



JQUERY - AJAX

Sisoft Technologies Pvt Ltd
SRC E7, Shipra Riviera Bazar, Gyan Khand-3, Indirapuram, Ghaziabad
Website: www.sisoft.in Email: info@sisoft.in
Phone: +91-9999-283-283

LEARNING TOPICS

- Introduction to JQuery
- The DOM (Document Object Modal)
- JQuery Functions
- JQuery Ajax Methods

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?”
- Current version is 1.3.2.
- Version 1.4 due out soon.



Introduction to jQuery

Installation - You just download the jquery-1.3.2.js file and put it in your website folder
Can access via URL

What jQuery Does

- “Unobtrusive” JavaScript – separation of behavior from structure
- CSS – separation of style 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

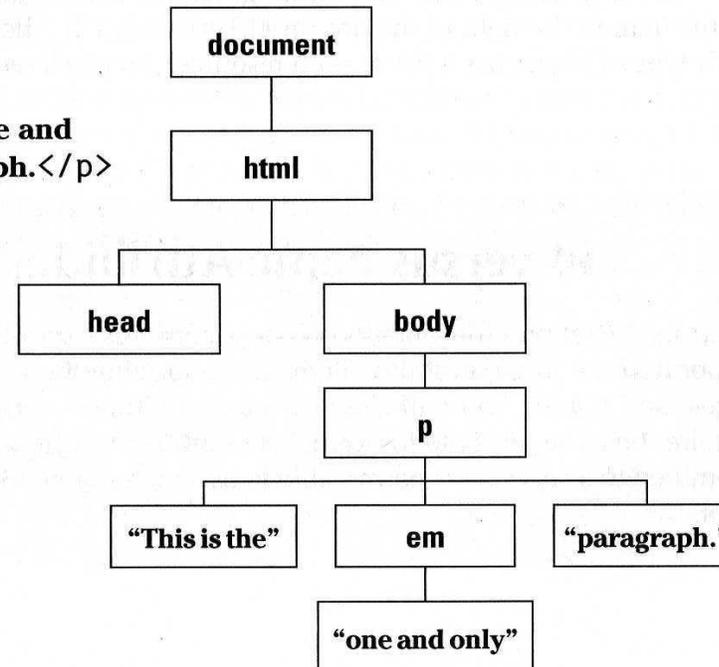
- Document Object Model
- jQuery is “DOM scripting”
- Heirarchal structure of a web page
- You can add and subtract DOM elements on the fly
- You can change the properties and contents of DOM elements on the fly

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.”

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)') elements that have at least one
- 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.

- `$(‘img’[alt])` `` elements having an alt attribute
- `$(‘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 attribute containing ‘ntpcug’

Useful jQuery Functions

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



jQuery AJAX Methods

AJAX is the art of exchanging data with a server, and update parts of a web page - without reloading the whole page.

The following table lists all the jQuery AJAX methods:

[\\$.ajax\(\)](#):- Performs an async AJAX request

```
$("#button").click(function(){
    $.ajax({url: "demo_test.txt",
success: function(result){
    $("#div1").html(result);
    }});
});
```

Syntax

```
$.ajax({name:value, name:value, ... })
```

The parameters specifies one or more name/value pairs for the AJAX request.

You can refer the website https://www.w3schools.com/jquery/ajax_ajax.asp for possible names/values given in the table.

`$.ajaxPrefilter()`

Handle custom Ajax options or modify existing options before each request is sent and before they are processed by `$.ajax()`

jQuery ajaxSetup() Method:- Sets the default values for future AJAX requests

```
$("#button").click(function(){
    $.ajaxSetup({url: "demo_ajax_load.txt",
success: function(result){
    $("#div").html(result);}}});
    $.ajax();
});
```

`$.ajaxTransport()`

Creates an object that handles the actual transmission of Ajax data

jQuery get() Method:- The \$.get() method loads data from the server using a HTTP GET request.

Syntax:- \$.get(*URL, data, function(data, status, xhr), dataType*)

```
$("#button").click(function(){
    $.get("demo_test.asp", function(data, status){
        alert("Data: " + data + "\nStatus: " +
status);
    });
});
```

jQuery getJSON() Method :- The getJSON() method is used to get JSON data using an AJAX HTTP GET request.

Syntax:- \$(*selector*).getJSON(*url, data, success(data, status, xhr)*)

```
$("#button").click(function(){
    $.getJSON("demo_ajax_json.js", function(result){
        $.each(result, function(i, field){
            $("#div").append(field + " ");
        });
    });
});
```



jQuery getScript() Method:- The getScript() method is used to get and execute a JavaScript using an AJAX HTTP GET request.

