JQuery

Notes by rajeeshvengalapurath@gmail.com

Introduction

JQuery

- Cross browser
- Extensible
- AJAX support
- Excellent documentation

Selectors list

https://api.jquery.com/category/selectors/

Sample

JavaScript

```
<script type="text/javascript">
window.onload = function(){
       if(document.addEventListener){ //IE8 or above

document.getElementById('JSButton').addEventListener('click',clickHandler,false)
//3rd parameter is rarelt used
       }
       else{
              document.getElementById('JSButton').attachEvent('onclick',
clickHandler)
       }
}

function clickHandler(){
       alert('JQuery Tutorial');
}
</script>
<input type="button" id="JSButton" value="jsButton">
```

JQuery

```
<script src="jquery-3.3.1.min.js"></script>
<script type="text/javascript">
              $(document).ready(function(){
              $('#JQueryButton').click(function(){
                     alert('JQuery Tutorial');
              });
       })
</script>
<input type="button" id="JQueryButton" value="JQueryButton">

</body>
```

Points

1.ready() function ensures that the DOM is fully loaded
2.All three of the following syntax are equal

- `$(document).ready(handler)`
- `$().ready(handler)`
- `$(handler)`

$(document).ready and $(window).load

$(document).ready is a jQuery event. It fires as soon as the DOM is loaded and ready to be manipulated by script. This is the earliest point in the page load process where the script can safely access elements in the page's html dom. This event is fired before all the images, css etc.. are fully loaded.

The following example works, because the jquery code that adds the event handler to the button is inside the ready() function, which ensures that the DOM is fully loaded before this piece of code is executed, so the JavaScript can find the button element in the DOM and adds the click event hanlder.

```
<html>
<head>
    <title></title>
    <script src="Scripts/jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('#button1').click(function () {
                alert('jQuery Tuorial');
            });
        });
    </script>
</head>
<body>
    <input id="button1" type="button" value="Click Me" />
</body>
</html>
```

In the following example, we have removed the ready() method. When you click the button now, you don't get the alert. This is because the jQuery code is present before the button element, so by the time the jQuery code is executed the button element is not loaded into DOM.

```
<html>
<head>
    <title></title>
    <script src="Scripts/jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $('#button1').click(function () {
            alert('jQuery Tuorial');
        });
    </script>
</head>
<body>
    <input id="button1" type="button" value="Click Me" />
</body>
</html>
```

To make this example work, you have 2 options
1. Place your jQuery code in $(document).ready function OR
2. Place your script at the bottom of the page just before the closing </body> element

$(window).load event fires when the DOM and all the content on the page (images, css etc) is fully loaded. Since the window load event waits for images, css etc to be fully loaded, this event fires after ready event.

The following example proves the above point. When you run the page with the following script, notice that the alert in ready function is displayed before the alert in load function.

```
<script type="text/javascript">
    $(window).load(function () {
        alert('Window loaded');
    });

    $(document).ready(function () {
        alert('DOM Loaded and ready');
    });
</script>
```

In most cases, the script can be run as soon as the DOM hierarchy has been fully constructed. So ready() is usually the best place to write your JavaScript code.

However, in your application there could be scenarios where you should be using $(window).load over $(document).ready. For example, let's say we want to display the actual image dimensions (Height and Width). To get the actual image dimensions, we will have to wait until the image is fully loded, so the jQuery code to get the height and width should be in $(window).load event.

Example : If you use $(document).ready() instead of $(window).load() the height and width will be displayed as 0.

```
<html>
<head>
    <title></title>
    <script src="Scripts/jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(window).load(function (){
            $('#div1').html("Height = " + $('#Image1').height()
                + "<br/>" + "Width = " + $('#Image1').width())
        });
    </script>
```

```
</head>
<body>
    <div id="div1"></div>
    <img src="Chrysanthemum.jpg" id="Image1" />
</body>
</html>
```

CDN

What is a CDN
CDN stands for Content Delivery Network. A CDN is a system of distributed servers that hosts resources such as images, CSS, JavaScript files etc.

Companies like Microsoft, Google, Yahoo etc have a free public CDN from which we can load jQuery instead of hosting it on our own web server.

