very powerful. But even in its simplest form this method can be very useful for us. When we'd like to add some movement to our otherwise static websites. In this video, I will walk you through the basic syntax of the animate method in jQuery.

HTML Sample Page

So let's get started. This is my slightly different HTML sample page, I've added a div with a class box and also 4 buttons each with an ID and you know, this is the HTML entity for an arrow. So I have four arrows for each of the directions. And now I will use these arrows to animate that box. So, because we could animate it on page load, but I just want to show you how the animation looks and it's just easier if we can trigger it.

Syntax

So inside our scripts.js, if I want to animate the box, I have to select it. So, class box and then use animate methods. Inside that method, inside first curly braces, we can specify different CSS properties that we want to animate, and will do that in a moment. And next there is a comma and we can specify the duration and how long the animation should take to complete. So let's say one second and then at the end, we can specify another function. Which will take place once the animation is complete. Now, this is the basic animate method syntax but I would like to use that inside a click event. So let me add a click event attached to the left arrow. So I have to select my left arrow. Click. And now, inside the click event, we need a function. And our animate method can go inside here. So let me just make it look better. And now, as I said, we can use different CSS properties and change their values. So if we want to move it to the left, we can say: left and then colon and inside quotes - single or double.

Movement Animation

We need to specify the value. So, if I say 40 pixels, that would move it to the right, it's like movement on the horizontal x-axis. So if you want to move it left, it has to be a negative value and that is basically it. Now we have to be mindful because these properties do not work exactly the same like in CSS, for example, if we want to animate border left, we would say, border left with camel case, or if we want to use the CSS syntax, we would specify border left, but in quotes. So I will use the camel case JavaScript syntax to specify different CSS parameters, you can use either of these but just note that if you're not using quotes, properties like border left, margin top, will use camel case. Like for the left, it's not an issue. It's just it's the same, it's simply left. If I save that, let's refresh my page and now when I click the right arrow, it goes to the left but it only does that once, okay. Because it changed its CSS property to negative 40 pixels and that's it. So if we do want to keep animating it every time I click the arrow, there is a nice trick or just a method we can use to specify instead of defining the exact value that CSS property should have after the animation. We can use the equal sign here so, minus and then equals and that will use the negative 40 pixels from its current position. Okay, so if I save that - it's the negative 40 pixels. And now from that position it uses another negative 40 pixels. We can make it work a bit faster if we decrease that time here. So 300 milliseconds, and let's see. Each time I click the box animates from its current position to a negative 40 pixels.

Multiple Animations - Font Size

So let's try adding another thing because that's what makes the animation method powerful, that you can animate different properties at once, it's not like fade in, that is only fades in, but you can do different things with that element. Let's try changing the font size. So, font size, colon and then in quotes, we can decrease it, similar way. So, minus and equals 3 pixels. So each time we click on the left arrow, the box will move to the left and also the font size of that element will be smaller by 3 pixels. So let's save that and see how that looks. Great. So now let's try doing the same for other arrows. This function, which will run once the animation is complete. It's just something you can do in addition, you don't need to use that. You can basically only use that part, the animate method, and then it would use the default timing and no additional callback function. So let's try creating something simple for the right arrow. So I will copy that. And now on for the right arrow, will animate the box without any additional functions. We can leave the 300 to make it quick. So now looking from the left, we have to use the positive value and if we want to make it larger than its current position, each time we'll use the plus equals and we can do the same for the font size. So now if I save that and refresh the font grows and then it gets smaller.

Animating objects with opacity

Let’s maybe now try doing the up and down buttons. When we click the up arrow, the top position has to have a negative value, if you want to move it up and then we can also change the opacity. So opacity, and each time it goes to the top, the opacity can increase by 0.1. Let's save that and try it. Let's move it only by 20 pixels - you can see that the opacity gets closer to no transparency. So closer to one opacity the box started with 50% opacity so when we refresh that's 50 percent now it's 60 70, 80 90 and that would be 100% opacity. And now for the down arrow, top and positive value, and we can decrease the opacity, simple as that. As you can see, the animate method can be very simple and you can animate any - almost any CSS property. What can't be animated is colors because jQuery cannot handle color animation on its own. You would need to use a separate plug-in but anything else your borders, fonts, all that, it can be changed, the position the opacity and and all that can be changed inside that method.

So just remember to use camel case or the standard CSS property in quotes and you can specify as many as you like. Next use the duration and then if you want to do another thing, when the animation ends, you can use the callback function as well. So hopefully that will give you a starting point to writing your own animations and animate any elements on your site, easily.