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 two-stage process:
  - Identify the page element that you wish to respond to an event.
  - Assign an event and define a function to run when the event occurs.

MOUSE EVENTS

- **.click**: The click fires after you click and release the mouse button.
- **.dblclick**: The dblclick fires after you click and release the mouse button twice.
- **.mousedown**: The mousedown fires after you press the mouse button down.
- **.mouseup**: The mouseup fires after you release the mouse button.
- **.mouseenter**: The mouseenter fires after you move the mouse over a page element.
- **.mouseover**: The mouseover fires after you move the mouse over a page element. It works on child elements also.
- **.mouseleave**: The mouseleave fires after you move the mouse off a page element.
- **.mouseout**: The mouseout fires after you move the mouse off a page element. It works on child elements also.
- **.mousemove**: The mousemove fires after you move the mouse.

DOCUMENT/WINDOW EVENTS

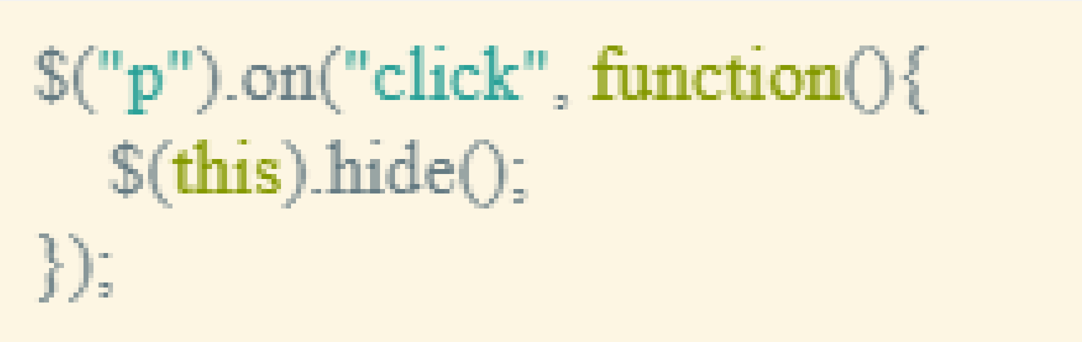
- **.load**: The load fires when the web browser finishes downloading all of a web page's files.
- **.resize**: The resize fires when you resize the browser window by clicking the maximise button, or by dragging the browser's resize handle.
- **.scroll**: The scroll fires when you drag the scroll bar, or use the keyboard (up/down/home/end) keys, or mouse scroll wheel to scroll a web page. If the web page has no scroll bars then no scroll event is ever triggered.
- **.unload**: The unload fires when you click a link to go to another page, close a browser tab, or close a browser window.

FORM EVENTS

- **.submit**: The submit fires whenever a visitor submits a form.
- **.reset**: The reset fires whenever a visitor clicks the reset button on a form.
- **.change**: The change fires whenever a form field value or status changes.
- **.focus**: The focus fires when you click into or select a form element.
- **.blur**: The blur fires when you exit a currently focused form element.

