

FIRST EDITION

COVERS  
ASP.NET 2.0 / 3.5 WITH  
JQUERY 1.3.2

# 51

## TIPS TRICKS AND RECIPES

SUPROTIM AGARWAL



with

# jQuery & ASP.NET

Controls

# 51 Tips, Tricks and Recipes with jQuery and ASP.NET Controls

Suprotim Agarwal

 | Knowledge Visuals

## ABOUT THE AUTHOR

---



*Suprotim Agarwal*, an ASP.NET Architecture MVP (**Microsoft Most Valuable Professional**), has been developing applications for over 10 years using Microsoft technologies. After working as a software engineer, manager, architect and a mentor, Suprotim now works as a Consultant to a large number of IT Firms, consulting them on implementing enterprise-level Microsoft solutions and other web-based applications. Not known to many, Suprotim also consults websites on Internet Marketing.

Suprotim is also the founder and primary contributor to [DotNetCurry](#), [SQLServerCurry](#) and [DevCurry](#) and has written over 700 articles and blog posts. When he is not churning out code, he spends his time checking out new gadgets, playing strategic video games and helps his wife (in eating) some deliciously cooked recipes from [Foodatarian](#)

## ABOUT THE REVIEWERS

---



*Govind Kanshi* strives to create efficient solutions for his customers using his experience and ability to dissect complex technologies available in the market. His experience ranges from working across startup, consulting to Fortune 100 firms as well as working in product delivery. He is comfortable creating and managing teams and is also comfortable working in individual capacity. His interests and expertise lies across Databases, Web applications, Virtualization and Performance tuning.



*Malcolm Sheridan* (MCSD) is an independent contractor who has been developing business line applications for the last decade. He has extensive experience using various development technologies but now concentrates on and favors the .Net framework and ASP.NET specifically. Most recently, Malcolm has been imperative in developing applications for clients in the telecommunications and banking sectors. In his spare time, he enjoys working on his F# and Silverlight.

## TABLE OF CONTENTS

---

### **Section 1 – jQuery and the ASP.NET TextBox**

- Recipe 1: Access Values from All or Selective TextBoxes using jQuery
- Recipe 2: Prevent Cut, Copy and Paste Operations in a TextBox using jQuery
- Recipe 3: Allow Only Alphanumeric Characters in a TextBox using jQuery
- Recipe 4: Make a TextBox ReadOnly at RunTime using jQuery
- Recipe 5: Shift Focus to the Next TextBox using the Enter Key using jQuery
- Recipe 6: Clone the Contents of a TextBox into another TextBox using jQuery
- Recipe 7: Auto Focus the First TextBox using jQuery
- Recipe 8: Click and Highlight Text in a TextBox using jQuery
- Recipe 9: Limit Number Of Characters In a Multiline TextBox using jQuery
- Recipe 10: AutoScroll a Multiline TextBox using jQuery
- Recipe 11: Synchronize Scrolling of Two Multiline TextBoxes using jQuery
- Recipe 12: Create a TextBox Watermark Effect using jQuery

### **Section 2: jQuery and the ASP.NET CheckBox**

- Recipe 13: Make the CheckBox a Required Field on your WebForm using jQuery
- Recipe 14: Check/Uncheck All Items in a CheckBoxList using jQuery
- Recipe 15: Use a CheckBox to Conditionally Enable or Disable controls using jQuery
- Recipe 16: Instantly Display the CheckBox Checked Items using jQuery
- Recipe 17: Performing Calculations with Check Boxes using jQuery

### **Section 3: jQuery and the ASP.NET RadioButton**

- Recipe 18: Display the Text and Value of the Selected RadioButton using jQuery
- Recipe 19: Display Text and Value of Multiple RadioButtonList using jQuery
- Recipe 20: Display and Hide a RadioButton Based on a condition using jQuery

