JQuery Notes

www.enosislearning.com

Contents

[What is jQuery? 3](#_Toc518980165)

[Using JQuery in Web Applications 4](#_Toc518980166)

[Step1 : Installing JQuery: 4](#_Toc518980167)

[Step2 : Adding Query library 4](#_Toc518980168)

[To use JQuery we need to link jquery into our page. 4](#_Toc518980169)

[Step3: Calling JQuery functions 5](#_Toc518980170)

[Using jQuery Selector 6](#_Toc518980171)

[The $() factory function: 6](#_Toc518980172)

[JQUERY SELECTION ELEMENTS EXAMPLES 7](#_Toc518980173)

[Examples: Selecting Elements in jQuery 7](#_Toc518980174)

[By Using Tag Name 7](#_Toc518980175)

[By Using CSS SELECTORS 7](#_Toc518980176)

[By Using CUSTOM SELECTORS 7](#_Toc518980177)

[Manipulating and Accessing Using CSS Class Names 8](#_Toc518980178)

[Manipulating CSS Styles with jQuery 8](#_Toc518980179)

[Adding, Removing, and Appending Elements and Content 9](#_Toc518980180)

[Dealing with Events in jQuery 10](#_Toc518980181)

[Example : Showing and Hiding Elements with jQuery 10](#_Toc518980182)

[jQuery Animations and Effects 11](#_Toc518980183)

[JQUERY EXAMPLES 12](#_Toc518980184)

[JQUERY EVENTS : CLICK 12](#_Toc518980185)

[JQUERY EVENTS : MOUSEENTER 12](#_Toc518980186)

[JQUERY EVENTS : FOCUS & BLUR 13](#_Toc518980187)

[JQUERY EVENTS: HIDE & SHOW 13](#_Toc518980188)

[JQUERY TOGGLE: 14](#_Toc518980189)

[JQUERY : SLIDEDOWN 14](#_Toc518980190)

[JQUERY : SLIDEDOWN 15](#_Toc518980191)

[JQUERY : ANIMATE 16](#_Toc518980192)

[JQUERY CODE EXAMPLES FOR ASP.NET CONTROLS 16](#_Toc518980193)

[Accessing asp.net elements using JQuery 16](#_Toc518980194)

[Simple Popup Window using jQuery UI Modal Dialog on Button Click 18](#_Toc518980195)

[Anatomy of jQuery 18](#_Toc518980196)

[JQUERY EVENTS 19](#_Toc518980197)

[**EVENTS** 19](#_Toc518980198)

[**DESCRIPTIONS** 19](#_Toc518980199)

[JQUERY VALIDATION TUTORIALS 20](#_Toc518980200)

[VALIDATION USING VALIDATION ENGINE CSS. 24](#_Toc518980201)

[MVC VALIDATION EXAMPLES 28](#_Toc518980202)

[JQUERY CONTROLS 31](#_Toc518980203)

[ToolTips 31](#_Toc518980204)

[**Date Control in JQuery : To display datetime** 32](#_Toc518980205)

[Slider Control in JQuery 32](#_Toc518980206)

[</html> 33](#_Toc518980207)

[Dailog Box in JQuery 33](#_Toc518980208)

[</html> 34](#_Toc518980209)

[**JQUERY DRAGGABLE ELEMENTS** 34](#_Toc518980210)

[JQUERY SERVER-SIDE CALLS 35](#_Toc518980211)

#

# What is jQuery?

**JQuery is a fast and concise JavaScript Library created by John Resig in 2006**

JQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.

jQuery is a JavaScript toolkit designed to simplify various tasks by writing less code. Here is the list of important core features supported by jQuery:

* **Event handling:** The jQuery offers an elegant way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers.
* **Animations:** The jQuery comes with plenty of built-in animation effects which you can use in your websites.
* **Lightweight:** The jQuery is very lightweight library - about 19KB in size ( Minified and gzipped ).
* **Cross Browser Support:** The jQuery has cross-browser support, and works well in IE 6.0+, FF 2.0+, Safari 3.0+, Chrome and Opera 9.0+

# Using JQuery in Web Applications

## Step1 : Installing JQuery:

|  |
| --- |
| This is very simple to do require setup to use jQuery library. You have to carry two simple steps:1. Go to the jquery.com to grab the latest version available.
2. Now put downloaded **jquery-\*.min.js** file in a directory of your website, e.g. /jquery.

