jQuery

The Magic of jQuery !!

By

M.Madadyar

http://www.madadyar.ir

JavaScript Libraries

- jQuery
- Angular
- Mootools
- Prototype
- YUI

jQuery is a lightweight,

open-source JavaScript library that

simplifies interaction between HTML and JavaScript

Introduction to jQuery

- Developed in 2006 by John Resig at Rochester Institute of Technology

- jQuery is a lightweight JavaScript library that emphasizes interaction between JavaScript and HTML

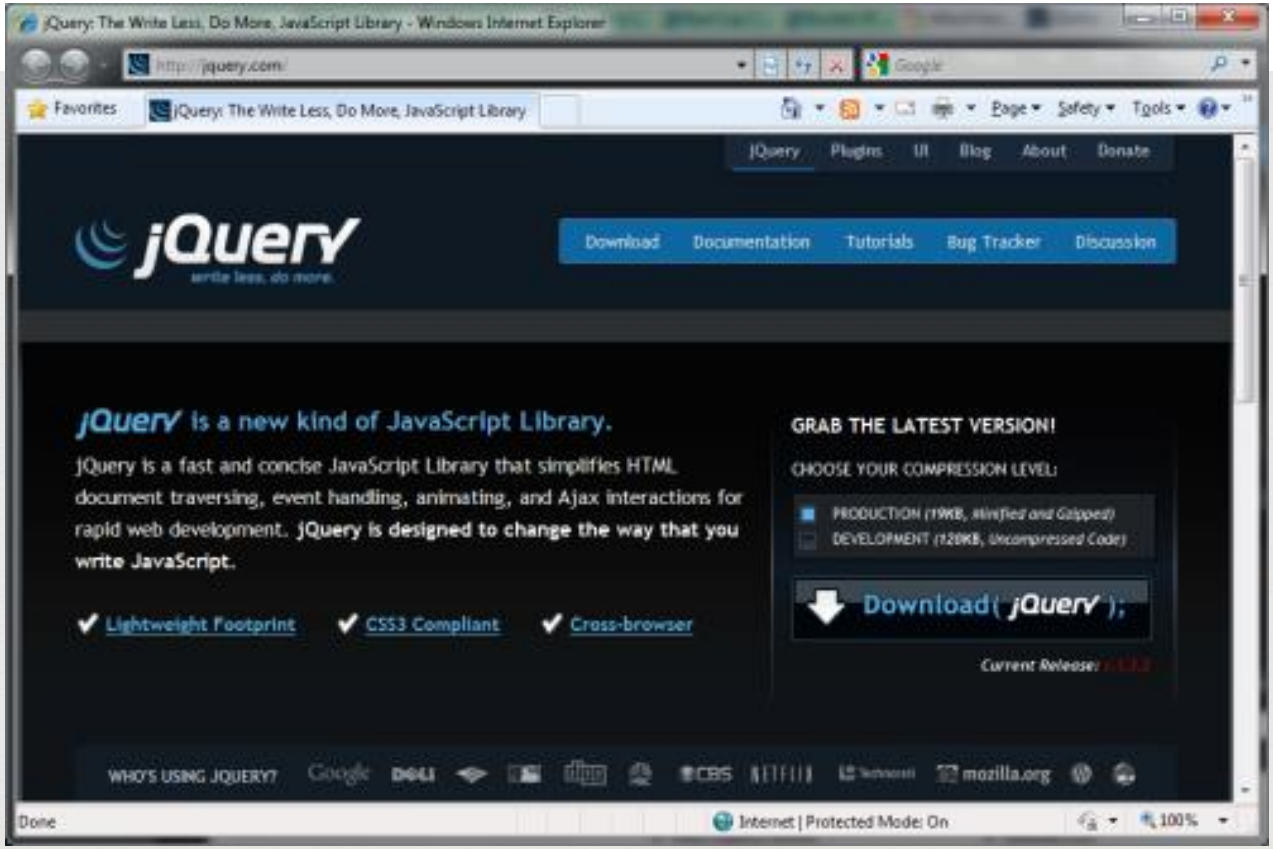
Helps web developers to create simple pieces of interaction without being forced to write long, complex, pieces of code

