JQuery

New Class Notes

By

Rakesh Singh (Sir)

SRI RAGHAVENDRA XEROX

Software languages material available

Beside Bangalore Ayyangar bakery, opp: C-dac, Ameerpet, Hyderabad

Cell: 9951596199

31/7/15 RS = 90/=

INTRODUCTION to jQuery

- jQuery is a lightweight cross-browser javascript library.

- jQuery is the most popular javascript library in use today.

- It is also called "Write less, Do More Library".

- It was released in Jaunary 2006 at Barcamp NYC by "John Resig".

- Barcamp is an international network of user-generated conferences primarily focused around technology and web.

- jQuery is free, opensource software.

* The official defination of jQuery :

- jQuery is a fast, small and feature-rich Javascript library. It makes things like HTML document traversal and manipulation, event handling, animation and Ajax interactions for rapid web development and it is much simpler with an easy to use API that works across a multitude [cross-browser] of browsers.

- With a combination of versality and extensibility jQuery has changed the way that millions of people write Javascript.

- John Resig

Objectives : Design and build rich interactive web applications.

- Creating interactive user interface.

- The jQuery library makes it easy to manipulate a page of HTML after its displayed by the browser.

- It also provides tools that help you listen for a user to interact with your page, tools that help you to create animation in your page and tools that let you communicate with a server without reloading the page [means by Ajax]

WHO'S USING jQuery ?

- Many companies are using jquery including :

1. Microsoft
2. Amazon
3. Google
4. Facebook
5. Twitter
6. Bank of America
7. BBC
8. ESPN
9. CBS News
10. Diga
11. Reuters
12. IBM
13. IBM
14. Netfix
15. Dell
16. Oracle

and many more....

Features of jQuery [Syllabus]

- DOM Element Selections Functions.
- DOM Attributes
- DOM Traversal And Modification.
- Event Handling
- CSS Manipulation
- Effect And Animations
- jQuery UI
  1. UI Library Based Effects.
  2. UI Widgets
  3. UI Interactions
  4. UI Themes
  5.
- jQuery Ajax
- Extensibility Through Plugins
- Cross Browser Support.

3.08.15

The Latest Version of Jquery library :

1) 1.X Edition : jquery-1.11.3.js [Development version] uncompress code
(old to New) jquery-1.11.3.min.js [Production version]

2) 2.X Edition : jquery-2.1.4 js [Development version]
(All latest version). jquery-2.1.4.min.js [Production version]

- Now the downloaded library file of jquery, we can put [scripts] in particular Folder of our website to use it.

VS2013 (For creating website).

↳ file → New Website → ASP.NET Empty Web Site

[HTTP]

http://localhost/WebSite;jQuery 03 Aug 15

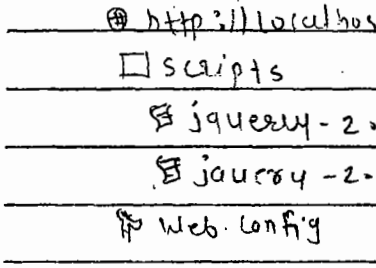
[OK]

Ex: Creating A new ASP.NET website in which we may use jquery library to perform any such operation based on client side script.

→ These are the following steps to create new ASP.NET website in the visual studio IDE.

1. Open the Visual Studio
2. Open Menu → File → New → Website, it opens a new window automatically called as "New Web Site" template window; in which select a language "Visual C#", then choose a template [ASP.NET Empty Web Site], select web location as "HTTP" then specify the website path [virtual path] with their name as [http://localhost/WebSite;jQuery 03 Aug 15].
3. Click Ok

4. Now open the solution explorer, right click on "Application Root", select → Add → New Folder then specify the folder name "scripts" and then downloaded jquery library file copy from the system and paste into the scripts folder.