The downloaded file name jquery-\*.min.js may vary for your version. Your minified version would be kind of unreadable which would not have any new line or unnecessary words in it.The jQuery does not require any special installation and very similar to JavaScript, we do not need any compilation or build phase to use jQuery. |

## Step2 : Adding Query library

##  To use JQuery we need to link jquery into our page.

Now you can include *jquery* library in your HTML file as follows:

jquery-1.3.2.js : 100kb

jquery-1.3.2.min.js [Minified version] : 30kb

|  |
| --- |
| <html><head><title>The jQuery Example</title><script type="text/javascript" src="/jquery/jquery-1.3.2.min.js"></script><script type="text/javascript"> // you can add our javascript code here </script></head><body></body></html> |

## Step3: Calling JQuery functions

|  |
| --- |
| $(document).ready(function() {alert(‘test’) // do stuff when DOM is ready}); |
| **To call upon any jQuery library function, use HTML script tags as shown below:**<html><head><title>The jQuery Example</title><script type="text/javascript" src="/jquery/jquery-1.3.2.min.js"></script><script type="text/javascript" language="javascript">$(document).ready(function() { $("#btnSave").click(function() { alert($("#txt1").val()); alert("Hello world!"); }); $("button").click(function() {alert("Hello world!"); });});</script></head><body><div id="newdiv">Click on this to see a dialogue box.</div></body></html> |

# Using jQuery Selector

A jQuery Selector is a function which makes use of expressions to find out matching elements from a DOM based on the given criteria.

The $() factory function:

All type of selectors available in jQuery, always start with the dollar sign and parentheses: $().

The factory function $() makes use of following *three building blocks* while selecting elements in a given document:

|  |  |
| --- | --- |
| **JQuery** | **Description** |
| By Tag Name: | Represents a tag name available in the DOM. For example **$('p')**selects all paragraphs in the document. |
| By Tag ID: | Represents a tag available with the given ID in the DOM. For example**$('#some-id')** selects the single element in the document that has an ID of some-id. |
| By Tag Class: | Represents a tag available with the given class in the DOM. For example **$('.some-class')** selects all elements in the document that have a class of some-class.**<span class="myClass">span class="myClass"</span>****<script>****$( ".myClass" ).css( "border", "3px solid red" );****</script>** |

All the above items can be used either on their own or in combination with other selectors. All the jQuery selectors are based on the same principle except some tweaking.

# JQUERY SELECTION ELEMENTS EXAMPLES

# Examples: Selecting Elements in jQuery

The jQuery library allows you to select elements in your HTML by wrapping them in $("") (you could also use single quotes), which is the jQuery wrapper. Here are some examples of “wrapped sets” in jQuery:

## By Using Tag Name

|  |
| --- |
| $("div"); // selects all HTML div elements$("#myElement"); // selects one HTML element with ID "myElement"$(".myClass"); // selects HTML elements with class "myClass"$("p#myElement"); / / selects paragraph elements with ID "myElement"$("ul li a.navigation"); // selects anchors with class "navigation" that are nested in list items |

##

## By Using CSS SELECTORS

## jquery supports the use of all CSS selectors, even those in CSS3. Here are some examples of alternate selectors:

|  |
| --- |
| $("p > a"); // selects anchors that are direct children of paragraphs$("input[type=text]"); // selects inputs that have specified type$("a:first"); // selects the first anchor on the page$("p:odd"); // selects all odd numbered paragraphs$("li:first-child"); // every list item that's first child in a list |

## By Using CUSTOM SELECTORS

## jQuery also allows the use of its own custom selectors. Here are some examples:

|  |
| --- |
| $(":animated"); // selects elements currently being animated$(":button"); // selects any button elements (inputs or buttons)$(":radio"); // selects radio buttons$(":checkbox"); // selects checkboxes$(":checked"); // selects selected checkboxes or radio buttons$(":header"); // selects header elements (h1, h2, h3, etc.) |

## Manipulating and Accessing Using CSS Class Names

jQuery allows you to easily add, remove, and toggle CSS classes, which comes in handy for a variety of practical uses. Here are the different syntaxes for accomplishing this:

|  |
| --- |
| $("div").addClass("content"); // adds class "content" to all <div> elements$("div").removeClass("content"); // removes class "content" from all <div> elements$("div").toggleClass("content"); // toggles the class "content" on all <div> elements (adds it if it doesn't exist, and removes it if it does) |

You can also check to see if a selected element has a particular CSS class, and then run some code if it does. You would check this using an if statement. Here is an example:

if ($("#myElement").hasClass("content"))

{

 // do something here

}

You could also check a set of elements (instead of just one), and the result would return “true” if any one of the elements contained the class.

**Manipulating CSS Styles with jQuery**

CSS styles can be added to elements easily using jQuery, and it’s done in a cross-browser fashion. Here are some examples to demonstrate this:

|  |
| --- |
| $("p").css("width", "400px"); // adds a width to all paragraphs$("#myElement").css("color", "blue") // makes text color blue on element #myElement$("ul").css("border", "solid 1px #ccc") // adds a border to all lists |

