

TUTORIAL 6

Introduction to jQuery

What is jQuery and why should you use it?

jQuery is designed to simplify the process of coding with JavaScript. It builds on top of JavaScript and is used to interact with elements in your HTML page. It's a lot easier to read than JavaScript and you can achieve a lot with just a few lines of code. Most common jQuery effects are drop down menus, drag and drop elements, animations and form validation.

Let's say we wanted to change the background colour of an element when it's clicked. In this example when we click a paragraph of text the background colour will change to red.

Javascript version

```
var i, paragraph = document.getElementsByTagName('p');
for(i = 0; i < divs.length; i++) {
    paragraph[i].onclick = function() {
        this.style.backgroundColor = 'red';
    }
}
```

... ouch!

jQuery version

```
$('#p').click(function() {
    $(this).css('background-color', 'red');
});
```

... much better!

As you can see the jQuery version is shorter and easier to read. This is what makes jQuery so great – and easy to learn.

jQuery doesn't replace Javascript it just simplifies it. There will be times when you will use plain old Javascript – displaying an alert box for example.

Make your page come alive

Adding jQuery to our HTML page.

All the tutorial files are available at <http://design.morsolutions.net/dojo/march24.zip>

1. Link the jQuery library to your HTML file by adding the `<script>` tag. Type the following just before the closing `</HEAD>` tag.

```
<script src="js/jquery-1.7.2.min.js"></script>
```

2. Create a blank document inside the 'js' folder and call it 'scripts.js' and link it to your HTML as well. NOTE: make sure you add it after the jQuery library!

```
<script src="js/scripts.js"></script>
```

3. In the scripts.js file add the following code:

```
$(document).ready(function() {  
    // your jQuery code will go in here!  
});
```

This does two things:

1. First, it ensures that the code does not run until the document is ready.
2. It also confirms that any elements being accessed are actually there, so the script won't return any errors.

Selecting stuff.

To anything with our HTML page we need to be able to select it first. It could be anything from a heading, a paragraph of text or an image on the page.

To select something using jQuery we start with `$(“ ”)`. Inside the quotes we can add the item we want to target. Some common items are...

1. `$(“section”);` selects all HTML section elements
2. `$(“#myElement”);` selects one HTML element with ID "myElement"
3. `$(“.myClass”);` selects HTML elements with class "myClass"

For now let's target the `<header>` tag and change the background colour when it's clicked.

In the scripts.js file add the following code inside the `$(document).ready()` function.

```
$(“header”).click(function() {  
    $(this).css('background-color', 'blue');  
});
```

Line one targets the <header> element and tells it to wait until it has been clicked. This is called an event listener.

Once the event is activated we run the function. Inside this function we select the item that's been clicked and call a jQuery specific function called `css()`.

Inside `css()` we can pass the name of the CSS style we want to apply and its value. In this case we're changing the background colour and applying the value 'blue'.

TRY FOR YOURSELF

Target other elements on the page and change how they look using the `css()` function. Here are some examples:

1. `css('font-size','48px')`
2. `css('color','red')`
3. `css('border','1px solid green')`

Try other event listeners: `hover`, `keydown`, `mousemove`, `resize`, `scroll` ... there's a lot more!

Animation.

One thing that jQuery is great at is animation. You can slide elements, animate elements, and even stop animations in mid-sequence.

Lets hide the content of our sub headings when the heading itself is clicked.

Add the following to your `index.html` file just after the closing `<p>` of the 'Lorem ipsum' text.

```
<article>
  <h2>Heading</h2>
  <p>Add in your own text here...<p>
</article>
```

... add more than just one article.

jQuery comes with some prebuilt effects. In the `scrips.js` file add the following code:

```
$( "h2" ).click(function() {
    $(this).next( 'p' ).slideUp();
});
```

We can pass arguments into the `slideUp()` function that make it perform in certain ways.

Try adding 'slow', 'fast' or time to the function. Note: for 1 second write 1000 (milliseconds) with no quotes...

```
$( "h2" ).click(function() {
    $(this).next( 'p' ).slideUp( 'slow' );
```

```
});

$("h2").click(function() {
    $(this).next('p').slideUp(6000); // 6 seconds
});
```

TRY FOR YOURSELF

Try using some of the other jQuery events: `fadeIn()`, `fadeOut()`, `show()`, `hide()`.

You might need to use the `toggle()` function to switch between to effects. So when the `slideUp()` effect is complete you can click again to call the `slideDown()`.

TRY FOR YOURSELF

```
$("h2").toggle(

function() {
    $(this).next('p').slideUp();
},

function() {
    $(this).next('p').slideDown();
}

);
```

Chaining things together

Another great thing about jQuery is that you can chain functions together. You've already done it! When you selected the `<header>` then added the `css()` and `slideUp()` functions. To chain each of these functions we simply inserted a full stop (`.`) in between them.

We can make complex selections using this method:

```
$("h2").click(function() {
    $(this).css('color', 'red').next('p').slideUp();
});
```

TRY FOR YOURSELF

With the events you've learned try making several things happen when an event is triggered (like a click)

That's it for today. Thank you for joining us at Drogheda Coder DoJo.