Advantages of using a CDN
1. Distributed CDN servers : The jQuery file can be downloaded from the CDN server that is closest to the user

2. Browser Caching : jQuery is used on many popular websites. If a user has already visited a webpage that uses jQuery from a CDN, and then if he arrives at your page, the jQuery file has already been cached by the browser so there is no need to download it again.

3. Parallel Downloads : There is a browser limit on how many files can be concurrently downloaded from a given domain. This number varies from browser to browser. For example, if the browser allows only 2 concurrent downloads from a given domain, the 3rd download is blocked until one of the previous files has been fully downloaded. Since the jQuery file is on a CDN, it is being downloaded from a different domain. So this means the browser allows another 2 parallel downloads from the CDN server.

4. Reduced server load : The HTTP request for jQuery file is handled by the CDN server, so the load on your web server is reduced. This also means there is a saving on your website bandwidth consumption which in turn will reduce your hosting cost.

Disadvantages of using a CDN
Your clients may block the CDN. So you may have to request your clients to whitelist the CDN.

What if the required jQuery file cannot be downloaded from CDN
Let assume that, the CDN is down or because of some network issue we are not able to download jQuery from CDN. In this case we will have to fallback to use jQuery file that we hosted on our own server.

Here is the code that falls back to use jQuery on your web server, if it can't be downloaded from CDN. If jQuery is successfully downloaded, jQuery property is added to the window object. If this property is not found then jQuery is not downloaded. So in this case we are writing a script tag to fallback to the local jQuery file.

```
<script src="http://ajax.aspnetcdn.com/ajax/jQuery...">
</script>

<script>
    window.jQuery || document.write("<script
                                        src='Scripts/jquery-2.1.3.js'><\/script>");
</script>
```

jQuery Selectors

What are jQuery selectors

One of the most important concept in jQuery is selectors. jQuery selectors allow you to select and manipulate HTML elements.

Different selectors in jQuery
jQuery selectors allow you to select html elements in the DOM by

1. Element ID
2. Element Tag Name
3. Element Class Name
4. Element attribute
5. Element Attribute values and many more

Id selector in jquery
To find an HTML element by ID, use the jQuery #id selector

Example : The following example finds button with ID button1 and attaches the click event handler.

```
<html>
<head>
    <title></title>
    <script src="jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('#button1').click(function () {
                alert('jQuery Tuorial');
            });
        });
    </script>
</head>
<body>
    <input id="button1" type="button" value="Click Me" />
</body>
</html>
```

Changes the background colour of the button to yellow

```
$(document).ready(function () {
    $('#button1').css('background-color', 'yellow');
});
```

Important points to remember about jQuery #id selector

1. jQuery #id selector uses the JavaScript document.getElementById() function
2. jQuery #id selector is the most efficient among all jQuery selectors. If you know the id of an element that you want to find, then always use the #id selector.
3. HTML element IDs must be unique on the page. jQuery #id selector returns only the first element, if you have 2 or more elements with the same ID.

```
<html>
<head>
    <title></title>
    <script src="jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('#button1').css('background-Color', 'yellow');
        });
    </script>
</head>
<body>
    <input id="button1" type="button" value="Click Me" />
    <input id="button1" type="button" value="Click Button" />
</body>
</html>
```

4. JavaScript's document.getElementById() function throws an error if the element with the given id is not found, where as jQuery #id selector will not throw an error. To check if an element is returned by the #id selector use length property.

```
<html>
<head>
    <title></title>
    <script src="jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            if ($('#button1').length > 0) {
                alert('Element found')
            }
            else {
                alert('Element not found')
            }
        });
    </script>
</head>
<body>
```

```
    <input id="button1" type="button" value="Click Me" />
</body>
</html>
```

5. JavaScript's document.getElementById() and jQuery(#id) selector are not the same. document.getElementById() returns a raw DOM object where as jQuery('#id') selector returns a jQuery object that wraps the DOM object and provides jQuery methods. This is the reason you are able to call jQuery methods like css(), click() on the object returned by jQuery. To get the underlying DOM object from a jQuery object write $('#id')<0>

6. document.getElementById() is faster than jQuery('#id') selector. Use document.getElementById() over jQuery('#id') selector unless you need the extra functionality provided by the jQuery object.

jQuery Element Selector

To select the elements by tag name use jQuery Element Selector

Syntax : $(element)