#

# Adding, Removing, and Appending Elements and Content

There are a number of ways to manipulate groups of elements with jQuery, including manipulating the content of those elements (whether text, inline elements, etc). Get the HTML of any element (similar to innerHTML in JavaScript):

|  |
| --- |
| var myElementHTML = $("#myElement").html(); // variable contains all HTML (including text) inside #myElementIf you don’t want to access the HTML, but only want the text of an element:var myElementHTML = $("#myElement").text(); // variable contains all text (excluding HTML) inside #myElementUsing similar syntax to the above two examples, you can change the HTML or text content of a specified element:$("#myElement").html("<p>This is the new content.</p>"); // content inside #myElement will be replaced with that specified$("#myElement").text("This is the new content."); // text content will be replaced with that specifiedTo append content to an element:$("#myElement").append("<p>This is the new content.</p>"); // keeps content intact, and adds the new content to the end$("p").append("<p>This is the new content.</p>"); // add the same content to all paragraphsjQuery also offers use of the commands appendTo(), prepend(), prependTo(), before(), insertBefore(), after(), insertAfter(), which work similarly to append but with their own unique characteristics that go beyond the scope of this simple tutorial. |

# Dealing with Events in jQuery

Specific event handlers can be established using the following code:

$("a").click (function () {

 // do something here // when any anchor is clicked

});

The code inside function() will only run when an anchor is clicked. Some other common events you might use in jQuery include:

blur, focus, hover, keydown, load, mousemove, resize, scroll, submit, select.

**Example : Showing and Hiding Elements with jQuery**

## The syntax for showing, hiding an element (or toggling show/hide) is:

|  |
| --- |
| $("#myElement").hide("slow", function() { // do something once the element is hidden}$("#myElement").show("fast", function() { // do something once the element is shown}$("#myElement").toggle(1000, function() { // do something once the element is shown/hidden} |

Remember that the “toggle” command will change whatever state the element currently has, and the parameters are both optional. The first parameter indicates the speed of the showing/hiding. If no speed is set, it will occur instantly, with no animation.

A number for “speed” represents the speed in milliseconds. The second parameter is an optional function that will run when the command is finished executing.

You can also specifically choose to fade an element in or out, which is always done by animation:

$("#myElement").fadeOut("slow", function() {

 // do something when fade out finished

}

$("#myElement").fadeIn("fast", function() {

 // do something when fade in finished

}

To fade an element only partially, either in or out:

$("#myElement").fadeTo(2000, 0.4, function() {

 // do something when fade is finished

}

The second parameter (0.4) represents “opacity”, and is similar to [the way opacity is set in CSS](http://www.impressivewebs.com/css-opacity-reference). Whatever the opacity is to start with, it will animate (fadeTo) until it reaches the setting specified, at the speed set (2000 milliseconds). The optional function (called a “callback function”) will run when the opacity change is complete. This is the way virtually all callback functions in jQuery work.

# jQuery Animations and Effects

You can slide elements, animate elements, and even [stop animations](http://api.jquery.com/stop/) in mid-sequence. To slide elements up or down:

|  |
| --- |
| $("#myElement").slideDown("fast", function() { // do something when slide down is finished}$("#myElement").slideUp("slow", function() { // do something when slide up is finished}$("#myElement").slideToggle(1000, function() { // do something when slide up/down is finished} |

To animate an element, you do so by telling jQuery the CSS styles that the item should change to. jQuery will set the new styles, but instead of setting them instantly (as CSS or raw JavaScript would do), it does so gradually, animating the effect at the chosen speed:

$("#myElement").animate(

 {

 opacity: .3,

 width: "500px",

 height: "700px"

 }, 2000, function() {

 // optional callback after animation completes

 }

);

**JQUERY EXAMPLES**

**JQUERY EVENTS : CLICK**

|  |
| --- |
| <!DOCTYPE html><html><head><script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script><script>$(document).ready(function(){ $("p").click(function(){ $(this).hide(); });});</script></head><body><p>If you click on me, I will disappear.</p><p>Click me away!</p><p>Click me too!</p></body></html> |

## JQUERY EVENTS : MOUSEENTER

|  |
| --- |
| <!DOCTYPE html><html><head><script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script><script>$(document).ready(function(){  $("#p1").mouseenter(function(){alert("You entered p1!"); });});</script></head><body><p id="p1">Enter this paragraph.</p></body></html>In Similar Way the [**jQuerymouseenter()**](http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_mouseenter)[**jQuerymouseleave()**](http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_mouseleave)[**jQuerymousedown()**](http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_mousedown)[**jQuerymouseup()**](http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_mouseup)[**jQuery hover()**](http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_hover) |

## JQUERY EVENTS : FOCUS & BLUR

Color changing on focus

|  |
| --- |
| <!DOCTYPE html><html><head><script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script><script>$(document).ready(function(){ $("input").focus(function(){ $(this).css("background-color","#cccccc"); }); $("input").blur(function(){ $(this).css("background-color","#ffffff"); });});</script></head><body>Name: <input type="text" name="fullname"><br>Email: <input type="text" name="email"></body></html> |

## JQUERY EVENTS: HIDE & SHOW

|  |
| --- |
| <!DOCTYPE html><html><head><script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script><script>$(document).ready(function(){ $("#hide").click(function(){ $("p").hide(); }); $("#show").click(function(){ $("p").show(); });});</script></head><body><p>If you click on the "Hide" button, I will disappear.</p><button id="hide">Hide</button><button id="show">Show</button></body></html> |

