Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Introduction to jQuery

Lab Book

Copyright © 2011 IGATE Corporation (a part of Capegemini Group). All rights reserved.

No part of this publication shall be reproduced in any way, including but not limited to photocopy, photographic, magnetic, or other record, without the prior written permission of IGATE Corporation (a part of Capegemini Group).

IGATE Corporation (a part of Capegemini Group) considers information included in this document to be confidential and proprietary.

Document Revision History

Date	Revision No.	Author	Summary of Changes
05-Oct-2011	1.0	Karthik M	Initial Draft
May- 2016	1.1	Anjulata	Removed some assignments as per new ToC

Table of Contents

Getting Started

Overview

This lab book is a guided tour for learning jQuery. It comprises solved examples and 'To Do' assignments. Follow the steps provided in the solved examples and work out the 'To Do' assignments given.

Setup Checklist for jQuery

Here is what is expected on your machine in order for the lab to work.

Minimum System Requirements

- Intel Pentium 90 or higher (P166 recommended)
- Microsoft Windows XP or above.
- Memory: 256MB of RAM (512MB or more recommended)
- Internet Explorer 9.0 or higher

Please ensure that the following is done:

- A text editor like Notepad, Editplus

Instructions

- For all coding standards refer Appendix A. All lab assignments should refer coding standards.
- Create a directory by your name in drive <drive>. In this directory, create a subdirectory jQuery_assign. For each lab exercise create a directory as lab <lab number>.
- You may also look up the on-line help provided in the http://www.jquery.com.
- The faculty will introduce you to the editor to be used.

Learning More (Bibliography if applicable)

- Visual QuickStart Guide jQuery by Steven Holzner
- jQuery in Action by Bear Bibeault and Yehuda Katz
- Learning jQuery 1.3 by Jonathan Chaffer and Karl Swedberg

Lab 1. jQuery Basics

Goals	• Understand the process of creating a jQuery page and viewing it in a browser window. • Understanding Content Delivery Network (CDN) and using the hosted jQuery file from Google or Microsoft
Time	20 minutes

1.1: Creating jQuery Page

Create a web page to display the message box 'Hello World'.

Solution:

Step 1: Download the latest java script file (jquery-1.12.3.min.js) minified version from http://www.jquery.com

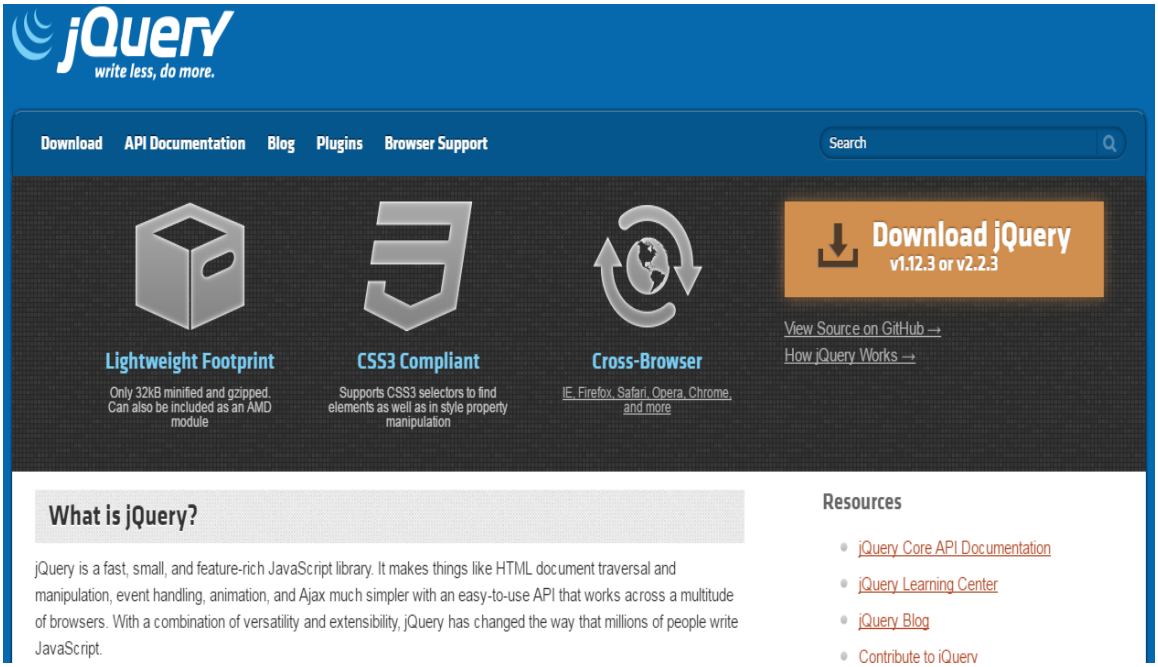

Figure 1: jquery.com

Step 2: Create a folder named scripts under jQueryDemos in C:\ or D:\ and place the downloaded script there.

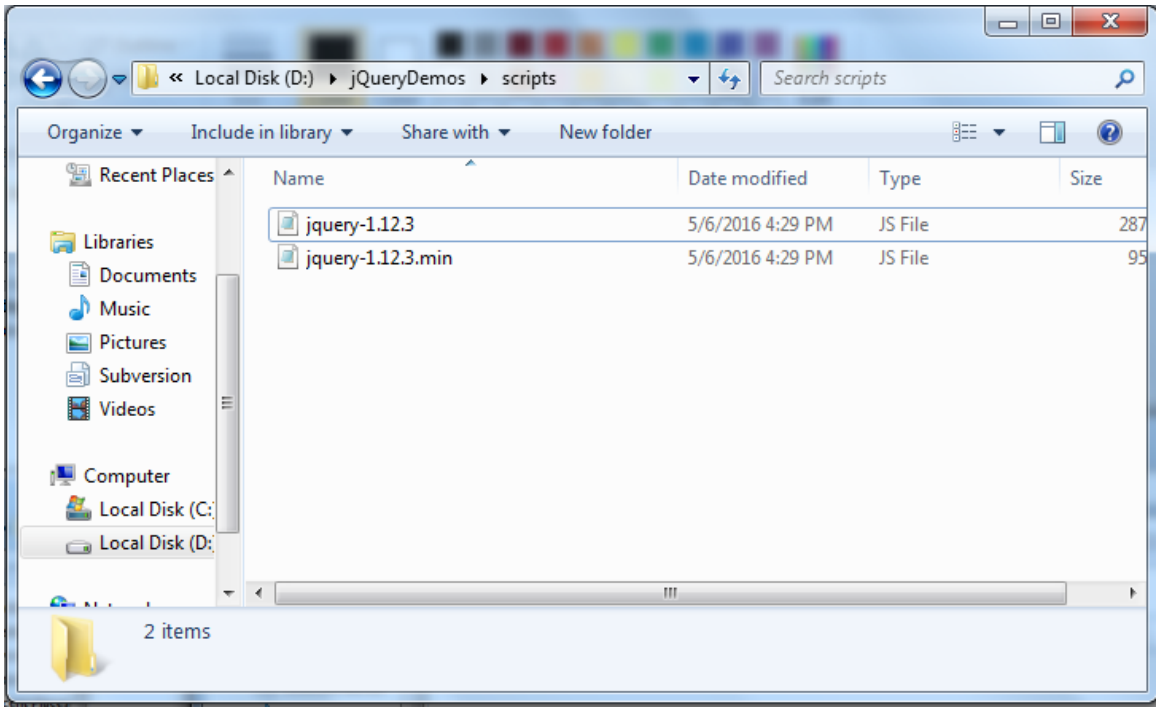

Figure 2: jQuery Folder Structure

Step 3: Open any text editor like notepad and type the following code and save it with .html extension in jQueryDemos folder.

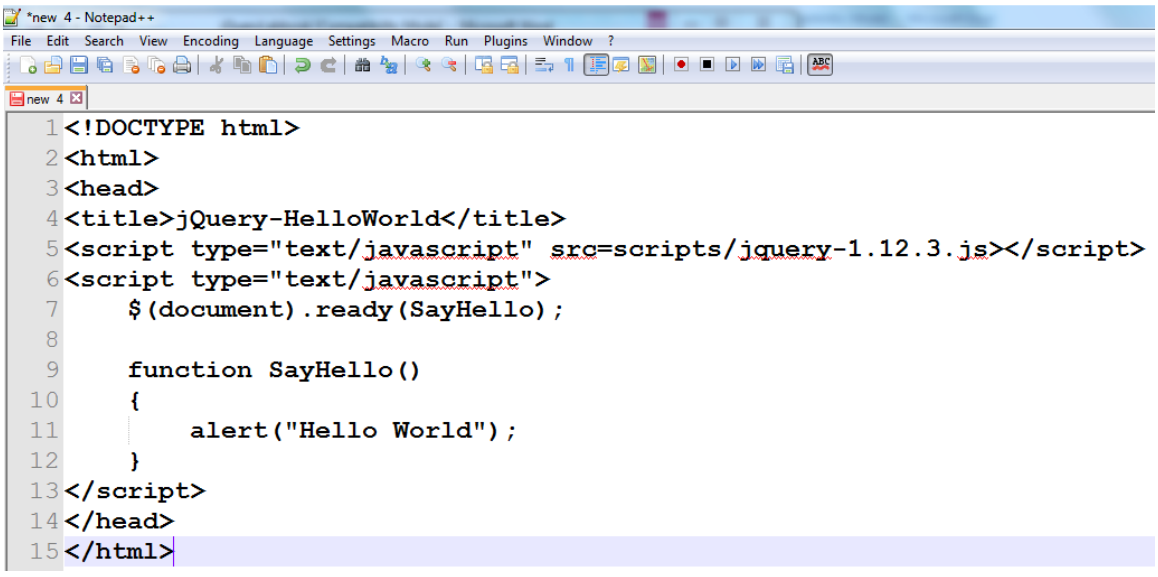

Figure 3: Save contents as an *.html* file

Step 4: Open the file which we have saved, in browser. Output will appear as shown in the following figure:

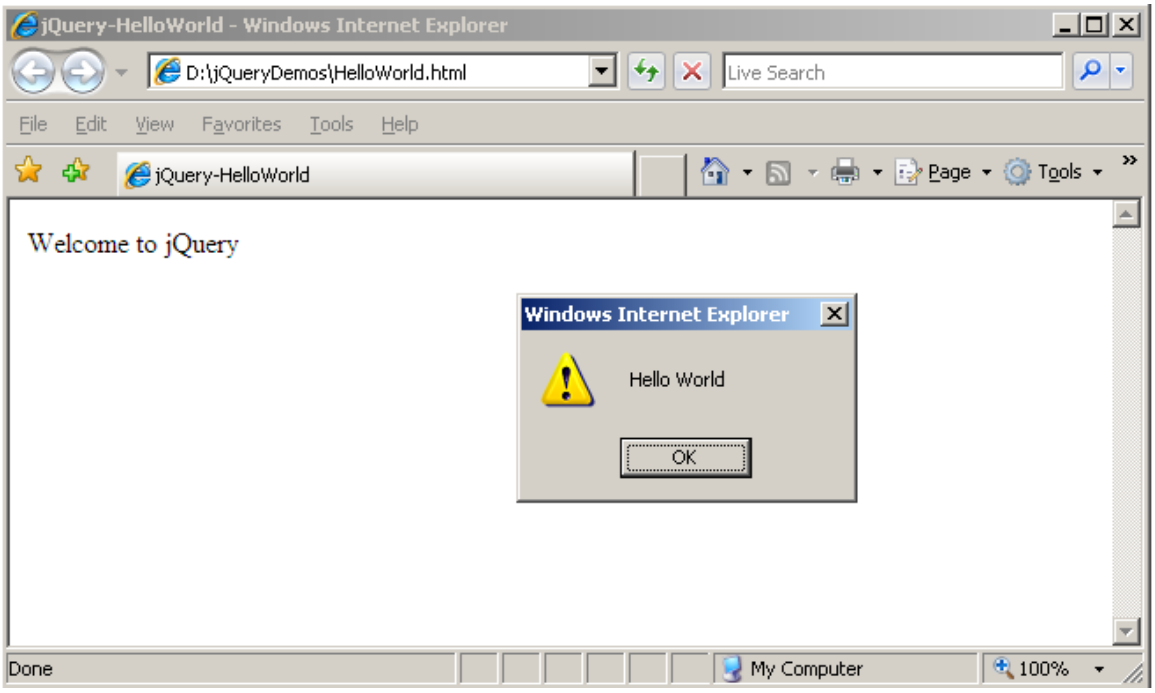

Figure 4: HelloWorld.html in a browser

1.2: Accessing jQuery Script from CDN

Create a jQuery web page and access the jQuery script from Google

Solution:

Step 1: Open any text editor like notepad and type the following code and save it with .html extension in jQueryDemos folder

```
jQueryCDN.html
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>jQuery-HelloWorld</title>
5  <script type="text/javascript"
6  src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.2/jquery.min.js">
7  </script>
8  <script type="text/javascript">
9      $(document).ready(function(){
10         alert("jQuery is awesome");
11     });
12 </script>
13 </head>
14 </html>
```

Figure 5 Accessing Script from CDN

Step 2: Output will appear as shown in the following figure:

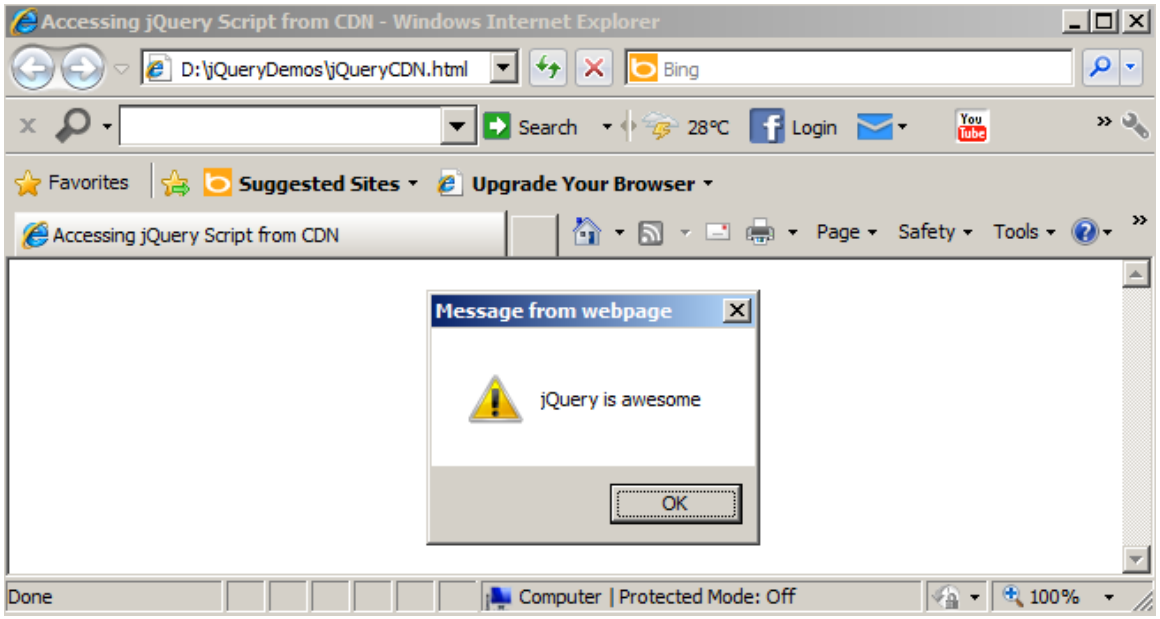

Figure 6 jQueryCDN.html in browser

Script can be also accessible from

Microsoft jquery Script: http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.12.3.min.js

jQuery Domain : http://code.jquery.com/jquery-1.12.3.min.js

Lab 2. jQuery Selectors

Goals	• At the end of this lab session, you will be able to perform the following topics using jQuery Selectors: • Selecting nodes using Tag Name • Selecting Nodes by ID • Selecting Nodes by Class Name • Selecting Nodes by Attribute Value • Selecting Input Nodes
Time	60 minutes

2.1: Selecting Nodes using tag name

Create a jQuery web page and select all the paragraphs apply blue background and white foreground to it. Select the paragraphs under div tag and change the text in that paragraph to uppercase. Navigate through all the paragraphs and print the contents within it in a message box

Solution:

Step 1: Open any text editor like notepad and type the following code and save it with .html extension in jQueryDemos folder

```
<html>
<head>
<title>Selectors By Tag Name</title>
<script type="text/javascript" src="scripts/jquery-1.12.3.min.js"></script>
<script type="text/javascript">
$(document).ready(function()
{
        //Selecting all the paragraphs
        var paragraphs = $('p');

        //Applying Styles to all the paragraphs
        paragraphs.css('background-color','blue');
        paragraphs.css('color','white');

        //Applying Styles to the paragraphs under div tag (descendants)
        var divParagraphs= $('div p');
        divParagraphs.css('text-transform','uppercase');
```

```
//Navigating though all paragraph tags and print the contents within it using
//message box
        var collection = $('div,p');
        paragraphs.each(function(){
        alert($(this).html());
        });

});
</script>
</head>
<body>
<p>First Paragraph</p>
<p>Second Paragraph</p>

<div>
<p>First Paragraph - inside div tag </p>
<p>Second Paragraph - inside div tag </p>
</div>

</body>
</html>
```

Step 2: Output will appear as shown in the following figure

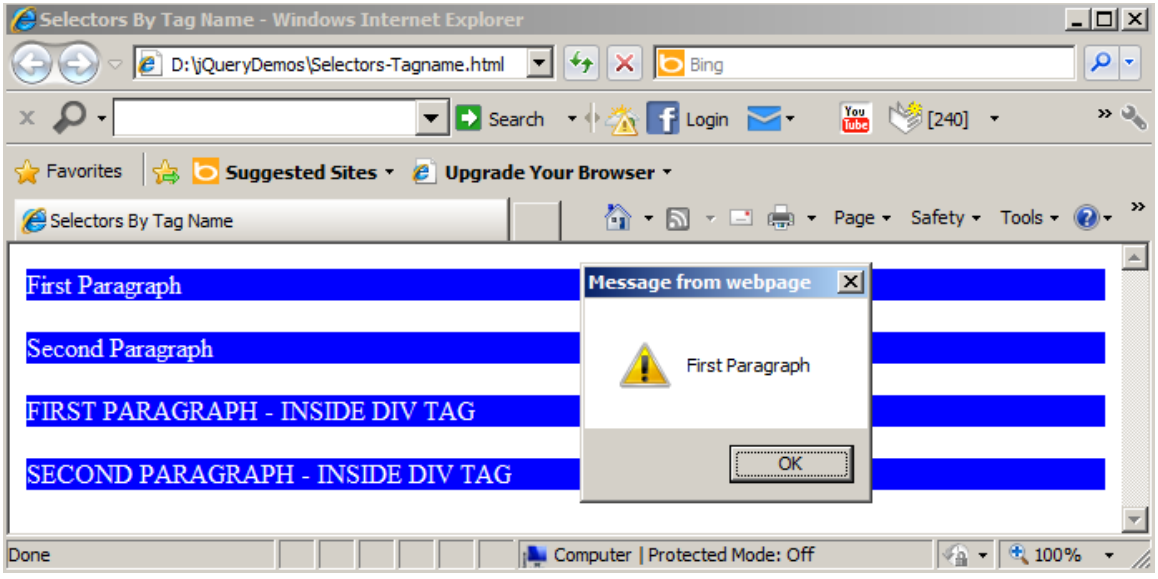

Figure 7 Selectors-TagName.html in browser

2.2: Selecting Nodes by Element ID

Create a jQuery web page to retrieve the contents from one div tag in html format and apply the retrieved content into another div tag in text format