Introduction to jQuery

- Installation
  - just download the jquery-1.x.x.js file and put it in your website folder
- Using jQuery
  - Simple Editor Like Notepad !
  - RapidCSS
  - VS
  - And etc…

Download the latest version from

http://jquery.com



You can also reference it from Google

```
<script src="http://ajax.googleapis.com/
          ajax/libs/jquery/1.2.6/
          jquery.min.js">
</script>
```

Using jQuery in HTML

- <html>
- <head>
- <script src="jquery.min.js" ></script>
- ***<script type="text/javascript">***

  JQUERY CODES HERE

- ***</script>***
- </head>
- <body>
- </body>
- </html>

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

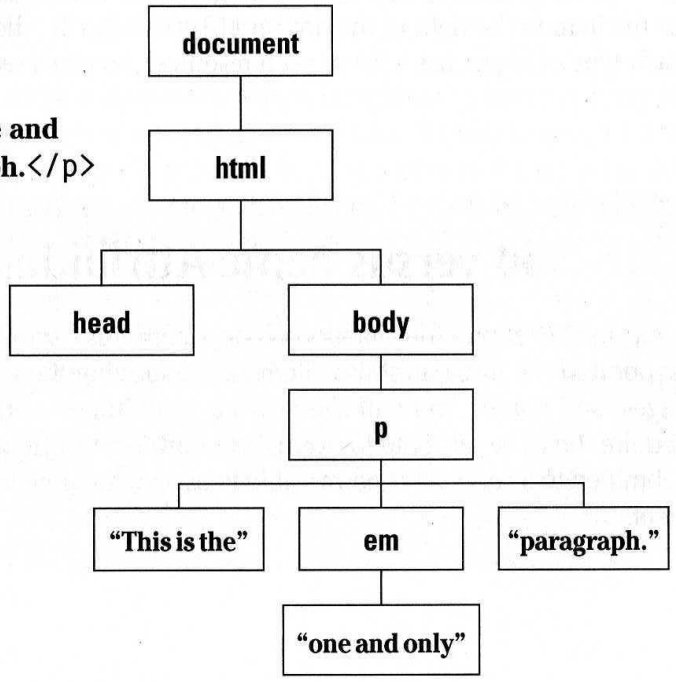
- Document Object Model
- jQuery is “DOM scripting”
- Heirarchal structure of a web page
- You can add and subtract DOM elements on the fly
- You can change the properties and contents of DOM elements on the fly

The DOM

"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>
```



jQuery's programming philosophy is:

GET >> ACT

- Select / Create Elements
- Change Content / Style
- Attach Custom Functions to Events
- Animate Elements

Create Element

```
$("<Tag Name/>")
```

```
$("<div/>")
```

```
function appendText() {
  var txt1 = "<p>Text.</p>";               // Create element with HTML
  var txt2 = $("<p></p>").text("Text.");   // Create with jQuery
  var txt3 = document.createElement("p");  // Create with DOM
  txt3.innerHTML = "Text.";
  $("body").append(txt1, txt2, txt3);      // Append the new elements
}
```

Basic Selectors

By Tag:

```
$("div")
// <div>Hello jQuery</div>
```

By ID:

```
$("#usr")
// <span id="usr">John</span>
```

By Class:

```
$(".menu")
// <ul class="menu">Home</ul>
```

Yes, jQuery implements CSS Selectors!

More Precise Selectors

```
$(“div.main”)      // tag and class
$(“table#data”)    // tag and id
```

Combination of Selectors

```
// find by id + by class
$("#content, .menu")
// multiple combination
$("h1, h2, h3, div.content")
```

Forms Selectors

```
$("input:checkbox")     // checkboxes
$("input:radio")        // radio buttons
$(":button")            // buttons
$(":text")              // text inputs
```

A Selector returns a pseudo array of jQuery objects