HOW TO USE IQUERY LIBRARY

- The first and most important thing is to include jquery library in an .aspx page [Web page] or any html page as following: [Add-New Items-Web Form] and write following code in it.

```
<head>
<script src="scripts/jquery-2.1.4.js" type="text/javascript">
</script>
<script type="text/javascript">
// write JavaScript code using jQuery Library [jQuery code]
</script>
```

2nd Process of Downloading [Best option]

DOWNLOADING AND INSTALLING JQUERY LIBRARY VIA MANAGE NEW-G NUGET PACKAGES:

- It is a collection of tools to automate the process of downloading, installing, upgrading, configuring and removing packages from a visual studio

FINDING A PACKAGE:

- In Solution Explorer right click the Application Root and click "Manage Nuget Packages" and then locate the jQuery Library from the list of online packages and then click install button.

http://localhost/website/jquery03Aug15

web.config → Right Click → Add Nuget Packages → jQuery [Install] →

tick will come ☺ → Close after that all jquery library creates automatically.

- If you want to uninstall follow the same steps → Uninstall

4.8.15

USING JQUERY WITH A CDN: [Content Delivery Network]

- CDNs can offer a performance benefits by hosting jquery on servers spread across the globe. This also offer an advantage that if the visitor to your web page has already downloaded a copy of jquery from the same CDN, it won't have to be re-downloaded.

JQUERY'S CDN PROVIDED BY MaxCDN:

- To use the jquery CDN just refresh the file directly from http://code.jquery.com in the script tag as following:

<script src="http://code.jquery.com/jquery-2.1.4.js" type="text/javascript"></script> // uncompressed

<script src="http://code.jquery.com/jquery-2.1.4.

To see the all available file and version, visit http://code.jquery.com

OTHER CDN

- The following CDN also host compressed and uncompressed versions of jquery releases.

Note: There may be delays between a jquery release and its availability there. Please be patient, they receive the files at the same time the blog post is made public. Beta and relase candidates are not hosted by these CDNs.

1) Google CDN
2) Microsoft CDN
3) CDNJS CDN
4) jsDelivr CDN

1) Google CDN: All the hosted libraries on the google CDNs can be found in the google developers link [i.e http://developers.google.com/speed/libraries/].

- In this above link locate the jquery hosted library and then use it as like in the script tag of our web page.
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery.js" type="text/javascript"></script> //uncompressed

2) Microsoft CDN: All the hosted libraries on the microsoft CDN can be found in the following link i.e. http://www.asp.net/ajax/cdn

- In the above link locate the jquery releases on the CDN. and then used in the script tag as following:

<script src = "http://ajax.aspnetcdn.com/ajax/jQuery/jquery-2.1.4.js" type = "text/javascript"></script>

3> CDNJS CDN: All the hosted libraries on the CDNJS can be found in the following link
http://cdnjs.com/libraries/jquery/

- In the above link locate the jquery hosted library with version 2.1.4 or 1.11.3 and use in script tag as following
<script src = "https://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.4/jquery.js" type = "text/javascript"></script>

4> jsDelivr CDN: It is free open source CDN for javascripts library, jquery plugins, css frameworks and many more...

- These all hosted libraries can be found in the following link: http://www.jsdelivr.com, in which locate the jquery library, CDN files and use in a script tag as follows
<script src = "http://cdn.jsdelivr.net/jquery/2.1.4/jquery.js" type = "text/javascript"></script>

CALLING A JQUERY LIBRARY FUNCTION: As almost everything we do, when using jquery reads or manipulates the document object model [DOM] we need to make sure that this & we start adding events etc as soon as the DOM is ready.

- If we want an event to work on our page, we should call it inside the $(document).ready() function. Everything inside it will load as soon as the DOM is loaded.

- To do this we register a ready event method for the document object as following

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

```
<script type="text/javascript"> $(document).ready(function(){
// Do something when the DOM is Ready
});
</script>
```

JQUERY SYNTAX

- The jquery syntax is for selecting html document and perform some action on the selected element(s).

- The basic syntax is $(selector).action();

$: It is a selector function and it is a short hand notation of an actual jquery selector function i.e. jQuery()

- This function is used to select html document element [S].

- Selector is an expression allows us to select the element [S] and filter out the element.

- Action is nothing but jquery method inorder to perform any operation on the selected element.
Ex: $("P").hide(); It hides all the elements which are selected by or matched by <P> element.