- $('td') // Selects all td elements
- $('div a') // Select all anchor elements that are descendants of div element
- $('div, span, a') // Selects all div, span and anchor elements

Alerts the total count of td elements on the page

```
<script type="text/javascript">
    $(document).ready(function () {
        alert($('td').length);
    });
</script>
```

Selects all the tr elements on the page and changes their background colour to red

```
<script type="text/javascript">
    $(document).ready(function () {
        $('tr').css('background-Color', 'red');
    });
</script>
```

Alerts the HTML content of the table

```
<script type="text/javascript">
    $(document).ready(function () {
        alert($('table').html());
    });
</script>
```

Alerts the HTML content of each table row

```
<script type="text/javascript">
    $(document).ready(function () {
        $('table tr').each(function () {
            alert($(this).html());
        });
    });
</script>
```

Select and changes the background colour of all the div, span and anchor elements

```
<script type="text/javascript">
    $(document).ready(function () {
        $('div, span, a').css('background-Color', 'yellow');
    });
</script>
```

Select all anchor elements that are descendants of div element

```
<html>
<head>
    <title></title>
    <script src="jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('div a').css('background-Color', 'yellow');
        });
    </script>
</head>
<body>
    <div>
        <a href="http://pragimtech.com">PragimTech</a>
    </div>
    <br />
    <a href="http://microsoft.com">Microsoft</a>
</body>
</html>
```

Changes the background color of even rows to gray and odd rows to yellow in both the tables.

```
<script type="text/javascript">
    $(document).ready(function () {
        $('tr:even').css('background-Color', 'gray');
        $('tr:odd').css('background-Color', 'yellow');
    });
</script>
```

To change the background color of even rows to gray and odd rows to yellow just for one of the table, use #id selector along with element selector.

```
<script type="text/javascript">
```

```
    $(document).ready(function () {
        $('#table1 tr:even').css('background-Color', 'gray');
        $('#table1 tr:odd').css('background-Color', 'yellow');
    });
</script>
```

jQuery class selector

i.e selecting elements using their class name

```
Syntax : $('.class')
```

jQuery class selectors uses JavaScript's native getElementsByClassName() function if the browser supports it.

```
$('.small') // Selects all elements with class small
$('.small,.big') // Selects all elements with class small or big
$('div.small,.big') // Selects div elements with class small and any element with
class big
```

Selects all elements with class "small" and sets 5px solid red border

```
<html>
<head>
    <title></title>
    <script src="Scripts/jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('.small').css('border', '5px solid red');
        });
    </script>
</head>
<body>
    <span class="small">
        Span 1
    </span>
    <br /><br />
    <div class="small">
        Div 1
    </div>
    <br />
    <span class="big">
        Span 2
    </span>
    <p class="big">This is a paragraph</p>
</body>
</html>
```

Selects all elements with class "small" or "big" and sets 5px solid red border

```
<script type="text/javascript">
    $(document).ready(function () {
        $('.small, .big').css('border', '5px solid red');
    });
</script>
```

Selects all elements with class "small" and all span elements with class "big" and sets 5px solid red border

```
<script type="text/javascript">
    $(document).ready(function () {
        $('.small, span.big').css('border', '5px solid red');
    });
</script>
```

Selects all elements with class small that are nested in a an element with id=div2 and sets 5px solid red border

```
<html>
<head>
    <title></title>
    <script src="Scripts/jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('#div2 .small').css('border', '5px solid red');
        });
    </script>
</head>
<body>
    <div id="div1" class="small">
        DIV 1
    </div>
    <br />
    <div id="div2">
        Div 2
        <br />
        <div class="small">
            DIV 3
        </div>
        <br />
        <span class="small">
            SPAN
        </span>
    </div>
</body>
</html>
```

Selects all elements with class small and sets 5px solid red border. Notice div1 has 2 classes - small and big.

```
<html>
<head>
    <title></title>
    <script src="Scripts/jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('.small').css('border', '5px solid red');
        });
    </script>
</head>
<body>
    <div class="small big">
        DIV 1
    </div>
    <br />
    <div class="small">
        DIV 2
    </div>
</body>
</html>
```