```
$("div").length
```

Returns number of selected elements.

It is the best way to check selector.

Selecting Elements

- $(selector)
- selector:
  - **$('#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 <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*=bk])	<a> elements with an href attriute containing 'bk'

Formatting Elements

- .html()
- .val() (form elements)
- .text()

- .css(property, value)
- .addClass('class')
- .removeClass('class')

Getting and Setting Inner Content

```
$(“p”).html(“<div>Hello $!</div>”);
```

Getting and Setting Values

Form elements

```
// get the value of the checked checkbox
 $(“input:checkbox:checked”).val();
```

```
// set the value of the textbox
 $(“:text[name=‘txt’]”).val(“Hello”);
```

Inserting new Elements

```
// select > append to the end
$(“h1”).append(“<li>Hello $!</li>”);


// select > append to the beginning
$(“ul”).prepend(“<li>Hello $!</li>”);
```

Replacing Elements

```
// select > replace
$("h1").replaceWith("<div>Hello</div>");
```

```
$("p").each(function(){
$(this).replaceWith("<div>"
                                    +
$(this).html()
                          + "</div>");
});
```

Deleting Elements

```
// remove all children
$("#mainContent").empty();
```

```
// remove selection
$("p a").remove();
```

Handling attributes

```
$("a").attr("href","home.htm");
 // <a href="home.htm">…</a>
```

```
// remove attribute - enable
 $(":button").removeAttr("disabled")
```

```
$("img").attr({
        "src" : "/images/smile.jpg",
        "alt" : "Smile",
        "width"  : 10,
        "height" : 10
 });
```

CSS Manipulations

```
// get style
$("div").css("background-color");
```

```
// set style
$("div").css("float", "left");
```

```
// set multiple style properties
$("div").css({"color":"blue",
              "padding": "1em"
              "margin-right": "0",
              "marginLeft": "10px"});
```

Handling CSS Classes

```
// add and remove class
$("p").removeClass("blue").addClass("red");
```

```
// add if absent, remove otherwise
$("div").toggleClass("main");
```

```
// test for class existance
 if ($("div").hasClass("main")) { //… }
```

When the DOM is ready…

```
$(document).ready(function(){
  //…
});
```

- **Fires when the document is ready for programming.**

EVENTS DEMO

Events Helpers

```
// attach          / trigger
elem.focus(fn)   / elem.focus()
elem.click(fn)   / elem.click()
elem.change(fn) / elem.change()
```

And many others…

Events Triggering

```
$(“div”).trigger(“click”);
```

Triggers browser’s event action as well.

Showing or Hiding Element

```
// just show
$("div").show();

// reveal slowly, slow=600ms
$("div").show("slow");

// hide fast, fast=200ms
$("div").hide("fast");

// hide or show in 100ms
$("div").toggle(100);
```

Sliding Elements

```
$(“div”).slideUp();
$(“div”).slideDown(“fast”);
$(“div”).slideToggle(1000);
```

Fading Elements

```
$(“div”).fadeIn(“fast”);

$(“div”).fadeOut(“normal”);

// fade to a custom opacity
 $(“div”).fadeTo (“fast”, 0.5);
```

Fading === changing opacity

Custom Animation

```
// .animate(options, duration)
$("div").animate({
                    width: "90%",
                    opacity: 0.5,
                    borderWidth: "5px"
                    }, 1000);
```

Controlling Animations Sync

```
$("div")
    .animate({width: "90%"},
                {queue:true, duration:1000})
    .animate({opacity : 0.5});
```

The first animation will be performed immediately without queuing

jQuery AJAX

```
$("div").load("content.htm");


// passing parameters
 $("#content").load("getcontent.aspx",
                                {"id":"33",
                              "type":"main"});
```

jQuery AJAX

HTTP Request: GET vs. POST

Two commonly used methods for a request-response between a client and server are: GET and POST.

- **GET** - Requests data from a specified resource
- **POST** - Submits data to be processed to a specified resource

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

```
$("button").click(function(){
  $.post("demo_test_post.asp",
  {
    name: "Donald Duck",
    city: "Duckburg"
  },
  function(data, status){
    alert("Data: " + data + "\nStatus: " + status);
  });
});
```

Continue by Examples