#### **Section 4: jQuery and the ASP.NET Panel**

Recipe 21: Display and Hide Panels with Animation using jQuery

Recipe 22: Smooth Cascading Panels using jQuery

Recipe 23: Create a Slide Show with Panels using jQuery

#### **Section 5: jQuery and the ASP.NET DropDownList**

Recipe 24: Retrieve the Selected Text and Value of a DropDownList using jQuery

Recipe 25: Sort the Items of a DropDownList using jQuery

Recipe 26: Add Items to the DropDownList using jQuery

Recipe 27: Search and Filter Items of a DropDownList using jQuery

Recipe 28: Convert the DropDownList to emulate a MultiSelect ListBox using jQuery

#### **Section 6: jQuery and the ASP.NET GridView**

Recipe 29: Pass Multiple Values from a GridView to Another Page using jQuery

Recipe 30: Efficient Server Side Paging using jQuery

Recipe 31: Handling the Master-Detail scenario in a GridView using jQuery

Recipe 32: Click and Retrieve the Value of a GridView Cell using jQuery

Recipe 33: Highlight and Remove GridView Rows using jQuery

Recipe 34: Highlight and Remove GridView Columns using jQuery

Recipe 35: How to Search and Highlight a GridView Cell using jQuery

Recipe 36: Quickly add Events to Controls inside the GridView using jQuery

#### **Section 7: jQuery and the ASP.NET Image Control**

Recipe 37: Click and Increase the Size of an Image using jQuery

Recipe 38: Change Image Opacity on MouseOver using jQuery

Recipe 39: Create an Image SlideShow using jQuery

Recipe 40: Image Swapping using jQuery

Recipe 41: Add/Remove Hyperlinks on Images using jQuery

## **Section 8: jQuery and the ASP.NET Bulleted List Control**

Recipe 42: Convert Bulleted List Items to Hyperlinks using jQuery

Recipe 43: Add New Items to a BulletedList Control and apply CSS on them using jQuery

Recipe 44: Sort Items of a BulletedList Control using jQuery

## **Section 9: jQuery and ASP.NET Hyperlink Control**

Recipe 45: Implementing KeyBoard Shortcuts on Hyperlink Control using jQuery

Recipe 46: Change the URL of a Hyperlink Control using jQuery

Recipe 47: Text Link Ad Rotation using jQuery

## **Section 10: jQuery, AJAX and ASP.NET**

Recipe 48: Consume an ASP.NET JSON Web Service using jQuery

Recipe 49: jQuery Tabs and Lazy Loading

Recipe 50: Populate a ListBox from JSON results using jQuery

Recipe 51: Serialize only Selected Elements of a Form and Submit it using jQuery

## **Appendix A: Useful jQuery Plug-ins**

## **Appendix B: jQuery Learning Resources**

## **Appendix C: Server-side Code and Other Resources**

## SAMPLE CHAPTERS

Note: JavaScript should be kept in separate files. For readability, it has been included inline.

### Recipe 2: Prevent Cut, Copy and Paste Operations in a TextBox using jQuery

#### Challenge:

Users should be prevented from doing Cut, Copy and Paste operations in an ASP.NET TextBox.

#### Solution:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
    <title>Prevent Cut, Copy and Paste Operations in a TextBox</title>
    <link href="../CSS/Demos.css" rel="stylesheet" type="text/css" />
    <script type="text/javascript"
        src='../Scripts/jquery-1.3.2.min.js'>
    </script>