Selects all elements that has both the classes - small and big. There should be no space between the class names.

```
<script type="text/javascript">
    $(document).ready(function () {
        $('.small.big').css('border', '5px solid red');
    });
</script>
```

If you have a space between the two class names then we are trying to find descendants, i.e. find elements with class big that are descendants of an element with class small.

```
<html>
<head>
    <title></title>
    <script src="Scripts/jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('.small .big').css('border', '5px solid red');
        });
    </script>
</head>
<body>
    <div class="small big">
        DIV 1
    </div>
    <br />
    <div class="small">
        DIV 2
        <div class="big">
            DIV 3
        </div>
    </div>
</body>
</html>
```

Another way to selects all elements that has both the classes - small and big is by using filter method. But this approach is slower because it will first create a list of objects with class "small" and then removes elements that does not have class "big"

```
<script type="text/javascript">
    $(document).ready(function () {
        $('.small').filter('.big').css('border', '5px solid red');
    });
</script>
```

jQuery attribute selector

1. That have specified attribute
2. That have specified attribute values

Syntax :

$('<attribute>')
$('<attribute="value">')

$('<title>') // Selects all elements that have title attribute
$('div<title>') // Selects all div elements that have title attribute
$('<title="divTitle">') // Selects all elements that have title attribute value - divTitle
$('div<title="divTitle">') // Selects all div elements that have title attribute value - divTitle

Selects all elements with title attribute and sets 5px solid red border

```
<html>
<head>
    <title></title>
    <script src="Scripts/jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('<title>').css('border', '5px solid red');
        });
    </script>
</head>
<body>
    <div title="div1Title">
        DIV 1
    </div>
    <br />
    <div title="div2Title">
        DIV 2
    </div>
    <p title="pTitle">
        This is a paragraph
    </p>
    <span title="div1Title">
        SAPN 1
    </span>
    <br /><br />
    <span>
        SPAN 2
    </span>
</body>
</html>

Selects all div elements with title attribute and sets 5px solid red border
<script type="text/javascript">
    $(document).ready(function () {
        $('div<title>').css('border', '5px solid red');
    });
</script>
```

Selects all elements with title attribute value - div1Title, and sets 5px solid red border

```
<script type="text/javascript">
    $(document).ready(function () {
        $('<title="div1Title">').css('border', '5px solid red');
    });
</script>

Selects all div elements with title attribute value - div1Title, and sets 5px solid
red border
<script type="text/javascript">
    $(document).ready(function () {
        $('div<title="div1Title">').css('border', '5px solid red');
    });
</script>
```

Selects all div elements with both title and style attributes, and sets 5px solid black border

```
<html>
<head>
    <title></title>
    <script src="Scripts/jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('div<title><style>').css('border', '5px solid black');
        });
    </script>
</head>
<body>
    <div title="divTitle" style="background-color:red">
        Red DIV
    </div>
    <br />
    <div title="divTitle" style="background-color:green">
        Green DIV
    </div>
    <br />
    <div title="divTitle">
        Normal Div
    </div>
    <br />
    <div>
        Normal Div without any attributes
    </div>
</body>
</html>
```

Selects all div elements with title attribute value - divTitle, and style attribute value - background-color:red, and sets 5px solid black border

```
<script type="text/javascript">
    $(document).ready(function () {
        $('div<title="divTitle"><style="background-color:red">').css('border', '5px
solid black');
    });
</script>
```

Selects all div elements with either title or style attributes, and sets 5px solid black border

```
<script type="text/javascript">
    $(document).ready(function () {
        $('div<title>,<style>').css('border', '5px solid black');
    });
</script>
```

jQuery attribute value selector

Attribute Equals Selector <name="value">
Attribute Not Equal Selector <name!="value">
Attribute Contains Selector <name*="value">
Attribute Contains Word Selector <name~="value">
Attribute Contains Prefix Selector <name|="value">
Attribute Starts With Selector <name^="value">
Attribute Ends With Selector <name$="value">

$('<title="div1Title">') // Selects all elements that have title attribute value equal to div1Title
$('<title!="div1Title">') // Selects all elements that have title attribute value not equal to div1Title
$('<title*="Title">') // Selects all elements that have title attribute value containing the given substring - Title
$('<title~="mySpan">') // Selects all elements that have title attribute value containing the given word - mySpan, delimited by spaces
$('<title|="myTitle">') // Selects all elements that have title attribute value equal to myTitle or starting with myTitle followed by a hyphen (-)
$('<title^="div">') // Selects all elements that have title attribute value starting with div
$('<title$="Heading">') // Selects all elements that have title attribute value ending with Heading

