jQuery - Javascript's Best Friend

If you're writing javascript and haven't tried jQuery you've been eoither asleep, not listening, or just don't want to be the best of the best, ..Maverick!

It adds so much functionality that I rarely write websites without it and for once me and Microsoft agrees - they'll be shipping jQuery with Visual Studio soon.

In this example, we'll use jQuery to simplify the AJAX call we made in the previous lecture, and update the page with the server time.

All of this is possible without jQuery, but its much simpler and cleaner using it.

Once you've downloaded jQuery, add it to your page like this:

```
<script src="/js/jquery.js" type="text/javascript"></script>
```

Making An AJAX Call with jQuery

In the HTML above, we have 2 things:

* A form field that calls the JS function
* A form field that will display the returned data

When the user enters text in the first filed we're going to get the server's current time and show it in the other filed. We'll do this with a combination of Javascript and PHP, but you could replace PHP with any server-side language you want.

Making An AJAX Call with jQuery

Although the syntax may look a bit confusing, the entire ajaxFunction function is basically one function call

```
<script type="text/javascript">
function ajaxFunction()
{
     // get the current time from the server and display it to the user
     jQuery.post('time.php',
     function (resp)
     {
          // update the filed with the response
          jQuery('#time') .val(resp);
     });
}
</script>
```

Making An AJAX Call with jQuery

```
jQuery.post();
```

jQuery's post function sends a request to the URL you've specified exactly as if you've visited the URL in the browser. As such, it will return the same data that the browser would (usually HTML source code, but it doesn't have to be - more on that later). The trick is that we'll then have the data in a variable, and we can do whatever we want with it.

In our example, we're passing 2 parameters to jQuery's post function - the URL and a javascript function.

Making An AJAX Call with jQuery

The first is self explanatory - it can be any normal URL (i.e. “http://www.google.com”), or it can be relative (i.e. “/posts/1″ or “time.php”).

That’s the URL we’re asking for the data from, and its going to execute just like we typed it into the address bar.

(Note: If you want to send POST variables you can specify them as one of the parameters. More information is available in the jQuery docs, and we’ll cover it in a future lesson.)

Making An AJAX Call with jQuery

The second parameter may be unfamiliar to you, but in Javascript you can write functions inline, which is often convenient.

Since we won't be able to use this function elsewhere, we don't name it.

However, jQuery will automatically pass the server response as the first parameter, which in our case is the resp parameter.

Here's the function by itself, for reference:

```
function (resp)
{
     // update the filed with the response
     jQuery('#time') .val(resp);
});
```

This function highlights one of the coolest features of jQuery: how simple it is to manipulate the document. Here's whats going on:

JQuery Constructor:

```
jQuery('#time')
```

This code constructs a new jQuery object. Its parameter can be any XPath or CSS3 selector, among others. In short, you can reference elements the same way you would in a CSS file. It returns a jQuery object, which we can apply any jQuery function to.

Note: We used the field's id as the selector, and since there's only one field with that id the jQuery object returned is a collection of only one object. However, if we had used a more general selector, i.e. jQuery('p'), our jQuery object would contain several elements. Any maniuplation done to a jQuery object will be performed on every element in the collection. This isn't important now, but its fundamental to understanding jQuery.

JQuery Constructor:

The second part of the statement:

```
.val(resp);
```

Replaces the VALUE of the matched elements with whatever we pass to it. In this case, the matched element is the filed with the id “time”, and we’re setting its value to be the server’s response (the current server time).

To set the HTML of an element (most common) we would use the following:

```
.html(resp);
```

That’s it for the Javascript, all that’s left is the server side code (PHP in this example).

Conclusion

That's how to make a basic AJAX call using PHP and jQuery.

In our example we simply returned the server time, which isn't incredibly exciting. But the server side code could be doing any number of things, such as looking up a definition in a database, page content, or any other data you could retreive using PHP.