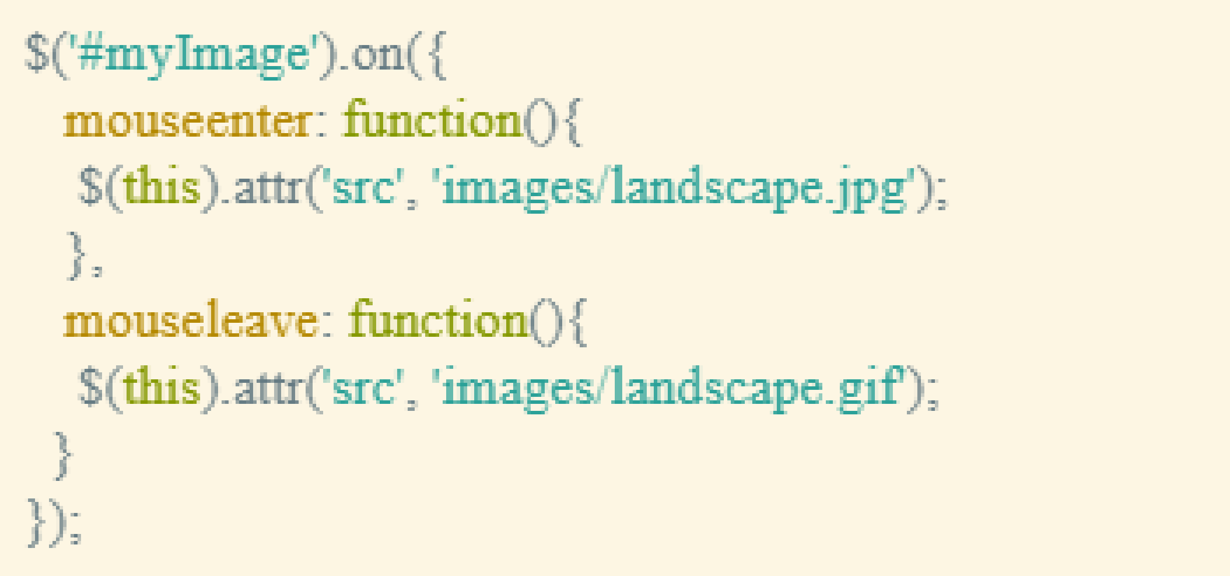
WRITING AN EVENT HANDLER

- 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();
});
```

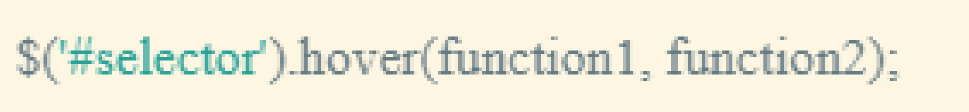
WRITING AN EVENT HANDLER



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

JQUERY HOVER

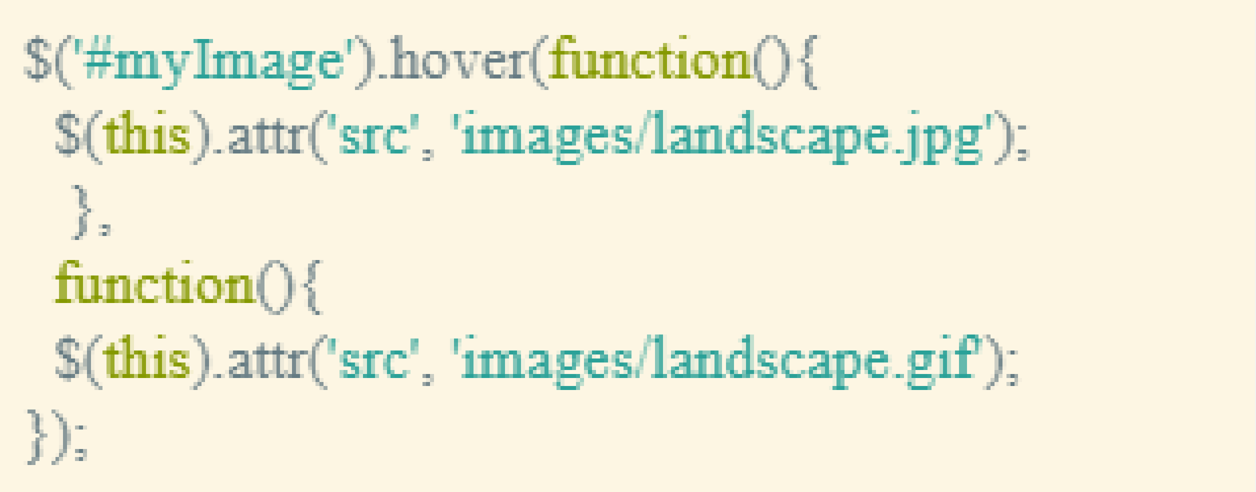
- The mouseenter and mouseleave events are frequently used together.
- For example, when you mouse over a button, a menu might appear, move your mouse off the button, and the menu disappears.
- jQuery provides a shortcut of referring to both. The .hover() method binds handlers for both mouseenter and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