Selects all elements that have title attribute value equal to div1Title and sets 5px solid red border

```
<html>
<head>
    <title></title>
    <script src="Scripts/jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('<title="div1Title">').css('border', '5px solid red');
        });
    </script>
</head>
<body>
    <div title="div1Title">
        DIV 1
    </div>
    <br />
    <div title="div2Title">
        DIV 2
    </div>
    <p title="myTitle-paragraph">
        This is a paragraph
    </p>
    <p title="myTitleHeading">
        This is a paragraph
    </p>
    <span title="div1Title">
        SAPN 1
    </span>
    <br /><br />
    <span title="mySpan Heading">
        SPAN 2
    </span>
</body>
</html>
```

Selects all div elements that have title attribute value not equal to div1Title and sets 5px solid red border

```
<script type="text/javascript">
    $(document).ready(function () {
        $('div<title!="div1Title">').css('border', '5px solid red');
    });
</script>
```

THIS IS
$('div<title!="div1Title">').css('border', '5px solid red');
EQUIVALENT TO
$('div:not(<title="div1Title">)').css('border', '5px solid red');

Selects all elements that have title attribute value containing the given substring - Title, and sets 5px solid red border

```
<script type="text/javascript">
    $(document).ready(function () {
        $('<title*="Title">').css('border', '5px solid red');
    });
</script>
```

Selects all elements that have title attribute value containing the given word - mySpan, delimited by spaces, and sets 5px solid red border

```
<script type="text/javascript">
    $(document).ready(function () {
        $('<title~="mySpan">').css('border', '5px solid red');
    });
</script>
```

Selects all elements that have title attribute value equal to myTitle or starting with myTitle followed by a hyphen (-) and sets 5px solid red border

```
<script type="text/javascript">
    $(document).ready(function () {
        $('<title|="myTitle">').css('border', '5px solid red');
    });
</script>
```

Selects all elements that have title attribute value starting with div and sets 5px solid red border

```
<script type="text/javascript">
    $(document).ready(function () {
        $('<title^="div">').css('border', '5px solid red');
    });
</script>
```

Selects all elements that have title attribute value ending with Heading and sets 5px solid red border

```
<script type="text/javascript">
    $(document).ready(function () {
        $('<title$="Heading">').css('border', '5px solid red');
    });
</script>
```

jQuery case insensitive attribute selector

The following example, only selects DIV 1. This is because jQuery attribute value selector is case-sensitive.

```
<html>
<head>
    <title></title>
    <script src="Scripts/jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('div<title="DivTitle">').css('border', '3px solid red');
        });
    </script>
</head>
<body>
    <div title="DivTitle">
        DIV 1
    </div>
    <br />
    <div title="DIVTITLE">
        DIV 2
    </div>
    <br />
    <div title="divtitle">
        DIV 3
    </div>
</body>
</html>
```

Use the following code to make the jQuery attribute value selector case-insensitive

```
<script type="text/javascript">
    $(document).ready(function () {
        $('div<title>').filter(function () {
            return $(this).attr('title').toLowerCase() == 'divtitle';
        }).css('border', '3px solid red');
    });
</script>
```

The above script should select all the 3 divs.

Now let us look at an example of making attribute contains selector <name*="value">, case-insensitive.

The following example, selects only DIV 1 element. This is because the attribute contains selector is case-sensitive.

```
<html>
<head>
    <title></title>
    <script src="Scripts/jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('div<title*="Div">').css('border', '3px solid red');
        });
    </script>
```

```
</head>
<body>
    <div title="DivTitle1">
        DIV 1
    </div>
    <br />
    <div title="DIVTITLE2">
        DIV 2
    </div>
    <br />
    <div title="divtitle3">
        DIV 3
    </div>
</body>
</html>
```

To make attribute contains selector case-insensitive, use filter() method and regular expression as shown below.

```
<script type="text/javascript">
    $(document).ready(function () {
        $('div<title>').filter(function () {
            return (/Div/i).test($(this).attr('title'));
        }).css('border', '3px solid red');
    });
</script>

OR

<script type="text/javascript">
    $(document).ready(function () {
        $('div<title>').filter(function () {
            return RegExp('Div','i').test($(this).attr('title'));
        }).css('border', '3px solid red');
    });
</script>
```

The above script should select all the 3 divs.