Syntax:- `$(selector).getScript(url, success(response, status))`

```
$("#button").click(function(){  
    $.getScript("demo_ajax_script.js");  
});
```

jQuery param() Method:-

- The param() method creates a serialized representation of an array or an object.
- The serialized values can be used in the URL query string when making an AJAX request.

Syntax:- `$.param(object, trad)`

```
$("#button").click(function(){  
    $("#div").text($.param(personObj));  
});
```

jQuery param() Method:-

- The param() method creates a serialized representation of an array or an object.
- The serialized values can be used in the URL query string when making an AJAX request.

Syntax:- `$.param(object, trad)`

```
$("#button").click(function(){
    $("#div").text($.param(personObj));
});
```

jQuery post() Method:-The \$.post() method loads data from the server using a HTTP POST request.

Syntax:- `$(selector).post(URL, data, function(data, status, xhr), dataType)`

Load data from the server using a HTTP POST request:

```
$("#button").click(function(){
    $.post("demo_test.asp", function(data, status){
        alert("Data: " + data + "\nStatus: " + status);
    });
});
```

Change the text of a <div> element using an AJAX POST request:

```
$("input").keyup(function()){  
    var txt = $("input").val();  
    $.post("demo_ajax_gethint.asp", {suggest:  
txt}, function(result){  
        $("span").html(result);  
    });  
});
```

jQuery ajaxComplete() Method:-

The ajaxComplete() method specifies a function to be run when an AJAX request completes.

Note: As of jQuery version 1.8, this method should only be attached to document.

Unlike ajaxSuccess(), functions specified with the ajaxComplete() method will run when the request is completed, even it is not successful.

Syntax:- `$(document).ajaxComplete(function(event,xhr,options))`

Example

Show a "loading" indicator image while an AJAX request is going on:

```
$(document).ajaxStart(function(){  
    $("#wait").css("display", "block");  
});
```

```
$(document).ajaxComplete(function(){  
    $("#wait").css("display", "none");  
});
```

jQuery ajaxError() Method:- The ajaxError() method specifies a function to be run when an AJAX request fails.

Syntax:- `$(document).ajaxError(function(event,xhr,options,exc))`

```
$(document).ajaxError(function(){  
    alert("An error occurred!");  
});
```

jQuery ajaxSend() Method:-

The ajaxSend() method specifies a function to run when an AJAX requests is about to be sent.

Syntax:- `$(document).ajaxSend(function(event,xhr,options))`

```
$(document).ajaxSend(function(e, xhr, opt){
    $("div").append("<p>Requesting: " +
    opt.url + "</p>");
});
```

jQuery ajaxStart() Method:-

The ajaxStart() method specifies a function to be run when an AJAX request starts.

Syntax:- `$(document).ajaxStart(function())`

```
$(document).ajaxStart(function(){
    $(this).html("<img src='demo_wait.gif'>");
});
```

jQuery ajaxStop() Method:-

- The ajaxStop() method specifies a function to run when ALL AJAX requests have completed.
- When an AJAX request completes, jQuery checks if there are any more AJAX requests. The function specified with the ajaxStop() method will run if no other requests are pending.

Syntax:- `$(document).ajaxStop(function())`

```
$(document).ajaxStop(function(){  
    alert("All AJAX requests completed");  
});
```

jQuery ajaxSuccess() Method:-

The ajaxSuccess() method specifies a function to be run when an AJAX request is successfully completed.

Syntax:- `$(document).ajaxSuccess(function(event,xhr,options))`

```
$(document).ajaxSuccess(function(){  
    alert("AJAX request successfully completed");  
});
```

jQuery load() Method:-

The load() method loads data from a server and puts the returned data into the selected element.

Syntax: `$(selector).load(url, data, function(response, status, xhr))`

```
$("#button").click(function(){
    $("#div1").load("demo_test.txt");
});
```

jQuery serialize() Method:-

- The serialize() method creates a URL encoded text string by serializing form values.
- You can select one or more form elements (like input and/or text area), or the form element itself.
- The serialized values can be used in the URL query string when making an AJAX request.

Syntax:- `$(selector).serialize()`

```
$("#button").click(function(){
    $("#div").text($("#form").serialize());
});
```

jQuery serializeArray() Method:-

- The serializeArray() method creates an array of objects (name and value) by serializing form values.
- You can select one or more form elements (like input and/or text area), or the form element itself.

Syntax:- `$(selector).serializeArray()`

```
$("#button").click(function(){  
    var x = $("#form").serializeArray();  
    $.each(x, function(i, field){  
        $("#results").append(field.name + ":" +  
field.value + " ");  
    });  
});
```



Thank You

Sisoft Technologies Pvt Ltd
SRC E7, Shipra Riviera Bazar, Gyan Khand-3, Indirapuram, Ghaziabad
Website: www.sisoft.in Email: info@sisoft.in
Phone: +91-9999-283-283