    <script type="text/javascript">
        $(function() {
            $('input[id$=tbl]').bind('cut copy paste', function(e) {
                e.preventDefault();
                alert('You cannot ' + e.type + ' text!');
            });
        });
    </script>
</head>
<body>
    <form id="form1" runat="server">
        <div class="bigDiv">
            <h2>Prevent Cut, Copy and Paste Operations in a TextBox</h2>
            <br /><br />
            <asp:Label ID="lblEmail" runat="server"
                Text="Re-Enter your Email Address"></asp:Label>
            <asp:TextBox ID="tbl" runat="server"
                Text="Try copying and pasting text here"
                ToolTip="Try Copy/Cut/Paste in textbox"/>
            <br /><br /><br />
            Tip: Text can only be entered in this box. It cannot be
            copied and pasted from a different textbox
        </div>
    </form>
</body>
</html>
```

**Explanation:**

This recipe handles a typical requirement where you are asked to 'confirm' an email address and want the user to type it manually, instead of copying and pasting it.

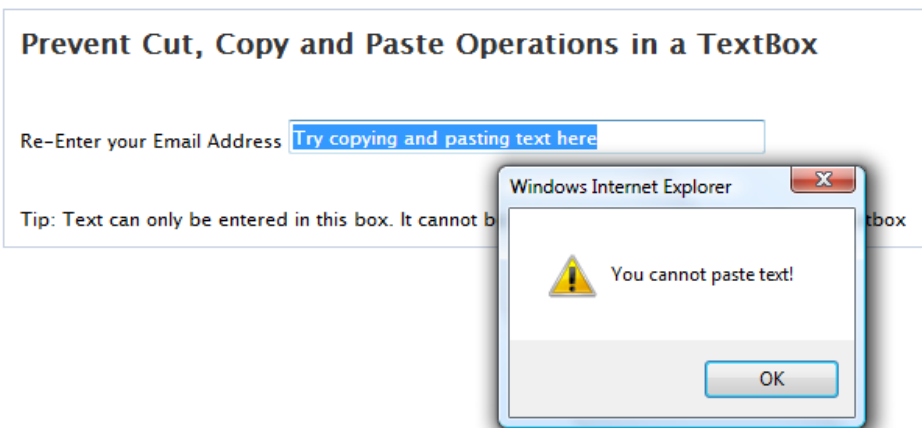
The jQuery *bind()* function binds one or more events to a handler. Observe how convenient it is to list multiple events (cut copy paste) together and bind it to a handler.

```
$('input[id$=tbl]').bind('cut copy paste', function(e) {})
```

If the user performs any of these events on the textbox, the default behavior is prevented using *e.preventDefault()* and the user is alerted. The *e.type* describes the type of event performed.

```
$('input[id$=tbl]').bind('cut copy paste', function(e) {
    e.preventDefault();
    alert('You cannot ' + e.type + ' text!');
});
```

The screenshot shown below represents how the user is informed when he/she tries to copy text from the textbox.



Text can be edited in this textbox, but cut/copy/paste actions are prevented.

[Live Demo](#)
**Browsers Supported:**

IE7, IE 8, Firefox 3, Chrome 2, Safari 4



**Useful Links:**

<http://docs.jquery.com/Events/bind>

**Recipe 3: Allow Only Alphanumeric Characters in a TextBox using jQuery****Challenge:**

Users should be allowed to enter only Alphabets and Numbers in a TextBox. Non-alphanumeric characters should be disallowed. You also want to prevent users from copying and pasting non-alphanumeric characters into the TextBox.

**Solution:**

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
    <title>Allow Only Alphanumeric Characters in a TextBox</title>
    <link href="../../CSS/Demos.css" rel="stylesheet" type="text/css" />
    <script type='text/javascript'
        src='../Scripts/jquery-1.3.2.min.js'>
    </script>

    <script type="text/javascript">
        $(function() {
            $('input.alpha[id=tb1]').bind('keyup blur', function() {
                if (this.value.match(/^[a-zA-Z0-9 ]/g)) {
                    this.value = this.value.replace(/^[a-zA-Z0-9 ]/g, '');
                }
            });
        });
    </script>
</head>
<body>
    <form id="form1" runat="server">
        <div class="bigDiv">
            <h2>Allow Only Alphanumeric Characters in a TextBox</h2>
            <asp:TextBox ID="tb1" runat="server" class="alpha"
                Text="" ToolTip="Try entering Non alphanumeric char"/>
            <br /><br />
            Tip: Examples of some non-alphanumeric characters <br />
            are ~!@~#$%^&* and so on.
        </div>
    </form>
</body>
</html>
```

**Explanation:**

The code shown above matches a regular expression `/[^a-zA-Z0-9]/g` that finds all characters that are not alphabets or numbers. If the user enters a character that is not an alphabet or number, the character is immediately replaced with an empty string. The blur event detects characters when the textbox loses focus whereas the keyup event detects characters after the character has been entered. I have observed people using only the keypress event to handle this requirement, but remember that in IE, keypress does not behave as expected for non-character keys.