## JQUERY TOGGLE:

|  |
| --- |
| <!DOCTYPE html><html><head><script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script><script>$(document).ready(function(){ $("button").click(function(){ $("p").toggle(); });});</script></head><body><button>Toggle</button><p>This is a paragraph with little content.</p><p>This is another small paragraph.</p></body></html> |

## JQUERY : SLIDEDOWN

|  |
| --- |
| <!DOCTYPE html><html><head><script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script><script>$(document).ready(function(){$("#flip").click(function(){ $("#panel").slideDown("slow"); });});</script><style type="text/css">#panel,#flip{padding:5px;text-align:center;background-color:#e5eecc;border:solid 1px #c3c3c3;}#panel{padding:50px;display:none;}</style></head><body><div id="flip">Click to slide down panel</div><div id="panel">Hello world!</div></body></html> |

## JQUERY : SLIDEDOWN

|  |
| --- |
| <!DOCTYPE html><html><head><script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script><script>$(document).ready(function(){$("#flip").click(function(){ $("#panel").slideUp("slow"); });});</script><style type="text/css">#panel,#flip{padding:5px;text-align:center;background-color:#e5eecc;border:solid 1px #c3c3c3;}#panel{padding:50px;}</style></head><body><div id="flip">Click to slide up panel</div><div id="panel">Hello world!</div></body></html> |

## JQUERY : ANIMATE

|  |
| --- |
| <!DOCTYPE html><html><head><script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script><script>$(document).ready(function(){ $("button").click(function(){ $("div").animate({left:'250px'}); });});</script></head><body><button>Start Animation</button><p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p><div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div></body></html> |

## JQUERY CODE EXAMPLES FOR ASP.NET CONTROLS

## Accessing asp.net elements using JQuery

jQuery code example for ASP.NET controls that we use on daily basis. Any ASP.NET control, always use ClientID.

As when Master pages are used then the ID of the ASP.NET controls is changed at run time. With ASP.NET 4.0, this is changed and now you have control over the Client ID.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Get label value:

|  |  |
| --- | --- |
| 1 | $('#<%=Label1.ClientID%>').text(); |

Set label value:

|  |  |
| --- | --- |
| 1 | $('#<%=Label1.ClientID%>').text("New Value"); |

Get Textbox value:

|  |  |
| --- | --- |
| 1 | $('#<%=TextBox1.ClientID%>').val(); |

Set Textbox value:

|  |  |
| --- | --- |
| 1 | $('#<%=TextBox1.ClientID%>').val("New Value"); |

Get Dropdown value:

|  |  |
| --- | --- |
| 1 | $('#<%=DropDownList1.ClientID%>').val(); |

Set Dropdown value:

|  |  |
| --- | --- |
| 1 | $('#<%=DropDownList1.ClientID%>').val("New Value"); |

Get text of selected item in dropdown:

|  |  |
| --- | --- |
| 1 | $('#<%=DropDownList1.ClientID%>option:selected').text(); |

Get Checkbox Status:

|  |  |
| --- | --- |
| 1 | $('#<%=CheckBox1.ClientID%>').attr('checked'); |

Check the Checkbox:

|  |  |
| --- | --- |
| 1 | $('#<%=CheckBox1.ClientID%>').attr('checked',true); |

Uncheck the Checkbox:

|  |  |
| --- | --- |
| 1 | $('#<%=CheckBox1.ClientID%>').attr('checked',false); |

Get Radiobutton Status:

|  |  |
| --- | --- |
| 1 | $('#<%=RadioButton1.ClientID%>').attr('checked'); |

Check the RadioButton:

|  |  |
| --- | --- |
| 1 | $('#<%=RadioButton1.ClientID%>').attr('checked',true); |

Uncheck the RadioButton:

|  |  |
| --- | --- |
| 1 | $('#<%=RadioButton1.ClientID%>').attr('checked',false); |

Disable any control:

|  |  |
| --- | --- |
| 1 | $('#<%=TextBox1.ClientID%>').attr('disabled', true); |

Enable any control:

|  |  |
| --- | --- |
| 1 | $('#<%=TextBox1.ClientID%>').attr('disabled', false); |

Make textbox read only:

|  |  |
| --- | --- |
| 1 | $('#<%=TextBox1.ClientID%>').attr('readonly', 'readonly'); |

 |

# Simple Popup Window using jQuery UI Modal Dialog on Button Click

|  |
| --- |
| <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></script><script src="http://ajax.aspnetcdn.com/ajax/jquery.ui/1.8.9/jquery-ui.js" type="text/javascript"></script><link href="http://ajax.aspnetcdn.com/ajax/jquery.ui/1.8.9/themes/start/jquery-ui.css"rel="stylesheet" type="text/css" /><script type="text/javascript"> $("[id\*=btnPopup]").live("click", function () { $("#dialog").dialog({ title: "jQuery Dialog Popup", buttons: { Close: function () { $(this).dialog('close'); } } }); return false; });</script><div id="dialog" style="display: none"> This is a simple popup</div><asp:Button ID="btnPopup" runat="server" Text="Show Popup" /> |

#

# Anatomy of jQuery

The jQuery architecture revolves around the following areas.



