



INTRODUCTION TO JQUERY (PART TWO)

OVERVIEW

- Events
- Image rollover
- Image replacement
- Plugins

EVENTS

- Web browsers are programmed to recognise basic actions like the page loading, someone moving a mouse, typing a key, or resizing the browser window
- Preparing a web page to respond to an event is a three-stage process:
 - Select the element node(s) you want the script to respond to.
 - Indicate which event on the selected node(s) will trigger the response.
 - State the code you want to run when the event occurs.

WRITING AN EVENT HANDLER

- We are going to handle events using the on() Method.
- The on() method attaches one or more event handlers for the selected elements.
- Event handlers attached using the on() method will work for both current and future elements.
- For example, if we want to attach a click event to a <p> element:

```
$("p").on("click", function(){  
    $(this).hide();  
});
```

WRITING AN EVENT HANDLER

```
$('#myImage').on({  
  mouseenter: function(){  
    $(this).attr('src', 'images/landscape.jpg');  
  },  
  mouseleave: function(){  
    $(this).attr('src', 'images/landscape.gif');  
  }  
});
```

JQUERY HOVER

- The mouseenter and mouseleave events are frequently used together.
- For example, when you mouse over a button, a menu might appear, move your mouse off the button, and the menu disappears.
- jQuery provides a shortcut of referring to both. The .hover() method binds handlers for both mouseenter and mouseleave events.
- jQuery's .hover() method works like any other event, except it accepts two functions.

```
$('#selector').hover(function1, function2);
```

JQUERY HOVER

- In the following example, if we move the mouse over the image it changes; move the mouse away and the original image is restored.

```
$('#myImage').hover(function(){  
    $(this).attr('src', 'images/landscape.jpg');  
    },  
    function(){  
    $(this).attr('src', 'images/landscape.gif');  
    });
```

IMAGE ROLLOVER

- We will now look at a method of performing image rollovers. This example uses the opacity property in CSS to control the display (1 = fully visible; 0 = fully invisible).

```
$(document).ready(function() {  
    $('#gallery img').css('opacity', 0.6);  
  
    $('#gallery img').on('mouseover', function() {  
        $(this).css('opacity', 1);  
    });  
  
    $('#gallery img').on('mouseout', function() {  
        $(this).css('opacity', 0.6);  
    });  
});
```


IMAGE ROLLOVER

- In this example, we also use the `.fadeTo()` method to fade in and out the opacity of the images.

```
$(document).ready(function() {  
  $('#gallery img').css('opacity', 0.6);  
  
  $('#gallery img').on('mouseover', function() {  
    $(this).stop().fadeTo(1200, 1);  
  });  
  
  $('#gallery img').on('mouseout', function() {  
    $(this).stop().fadeTo(1200, 0.6);  
  });  
});
```

IMAGE REPLACEMENT

- Because the images are included in links in the previous examples, if you clicked an image then you would follow the link and a new page would open displaying a bigger image of the link/image clicked.
- We will now look at another example that will ignore the click event on the thumbnail image (and associated link) and instead get the link's href value and display it on the current page (i.e. replace the larger image with the href value of the chosen link).
- As the image is being replaced, it fades into view slowly as opposed to immediately.

IMAGE REPLACEMENT

```
$('#gallery a').click(function(evt) {  
    evt.preventDefault();  
    var imgPath = $(this).attr('href');  
    var oldImage = $('#photo img');  
    var newImage = $('');  
    $('#photo').prepend(newImage);  
    oldImage.fadeOut(4500,function(){  
        $(this).remove();  
    });  
});  
$('#gallery a:first').click();
```

JQUERY PLUGINS

- jQuery plugins build on the capabilities of the jQuery library
- You can also find more examples on the jQuery plugin registry:
 - <https://plugins.jquery.com/>

EXAMPLE: ADDING A LIGHTBOX

- One very popular technique for viewing images is dimming the Web page and displaying the larger image version of the thumbnail as if it were floating on top of the browser window.
- One well known version of this method is a JavaScript plugin called Lightbox.
- Lightbox is small javascript library used to overlay images on top of the current page.

EXAMPLE: SETTING THE LIGHTBOX UP

- In the `<head>` of the document, before the other `<link>` tag, we will add: `<link href="css/lightbox.css" rel="stylesheet">`. This file contains all of the styles used to format the background that lies over the Web page, the pop up image, and the photo caption text.
- We will attach the Lightbox plugin JavaScript file immediately after the `<script>` tag that attaches the jquery file.

`<script src="js/lightbox.min.js"></script>`

- Look inside the images folder to find *close.png*, *loading.gif*, *prev.png*, and *next.png*. These files are used in *lightbox.css*.

EXAMPLE: EXECUTING THE LIGHTBOX

- Add a **data-lightbox** attribute to any image link to activate Lightbox.
- For the value of the attribute, use a unique name for each image. For example:
 - `image #1`
- Optional: Set the **data-title** attribute if you want to show a caption.
- If you have a group of related images that you would like to combine into a set, use the same data-lightbox attribute value for all of the images. For example:

```
<a href="images/image-2.jpg" data-lightbox="roadtrip">image #2</a>  
<a href="images/image-3.jpg" data-lightbox="roadtrip">image #3</a>  
<a href="images/image-4.jpg" data-lightbox="roadtrip">image #4</a>
```

LIGHTBOX OPTIONS

- You can make changes by adjusting the LightBox CSS property values and by changing the images used.
- You can also make changes by adding JavaScript code to adjust fade speeds, etc...

DATE PICKER

- To aid the user adding a date, we can add a date picker. For this we will use the Bootstrap datepicker and call it using jQuery.
- You will need the following files:
 - `css/css/bootstrap-datepicker.min.css`
 - `js/js/bootstrap-datepicker.min.js`

DATE PICKER

- To call the datepicker, you need (JavaScript):

```
<script>
$(document).ready(function () {
    $('#dob').datepicker();
});
</script>
```

DATE PICKER

- Again, there are options with the date picker such as:
 - Format
 - StartDate
 - EndDate
 - StartView.

HTML5 VALIDATION PLUGIN

- **h5Validate** is a jQuery plugin that understands HTML5 forms and knows how to validate them, even in browsers that don't yet support HTML5. In browsers that do support HTML5, h5Validate adds some much-needed features, such as the ability to customize the user interface when an input fails validation.
- You will need the following file:
 - `jquery.h5validate.js`

HTML5 VALIDATION PLUGIN

- You can invoke the plugin as follows:

```
$(document).ready(function () {  
    $('form').h5Validate();  
});
```

- To add colour to the elements causing the error add an errorClass.

```
$(document).ready(function () {  
    $('form').h5Validate({errorClass:'validationError' });  
});
```

HTML5 VALIDATION PLUGIN

- To display the title attribute values as error messages on the web page, we assign the **data-h5-errorid** attribute in the input field and we add a corresponding error div.
- The data-h5-errorid attribute is assigned a value and that value is the id of the error div.
- The error div includes a class to style the error message (if displayed).