As [Yehuda Katz](#) says – “My mantra has always been: *keydown/keyup* if you want to know the key that was pressed; *keypress* if you want to know what text was detected”



If you want to accept only numbers, use `/^[0-9]/g` and for accepting only alphabets, use `/^[a-zA-Z]/g`

**[Live Demo](#)****Browsers Supported:**

IE 7, IE 8, Firefox 3, Chrome 2, Safari 4

**Useful Links:**

<http://docs.jquery.com/Events/bind>

<http://docs.jquery.com/Events/keyup>

## Recipe 9: Limit Number Of Characters In a Multiline TextBox using jQuery

**Challenge:**

You want to restrict the user from entering more than 50 characters in an ASP.NET Multiline TextBox. The ASP.NET Multiline TextBox ignores the MaxLength property.

**Solution:**

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
  <title>Limit Characters in a Multiline TextBox</title>
  <link href="../../CSS/Demos.css" rel="stylesheet" type="text/css" />
  <script type='text/javascript'
    src='../Scripts/jquery-1.3.2.min.js'>
```

- [10]-

```

</script>

<script type="text/javascript">
    $(function() {
        var limit = 50;

        $('#textarea[id$=tb1]').keyup(function() {
            var len = $(this).val().length;
            if (len > limit) {
                this.value = this.value.substring(0, limit);
            }
            $('#charLeft').text(limit - len + " characters left");
        });
    });
</script>
</head>
<body>
    <form id="form1" runat="server">
        <div class="smallDiv">
            <h2>Type into this textbox which accepts 50 characters overall</h2>
            <asp:TextBox ID="tb1" runat="server" TextMode="MultiLine"/><br />
            <span id="charLeft"></span>
        </div>
    </form>
</body>
</html>

```

**Explanation:**

In this example, we first capture the keyup event and calculate the number of characters in the textbox.

```

$('#textarea[id$=tb1]').keyup(function() {
    var len = $(this).val().length;

```

If the character exceeds the limit of the textbox (50 characters), the additional characters entered by the user is disallowed.

```

if (len > limit) {
    this.value = this.value.substring(0, limit);
}

```

Observe how we are selecting the control using textarea `$('#textarea[id$=tb1]')`. This is because the ASP.NET Multiline TextBox renders as a textarea.

**[Live Demo](#)**

### Recommended Approach

Although the solution given above restricts the user from entering more than 50 characters, this behavior can be confusing for users who may not realize that they have reached the TextBox limit. Instead of disallowing extra characters, a slick way of handling this requirement would be to visually indicate to the user when the textbox limit has been exceeded. Then before submitting the form, give the user a chance to remove the extra text. The code shown below changes the background color of the textbox to red when the textbox limit is exceeded. The user is also prevented from submitting the form.

Here's the recommended solution:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
    <title>Limit Characters in a Multiline TextBox</title>
    <link href="../../CSS/Demos.css" rel="stylesheet" type="text/css" />
    <script type='text/javascript'
        src='../Scripts/jquery-1.3.2.min.js'>
    </script>

    <script type="text/javascript">
        $(function() {
            var limit = 50;
            var $tb = $('textarea[id$=tb1]');
            $tb.keyup(function() {
                var len = $(this).val().length;
                if (len > limit) {
                    $(this).addClass('exceeded');
                    $('#spn').text(len - limit + " characters exceeded");
                }
                else {
                    $(this).removeClass('exceeded');
                    $('#spn').text(limit - len + " characters left");
                }
            });

            $('input[id$=btnSubmit]').click(function(e) {
                var len = $tb.val().length;
                if (len > limit) {
                    e.preventDefault();
                }
            });
        });
    </script>
</head>
<body>
    <form id="form1" runat="server">
        <div class="smallDiv">
            <h2>Type into this textbox which accepts 50 characters overall</h2>
            <asp:TextBox ID="tb1" runat="server" TextMode="MultiLine"/><br />
            (This textbox accepts only 50 characters) <br />
            <span id="spn"></span> <br />
        </div>
    </form>
</body>
</html>
```

```

<asp:Button ID="btnSubmit" runat="server" Text="Submit"/>
<span id="error"></span>
<br /><br />
Tip: Clicking on the Submit button when number of characters are
less than 50, results in a postback to same page. If you exceed 50
characters, the exceeded characters are printed below the textbox
and a postback is prevented.
</div>
</form>
</body>
</html>