Solution:

Step 1: Open any text editor like notepad and type the following code and save it with .html extension in jQueryDemos folder

```
<html>
<head>
<title>Selectors By ElementID</title>
<script type="text/javascript" src="scripts/jquery-1.12.3.min.js"></script>
<script type="text/javascript">
$(document).ready(function()
{
        //Retrieving the content from testDiv in html format
        var data = $('#testDiv').html();

        //Applying the content to targetDiv in text format
        $('#targetDiv').text(data);
});
</script>
</head>
<body>
<div id="testDiv">
<b>This is my Test Div text</b>
</div>
<div id="targetDiv" />
</body>
</html>
```

Step 2: Output will appear as shown in the following figure

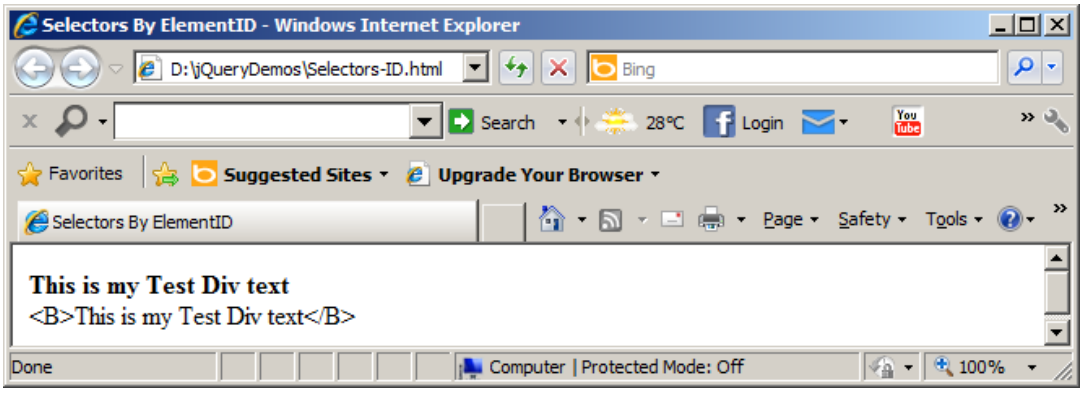

Figure 8 Selectors-ID.html in browser

2.3: Selecting Nodes by Class Name

Create a jQuery web page to apply styles to the elements that don't match a specific selector

Solution:

Step 1: Open any text editor like notepad and type the following code and save it with .html extension

```
<html>
<head>
<title>Selectors using class Name</title>
<style type="text/css">
.blueDiv{background-color:blue; color:white;}
.redDiv{background-color:red; color:white;}
</style>
<script type="text/javascript" src="scripts/jquery-1.12.3.min.js"></script>
<script type="text/javascript">
$(function ( ) {
        //Selects divs that do not have the class 'redDiv' using not filter
        var collection = $('div:not(.redDiv)');
        collection.css('border','5px solid green');
        collection.css('padding','5px');
});
</script>
</head>
<body>
<div class="blueDiv">
<p>First BlueDiv - Paragraph.</p>
</div>
<div class="redDiv">
<p>First RedDiv - Paragraph.</p>
</div>
<div class="blueDiv">
<p>Second BlueDiv - Paragraph.</p>
</div>
</body>
</html>
```

Step 2: Output will appear as shown in the following figure

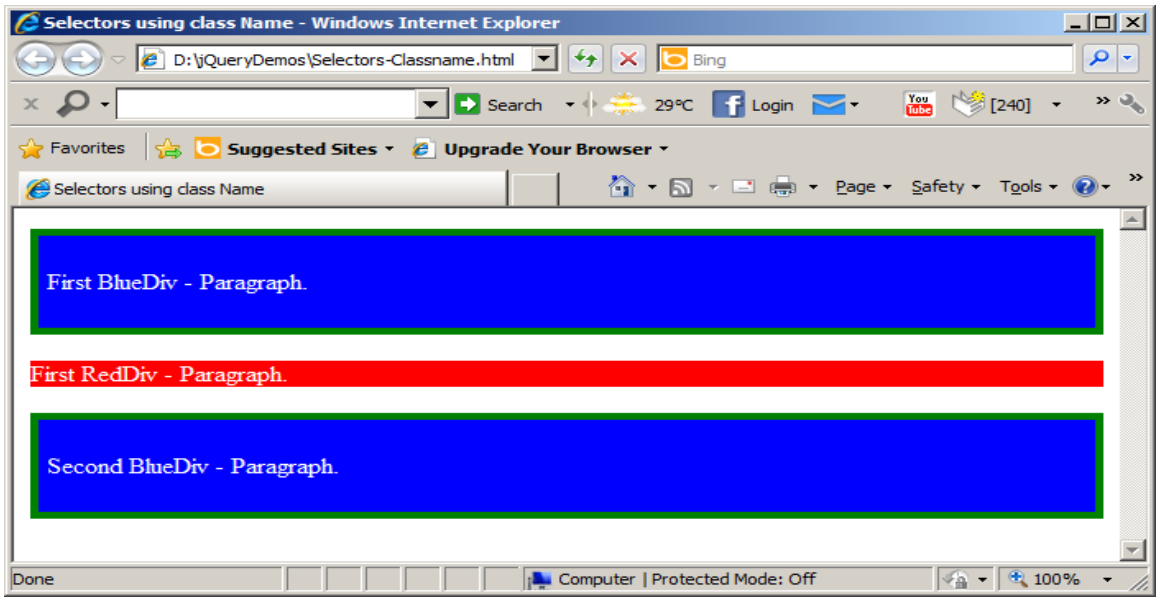

Figure 9 Selectors-Classname.html in browser

2.4: Selecting Nodes by Attribute Value

Create a jQuery web page to select elements based on attributes (input) and attributes values (text) and apply styles to it.

Solution:

Step 1: Open any text editor like notepad and type the following code and save it with .html extension

```
<html>
<head>
<title>Selectors By Attributes</title>
<style type="text/css">
.applyStyles {color:white;background-color:black;text-transform:uppercase;}
</style>
<script type="text/javascript" src="scripts/jquery-1.12.3.min.js"></script>
<script type="text/javascript">
        $(function ( ) {
                var textBoxes = $('input[type="text"]');
                textBoxes.addClass("applyStyles");
        });
</script>
</head>
<body>
<table> <tr>
<td align="right"><span>Name : </span></td>
<td><input type="text"/></td> </tr>
<tr>
<td align="right"><span>City : </span></td>
<td><input type="text"/></td>  </tr>
</body> </html>
```

Step 2: Output will appear as shown in the following figure

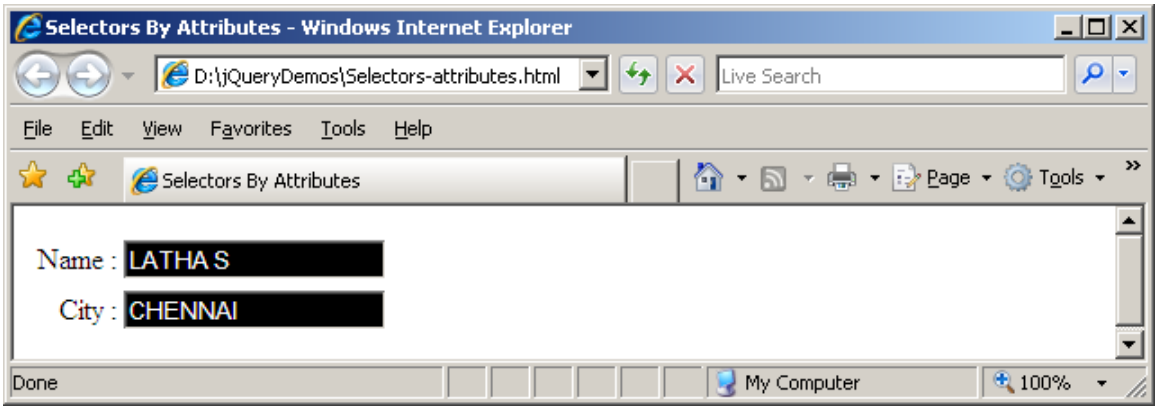

Figure 10 Selectors-attributes.html in browser

2.5: Selecting Form Elements by type

Create a jQuery web page to select the checked radio button and display its index position and value

Solution:

Step 1: Open any text editor like notepad and type the following code and save it with .html extension

```
<html>
<head>
<title>Input Selectors</title>
<script type="text/javascript" src="scripts/jquery-1.12.3.min.js"></script>
<script type="text/javascript">
$(document).ready(function()
{
        var $radChecked = $(':radio:checked');
        $("#targetDiv").text('')
        .append("<b>Index: </b>" +$radChecked.index() + "<br/>")
        .append("<b>Value: </b>" +$radChecked.val());
});
</script>
</head>
<body>
<table>
<tr>
 <td>
<input type="radio" name="gender" value="Male">Male
<input type="radio" name="gender" value="Female"
checked="checked">Female
</td>
 </tr>
<tr>
 <td>
<div id="targetDiv" />
 </td>
 </tr>
</table>
</body>
</html>
```

