

JavaScript Workshop

jQuery Basics pt. 2

DOM Traversal Methods

```
<ul>  
  <li class = "first" >item 1</li>  
  <li class="second" >item 2</li>  
  <li class = "third" >item 3</li>  
</ul>
```

`$("li").eq(2)` returns the third ``

`$("li").lt(2)` returns the first and second ``

`$("li").gt(1)` returns the third ``

`$(document).find(".second")` returns the second ``

`$("li.second").siblings()` returns the first and third ``

`$("li.third").prev()` returns the second ``

`$("li.first").next()` returns the second ``

Events - .bind()

- `.bind (eventType [, eventData], handler)`
- `eventType`: A string containing a JavaScript event type such as `click` or `submit`.
- Attaches a handler to an event for the elements. The `.bind()` method is the primary means of attaching behavior to a document. All JavaScript event types are allowed for `eventType`, such as:

change, click, dblclick, error, focus, keydown, keypress, keyup, load, mouseover, mouseout, mouseup, resize, scroll, submit, unload

- jQuery provides shortcut methods for binding each of these event types such as `.click()` for `.bind('click')`.

Events - .hover()

- Binds two event handlers to the matched elements, to be executed when the mouse pointer enters and leaves the elements.

```
.hover(handlerIn, handlerOut)
```

- The hover() method binds handlers for both mouseover and mouseout events. You can use it to simply apply behavior to an element during the time the mouse is within the element.

```
$(document).ready(function() {  
  
    $('#item1, #item2').hover(function() {  
        $(this).addClass('hover');  
    }, function() {  
        $(this).removeClass('hover');  
    });  
});
```

Events – The Event object

- The callback function takes a single parameter; when the handler is called the JavaScript event object will be passed through it. The event object is often unnecessary, but at times it may be necessary to gather more information about the user's environment at the time of the event. JavaScript provides some properties and methods such as:

`.target`: DOM element that initiated the event

`.pageX` & `.pageY`: x and y coordinates of the mouse cursor

`.preventDefault()`: default action of event will not be triggered

`.stopPropagation()`: prevents event from bubbling up to the DOM tree

```
$('#foo').click(function(evt) {  
    alert('Mouse cursor s at ' + evt.pageX + ' , ' + evt.pageY)  
});
```

Events - .unbind()

- Removes a previously attached event handler from the element
- `.unbind ([eventType [, handler]]); .unbind(event)`

`$('#foo').unbind();` --- removes all handlers attached to element

`$('#foo').unbind('click')` --- click event handlers will be removed

`$('#foo').unbind('click', handler)` --- 'handler' for click event removed

Effects

- jQuery provides some effect methods:

.hide() – style attribute to display:none

.show() – restores the visibility display

.toggle() – displays or hides

.slideDown() – displays with a sliding motion

.slideUp() – hides with a sliding motion

.slideToggle() – displays or hides with a sliding motion

.fadeIn() – gradually increase opacity

.fadeOut() - gradually decrease opacity

.animate() – modify position: (left, top, right, bottom) or opacity

most methods accept optional **speed** parameter: 'slow', 'normal', 'fast'

DOM Manipulation Methods

- `.attr(attribute)` – gets value of attribute for the first element in the set
- `.removeAttr(attribute)` – removes attribute from each element in the set
- `.addClass(class)` – adds class to each element in the set
- `.removeClass(class)` – removes class from each element in set
- `.text(text)` – sets the content of each element in the set
- `.prepend(content)` – inserts content at the beginning of the element
- `.append(content)` – inserts content at the end of the element
- `.before(content)` – inserts content before the element
- `.after(content)` – inserts content after the element
- `.clone()` – creates a copy of the matched elements
- `.remove()` - removes the set of matched elements

Extend jQuery: Plug-ins

- jQuery has an architecture to allow for extensibility through the use of plug-ins. Plug-ins range from small selector helpers to full-scale, user-interface widgets.

Find a Plug-in:

<http://jquery.com/plugins/>

Use a Plug-in:

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

- JavaScript Workshop website:
<http://www.javascriptworkshop.com>
- Any questions?
frank@javascriptworkshop.com - Frank Stepanski
- Resources:
http://docs.jquery.com/Main_Page
<http://docs.jquery.com/Tutorials>