```

When the number of characters exceeds the textbox limit, we add the *exceeded* class to the textbox, which turns the textbox background to red, indicating the user that the limit has been exceeded.

```

if (len > limit) {
    $(this).addClass('exceeded');
}

```

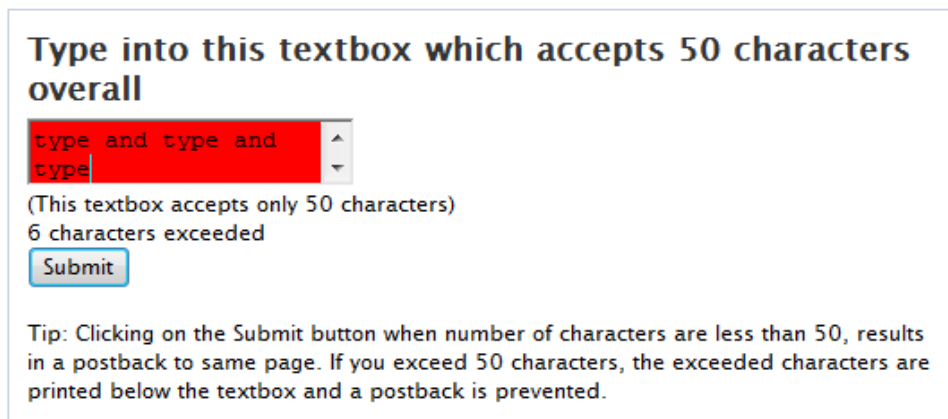
The exceeded class declared in Demos.css looks like this

```

.exceeded
{
    background-color:red;
}

```

The result is shown here:



**Type into this textbox which accepts 50 characters overall**

type and type and type

(This textbox accepts only 50 characters)  
6 characters exceeded

Submit

Tip: Clicking on the Submit button when number of characters are less than 50, results in a postback to same page. If you exceed 50 characters, the exceeded characters are printed below the textbox and a postback is prevented.

When the user tries and submits the form now, the length of the TextBox is calculated and the user is prevented from submitting the form, since the textbox now exceeds the permissible length.

```

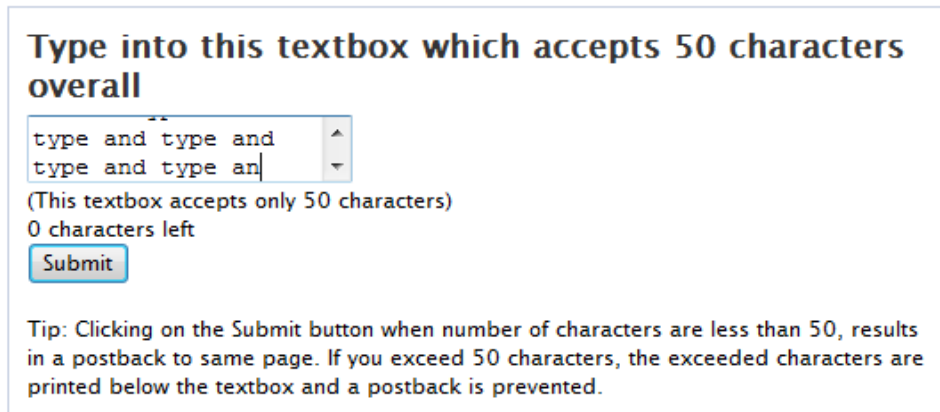
$('input[id$=btnSubmit]').click(function(e) {
    var len = $tb.val().length;
    if (len > limit) {
        e.preventDefault();
    }
});