Step 2: Output will appear as shown in the following figure

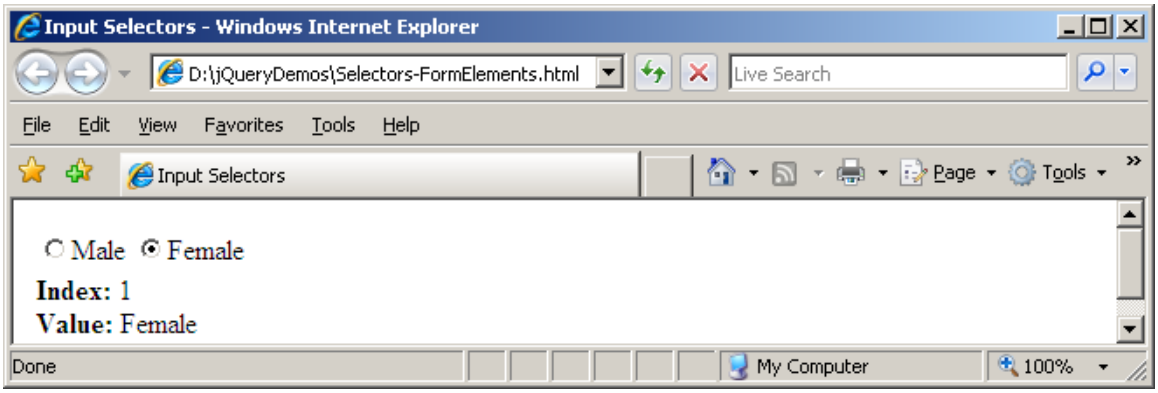

Figure 11 Selectors-FormElements.html in browser

:text selects <input type="text" />
:password selects <input type="password" />
:radio selects <input type="radio" />
:checkbox selects <input type="checkbox" />
:submit selects <input type="submit" />
:image selects <input type="image" />
:reset selects <input type="reset" />
:button selects <input type="button" />
:file select <input type="file" />
:hidden selects <input type="hidden" />
:input filter selects all input, textarea, button, and select elements.

TODO Assignments

Assignment-1: Use the following html code snippet

```
<table id="region" border="2" bordercolor="black">
<thead>
<tr>
<th>State</th>
<th>Capital</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andhrapradesh</td>
<td>Hyderabad</td>
</tr>
<tr>
<td>Karnataka</td>
<td>Bangalore</td>
</tr>
<tr>
<td>Maharashtra</td>
<td>Mumbai</td>
</tr>
<tr>
<td>Tamilnadu</td>
<td>Chennai</td>
</tr>
</tbody>
</table>
<div id="StatesDiv"></div>
<div id="CapitalDiv"></div>
```

- Retrieve the states from the table and print the states in ordered list
- Retrieve the 4th State's Capital and print it

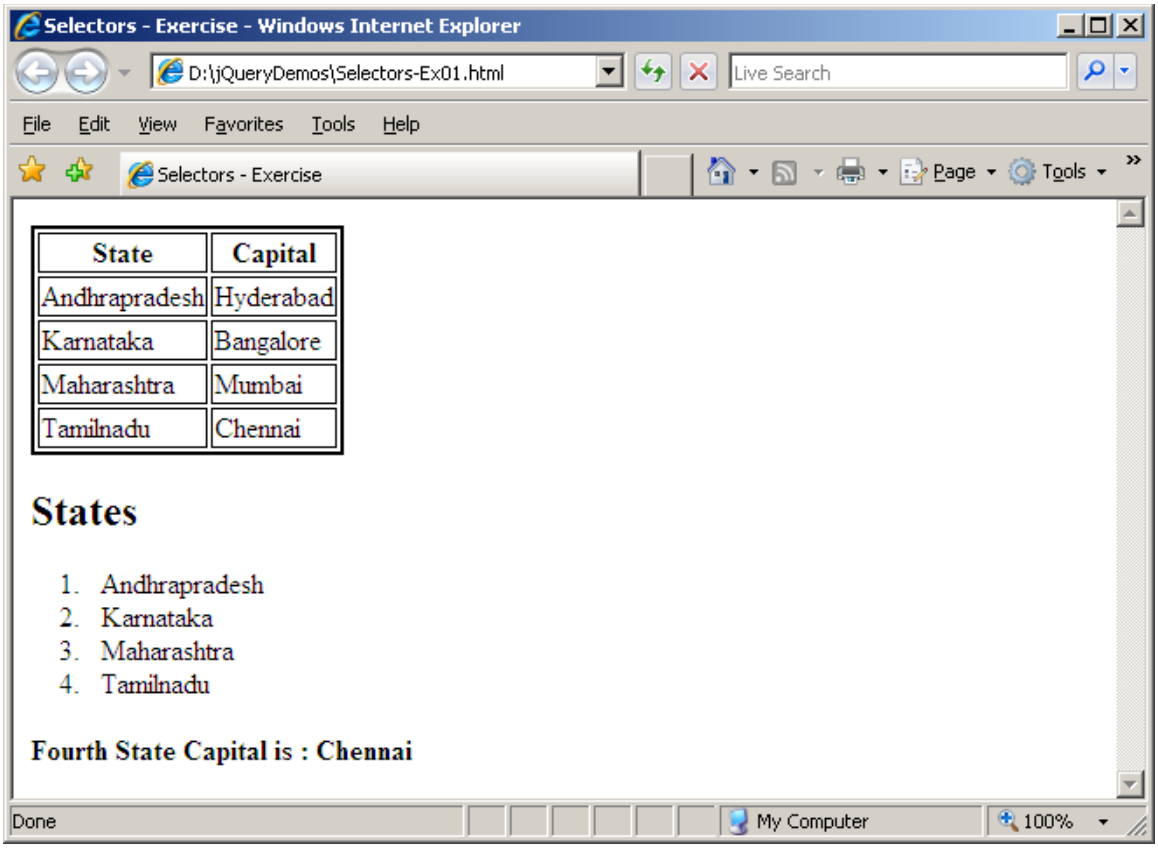

Figure 12 Assignment output

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Capgemini Public

Lab 3. Working with JSON

Goals	• At the end of this lab session, you will be able to • Work with JSON Object • Using JSON in jQuery
Time	20 minutes

3.1: Working with JSON Object

Create a webpage to store the employee details as JSON Object and retrieve the employee details and print it

Solution:

Step 1: Open any text editor like notepad and type the following code and save it with .html extension

```
<html>
<head>
<title>JSON Introduction</title>
<script type="text/javascript">
var jsonObject = {"EmployeeID":714709, "Name":"Karthik M",
"Department":"Training", "ISContract":false}
var parsedJson = eval(jsonObject);
var contractEmployee = ""
if(parsedJson.ISContract)
 contractEmployee = "Yes";
 else
 contractEmployee = "No";
var result = "
<table border='1'><tr><td>EmployeeID</td><td>Name</td>
<td>Department</td><td>Contract Employee</td><tr>"
                    +"<td>"+parsedJson.EmployeeID+"</td>"
                    +"<td>"+parsedJson.Name+"</td>"
                    +"<td>"+parsedJson.Department+"</td>"
                    +"<td>"+contractEmployee+"</td>"
                    +"</tr></table>";
document.write(result);
</script>
</head>
<body>
</body>
</html>
```

Step 2: Output will appear as shown in the following figure

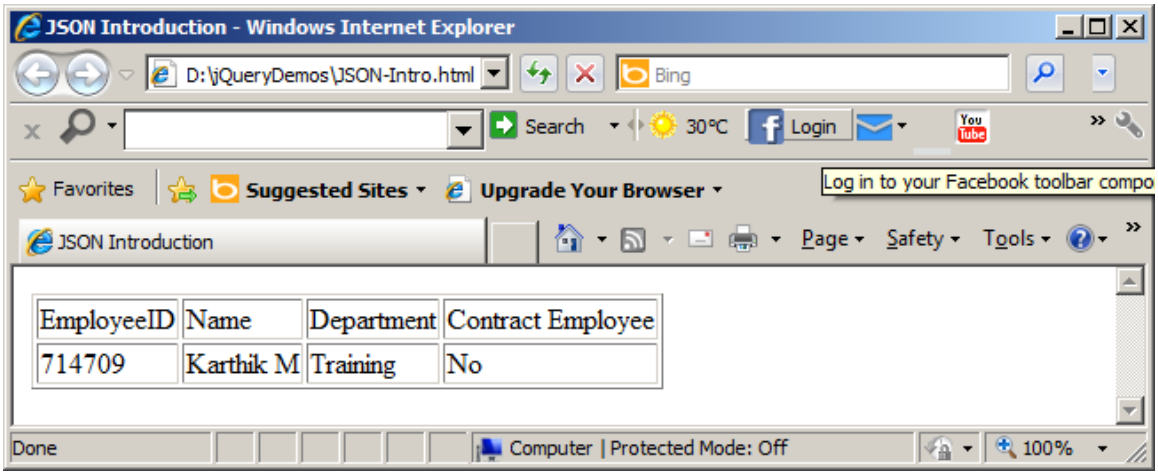

Figure 13 JSON-Intro.html in browser

3.2: Working with JSON Object and Array

Create a webpage to illustrate the usage of JSON object and Array types

Solution:

Step 1: Open any text editor like notepad and type the following code and save it with .html extension

```
<html>
<head>
<title>JSON Array and Object</title>
<script type="text/javascript">
var jsonObject = {"EmployeeID":"714709","Name":"Karthik",
  "office":{
          "company":"Capgemini ",
          "city":"Bangalore"
   },
   "Hobbies":["Programming","Music"]
}
var parsedJson = eval(jsonObject);
var result = "Name : "+parsedJson.Name+"<br/>Company :
"+parsedJson.office.company+"<br/>Hobby :
"+parsedJson.Hobbies[0];
document.write(result);
</script>
</head>
<body>
</body>
</html>
```

Step 2: Output will appear as shown in the following figure

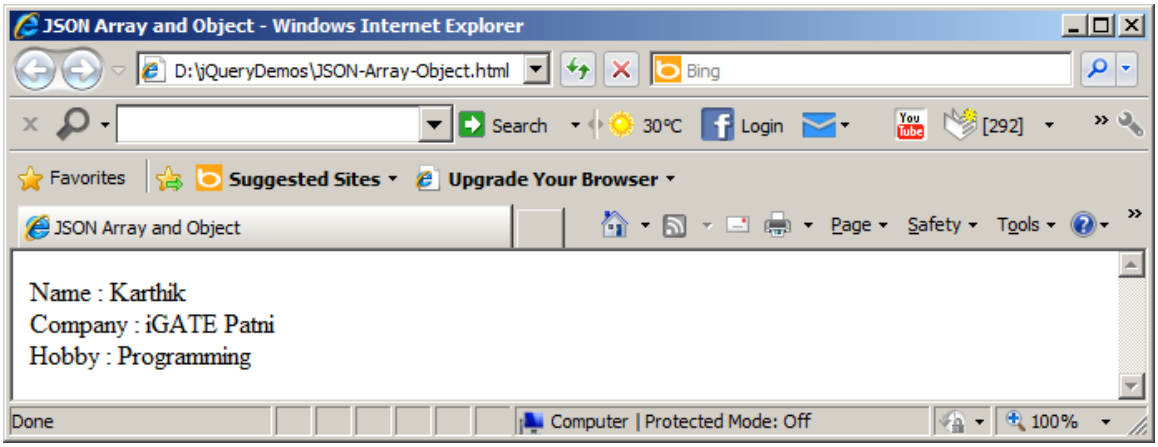

Figure 14 JSON-Array-Object.html in browser

3.3: Using JSON object in jQuery

Create a jQuery webpage to format a div tag with some list of css properties in JSON format.

Solution:

Step 1: Open any text editor like notepad and type the following code and save it with .html extension

```
<html>
<head>
<title>Using JSON in jQuery</title>
<script type="text/javascript" src="scripts/jquery-1.12.3.min.js"></script>
<script type="text/javascript">
$(document).ready(function()
{
        var collection = $('#targetDiv');
        //using JSON to avoid Method chaining
        collection.css({"border":"5px solid green",
                                "padding":"10px",
                                "background-color":"blue",
                                "color":"white",
                                "font-size":"24pt",
                                "font-family":"verdana",
                                "font-weight":"bold",
                                "text-transform":"uppercase"
                        });
});
</script>
</head>
```

```
<body>
<div id="targetDiv">
<p>Using JSON in jQuery</p>
</div>
</body>
</html>
```

Step 2: Output will appear as shown in the following figure

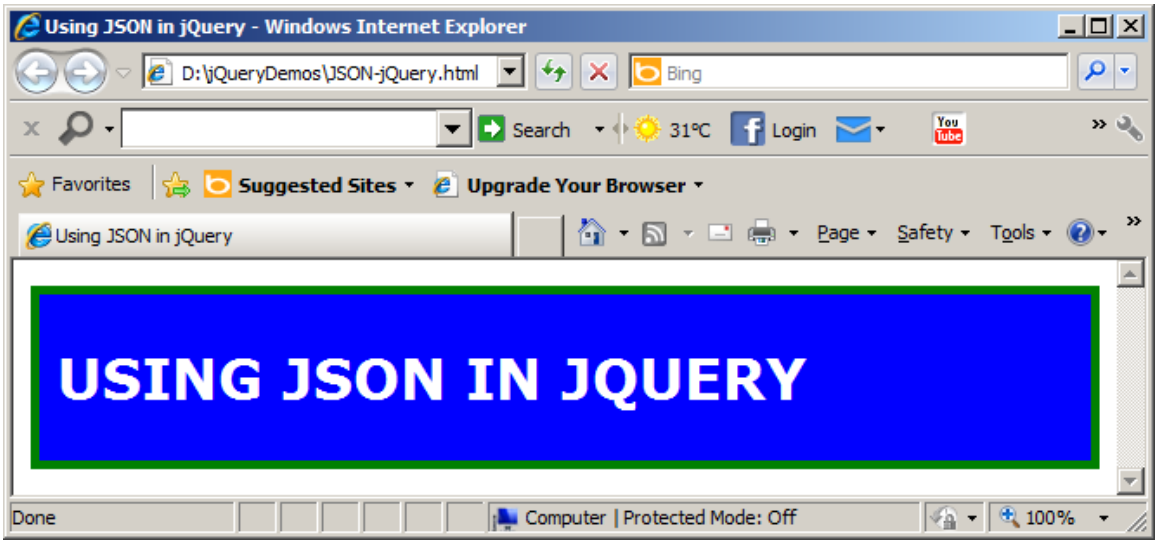

Figure 15 JSON-jQuery.html in browser

Lab 4. Interacting with DOM

Goals	• At the end of this lab session, you will be able to • Perform DOM Manipulations • Interact with DOM elements using jQuery
Time	30 minutes

4.1: Iterate Nodes using jQuery

Create a jQuery webpage to iterate through the DOM tree using jQuery

Solution:

Step 1: Open any text editor like notepad and type the following code and save it with .html extension

```
<html>
<head>
<title>DOM Iterations</title>
<style type="text/css">
.main{background-color:blue;color:white}
.company{background-color:yellow;color:black}
.sales{background-color:crimson;color:white}
.hr{background-color:green;color:white}
.training{background-color:brown;color:white}
</style>
<script type="text/javascript" src="scripts/jquery-1.12.3.min.js"></script>
<script type="text/javascript">
$(document).ready(function()
{
	var main = $("#mainDiv").children();
	var company = $("#company").children();

	$("#targetDiv").append("<h2> MainDiv has "+main.length+"
children</h2>");
	$("#targetDiv").append("<h2> Company has "+company.length+"
children(s)</h2>");

	var result="";
	$(company).each(function(index)
	{
		result+=index+" "+$(this).text()+"<br/>";
	});
	$("#targetDiv").append("<h2>"+result+"</h2>");
});
</script>
</head>
```

```
<body>
<div id="mainDiv" class="main">Main
        <div id="company" class="company">Capgemini
                <div id="sales" class="sales">Sales Team</div>
                <div id="hr" class="hr">HR Team</div>
                <div id="training" class="training">Training</div>
        </div>
</div>
<div id="targetDiv"></div>
</body>
</html>
```

Step 2: Output will appear as shown in the following figure

Figure 16 DOM-Iteration.html in browser

4.2: Sort list items using jQuery

Create a jQuery webpage to sort list items and attach style (change the list item bullet to icon) to the sorted list using jQuery

Solution:

Step 1: Open any text editor like notepad and type the following code and save it with .html extension

```
<html>
<head>
<title>Sorting list items using jQuery</title>
<style type="text/css">
.java{
list-style-image:url('images/java.png');
}
</style>
<script type="text/javascript" src="scripts/jquery-1.12.3.min.js"></script>
<script type="text/javascript">
$(document).ready(function()
{
	if(confirm('Are you sure to sort these JEE technologies?'))
	{
		//Applying JavaScript built-in sort() function
		var sortedjava = $('.ilearn_menu li').sort(function(o, n) {
		return $(o).text() < $(n).text() ? -1 : 1;
		});
		//Applying Style
		sortedjava.addClass('java');
		//Removing Node
		$('#sourceDiv').remove();
		//Appending the sorted data in targetDiv
		$('#targetDiv').append(sortedjava);
	}
});
</script>
</head>
<body>
<div id="sourceDiv">
<ul class="ilearn_menu">
<li>Struts</li>
<li>JSP</li>
<li>Servlets</li>
<li>EJB</li>
<li>RMI</li>
</ul>
</div>
<div id="targetDiv"></div>
</body>
</html>
```