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

JQUERY HOVER

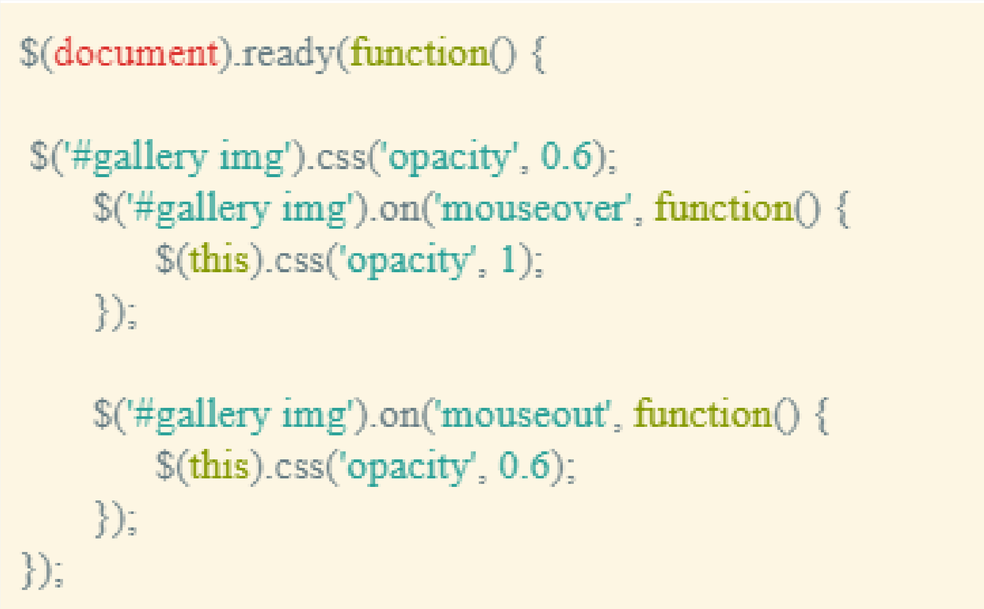
- In the following example, if we move the mouse over the image it changes; move the mouse away and the original image is restored.



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

IMAGE ROLLOVER

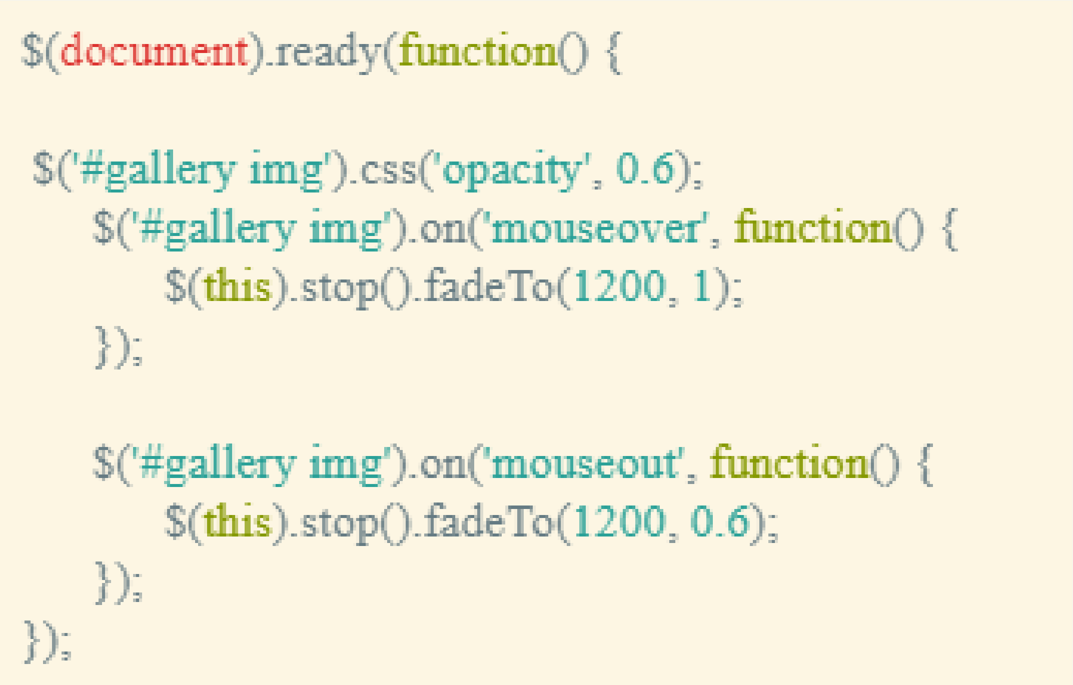
- We will now look at a method of performing image rollovers. This example uses the opacity property in CSS to control the display (1 = fully visible; 0 = fully invisible).