```

When the user removes the extra text, the *exceeded* class is removed.

```
$(this).removeClass('exceeded');
```

The result is shown here:



**Type into this textbox which accepts 50 characters overall**

type and type and  
type and type an

(This textbox accepts only 50 characters)  
0 characters left

Tip: Clicking on the Submit button when number of characters are less than 50, results in a postback to same page. If you exceed 50 characters, the exceeded characters are printed below the textbox and a postback is prevented.

The form can now be submitted.

#### [Live Demo](#)

#### Browsers Supported:

IE 7, IE 8, Firefox 3, Chrome 2, Safari 4

#### Useful Links:

<http://docs.jquery.com/Attributes/addClass>

<http://docs.jquery.com/Attributes/removeClass>

## Recipe 14: Check/Uncheck All Items in a CheckBoxList using jQuery

#### Challenge:

You want to check/uncheck all the checkboxes at once, in the easiest possible way.

**Solution:**

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Check/Uncheck All CheckBoxes At Once</title>
    <link href="../../CSS/Demos.css" rel="stylesheet" type="text/css" />
    <script src='../Scripts/jquery-1.3.2.min.js'
        type='text/javascript'>
    </script>

    <script type="text/javascript">
        $(function() {
            var $chkBox = $("input:checkbox[id$=chkAll]");
            var $tblChkBox = $("table.chk input:checkbox");

            $chkBox.click(function() {
                $tblChkBox
                    .attr('checked', $chkBox
                        .is(':checked'));
            });

            // Unchecks chkAll when a checked CheckBox in cbList
            // is unchecked
            $tblChkBox.click(
                function(e) {
                    if (!$(this)[0].checked) {
                        $chkBox.attr("checked", false);
                    }
                }
            );
        });
    </script>
</head>
<body>
    <form id="form1" runat="server">
        <div class="smallDiv">
            <h2>Check/Uncheck all checkboxes at once using the
                'Do All' checkbox</h2><br />
            <asp:CheckBox ID="chkAll" runat="server"
                Text="Do All"
                ToolTip="Click here to check/uncheck all checkboxes at once"/>
            <br /><hr />
            <asp:CheckBoxList ID="cbList" runat="server" class="chk">
                <asp:ListItem Text="Option One" Value="One" />
                <asp:ListItem Text="Option Two" Value="Two" />
                <asp:ListItem Text="Option Three" Value="Three" />
                <asp:ListItem Text="Option Four" Value="Four" />
                <asp:ListItem Text="Option Five" Value="Five" />
            </asp:CheckBoxList>
            <br /><br />
            Tip: Clicking on any of the individual 'checked' checkboxes,
                unchecks the 'Do All' checkbox.
        </div>
    </form>

```

```
</body>
</html>
```

### Explanation:

In the Section 2 introduction, we have seen how the CheckBoxList renders as a group of input elements of the type checkbox, inside a HTML table.

In this example, when the user clicks the checkbox (Do All), we reference all input elements of the checkbox type inside the table, and set the 'checked' value of the referenced elements to true or false, based on the checked value of the 'Do All' checkbox.