Wed
5.8.15

JQUERY SELECTORS:

- Jquery Selectors are one of the most important parts of the jquery library

- JQuery selectors allows us to select and manipulate html DOM elements as a group or as a single element.

- Jquery selectors are required at every step while using jquery. selectors allows us to get the exact element we want from our html document.

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

- These are the following major selectors supported by jquery:
1. TagID - select an element by using their Id.
$("#") - It selects an element that an id of id1

2. TagName : Selects Elements by using their tagname.

$("div") - it selects all the elements matched by

$("input") - it selects all the elements matched by <input>

3. Class Name : select elements by using their class name.

$(".class1") - it selects all the elements that have class of class1

$("div.class1") - it selects all the elements matched by that have class of class1.

$("#id1.class1") - it selects an element that have an id of id1 with class of class1.

4. Multiple Elements : select multiple elements seperated by a (,) comma seperator

$("div, p") - it selects all the elements matched as well as <p>

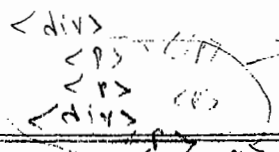
$(".class1, p, div") : it selects all elements that have class of class1 and all the elements matched by <p> as well as

5. Children Elements : select children elements.

$("div > p") : it select all elements matched by <p> that are children of each element matched by

~~$("p > *")~~

$("p > .class1") - it selects all elements that have class of class1 and that are children of each element matched by <p>.



6. Descendent Elements - Select descendent elements matched by <p> that are descendent elements.

7. All Elements: Select all elements

$("*") - it selects all elements of html dom

$("div > *") - it select all children elements of each element matched by .

$("div *") - it selects descendent elements of each element matched by

8. Current Element: selects current element into their action.

$(this) - it selects the current element into their operation.

alert function does not have any alternative in jQuery.

Examples: UI Design

jQuery Example

Enter Value 1:

Enter Value 2:

Add

Result:

Source view (default.aspx)

(body)

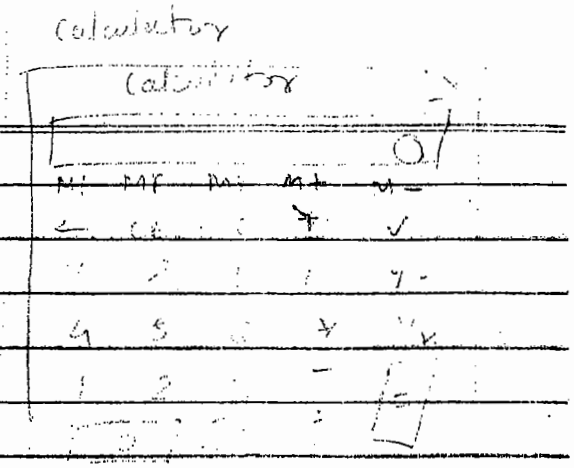
<h1> jQuery Example </h1>

 Enter Value 1 : <input type="text" id="txtValue1" />

 Enter Value2 <input type="text" id="txtValue2" />

```
<br/> <br/>
<input type="button" id="bthAdd" value="Add" />
<br/><br/>
<b>Result:</b><input type="text" id="txtResult" readonly=
"readonly" />
</div>

jQuery code
<head>
<script src="scripts/jquery-2.1.4.js" type="text/javascript">
</script>
<script type="text/javascript">
$(document).ready(function() { // To make sure whether the document is fully loaded or not

    $("#txtValue1").Focus();
    $("#bthAdd").click(function(){

    var value1=$("#txtValue1").val();
    var value2=$("#txtValue2").val();
    var error = "";
if (value1 == "")
error += "Please enter Value1" + "\n";
if (value2 == "")
error+="Please enter value2" + "\n";
if (error != "") {
    alert(error);
    return;
}
var result= parseFloat(value1) + parseFloat(value2);
```

```
$("#txtResult").val(result);
});
});
</script>
```