Step 2: Output will appear as shown in the following figure

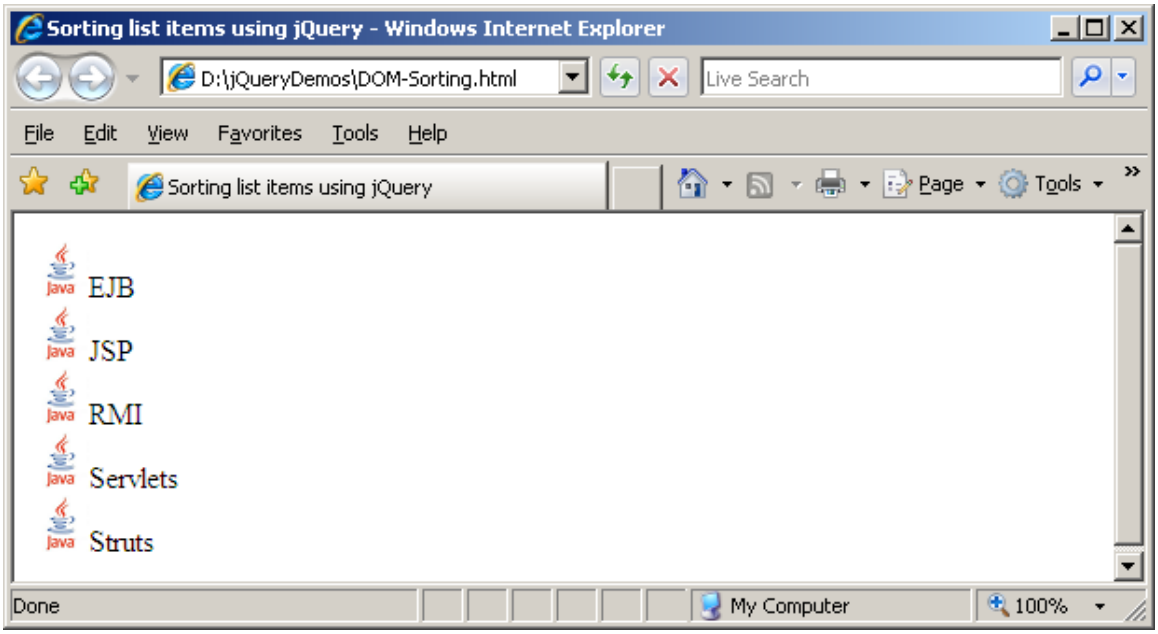

Figure 17 DOM-Sorting.html in browser

Note: Save this image  under D:\jQueryDemos\images

Stretched Assignments

Assignment-1: Sort the same list listed above in descending order

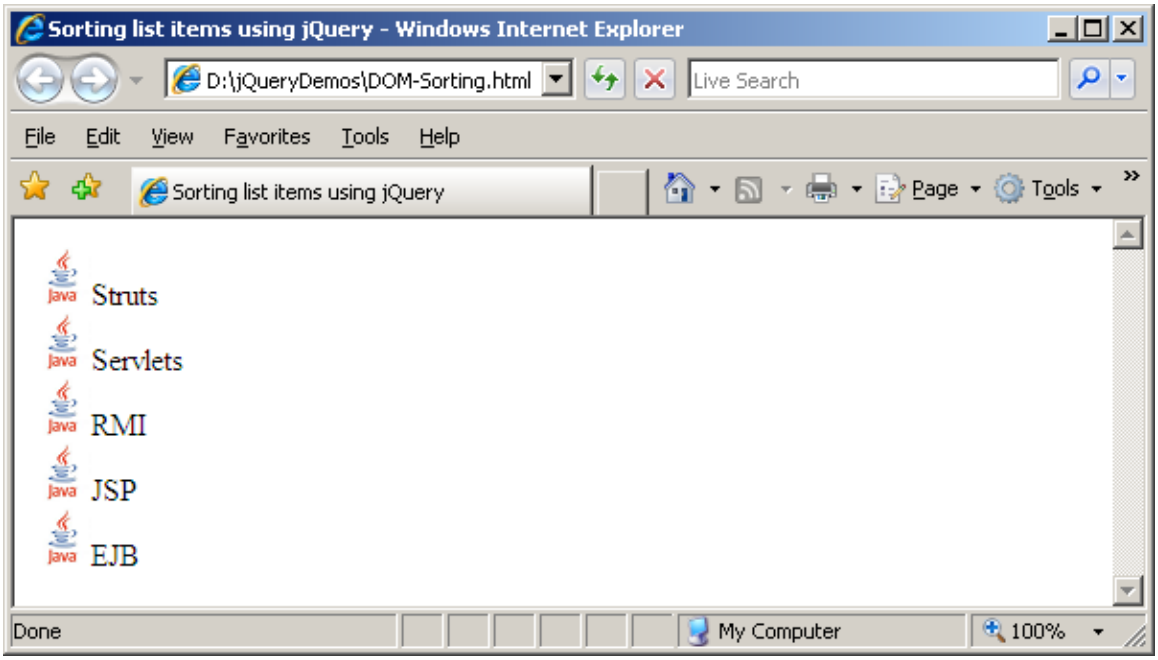

Figure 18 Sorting list items in descending order

TODO Assignments

Assignment-1: Create an unordered list like given below detach mainframe technologies list from last and add it before dotnet technologies

HTML Code

```
<div>
<ul class="ilearn_menu">
    <li class="dotnet">Dotnet Technologies
        <ul>
            <li>CSHARP</li>
            <li>ASP.NET</li>
        </ul>
    </li>
    <li class="jee">JEE Technologies
        <ul>
            <li>RMI</li>
            <li>Servlet</li>
        </ul>
    </li>
    <li class="mainframe">MainFrame Technologies
        <ul>
            <li>JCL</li>
            <li>COBOL</li>
        </ul>
    </li>
</ul>
</div>
```

Before Detaching

- Dotnet Technologies
  - CSHARP
  - ASP.NET
- JEE Technologies
  - RMI
  - Servlet
- MainFrame Technologies
  - JCL
  - COBOL

After Detaching

- MainFrame Technologies
  - JCL
  - COBOL
- Dotnet Technologies
  - CSHARP
  - ASP.NET
- JEE Technologies
  - RMI
  - Servlet

Figure 19 DOM Assignment

Lab 5. Handling Events

Goals	• At the end of this lab session, you will be able to • Handle events in jQuery • Perform animations using jQuery
Time	90 minutes

5.1: Adding and Removing Style at runtime using Click Event

Create a jQuery webpage to add and remove style from div tag when a button is clicked

Solution:

Step 1: Open any text editor like notepad and type the following code and save it with .html extension

```
<html>
<head>
<title>Click Event</title>
<style type="text/css">
.custom{background-color:blue;color:white;font-size:24pt;border:10px solid green}
</style>
<script type="text/javascript" src="scripts/jquery-1.12.3.min.js"></script>
<script type="text/javascript">
$(document).ready(function()
{
	$('#btnAddRemove').click(function(){
		$('#targetDiv').toggleClass('custom');
	});
});
</script>
</head>
<body>
<p>
<input type="button" id="btnAddRemove" value="Add Style / Remove Style"/>
</p>
<div id="targetDiv">
Div tag formatted with CSS Style
</div>
</body>
</html>
```

Step 2: Output will appear as shown in the following figure

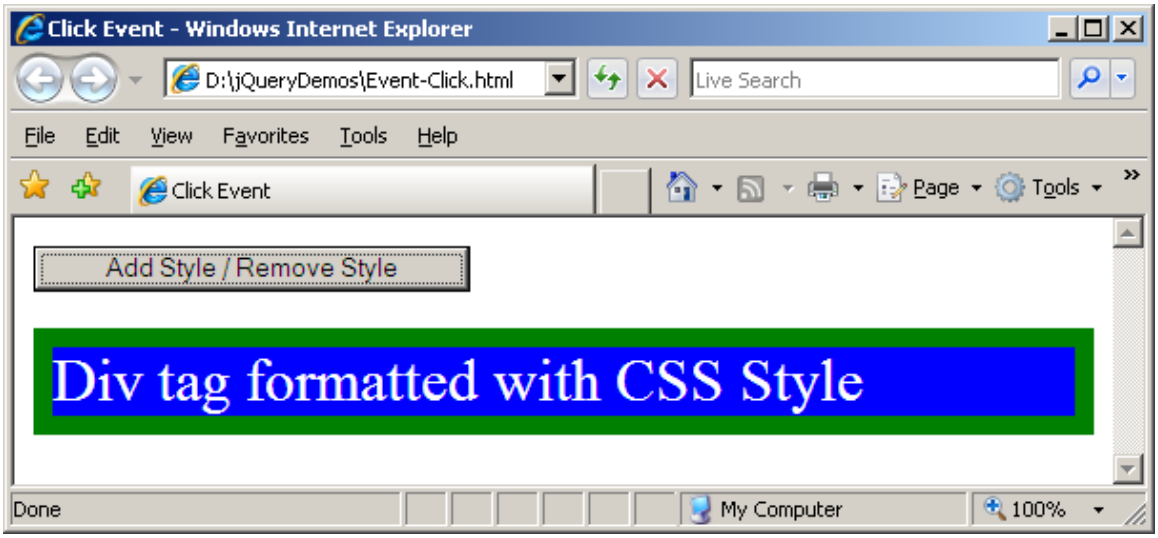

Figure 20 Event-Click.html in browser

5.2: Preventing Cut-Copy-Paste in textbox

Create a jQuery webpage which prevent Cut, Copy, Paste operations in a textbox

Solution:

Step 1: Open any text editor like notepad and type the following code and save it with .html extension

```
<html>
<head>
<title>Preventing Cut-Copy-Paste</title>
<script type="text/javascript" src="scripts/jquery-1.12.3.min.js"></script>
<script type="text/javascript">
$(document).ready(function()
{
		$('input[id="txtSample"]').bind('cut copy paste', function(e) {
			e.preventDefault();
			alert('You cannot ' + e.type + ' text!');
		});
});
</script>
</head>
<body>
<div>
Special Textbox : <input id="txtSample" type="text"/>
</div>
</body>
</html>
```