```
$chkBox.click(function() {
    $tblChkBox
        .attr('checked', $chkBox
            .is(':checked'));
});
```

The output is as shown below:

### Check/Uncheck all checkboxes at once using the 'Do All' checkbox

☒ Do All

---

☒ Option One  
☒ Option Two  
☒ Option Three  
☒ Option Four  
☒ Option Five

Tip: Clicking on any of the individual 'checked' checkboxes, unchecks the 'Do All' checkbox.

When the 'Do All' checkbox is unchecked, the CheckBoxList is unchecked too.

The code also takes into account a scenario when a user unchecks any of the individual checkboxes in the CheckBoxList, the 'Do All' checkbox is also unchecked. This is achieved using the following code:

```
$tblChkBox.click(
function(e) {
    if (!$(this)[0].checked) {
        $chkBox.attr("checked", false);
    }
});
```



Thanks to Tim Hobbs for sharing this tip.

### [Live Demo](#)

#### **Browsers Supported:**

IE 7, IE 8, Firefox 3, Chrome 2, Safari 4

#### **Useful Links:**

[Check/Uncheck all Items in an ASP.NET CheckBox List using jQuery](#)

<http://docs.jquery.com/Selectors/checked>

## Recipe 18: Display the Text and Value of the Selected RadioButton using jQuery

#### **Challenge:**

Display the text and value of the selected radio button.

#### **Solution:**

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Retrieve the Text and Value of the selected RadioButton</title>
    <link href="../../CSS/Demos.css" rel="stylesheet" type="text/css" />
    <script src='../Scripts/jquery-1.3.2.min.js'
        type='text/javascript'>
    </script>

    <script type="text/javascript">
        $(function() {
            var $radBtn = $("table.tbl input:radio");
            $radBtn.click(function() {
                var $radChecked = $(':radio:checked');
                $("#para").text('')
                    .append("<b>Index: </b>" +
                        $radChecked.val() + "<br/>")
                    .append("<b>Value: </b>" +
                        $radChecked.next().text());
            });
        });
    </script>
</head>
```

```

<body>
  <form id="form1" runat="server">
    <div class="smallDiv">
      <h2>Select a RadioButton to display its Text and Value</h2><br />
      The site I like the most : <br />
      <asp:RadioButtonList ID="rbl" runat="server" class="tbl"
        ToolTip="Select a Radio Button to see its text and value">
        <asp:ListItem Text="dotnetcurry.com" Value="0"></asp:ListItem>
        <asp:ListItem Text="devcurry.com" Value="1"></asp:ListItem>
        <asp:ListItem Text="sqlservercurry.com" Value="2"></asp:ListItem>
      </asp:RadioButtonList>
      <br />
      <p id="para"></p>
    </div>
  </form>
</body>
</html>

```

**Explanation:**

Retrieving the index and value of a radio button is a very common requirement in our applications and can be easily achieved using jQuery. In this example, we first retrieve the radio buttons using

```
var $radBtn = $("table.tbl input:radio");
```

Since this is a RadioButtonList, the control gets rendered as a table. When the user clicks on any of the radio buttons, we first store the checked radiobuttons in the `$radChecked` variable to improve selector performance

```
var $radChecked = $(':radio:checked');
```

We then use

`$radChecked.val()` to retrieve the value and `$radChecked.next().text()` to retrieve the text.

Observe how we are accessing the value of a radiobutton using

```
$radChecked.next().text().
```

This is because the RadioButton gets rendered like this:

```

<td>
<input id="rbl_0" type="radio" name="rbl" value="0" />
<label for="rbl_0">dotnetcurry.com</label>
</td>

```

To access the value, which is inside the label, we use `next()` which finds the very next sibling of each radiobutton, which in our case, is the label control. The value and text is displayed on the page using a paragraph (para). The output looks similar to this:

### Select a RadioButton to display its Text and Value

The site I like the most :

- ☒ dotnetcurry.com
- ☐ devcurry.com
- ☐ sqlservercurry.com

**Index:** 0

**Value:** dotnetcurry.com

[Live Demo](#)

#### Browsers Supported:

IE 7, IE 8, Firefox 3, Chrome 2, Safari 4

#### Useful Links:

<http://docs.jquery.com/Selectors/checked>

[Order your copy now!](#)