# JQUERY EVENTS

|  |  |
| --- | --- |
| **EVENTS** | **DESCRIPTIONS** |
| [.bind()](https://api.jquery.com/bind/) | Attach a handler to an event for the elements. |
| [.blur()](https://api.jquery.com/blur/) | Bind an event handler to the “blur” JavaScript event, or trigger that event on an element. |
| [.change()](https://api.jquery.com/change/) | Bind an event handler to the “change” JavaScript event, or trigger that event on an element. |
| [.click()](https://api.jquery.com/click/) | Bind an event handler to the “click” JavaScript event, or trigger that event on an element. |
| [.contextmenu()](https://api.jquery.com/contextmenu/) | Bind an event handler to the “contextmenu” JavaScript event, or trigger that event on an element. |
| [.dblclick()](https://api.jquery.com/dblclick/) | Bind an event handler to the “dblclick” JavaScript event, or trigger that event on an element. |
| [.delegate()](https://api.jquery.com/delegate/) | Attach a handler to one or more events for all elements that match the selector, now or in the future, based on a specific set of root elements. |
| [.die()](https://api.jquery.com/die/) | Remove event handlers previously attached using .live() from the elements. |
| [.error()](https://api.jquery.com/error/) | Bind an event handler to the “error” JavaScript event. |
| .focus() | Bind an event handler to the “focus” JavaScript event, or trigger that event on an element. |
| .focusin() | Bind an event handler to the “focusin” event. |
| .focusout() | Bind an event handler to the “focusout” JavaScript event. |
| .hover() | Bind one or two handlers to the matched elements, to be executed when the mouse pointer enters and leaves the elements. |
| jQuery.proxy() | Takes a function and returns a new one that will always have a particular context. |
| .keydown() | Bind an event handler to the “keydown” JavaScript event, or trigger that event on an element. |
| .keypress() | Bind an event handler to the “keypress” JavaScript event, or trigger that event on an element. |
| .keyup() | Bind an event handler to the “keyup” JavaScript event, or trigger that event on an element. |
| .live() | Attach an event handler for all elements which match the current selector, now and in the future. |
| .load() | Bind an event handler to the “load” JavaScript event. |
| .mousedown() | Bind an event handler to the “mousedown” JavaScript event, or trigger that event on an element. |
| .mouseenter() | Bind an event handler to be fired when the mouse enters an element, or trigger that handler on an element. |
| .mouseleave() | Bind an event handler to be fired when the mouse leaves an element, or trigger that handler on an element. |
| .mousemove() | Bind an event handler to the “mousemove” JavaScript event, or trigger that event on an element. |
| .mouseout() | Bind an event handler to the “mouseout” JavaScript event, or trigger that event on an element. |
| .mouseover() | Bind an event handler to the “mouseover” JavaScript event, or trigger that event on an element. |
| .mouseup() | Bind an event handler to the “mouseup” JavaScript event, or trigger that event on an element. |
| .off() | Remove an event handler. |
| .on() | Attach an event handler function for one or more events to the selected elements. |
| .one() | Attach a handler to an event for the elements. The handler is executed at most once per element per event type. |
| .ready() | Specify a function to execute when the DOM is fully loaded. |
| .resize() | Bind an event handler to the “resize” JavaScript event, or trigger that event on an element. |
| .scroll() | Bind an event handler to the “scroll” JavaScript event, or trigger that event on an element. |
| .select() | Bind an event handler to the “select” JavaScript event, or trigger that event on an element. |
| .submit() | Bind an event handler to the “submit” JavaScript event, or trigger that event on an element. |
| .toggle() | Bind two or more handlers to the matched elements, to be executed on alternate clicks. |
| .trigger() | Execute all handlers and behaviors attached to the matched elements for the given event type. |
| .triggerHandler() | Execute all handlers attached to an element for an event. |
| .unbind() | Remove a previously-attached event handler from the elements. |
| .undelegate() | Remove a handler from the event for all elements which match the current selector, based upon a specific set of root elements. |
| .unload() | Bind an event handler to the “unload” JavaScript event. |

# JQUERY VALIDATION TUTORIALS

These examples will demonstrate how to create simple form in asp.net and adding validations using jquery.

**Syntax :**

$(document).ready(function ()
 {
     $('#btnSave').click(function ()
         {
 })

});

In the preceding syntax, the function is the keyword provided by the jQuery to declare a function followed by a "$" (dollar) symbol and the "$('#btnSave').click" means that this function is called on the ASP.Net button click that has the id "btnSave".