Visual Studio Keyboard Shortcuts

Convert Selected Text to Upper Case - CTRL + SHIFT + U
Convert Selected Text to Lower Case - CTRL + U

jQuery input vs :input

$(:input) selects all input, textarea, select and button elements where as $(input) just selects elements with an input tag.

```
<html>
<head>
    <title></title>
    <script src="jquery-1.11.2.js"></script>
</head>
<body style="font-family:Arial">
    First Name : <input type="text" value="John" />
    <br /><br />
    Last Name : <input type="text" value="Major" />
    <br /><br />
    Gender :
    <input type="radio" name="gender" checked="checked" value="Male">Male
    <input type="radio" name="gender" value="Female">Female
    <br /><br />
    Skills :
    <input type="checkbox" name="skills" checked="checked" value="JavaScript"
/>JavaScript
    <input type="checkbox" name="skills" checked="checked" value="jQuery" />jQuery
    <input type="checkbox" name="skills" value="C#" />C#
    <br /><br />
    Country:
    <select>
        <option selected="selected" value="USA">USA</option>
        <option value="India">India</option>
        <option value="UK">UK</option>
    </select>
    <br /><br />
    Summary :
    <br />
    <textarea>I am a Senior Dot Net Developer with 10 years experience</textarea>
    <br /><br />
    <input type="submit" value="submit" />
</body>
</html>
```

Now we want to get the text value from all the textboxes. On this page we have 2 textboxes
1. First Name
2. Last Name

jQuery code to get textbox value using $(input)

```
<script type="text/javascript">
    $(document).ready(function () {
        $('input[type="text"]').each(function () {
            alert($(this).val());
        });
    });
</script>
```

jQuery code to get textbox value using $(:input)

```
<script type="text/javascript">
    $(document).ready(function () {
        $(':input[type="text"]').each(function () {
            alert($(this).val());
        });
    });
</script>
```

Which one is better for performance $('input[type="text"]') or $(':input[type="text"]')
$('input[type="text"]') is better for performance over $(':input[type="text"]'). This is because $(':input[type="text"]') needs to scan all input elements, textarea, select etc, where as $('input[type="text"]') scans only input elements. So if you want to find elements with an input tag, it is always better to use $('input[type="text"]') over $(':input[type="text"]')

:checked selector

The :checked selector selects all checked checkboxes or radio buttons. Let us understand this with an example.

Selects all checked radio button elements

$('input[type="radio"]:checked')

Example : When you click the submit button without selecting any radio button, "No radio button checked" message should be displayed.

When you click the submit button after a radio button is checked, then a message stating "Male is checked" or "Female is checked" should be displayed.

```
<html>
<head>
    <title></title>
    <script src="jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('#btnSubmit').click(function () {
                var result = $('input[type="radio"]:checked');
                if (result.length > 0) {
                    $('#divResult').html(result.val() + " is checked");
                }
                else {
                    $('#divResult').html("No radio button checked");
                }
            });
        });
    </script>
</head>
<body style="font-family:Arial">
    Gender :
    <input type="radio" name="gender" value="Male">Male
    <input type="radio" name="gender" value="Female">Female
    <br /><br />
    <input id="btnSubmit" type="submit" value="submit" />
    <br /><br />
    <div id="divResult">
    </div>
</body>
</html>
```

Selects all checked checkbox elements
$('input[type="checkbox"]:checked')

Example : When you click the submit button without checking any checkbox, "No checkbox checked" message should be displayed.