Step 2: Output will appear as shown in the following figure

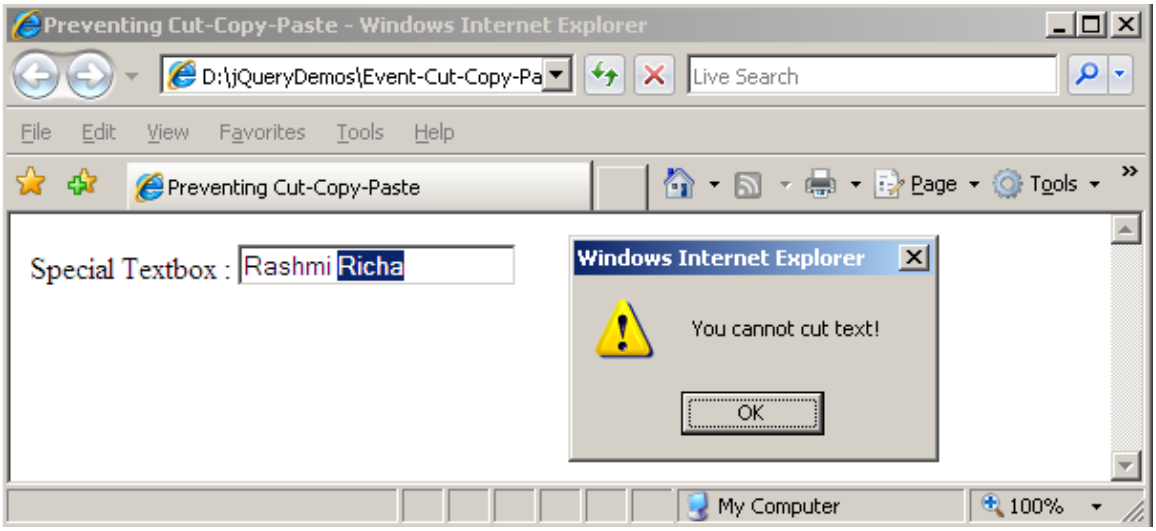

Figure 21Event-Cut-Copy-Paste.html in browser

5.3: Identifying mouse button

Create a jQuery webpage and identify which mouse button is pressed in the specified target

Solution:

Step 1: Open any text editor like notepad and type the following code and save it with .html extension

```
<html>
<head>
<title>Mouse Button identification</title>
<style type="text/css">
.custom{background-color:blue;color:white;font-size:24pt;border:10px solid
green}
</style>
<script type="text/javascript" src="scripts/jquery-1.12.3.min.js"></script>
<script type="text/javascript">
$(document).ready(function()
{
        $('#targetDiv').mousedown(function(event)
        {
                switch (event.which)
                {
```

```
                        case 1:
                                alert('Left mouse button pressed');
                                break;
                        case 2:
                                alert('Middle mouse button pressed');
                                break;
                        case 3:
                                alert('Right mouse button pressed');
                                break;
                        default:
                                alert('You have a strange mouse');
                }
        });
});
</script>
</head>
<body>
<div id="targetDiv" class="custom">
Click here to know which mouse button is pressed
</div>
</body>
</html>
```

Step 2: Output will appear as shown in the following figure

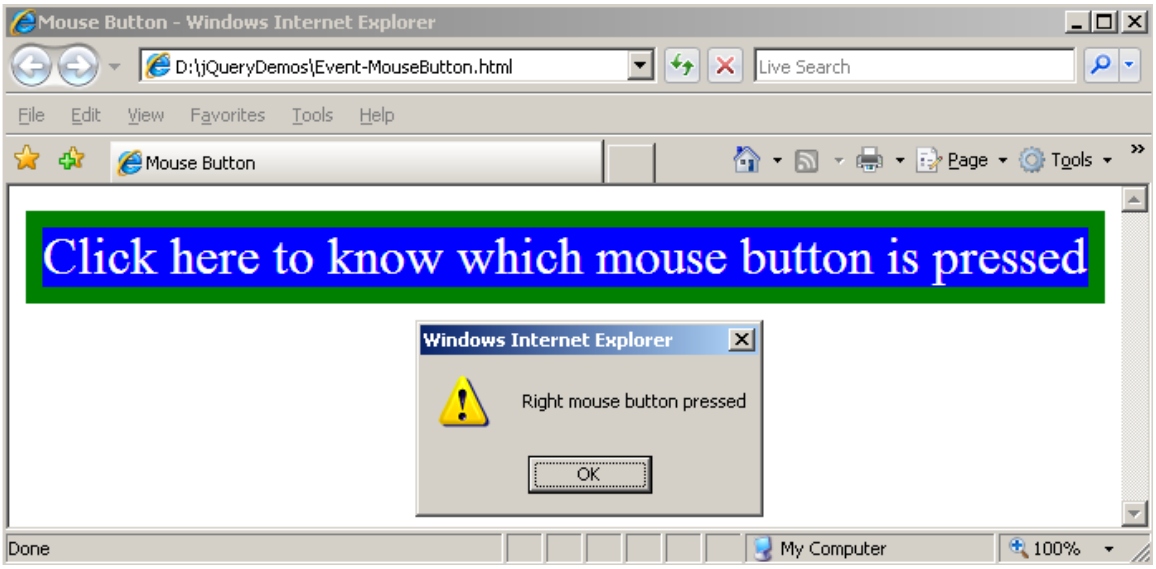

Figure 22 Event-MouseButton.html in browser

5.4: Limiting number of characters in textarea

Create a jQuery webpage which limits the number of character in textarea

Solution:

Step 1: Open any text editor like notepad and type the following code and save it with .html extension

```
<html>
<head>
<title>Limiting number of characters in textarea</title>
<style type="text/css">
.custom{background-color:red}
</style>
<script type="text/javascript" src="scripts/jquery-1.12.3.min.js"></script>
<script type="text/javascript">
$(document).ready(function()
{
	var limit = 25;
	var $tb = $('#comments');

	$tb.keyup(function(e) {
		var len = $(this).val().length;
		if (len > limit)
			this.value = this.value.substring(0, limit);
		else
			$('#charLeft').text(limit - len + " characters left");
	});
});
</script>
</head>
<body>
<div>
<h2>Type into this textbox which accepts 25 characters overall</h2>
<textarea id="comments" rows="5" cols="40"></textarea>
<br/>
<span id="charLeft"></span>
</div>
</body>
</html>
```

Step 2: Output will appear as shown in the following figure

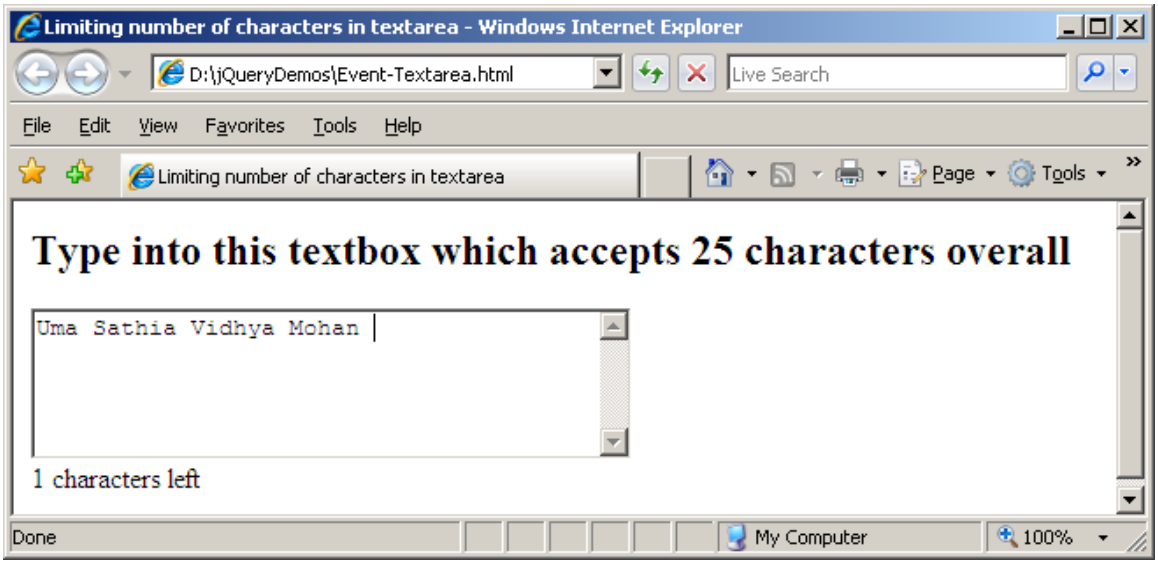

Figure 23 Event-Text area. html

5.5: Check/Uncheck all checkboxes at once

Create a jQuery webpage which Check/Uncheck all checkboxes at once

Solution:

Step 1: Open any text editor like notepad and type the following code and save it with .html extension

```
<html>
<head>
<title>Check/Uncheck all checkboxes at once</title>
<script type="text/javascript" src="scripts/jquery-1.12.3.min.js"></script>
<script type="text/javascript">
$(document).ready(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>
 <div class="smallDiv">
 <h2>Check/Uncheck all checkboxes at once</h2><br />
 <span title="Click here to check/uncheck all checkboxes at once">
<input id="chkAll" type="checkbox" name="chkAll" /><label>Do All</label>
</span>
<br /><hr />
<table id="cbList" class="chk" border="0">
<tr>
<td><input id="cbList0" type="checkbox" name="cbList0" /><label>Option
One</label></td>
</tr> <tr>
<td><input id="cbList1" type="checkbox" name="cbList1" /><label>Option
Two</label></td> </tr>
<tr>
<td><input id="cbList2" type="checkbox" name="cbList2" /><label>Option
Three</label></td></tr>
<tr>
<td><input id="cbList3" type="checkbox" name="cbList3" /><label>Option
Four</label></td> </tr>
</table>
</div>
</body>
</html>
```