**Example**

$(document).ready(function ()
 {
     $('#btnSave').click(function ()
         {
 alert("button is clicked")
 })
  });

**The source below is used to design the page**
The first page source code <body> tag will look as in the following:

|  |
| --- |
|         <bodybgcolor="#3366ff">            <form id="form2" runat="server">            <div>                <table><tr>                         <td>Name</td>                        <td><asp:TextBox ID="txtUserId" runat="server"></asp:TextBox></td>                    </tr>                    <tr>                        <td>Email Id </td>                        <td> <asp:TextBoxID="txtmail"runat="server"></asp:TextBox></td>                    </tr>                    <tr>                        <td>Gender</td>                        <td>                            <asp:DropDownListID="ddlType"runat="server">                                <asp:ListItemValue="0">-Select-</asp:ListItem>                                <asp:ListItemValue="1">Male</asp:ListItem>                                <asp:ListItemValue="2">Female</asp:ListItem>                            </asp:DropDownList>                        </td>                    </tr>                    <tr><td> **word**</td>                        <td><asp:TextBoxID="txt1"runat="server"TextMode="word"></asp:TextBox>                     </td>                    </tr>                    <tr>                        <td>Confirm word</td>                        <td><asp:TextBoxID="txt2"runat="server"TextMode="word"></asp:TextBox>                        </td>                    </tr>                    <tr>                        <td>                        </td>                        <td>                            <asp:ButtonID="btnSave"runat="server"Text="Create"/>                            <asp:ButtonID="Button1"runat="server"Text="Reset"/>                        </td>                    </tr>                </table>                <asp:LabelID="Label1"runat="server"Text="Label"></asp:Label>            </div>            </form>        </body> |

Look at the preceding source code closely; see the ids of the controls that play the important role in JavaScript validation by reading the ASP.Net control values in JavaScript code.
The design view of the preceding source code will look as in the following:



After creating the form in asp.net now write the validation code in the script tag using jquery

Reading ASP.Net Control Values in jQuery

|  |
| --- |
| $(document).ready(function ()  {     $('#btnSave').click(function ()         {    Var Name,word , gender, con, EmailId, emailExp;    Name = $("#txtUserId").val();    gender = $("#ddlType").val();    word= $("#txt1").val();    con = $("#txt2").val();    EmailId = $("#txtmail").val();    emailExp = /^([a-zA-Z0-9\_\.\-])+\@(([a-zA-Z0-9\-])+\.)+([com\co\.\in])+$/;  if (Name == ''&& gender == 0 && word== ''&& con == ''&&EmailId == '') {    alert("Enter All Fields");    return false;}if (Name == '') {    alert("Please Enter Login ID");    return false;}if (gender == 0) {   alert("Please Select gender");   return false;}if (word == '') {     alert("Please Enter word");     return false;}if ( word != ''&& con == '') {    alert("Please Confirm word");    return false;}if (word != con){    alert("word not match");    return false;}if (EmailId == '') {    alert("Email Id Is Required");    return false;}if (EmailId != '') {    if (!EmailId.match(emailExp))    {        alert("Invalid Email Id");        return false;     }  }  return true;  }) }); |

**Importance of returning false**It's very important to use the return false statement after the condition block that returns false so if validation determines that the business requirements are not met then the form cannot be submitted the condition that checks both text boxes for the word.

**VALIDATION USING VALIDATION ENGINE CSS.**

In JQuery we can us inbuilt validation css & functions for doing validation in asp.net