When you click the submit button after checking a checkbox, then a message stating the number of checkboxes checked and their values should be displayed.

```
<html>
<head>
    <title></title>
    <script src="jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('#btnSubmit').click(function () {
                var result = $('input[type="checkbox"]:checked');
                if (result.length > 0) {
                    var resultString = result.length + " checkboxe(s)
checked<br/>";
                    result.each(function () {
                        resultString += $(this).val() + "<br/>";
                    });
                    $('#divResult').html(resultString);
                }
                else {
                    $('#divResult').html("No checkbox checked");
                }
            });
        });
    </script>
</head>
<body style="font-family:Arial">
    Skills :
    <input type="checkbox" name="skills" value="JavaScript" />JavaScript
    <input type="checkbox" name="skills" value="jQuery" />jQuery
    <input type="checkbox" name="skills" value="C#" />C#
    <input type="checkbox" name="skills" value="VB" />VB
    <br /><br />
    <input id="btnSubmit" type="submit" value="submit" />
    <br /><br />
    <div id="divResult">
    </div>
</body>
</html>
```

Select values of checked checkboxes that are in different groups & How to pass a variable to jquery selector

If you have just one group of checkboxes on your page, to get all the checked checkboxes you can use $('input[type="checkbox"]:checked').

However, if you have more than one checkbox group, then $('input[type="checkbox"]:checked') is going to select all checked checkboxes from both the checkbox groups.

If you prefer to get checked checkboxes from a specific checkbox group, depending on which button you have clicked, you can use $('input[name="skills"]:checked') or $('input[name="cities"]:checked'). This will ensure that the checked checkboxes from only the skills or cities checkbox group are selected.

At the moment to get the checked checkboxes values, we are using a button click event. You can also use the click event of the checkbox to do this.

```
<html>
<head>
    <title></title>
    <script src="jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('input[name="skills"]').click(function () {
                getSelectedCheckBoxes('skills');
            });

            $('input[name="cities"]').click(function () {
                getSelectedCheckBoxes('cities');
            });

            var getSelectedCheckBoxes = function (groupName) {
                var result = $('input[name="' + groupName + '"]:checked');
                if (result.length > 0) {
                    var resultString = result.length + " checkbox(es)
checked<br/>";
                    result.each(function () {
                        resultString += $(this).val() + "<br/>";
                    });
                    $('#div' + groupName).html(resultString);
                }
                else {
                    $('#div' + groupName).html("No checkbox checked");
                }
            };
        });
    </script>
</head>
<body style="font-family:Arial">
```

```
    Skills :
    <input type="checkbox" name="skills" value="JavaScript" />JavaScript
    <input type="checkbox" name="skills" value="jQuery" />jQuery
    <input type="checkbox" name="skills" value="C#" />C#
    <input type="checkbox" name="skills" value="VB" />VB
    <br /><br />
    Preferred Cities :
    <input type="checkbox" name="cities" value="New York" />New York
    <input type="checkbox" name="cities" value="New Delhi" />New Delhi
    <input type="checkbox" name="cities" value="London" />London
    <br /><br />
    Selected Skills:<br />
    <div id="divskills"></div>
    <br />
    Selected Cities:<br />
    <div id="divcities"></div>
</body>
</html>
```

Get the checked checkbox text along with the value

Here is what we want to do. As we check the checkboxes, we want to display the count of checkboxes checked and their text and values.

```
<html>
<head>
    <title></title>
    <script src="jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('input[name="skills"]').click(function () {
                getSelectedCheckBoxes('skills');
            });

            $('input[name="cities"]').click(function () {
                getSelectedCheckBoxes('cities');
            });

            var getSelectedCheckBoxes = function (groupName) {
                var result = $('input[name="' + groupName + '"]:checked');
                if (result.length > 0) {
                    var resultString = result.length + " checkbox checked<br/>";
                    result.each(function () {
                        var selectedValue = $(this).val();
                        resultString += selectedValue + " - "
                            + $('label[for="option-' + selectedValue + '"]').text()
+ "<br/>";
                    });
                    $('#div' + groupName).html(resultString);
                }
                else {
                    $('#div' + groupName).html("No checkbox checked");
                }
            };
        });
    </script>
</head>
<body style="font-family:Arial">
    Skills :
    <input type="checkbox" name="skills" value="js" />
    <label for="option-js">JavaScript</label>
    <input type="checkbox" name="skills" value="jq" />
    <label for="option-jq">jQuery</label>
    <input type="checkbox" name="skills" value="cs" />
    <label for="option-cs">C#</label>
    <input type="checkbox" name="skills" value="vb" />
    <label for="option-vb">VB</label>
    <br /><br />
    Preferred Cities :
    <input type="checkbox" name="cities" value="ny" />
    <label for="option-ny">New York</label>
    <input type="checkbox" name="cities" value="nd" />
    <label for="option-nd">New Delhi</label>
    <input type="checkbox" name="cities" value="ln" />
    <label for="option-ln">London</label>
    <br /><br />
    Selected Skills:<br />
    <div id="divskills"></div>
    <br />
    Selected Cities:<br />
    <div id="divcities"></div>
</body>
</html>
```