Step 2: Output will appear as shown in the following figure

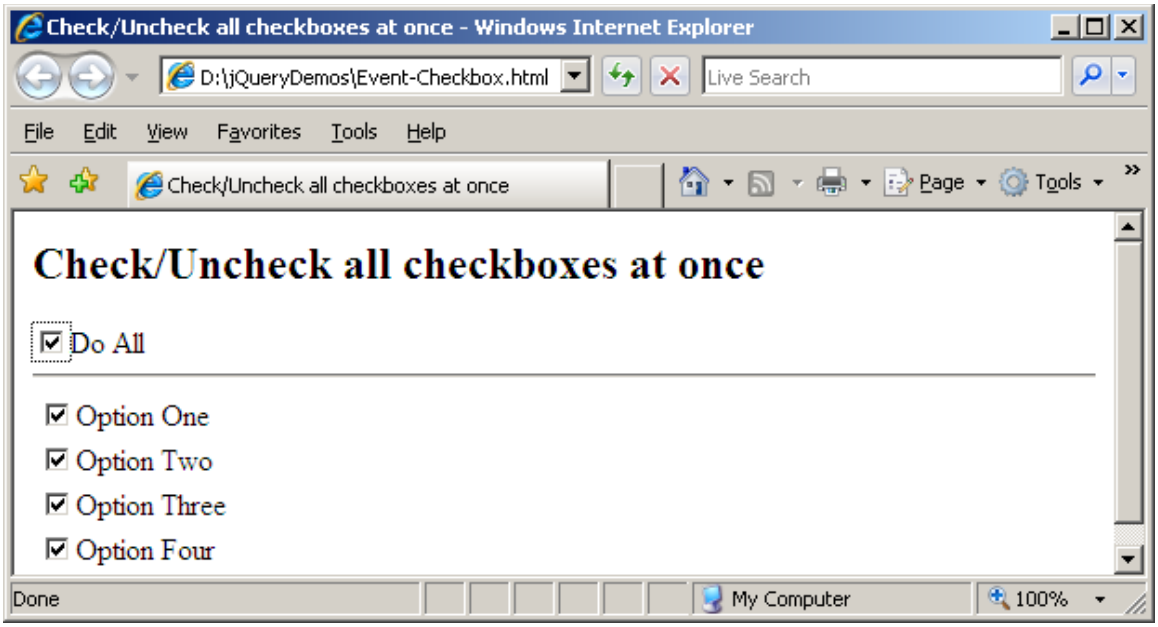

Figure 24 Event-Checkbox.html in browser

5.6: Event Triggering

Create a jQuery webpage which illustrates Event triggering (Invoke the second button Click event when the first button is clicked)

Solution:

Step 1: Open any text editor like notepad and type the following code and save it with .html extension

```
<html>
<head>
<title>Event Trigger</title>
<script type="text/javascript" src="scripts/jquery-1.12.3.min.js"></script>
<script type="text/javascript">
$(document).ready(function()
{
        $('#btnSecond').click(function(){
                alert('Second Button Clicked');
        });
        $('#btnFirst').click(function(){
                $('#btnSecond').trigger("click");
        });
});
</script>
</head>
<body><div>
<input id="btnFirst" type="button" value="First" />
<input id="btnSecond" type="button" value="Second" />
</div></body>
</html>
```

Step 2: Output will appear as shown in the following figure

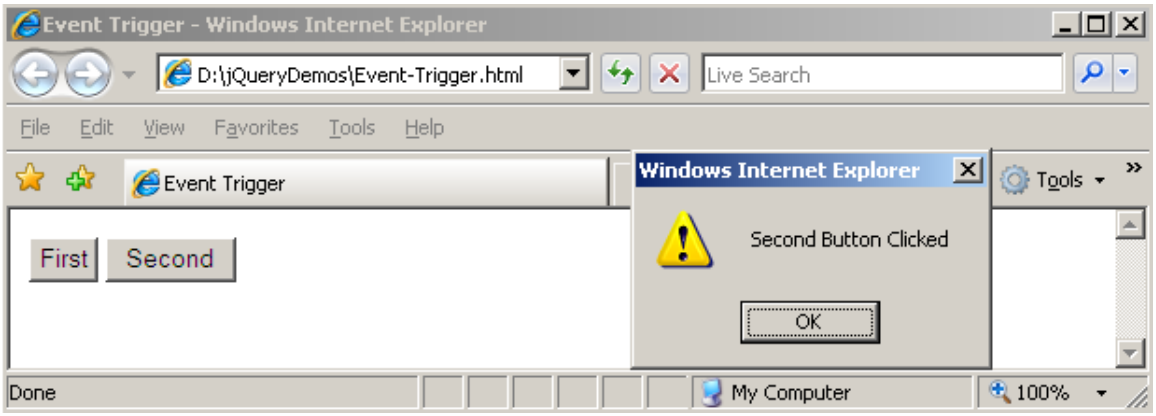

Figure 25 Event-Trigger.html in browser

TODO Assignments

Assignment-1: Create a jQuery webpage to Change the URL of a Hyperlink from google.com to bing.com and disable the button once it is clicked

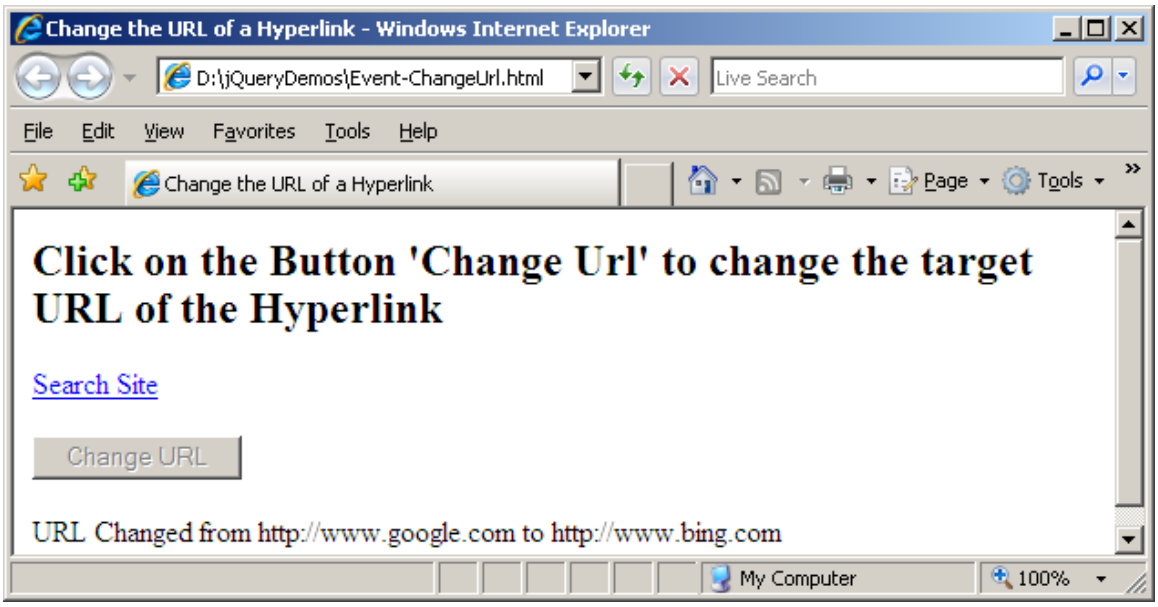

Figure 26 Assignment Output

Appendices

Appendix A: Coding Best Practices

Assign selectors to local variables instead of overusing it

```
$('#button').click(function()
 {
            $('#label').method();
            $('#label').method2();
}
```

```
$('#button').click(function()
 {
            var $label = $('#label');
            $label.method();
            $label.method2();
}
```

Best way is using chaining

```
$('#button').click(function()
{
            $('#label').method().method2();
}
```

All variables that are used to store/cache jQuery objects should have a name prefixed with a '$'.

```
var $MyTable = $("#MyTable");
```

Manipulation of the Document Object Model (DOM) can be costly and inefficient, regardless of whether it is undertaken through jQuery or JavaScript.

```
var $myList = $("#myList");
```

```
for (i=0; i<1000; i++){
   $myList.append("This is list item " + i);
}
```

This code adds 1000 lines to an HTML list. This is done with 1000 successive calls to the .append() method, and hence, 1000 manipulations to the DOM. Code needs to be re-written as given below

```
var $myList = $("#myList");
var li = "";
for (i=0; i<1000; i++){
  li += "<li>This is list item " + i + "</li>";
}
$myList.append(li);
```

Declare the styles in a class within a CSS StyleSheet file and use .addClass(), .removeClass() or .toggleClass() upon your selecteded object(s). For example, avoid this type of declaration:

```
  $("#MyTR").css({
    "background-color":"gray"
  });
```

Use instead:

```
  $("#MyTR").addClass("HighlightRow");

  /*In CSS File:*/
  .HighlightRow
  {
    background-color:gray;
  }
```

Use #ID selector wherever possible. It is the fastest. Ensure slower selectors are optimized for performance – Combine them with faster selectors where possible.

```
//Inefficient: scans DOM for all elements with oddrows class
```

```
$(".oddRows");

//More efficient: Searches only <tr> with oddrows class
$("tr.oddRows");

//More efficient: searches descendents of #MyTable
$("#MyTable tr.oddRows");

//Best: searches immediate children
$("#MyTable>tbody>tr.oddRows");
```

Don't use bare class selectors, like given below, this will end up looking at every single element to see if it has a class of "button".

```
$('.button').click(function() { /* do something */ });

//Instead use this
$('span.button').click(function() { /* do something */ });
```

Return 'false' to prevent default behavior

```
//use this

$('popup').click(function(){
   // Launch popup code
   return false;
});

//instead of this

$('popup').click(function(){
   // Launch popup code
});
```

While using $.ajax function for Ajax requests to server, you should avoid using the complete event to process response data. It will fire whether the request was successful or not. Rather than complete, use success.

Appendix B: Table of Figures