|  |
| --- |
| <%@ Page Language="C#" AutoEventWireup="true" CodeFile="JQuery\_ValidateCustomer.aspx.cs" Inherits="ValidateCustomer" %><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><html xmlns="http://www.w3.org/1999/xhtml"><head runat="server"><title>Show Inline validation Messages</title><link href="css/template.css" rel="stylesheet" type="text/css" /><link href="css/validationEngine.jquery.css" rel="stylesheet" type="text/css" /><script src="js/jquery-1.6.min.js" type="text/javascript"></script><script src="js/jquery.validationEngine-en.js" type="text/javascript" charset="utf-8"></script><script src="js/jquery.validationEngine.js" type="text/javascript" charset="utf-8"></script><script type="text/javascript"> jQuery(document).ready(function () { jQuery("#form1").validationEngine();});</script></head><body><form id="form1" runat="server"><table align="center"><tr><td colspan="2"><div style="border: 1px solid #CCCCCC; padding: 10px;"><table cellpadding="0" cellspacing="30" style=" background-color:White"><tr><td>First Name:</td><td><asp:TextBox ID="txtfname" runat="server" CssClass="validate[required]"/></td></tr><tr><td>Last Name:</td><td><asp:TextBox ID="txtlname" runat="server" CssClass="validate[required]"/></td></tr><tr ><td>Email:</td><td><asp:TextBox ID="txtemail" runat="server" CssClass="validate[required,custom[email]" /></td></tr><tr ><td>Url:</td><td><asp:TextBox ID="txtUrl" runat="server" CssClass="validate[required,custom[url]] text-input" /></td></tr><tr><td valign="top">Address:</td><td><asp:TextBox ID="txtaddress" runat="server" TextMode="MultiLine" Rows="8" Columns="26"/></td></tr><tr><td>State:</td><td><asp:DropDownList ID="ddlState" runat="server" CssClass="validate[required] radio"><asp:ListItem value="">Choose State</asp:ListItem><asp:ListItem Value="AL">Alabama</asp:ListItem><asp:ListItem value="AK">Alaska</asp:ListItem><asp:ListItem value="AL">Alabama </asp:ListItem><asp:ListItem value="AK">Alaska</asp:ListItem><asp:ListItem value="AZ">Arizona</asp:ListItem><asp:ListItem value="AR">Arkansas</asp:ListItem><asp:ListItem value="CA">California</asp:ListItem><asp:ListItem value="CO">Colorado</asp:ListItem><asp:ListItem value="CT">Connecticut</asp:ListItem><asp:ListItem value="DE">Delaware</asp:ListItem><asp:ListItem value="FL">Florida</asp:ListItem><asp:ListItem value="GA">Georgia</asp:ListItem><asp:ListItem value="HI">Hawaii</asp:ListItem><asp:ListItem value="ID">Idaho</asp:ListItem></asp:DropDownList></td></tr><tr><td>Zip:</td><td><asp:TextBox ID="txtZip" runat="server" CssClass="validate[required,custom[integer]] text-input"/></td></tr><tr><td> I Agree Conditions</td><td><input class="validate[required] checkbox" type="checkbox" id="agree" name="agree"/></td></tr><tr><td></td><td><asp:Button ID="btnSubmit" runat="server" Text="Submit" onclick="btnSubmit\_Click" /></td></tr><tr><td colspan="2"><asp:Label ID="lblResult" runat="server" Font-Bold="true"/></td></tr></table></div></td></tr></table></form></body></html> |

# MVC VALIDATION EXAMPLES

@model TestApp.Models.LoginModel

<script src="../../Scripts/jquery-1.7.1.min.js" type="text/javascript"></script>

<script src="../../Scripts/jquery.validate.js" type="text/javascript"></script>

<script>

$(document).ready(function(){

 $("#loginForm").validate();

});

</script>

<form id="loginForm">

<fieldset>

<ol>

<li>

@Html.LabelFor(m =>m.UserName)

@Html.TextBoxFor(m =>m.UserName,new{@class="required"})

</li>

<li>

@Html.LabelFor(m =>m.Password)

@Html.PasswordFor(m =>m.Password,new{@class="required"})

</li>

</ol>

<input type="submit" value="Submit"/>

</fieldset>

</form>

Below after few screenshots I have taken before and after the form submit.

After hitting the submit button



jquery.validate.js, that you should check out.

|  |
| --- |
| <head> <meta name="viewport" content="width=device-width" /> <title>LoginNew</title> <script src="~/Scripts/jquery-1.7.1.min.js"></script> <script src="~/Scripts/jquery.validate.min.js"></script> @\*<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>\*@ <script>  $(document).ready(function () {  $('#registerForm').validate({  messages: {  UserName: {  required: "Eh ? empty username ?",  minLength: "At least 4 characters are necessary"  },  Password: {  required: "password missing"  },  Email: {  required: "please provide us your email, we promise we wont spam !",  email: "email id is not in correct format"  }  }  });  });  </script></head><body> @using (Html.BeginForm(null, null, FormMethod.Get, new { @id = "registerForm" })) { @Html.ValidationSummary(true)  <fieldset> <legend>UserModel</legend> <div class="editor-label"> @Html.LabelFor(model => model.UserName) </div> <div class="editor-field"> @Html.TextBoxFor(model => model.UserName, new { @class = "required", minLength = "4"}) @Html.ValidationMessageFor(model => model.UserName) </div> <div class="editor-label"> @Html.LabelFor(model => model.Password) </div> <div class="editor-field"> @Html.TextBoxFor(model => model.Password, new { @class = "required", minLength = "4" }) @Html.ValidationMessageFor(model => model.Password) </div>  <div class="editor-label"> @Html.LabelFor(model => model.Email) </div> <div class="editor-field"> @Html.TextBoxFor(model => model.Email, new { @class = "required email" }) @Html.ValidationMessageFor(model => model.Email) </div> <p> <input type="submit" value="Create" /> </p> </fieldset> } <div> @Html.ActionLink("Back to List", "Index") </div></body></html> |

**Screengrab**



## JQUERY CONTROLS

## ToolTips

## [Tooltips](https://jqueryui.com/resources/demos/tooltip/default.html) can be attached to any element. When you hover the element with your mouse, the title attribute is displayed in a little box next to the element, just like a native tooltip.