Create same example as above

Add | sub | mul | div

all button on click the validation

Single line of code should not be repeated

6.8.15

omit it

```
$("#<%= txtValue1.ClientID %>").focus();
$("#<%= btnAdd.ClientID %>").click(function() {
var value1 = $("#<%= txtValue1.ClientID %>").val();
var value2 = $("#<%= txtValue2.ClientID %>").val();
var value2 = $("#<%=
```

6/8/15

Working with ASP.NET element to select any element by using their id attribute

```
<div align = "center">
<h1> jQuery Example </h1>
<b> Enter Value1 : </b> <asp : TextBox ID = "txtValue1" runat = "server" />
<br /> <br />
<b> Enter Value2 : </b> <asp : TextBox ID = "txtValue2" runat = "server" />
<br /> <br />
<asp : Button ID = "btnAdd" runat = "server" Text = "Add" />
<br /> <br />
<b> Result : </b> <asp : TextBox ID = "txtResult" runat = "server"
ReadOnly = "true" />
</div>
```

Jquery Code

```
<script src = "Scripts/jquery-2.1.4.js" type = "text/javascript">
</script>
<script type = "text/javascript">
$(document).ready(function() {
$("#<%= txtValue1.ClientID %>").focus();
$("#<%= btnAdd.ClientID %>").click(function() {

var value1 = $("#<%= txtValue1.ClientID %>").val();
var value2 = $("#<%= txtValue2.ClientID %>").val();

    var error = "";
    if (value1 == "")
    error += "Please Enter Value1" + "\n";
    if (value2 == "")
```

```
    error += "Please Enter Value2" + "\n";
    }
    if (error != "") {
    alert(error);
    return false;
    }

    var result = parseFloat(value1) + parseFloat(value2);
    $("#<%= txtResult.ClientID %>").val(result);
        return false;
    });
    });
</script>
```

Example No: 2 <HTML Source>

```
<body>
    <p>This is first Paragraph</p>
    <p>this is second Paragraph</p>
    <p>This is third Paragraph</p>
    <br/>
<input type="button" id="btn1" value="Show Content" />
```

Jquery code

```
<script src="scripts/jquery-2.1.4.js" type="text/javascript
"></script>
<script type="text/javascript">
$(document).ready(function() {
    $("#btn1").click(function() {
        var contents = "";
$("p").each(function() {
```

```
contents += $(this).html() + "\n";
    });

alert("Paragraph contents are :\n" + contents);
    });
});
</script>
```

Example No 3 : HTML source

```
<div class="class1">
This is Div 1
</div>
<div class="class2">
This is Div 2
</div>
<div class="class3">
This is Div 3
</div>

<style type="text/css">
.class1
    width : 150px;
    height : 100px;
    line-height : 100px;
    text-align : center;
    border : 5px solid red;
    background-color : blue;
```

```
color : white;
margine : 5px;
cursor : pointer;
}
</style>

<script type:"text/javascript">
$(document).ready(function(){
$('div.class1').click(function(){
alert($(this).html());
});
});
</script>
```

Fri
7/18/15

SELECTORS

9. $("input") → it selects all elements matched by <input>

10. $("input:text") → it selects all elements matched by <input> type of text

OR

$(":text") → it select all element which are type of text.

11. $("input:button") → it select all elements matched by <input> type of button.

OR

$(":button") → it selects all elements which are type of button

12. $("input:submit") → it selects all element matched by <input> type of submit

OR

$(":submit") → it selects all elements which are type of submit.

13. $("input:reset") → it selects all elements matched by <input> type of reset

OR

$(":reset") → it select all elements which are type of reset.

14. $("input:checkbox") → it selects all elements matched by <input> type of checkbox.

OR

$(":checkbox") → It selects all elements which are type of checkbox.

15. $("input:checkbox[name=Options]") → It selects all elements matched by <input> type of checkbox that used name attribute value equal to Options. [It means, it selects a group of checkboxes]

16. $("input:checkbox[name=Options]:checked") → It selects a group of checkboxes that are checked.

17. $("input:checkbox[name=Options]").not(":checked") → It selects a group of checkboxes which are not checked.