```
$(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 tofade in and out the opacity of the images.

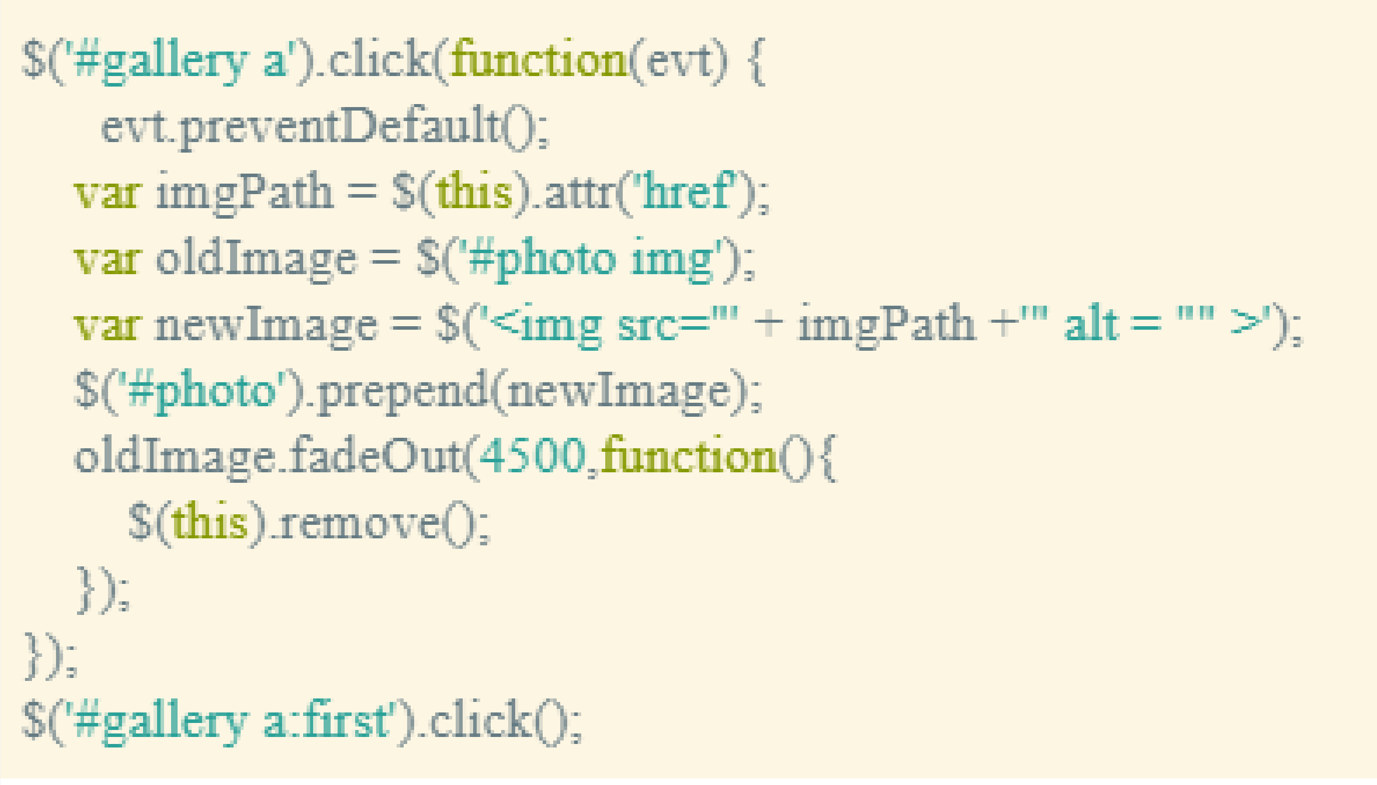


```
$(document).ready(function() {

 $('#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 = $('<img src="' + imgPath +'" alt = "" >');
    $('#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

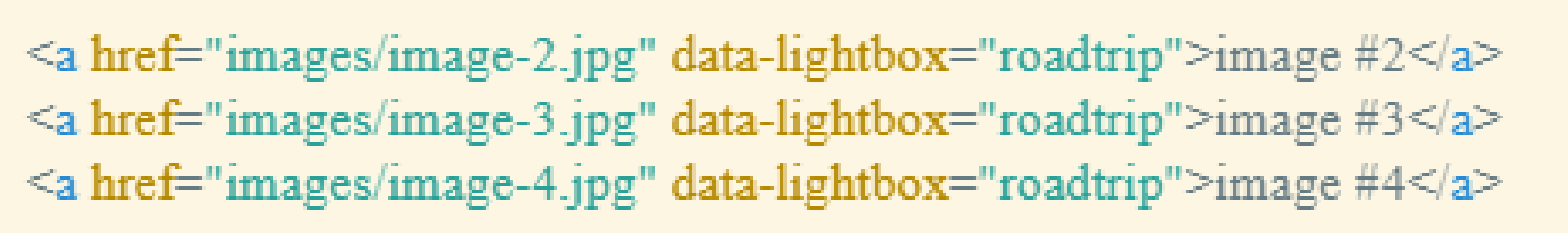
- One very popular technique for viewing images is dimming the Web page and displaying the larger image version of the thumbnail as if it were floating on top of the browser window.
- One well known version of this method is a JavaScript plugin called Lightbox.
- Lightbox is small javascript library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

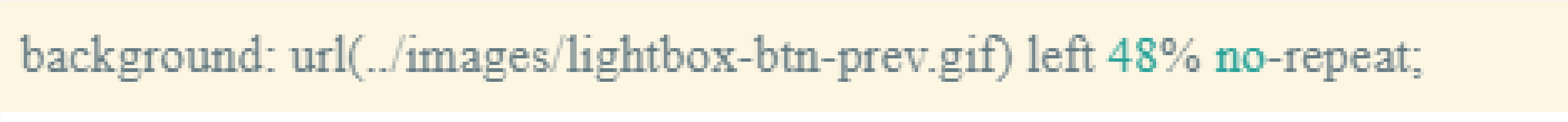
- Add a data-lightbox attribute to any image link to activate Lightbox.
- For the value of the attribute, use a unique name for each image. For example:
    - <a href="images/image-1.jpg" data-lightbox="image-1" data-title="My caption">image #1</a>
- Optional: Set the data-title attribute if you want 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>
```

