

jQuery for Forms

Perhaps, the most powerful form of front-end JavaScript right now is AJAX (Asynchronous JavaScript and XML). AJAX allows your page to interact with your database **without refreshing the page!** Sound cool? Yeah, that's because it is! You will learn AJAX soon. To make your transition easier, there are a couple of things you should master in jQuery regarding forms. Forms are 99% of what AJAX is used for. And jQuery has some very effective handlers and functions for forms. If you pay close attention to this section, you will have the foundations of AJAX under your belt. We are going to cover three things in this section: `.submit()`, `return false`, and `.serialize()`.

`.submit()`

We can use the jQuery command `.submit()` to either submit an actual HTML form, or we can use it as an event handler. What is meant by this is that `.submit()` has two purposes. Let's look at some sample code:

```
$('form').submit(function(){alert('you submitted me!');});
```

What this code means is that when any form element is submitted, an alert box pops up with a message. In this case, the `.submit()` function acted as an event handler; we wrote code for the **event** of a form being submitted. Now, take a look at this:

```
$( 'h4' ).click(function() {  
    $( 'form' ).submit();  
})
```

In this case, we tell the document: "whenever an *h4* tag gets clicked, submit any form." Notice this instance of `.submit()` code for the actual submission of a form, not for a handler to be attached. In order to understand this, be sure to experiment on your own using `.submit()` as both an event listener and as a method to submit a form (meaning, you actually code in the act of submitting the form).

Return false

Submitting a form, actually, redirects your browser to a different page! You know that form element...

```
<input type = 'submit'>
```

...button that you put in all of your forms? Well, that button actually tells the browser to navigate to wherever it was you told the form to go (specified by the *'action'* attribute of your `<form>` tag). What if there is a way not to allow that form to submit if you don't want it to? Like maybe, somebody forgot to fill out all of the information in your form or something. Well, you can code for that! Every time you don't want your form to be allowed to change pages (and thereby stopping all the information in your form from being submitted) create a `.submit()` **event listener** (see above) for your form and within the function part of the code, simply put: **return false**. It should look something like this:

```
$('form').submit(function() {  
    return false;  
});
```

.serialize()

This function isn't one which you will actually see the effectiveness of until you really get into the heavy use of AJAX and APIs. But once you're there, you will be glad you learned it. The function `.serialize()` is a function that encodes a set of form elements into a computer-friendly array. Why is this helpful? Because `.serialize()` transforms data that the user inputs into a format that can be easily passed to a back-end process and used by a non-human entity. To illustrate what is really happening, watch the corresponding video and type along with it. Bear in mind that your user will never interact with any form element after it has been serialized - it is mostly used for cleaning up data before it is passed to either an AJAX request or an API. That being said, it is still an extremely useful piece of code. That's why we teach it!