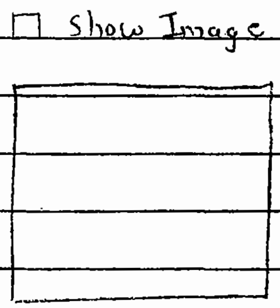
18. $("input:checkbox[name=Options]").is(":checked") → It checks whether any checkbox is checked in a group of checkboxes or not, if at least any one is checked returns true otherwise False.

19. $("input:checkbox[name=Options]").prop("checked", true) → It makes a group of checkboxes to be checked.

20. $("input:checkbox[name=Options]").prop("checked", False) → It makes a group of checkboxes to be unchecked.

unless complete page is fully not load we can't perform any operation on dom element

Examples : Design View



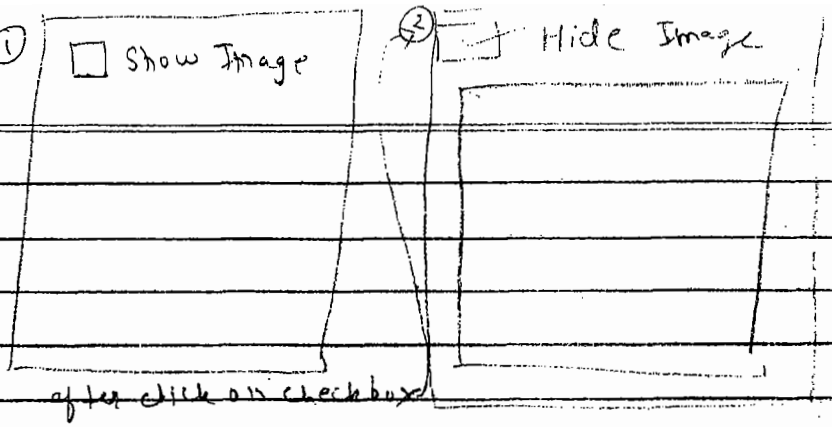
HTML source

```
<input type="Checkbox" id="chk1" />
<label For="chk1"> Show Image</label>
<br/>
<img id="img1" src="Images/Ads1.jpg" width="300" height=
"300" />
```

jquery code

```
<head>
<script src="Scripts/jquery-2.1.4.js" type="text/javascript">
</script>
<script type="text/javascript">
 $(document).ready(function(){
   $("#img1").hide();
   $("#chk1").change(function(){
   if($(this).is(":Checked")){
      $("#img1").show();
      $("label[For=chk1]").text("Hide Image");
    }
 else {
  $("#img1").hide();
  $("label[For=chk1]").text("Show Image");
   }
```

O/p: ① ☐ Show Image ② ☐ Hide Image




```
});
});
</script>
```

- Working above example using ASP.NET element.

```
<asp:CheckBox ID="chk1" runat="server" Text="Show Image" />
<br />
<asp:Image ID="img1" runat="server" ImageUrl="~/Images/Ads1.jpg" width="300" height="300" />

<script type="text/javascript">
$(document).ready(function(){
var img = $("#<%= img1.ClientID %>");  img.hide();
var chk = $("#<%= chk1.ClientID %>");
var lbl = $("lbl[for=" + chk.prop("id") + "]");
img.hide();
$("#<%= chk1.ClientID %>").change(function(){
    if ($(this).is(":checked")) {
        img.show();
        lbl.text("Hide Image");
    }
    else {
        img.hide();
        lbl.text("Show Image");
    }
});
});
</script>
```

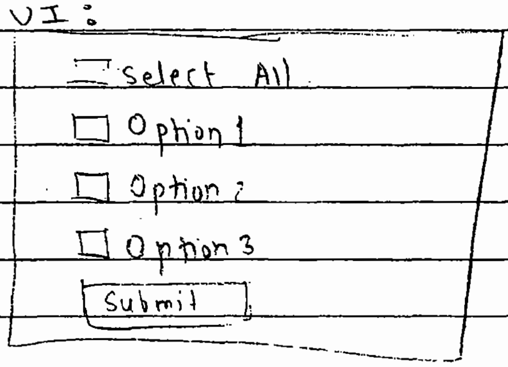
Same output will be generated as above.