EXAMPLE: EDIT LIGHTBOX CSS PROPERTIES

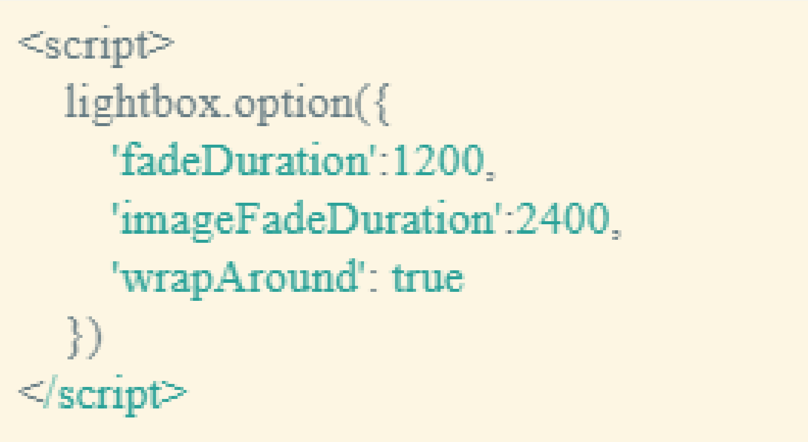
- **Changing the LightBox CSS property values**
- To use a different pointers/buttons/arrows, change the background property in the (lightbox.css file) to the following for prev (.lb-nav a.lb-prev):



```
background: url(../images/lightbox-btn-prev.gif) left 48% no-repeat;
```

EXAMPLE: ADDING OPTIONS

- We may add options to the lightbox plugin, as can be seen in the following example. For example instead of stopping when we reach the maximum number of images, the lightbox continues to iterate.
- All options can be found at the following site: http://lokeshdhakar.com/projects/lightbox2/#options



```
<script>
   lightbox.option({
      'fadeDuration':1200,
      'imageFadeDuration':2400,
      'wrapAround': true
   })
</script>
```