CSE 3002
INTERNET AND WEB PROGRAMMING

Module : 3

jQuery

Prof R Lokeshkumar

jQuery

The Way to JavaScript and Rich Internet Applications

Introduction to jQuery

- Developed by John Resig at Rochester Institute of Technology
- **“jQuery** is a lightweight JavaScript library that emphasizes interaction between JavaScript and HTML. It was released in January 2006 at BarCamp NYC by John Resig.”
- “jQuery is free, open source software Dual-licensed under the MIT License and the GNU General Public License.”
- “It’s all about simplicity. Why should web developers be forced to write long, complex, book-length pieces of code when they want to create simple pieces of interaction?”

Introduction to jQuery

- Installation – You just download the jquery.js file and put it in your website folder
    - Can access via URL

What jQuery Does

- “Unobtrusive” JavaScript – separation of <u>behavior</u> from structure
- CSS – separation of <u>style</u> from structure
- Allows adding JavaScript to your web pages
- Advantages over *just* JavaScript
  - Much easier to use
  - Eliminates cross-browser problems
- HTML to CSS to DHTML

5 Things jQuery Provides

- ▶Select DOM (Document Object Model) elements on a page – one element or a group of them
- ▶Set properties of DOM elements, in groups ("Find something, do something with it")
- ▶Creates, deletes, shows, hides DOM elements
- ▶Defines event behavior on a page (click, mouse movement, dynamic styles, animations, dynamic content)
- ▶AJAX calls

The DOM

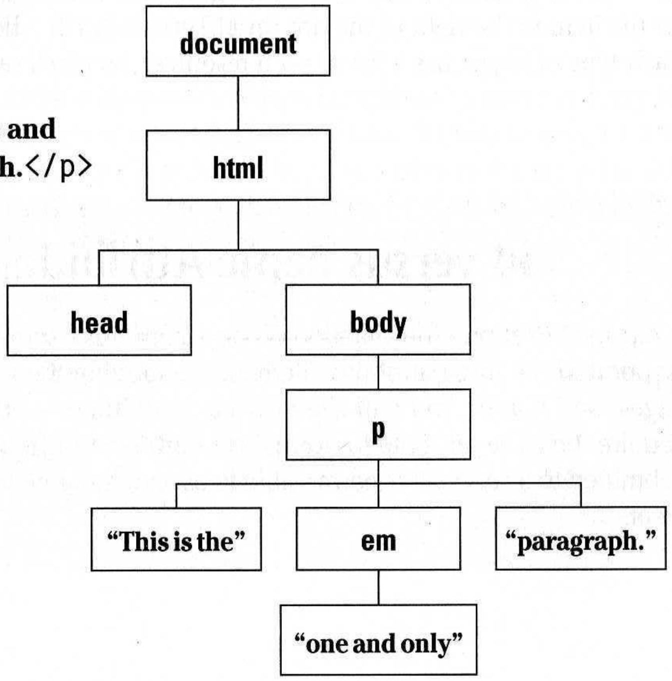
- Document Object Model
- jQuery is “DOM scripting”
- Heirarchal structure of a web page
- You can add and subtract DOM elements on the fly
- You can change the properties and contents of DOM elements on the fly

The DOM

- “The **Document Object Model** (**DOM**) is a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML and XML documents. Aspects of the DOM (such as its "Elements") may be addressed and manipulated within the syntax of the programming language in use.” Wikipedia

A simple HTML document node tree.

```
<html>
  <head></head>
  <body>
       <p>This is the <em>one and
         only</em> paragraph.</p>
  </body>
</html>
```



The jQuery Function

jQuery() = $()	
$(function)	The “Ready” handler
$(‘selector’)	Element selector expression
$(element)	Specify element(s) directly
$(‘HTML’)	HTML creation
$.function()	Execute a jQuery function
$.fn.myfunc(){}	Create jQuery function

The Ready Function

- Set up a basic HTML page and add jQuery
- Create a “ready” function
- Call a function
- 5 ways to specify the ready function
  - jquery(document).ready(function(){…};);
  - jquery().ready(function(){…};)
  - jquery(function(){…};)
  - jquery(dofunc);
  - $(dofunc);

Selecting Elements
Creating a “wrapped set”

- $(selector)
- selector:

