

Introduction to jQuery– Part Two

Website Development 2

Lecture Outline

- 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

- The following attaches two event handlers for the element with an ID of *myImage*.

```
$('#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');  
    });
```

Lecture Outline

- Events
- Image rollover
- Image replacement
- Plugins

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).

```
$('#gallery img').css('opacity', 0.6);

$('#gallery img').on({
  mouseover:function() {
    $(this).css('opacity', 1);
  },

  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.

```
$('#gallery img').css('opacity', 0.6);

$('#gallery img').on({
  mouseover:function() {
    $(this).stop().fadeTo(4000, 1);
  },
  mouseout:function() {
    $(this).stop().fadeTo(4000, 0.6);
  }
});
```

Lecture Outline

- Events
- Image rollover
- Image replacement
- Plugins

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 an example that will ignore the click event on the thumbnail image (and associated link) and instead get the link's *href* value. This *href* value becomes the source of a new image and it is displayed 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').on('click', function(evt) {  
    evt.preventDefault();  
    let imgPath = $(this).attr('href');  
    let newImage = $('<img src=' + imgPath + ' alt = "" >');  
    let oldImage = $('#photo img');  
    $('#photo').prepend(newImage);  
    oldImage.fadeOut(4500,function(){  
        $(this).remove();  
    });  
});  
  
$('#gallery a:first').click();
```

Lecture Outline

- Events
- Image rollover
- Image replacement
- Plugins

jQuery plugins

- jQuery plugins build on the capabilities of the jQuery library.
- In the labs, you can see how to integrate the following plugins:
 - LightBox.
 - DatePicker.
 - h5Validate.

jQuery plugins

Example One

- We will use the [Math Library](#).
- Include the following link:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/mathjs/5.4.2/math.js">  
</script>
```
- Examples:
 - `let a = math.factorial(5);`
 - `let b = math.multiply(4,3,2);`
 - `let c = math.chain(3).add(4).multiply(2).done();`

jQuery plugins

Example Two

- In this example, we will convert a list of linked titles into a list of linked thumbnails, using JavaScript DOM manipulation and the functions from the **youtube.JS** library ([read documentation](#)).
- Include the following link:
<script src="http://www.teaching-materials.org/common/youtube.js"></script>

jQuery plugins

Example Two

- Select every link on the page and iterate through each one
\$('a').each(function() {..});
- In each iteration of the loop:
 - Find the current href, and store into a variable.
let linkUrl = \$(this).attr('href');
 - Generate a thumbnail URL using `youtube.generateThumbnailUrl()`.
let thumbnailUrl = youtube.generateThumbnailUrl(linkUrl);
 - Create an IMG element and set the "src" of the IMG element.
let thumbnaillmg = \$('');
thumbnaillmg.attr('src', thumbnailUrl);
 - Replace the text with the IMG to the link.
\$(this).html(thumbnaillmg);

jQuery plugins

- We will now download and implement our own plugin.
 - <https://plugins.jquery.com>