Inserting, Replacing and Removing Elements

Hello and welcome to the lesson about inserting, replacing and removing elements from your websites using jQuery. In this video, I would like to show you how you can control the DOM and change whatever you want inside your website structure. So we're going to start by adding new elements onto the page and there are a few different ways to achieve this in jQuery. We are going to look at a few different functions and I'm going to explain what the differences are.

But first, a quick look at our HTML structure, I have three divs with a class box and also the class that specifies the color of the box. I have three paragraphs and an unordered list here. Very simple structure

Append Function

Now, inside our scripts.js. Let's start with a new selector. So let's select a second UL. So the second level list and we'll use the append function. Let me show you an example inside quotes. I can insert HTML here solely, just like that. Let's see how that looks. So you can see a new list item right here. So what append function does, you can add new elements. This would be the new element you're adding. As a child of your selected that's in front here element . So we selected a second level UL and inside that element as a child but as a last child we are appending this li.

Append to Function

And there is another very similar function which also achieves the same exact effect but has a quite different syntax - The append to method and this work the other way around. So let me just comment this out and now we can select our new HTML. o li, this is last and append this to - so append to our element. If I save and refresh. Not a li but UL, this code does exactly the same thing. It just works the other way around. The thing you want to append is at the front here and you can append it to an element here and you can use it like this as a selector, but you can also just simply define a selector like that. Okay, that would work the same way. And similar to append we can also go ahead and call a function called prepend. And that's going to add a new element as the first child within our selector. So let's copy that.

Prepend Function

And now instead of append, we can use prepend and this would be first. Let's have a look. You see it adds - that's our selector that UL right here, the second level UL, and it added a child element as the first child because our UL already has three children. One, two, three. And it added this part as a first child. So that's what prepend does. And we also have prepend to which works the same. Well, the opposite way as the append to, but it uses the same logic and syntax.

Prepend to Function

So instead of this code here, we could use this and use prepend to and - this is first. If I refresh nothing will change because this and this does exactly the same. It's just different logic. So whichever of this makes more sense to you whether you want to find an element and then prepend a first child to it or define what you want to prepend and then prepend to an element. Whichever makes more sense to you. This one you can use, the end result will be the same. All right. So now, you know, four functions you can use to add a new element as a child of another element.

Insert Function

The next two functions I want to show you. I'm not going to add a new element as a child, but as a sibling and there are two different functions we can use one is gonna insert a new element as a next sibling. And one is going to insert a new element as the previous sibling, before, and after. So let's try inserting an element before our - one of the boxes. Let's select our green box. Before. And just like we did with our lis, I can define a div with a class of green box and box because that will give it the styling we need. Second green, that would be the content of my div and closing tag. And my quote. Okay, so before the green box class element with a class, green box, we are inserting that div, that HTML. So let's save that. Okay, so you see, now if we make this smaller, we have a second green, which we've added with jQuery, and we can do the same with after. Let's try the second blue after the green, so box, and then blue box. Now, the document sees two green boxes because we have our first green box and then the second green and each both of them it added the new second blue. If I would comment this part and only leave our main green box - the first one, it will add a second one after that. Now, in all these methods, instead of passing in a string with the HTML code, you can also pass an element. This is super powerful and you can use it to move your content around. So, basically, that's my structure. Okay, once again I have orange, blue and green. And now, after my green box, I can insert instead of this div here I can insert a selector - my orange box. Just like that. And now if I refresh the page, this doesn't insert a new orange box, but it moves the content from its original position to after that element after the green box.

Target Function

But if instead of green box we would target - let's target paragraphs. Okay. And now, let's see this moves the element after the first paragraph, but we have more paragraphs and then if our selector, if we specified something that targets more elements that will clone that thing, okay? So we have one orange box, and we will place it after each P. It is after each P. So, we have the original one here and jQuery cloned it for us 2 times. So, as you can see, it's very, very powerful and basically allows you to move any content around. Okay? So now, you know how to add new elements, on the page, how to move elements. So now let me show you how to replace existing elements on the page, with either another element that's already on the page or with a completely new element that's created just for the purpose of replacing the other element. Now, there are two functions in jQuery that provide this functionality for you. First, let me comment this out. Let's save that, refresh that.

Replaced With Function

So, that's the original structure, the very first function we can use is called replaced with. So for example, I can select my UL. And then use replace with and replace it with H1 - Hello. Just like that. Okay, let's save that. And now I no longer have my UL here and I have my H1 tag and you can pass a new string of HTML, but you can also pass an existing element. So, let's say I want to replace my UL with my blue box, okay? So let's add this here, instead of that HTML. And that works exactly the same like for the before, after and prepend it will move the element, okay? It doesn't create a new element. It moves it instead of moving it before or after or inside as a child we can replace our selector with a new thing. And you can see that again, the item here, the existing element that we have now used to replace the UL is actually moved to the new position. So this is also consistent with how the append and prepend and before and after functions behave. But once you go ahead and have multiple elements here like P in our case, this will also clone that and replace that element. We have replace with and we have a function called replace all which works kind of like append to and prepend to. It's the same thing, but let's do the UL. It's the same thing, but just a different syntax. So instead of first defining the thing you want to be replaced, you first define what will be the new element. So let's say the H1.

Replace All Function

Okay, and we'll use replace all. And we can specify what do we want to replace. So inside, here, we can have our selector and our UL. And now, this will replace all ULs. Now, I personally find the replace with function to be more readable, but of course, you can choose whichever way you prefer and you will be able to achieve your replacements with either one of these functions. Okay, so now there is one more operation left which is the removing of an existing element. Now there are three different functions in jQuery that I want to show you for removing website elements.

Remove Function

And the most basic and simple function that you can use to remove elements is, you probably guessed it - remove. Okay, so if we target our UL again, we can simply call remove, just like that. And it will be removed from our page, that removes the parent and all its children. If I would to remove the main content, my main div, that would remove all the elements within that element. It's not going to just remove the parent wrapper and keep all the child elements, it's going to remove the whole hierarchy below that element along with the element itself.

Detach Function

Instead of remove, we can use a very similar function called detach. So for example, if I target my UL and call detach. We won't see any difference because it will look the same as a remove function. But if for example, our UL has some click event specified earlier in the code or any additional data using the detach actually keeps that data. So, for example, if I would specify a variable here, so my variable detached list and that would be that thing that we are removing/detaching from the page. I could then append this inside my content for example. So, in the main content div, append, meaning added as a direct child as the last element I could append my variable, so my detached the list. Okay, and now if I refresh my list would be as the last element, it got a bit confusing because we had ULs as sub items and then it place it all again, as one element, if I would add a class to my UL that would keep the structure, okay? If I would be more specific and detached my main UL list that would use the same thing again as it appends it. Because now we detached four different ULs and then append it all these again as the new element. So we have to be mindful of what we are detaching. But by using detach, we are still able to reuse that element and all the corresponding data, okay?

Empty Function

And the final way you can remove things from your site is the function called empty. Let's say, Target our paragraphs, and call the empty function. And what that does, it will look very similar to a remove and detach. But if we inspect the page those P tags are still here, they are just empty. If I would detach it or remove it we won't see them anymore. But empty function only empties the content of that element. So all the children of that paragraph and all the text that it contains, but the tag itself is still here. So that's some time, it may be useful, just something to know of so basically that's how the empty function works. It removes all child elements and all the text inside the element, but not the element itself.

So these were all the ways for inserting, replacing and removing elements I wanted to show you.