$(‘#id’)	id of element
$(‘p’)	tag name
$(‘.class’)	CSS class
$(‘p.class’)	<p> elements having the CSS class
$(‘p:first’) $(‘p:last’)	$(‘p:odd’) $(‘p:even’)
$(‘p:eq(2)’)	gets the 2nd <p> element (1 based)
$(‘p’)[1]	gets the 2nd <p> element (0 based)
$(‘p:nth-child(3))	gets the 3rd <p> element of the parent. n=even, odd too.
$(‘p:nth-child(5n+1)’)	gets the 1st element after every 5th one
$(‘p a’)	<a> elements, descended from a <p>
$(‘p>a’)	<a> elements, direct child of a <p>
$(‘p+a’)	<a> elements, directly following a <p>
$(‘p, a’)	<p> and <a> elements
$(‘li:has(ul)’)	<li> elements that have at least one <ul> descendent
$(‘:not(p)’)	all elements but <p> elements
$(‘p:hidden’)	only <p> elements that are hidden
$(‘p:empty’)	<p> elements that have no child elements

Selecting Elements, cont.

- $(‘img’[alt]) <img> elements having an alt attribute
- $(‘a’[href^=http://]) <a> elements with an href attribute starting with ‘http://’
- $(‘a’[href$=.pdf]) <a> elements with an href attribute ending with ‘.pdf’
- $(‘a’[href*=ntpcug]) <a> elements with an href attriute containing ‘ntpcug’

Useful jQuery Functions

▶.each()	iterate over the set
▶.size()	number of elements in set
▶.end()	reverts to the previous set
▶.get(n)	get just the nth element (0 based)
▶.eq(n)	get just the nth element (0 based) also .lt(n) & .gt(n)
▶.slice(n,m)	gets only nth to (m-1)th elements
▶.not(‘p’)	don’t include ‘p’ elements in set
▶.add(‘p’)	add <p> elements to set
▶.remove()	removes all the elements from the page DOM
▶.empty()	removes the contents of all the elements
▶.filter(fn/sel)	selects elements where the func returns true or sel
▶.find(selector)	selects elements meeting the selector criteria
▶.parent()	returns the parent of each element in set
▶.children()	returns all the children of each element in set
▶.next()	gets next element of each element in set
▶.prev()	gets previous element of each element in set
▶.siblings()	gets all the siblings of the current element

Formatting Elements

- .css(property, value)
- .html()
- .val()   (form elements)
- .text()
- .addClass(‘class’)
- .removeClass(‘class’)

Add Page Elements

- $(‘#target’).before(‘<p>Inserted before #target</p>’);
- $(‘#target’).after(‘<p>This is added after #target</p>’);
- $(‘#target’).append(‘<p>Goes inside #target, at end</p>’);
- $(‘#target’).wrap(‘<div></div>’);

Adding Events

- Mouseover events – bind, hover, toggle
- Button click events
- Keystrokes

Event Background

- DOM Level 2 Event Model
  - Multiple event handlers, or listeners, can be established on an element
  - These handlers cannot be relied upon to run an any particular order
  - When triggered, the event propagates from the top down (capture phase) or bottom up (bubble phase)
  - IE doesn’t support the “capture phase”

Basic Syntax of Event Binding

- $('img').bind('click',function(event){alert('Howdy';});
- $('img').bind('click',imgclick(event));
  - Allows unbinding the function
- $('img').unbind('click',imgclick());
- $('img').unbind('click');
- $('img').one('click',imgclick(event));
  - Only works once
- $('img').click(imgclick);
- $('img').toggle(click1, click2);
- $('img').hover(mouseover, mouseout);

Element Properties – “this”

- ▶this
- ▶this.id
- ▶this.tagName
- ▶this.attr
- ▶this.src
- ▶this.classname
- ▶this.title
- ▶this.alt
- ▶this.value (for form elements)

‘Event’ properties

- event.target ref to element triggering event
- Event.target.id id of element triggering event
- event.currentTarget
- event.type type of event triggered
- event.data second parm in the bind() func
- Various mouse coordinate properties
- Various keystroke related properties

Event Methods

- .stopPropagation()   no bubbling
- .preventDefault()   no <a> link, no <form> submit
- .trigger(eventType) does not actually trigger the event, but calls the appropriate function specified as the one tied to the eventType
- .click(), blur(), focus(), select(), submit()
  - With no parameter, invokes the event handlers, like trigger does, for all the elements in the wrapped set

Shortcut Event Binding

- .click(func)
- .submit(func)
- .dblclick(func)
- .mouseover(func)
- .mouseout(func)
- .select(func)

Useful Event Functions

- .hide() display:true
- .show() display:none
- .toggle(func1, func2) first click calls func1, next click executes func2
- .hover(over, out) mouseover, mouseout