|  |
| --- |
| <!DOCTYPE html><html xmlns="http://www.w3.org/1999/xhtml"><head> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1"> <title>jQuery UI Tooltip - Default functionality</title> <link rel="stylesheet" href="ToolTips/jquery-ui.css" /> <script src="ToolTips/jquery-1.12.4.js"></script> <script src="ToolTips/jquery-ui.js"></script> <script> $(function () { $(document).tooltip(); }); </script> <style> label { display: inline-block; width: 5em; } </style></head><body><p><label for="age">Your age:</label><input id="age" title="We ask for your age only for statistical purposes."></p><p>Hover the field to see the tooltip.</p> <p><label for="Name">Your Name:</label><input id="tName" title="Please enter your Name only for statistical purposes."></p><p>Hover the field to see the tooltip.</p></body></html> |
| Date Control in JQuery : To display datetime<html lang=”en”><head> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1"> <title>jQuery UI Datepicker - Default functionality</title> <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css"> <link rel="stylesheet" href="/resources/demos/style.css"> <script src="https://code.jquery.com/jquery-1.12.4.js"></script> <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script> <script> $(function () { $("#datepicker").datepicker(); }); </script></head><body> <p>Date: <input type="text" id="datepicker"></p> </body></html> |

## Slider Control in JQuery

|  |
| --- |
| <!doctype html><html lang="en"><head> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1"> <title>jQuery UI Slider - Default functionality</title> <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css"> <link rel="stylesheet" href="/resources/demos/style.css"> <script src="https://code.jquery.com/jquery-1.12.4.js"></script> <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script> <script> $(function () { $("#slider").slider(); }); </script></head><body><div id="slider"></div></body></html> |

## Dailog Box in JQuery

|  |
| --- |
| <html lang="en"><head> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1"> <title>jQuery UI Dialog - Default functionality</title> <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css"> <link rel="stylesheet" href="/resources/demos/style.css"> <script src="https://code.jquery.com/jquery-1.12.4.js"></script> <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script> <script> $(function () { $("#dialog").dialog(); }); </script></head><body> <div id="dialog" title="Basic dialog"> <p>This is the default dialog which is useful for displaying information. The dialog window can be moved, resized and closed with the 'x' icon.</p></div> </body></html> |
| **JQUERY DRAGGABLE ELEMENTS**<html xmlns="http://www.w3.org/1999/xhtml"><head runat="server"> <style> #contentArea{width: 300px;height: 100%;} .panel{border:1px solid;border-color:Gray;width:200px;height:100px;padding:5px;} </style> <title></title> <link rel="stylesheet" href="ToolTips/jquery-ui.css" /> <script src="ToolTips/jquery-1.12.4.js"></script> <script src="ToolTips/jquery-ui.js"></script>  <script language="javascript" type="text/javascript"> $(document).ready(function () { $(".panel").draggable(); }); </script></head><body> <form id="form1" runat="server"> <div align="center" id="contentArea"> <panel ID="Panel11" runat="server" class="panel">Lorem ipsum dolor sit amet, consectetur adipisicing.</panel><panel ID="Panel12" runat="server" class="panel">Ut enim ad minim veniam, quis nostrud.</panel><panel ID="Panel13" runat="server" class="panel">Duis aute irure dolor in reprehenderit.</panel><panel ID="Panel14" runat="server" class="panel">Excepteur sint occaecat cupidatat non proident</panel></div></form></body></html> |

## JQUERY SERVER-SIDE CALLS

jQuery allows you to call Server Side ASP.net methods from client side without any PostBack. Actually it is an AJAX call to the server but it allows us to call the method or function defined server side.

The following picture describes the syntax of the jQuery AJAX call.



 **HTML Markup**

The following HTML Markup consists of an ASP.Net TextBox and an HTML Button.

<div>

Your Name :

<asp:TextBox ID="txtUserName" runat="server"></asp:TextBox>

<input id="btnGetTime" type="button" value="Show Current Time"

    onclick = "ShowCurrentTime()" />

</div>

 **Client Side Script**

When the Button is clicked the ShowCurrentTime JavaScript function is executed which makes an AJAX call to the GetCurrentTime WebMethod. The value of the TextBox is passed as parameter to the WebMethod.

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js" type="text/javascript"></script>

<script type = "text/javascript">

function ShowCurrentTime() {

    $.ajax({

        type: "POST",

        url: "CS.aspx/GetCurrentTime",

        data: '{name: "' + $("#<%=txtUserName.ClientID%>")[0].value + '" }',

        contentType: "application/json; charset=utf-8",

        dataType: "json",

        success: OnSuccess,

        failure: function(response) {

            alert(response.d);

        }

    });

}

function OnSuccess(response) {

    alert(response.d);

}

</script>

**The WebMethod**

The following WebMethod returns a greeting message to the user along with the current server time. An important thing to note is that the method is declared as static (C#) and Shared (VB.Net) and is decorated with WebMethod attribute, this is necessary otherwise the method will not be called from client side jQuery AJAX call.

**C#**

[System.Web.Services.WebMethod]

public static string GetCurrentTime(string name)

{    return "Hello " + name + Environment.NewLine + "The Current Time is: "

       + DateTime.Now.ToString();}