jQuery :selected selector

To select all checked checkboxes or radio buttons, we use :checked selector. To select all selected options of a select element use :selected selector.

How to get selected option from single select dropdown in jquery : We want to get the selected option text and value

```
<html>
<head>
    <title></title>
    <script src="jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('#selectCountries').change(function () {
                var selectedOption = $('#selectCountries option:selected');
                $('#divResult').html('Value = ' + selectedOption.val()
                    + ', Text = ' + selectedOption.text());
            });
        });
    </script>
</head>
<body style="font-family:Arial">
```

```
    Country:
    <select id="selectCountries">
        <option selected="selected" value="USA">United States</option>
        <option value="IND">India</option>
        <option value="UK">United Kingdom</option>
        <option value="CA">Canada</option>
        <option value="AU">Australia</option>
    </select>
    <br /><br />
    <div id="divResult"></div>
</body>
</html>
```

How to get all selected options from multi select dropdown in jquery : We want to get all the selected options text and value.

```
<html>
<head>
    <title></title>
    <script src="jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('#selectCountries').change(function () {
                var selectedOptions = $('#selectCountries option:selected');
                if (selectedOptions.length > 0) {
                    var resultString = '';
                    selectedOptions.each(function () {
                        resultString += 'Value = ' + $(this).val() +
                            ', Text = ' + $(this).text() + '<br/>';
                    });
                    $('#divResult').html(resultString);
                }
            });
        });
    </script>
</head>
<body style="font-family:Arial">
    <select id="selectCountries" multiple="multiple">
        <option selected="selected" value="USA">United States</option>
        <option value="IND">India</option>
        <option value="UK">United Kingdom</option>
        <option value="CA">Canada</option>
        <option value="AU">Australia</option>
    </select>
    <br /><br />
    <div id="divResult"></div>
</body>
</html>
```

Please note : Hold down the CTRL key, to select more than one item.

jQuery enabled/disabled selector

Selects all disabled elements

```
$(':disabled')
```

Selects all disabled elements and sets a 3px solid red border

```
<html>
<head>
    <title></title>
    <script src="jquery-1.11.2.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $(':disabled').css('border', '3px solid red');
        });
    </script>
</head>
<body style="font-family:Arial">
    <table>
        <tr>
            <td>First Name </td>
            <td><input type="text" disabled="disabled" /></td>
        </tr>
        <tr>
            <td>Last Name </td>
            <td><input type="text" disabled="disabled" /></td>
        </tr>
        <tr>
            <td>Email </td>
            <td><input type="text" /></td>
        </tr>

        <tr>
            <td>Gender </td>
            <td>
                <select id="selectGender" disabled="disabled">
                    <option value="Male">Male</option>
                    <option value="Female">Female</option>
                </select>
            </td>
        </tr>
        <tr>
            <td>Comments</td>
            <td><textarea></textarea></td>
        </tr>
        <tr>
            <td colspan="2">
                <input type="submit" value="Submit" disabled="disabled"/>
            </td>
        </tr>
    </table>
</body>
</html>
```

Selects all input elements that are disabled and sets a 3px solid red border

```
<script type="text/javascript">
    $(document).ready(function () {
        $('input:disabled').css('border', '3px solid red');
    });
</script>
```

Selects all input elements with type=text that are disabled and sets a 3px solid red border

```
<script type="text/javascript">
    $(document).ready(function () {
        $('input<type="text">:disabled').css('border', '3px solid red');
    });
</script>
```

Selects all enabled elements

```
$(':enabled')
```

Selects all enabled textarea elements and sets a 3px solid red border

```
<script type="text/javascript">
    $(document).ready(function () {
        $('textarea:enabled').css('border', '3px solid red');
    });
</script>
```

EoF