8/8/15

```
{
var img .
var chk = $("#<%= chk1.ClientID %>");
var lbl = $("label[for=" + chk.prop("id") + "]");
image.hide();
chk.change( function(){
  if ( $(this).is(":checked")){
    img.show();
    lbl.text ("Hide Image");
  }
```

'id' for get prop pr
get value of certain prop.

UI:



HTML Source

```
<input type = "checkbox" id ="chkSelect"/>
<label For ="chkSelect"> Select All </label>
<br/>
<input type="checkbox" id="chk1" value ="Opt1" name=" options"/>
<label For="chk1">Option1 </label>
<br/>
<input type="checkbox" id="chk2" value="Opt2" name="Options"/>
<label For=" chk2">Option 2 </label>
<br/>
<input type="checkbox" id="chk3" value="Opt3" name="Options"/>
<label For=" chk 3"> Option3 </label>
<br/>
```

```
<input type="button" id="btnSubmit" value="submit" />
<br />
<span id="spanResult"></span>

jQuery code:

<script src="Scripts/jquery-2.1.4.js" type="text/javascript">
</script>
<script type="text/javascript">
$(document).ready(function() {
  $("#chkSelect").change(function() {
    if ($(this).is(":checked")) {
      $("input:checkbox[name=Options]").prop("checked", true);
$("input:checkbox[name=Options]").prop("checked", true);
$("label[for=chkSelect]").text("De-Select All");
}
else {
$("input:checkbox[name=Options]").prop("checked", false);
$("label[for=chkSelect]").text("Select All");
    }
  });

$("input:checkbox[name=Options]").change(function() {
  var chkCount = $("input:checkbox[name=Options]").length;
var chkCount = $("input:chkbox[name=Options]:checked").length;
    if (chkCount == chkdCount) {
      $("chkSelect").prop("checked", true);
$("label[for=chkSelect]").text("De-Select All");
    }
```

```
else {
$("#chkSelect").prop("checked", false);
$("label[for=chkselect]").text("select All");
    }
});

$("#btnSubmit").click(function() {
  if ($("input:checkbox[name=Options]").is(":checked")) {
    var texts="";
  $("input:checkbox[name=Options]:checked").each(function
    () {
  values+=$(this).val()+",";
  texts+=$("label[For=" + $(this).prop("id")+"]").text
    ()+",";
  });

values = values.substring(0, values.length-1);
texts = texts.substring(0, texts.length-1);

$("#spanResult").html("Selected Option Are :<br/>" + "Texts:"
+ texts+"<br/>" + "Values:" + values);
    }
  else {
$("#spanResult").html("<b style='color:red'>No option
Selected !!!</b>");
    }
});
});
</script>
```

o/p : Deselect Select All	Select All
Submit	Submit

Working with ASP.NET check box

select options

Option 1 Option 2 Option 3

submt

lblResult

```
<b> select options:. </b>
<br>
<asp: Panel ID="Panel1" runat="server">
<asp: CheckBox ID="Chk1" runat="server" Text="Option1" />
<asp: CheckBox ID="Chk2" runat="server" Text="Option2" />
<asp: CheckBox ID="Chk3" runat="server" Text="Option3" />
</asp: Panel>
<asp: Button ID="btnSubmit" runat="server" Text="submit"
OnClick="btnSubmit_Click" />
<br>
<asp: Label ID="lblResult" runat="server" />
```

Client Side Script

```
<script type="text/javascript">
$(document).ready(function() {
$("#<%= btnSubmit.ClientID %>").click(function() {
if ($("#<%= Panel1.ClientID %> > input:checkbox").
is(":checked") == false) {
```

```
$("#<%=lblResult.ClientID%>").html("");
alert("Please Select Option");
return false;
}
});
});
</script>
```

Server side script (Here we are checking that options are selected or not by server side)

```
Protected void btnSubmit_Click(object sender, Event Args e)
{
string selectedOptions = string.Empty;
foreach(Control ctrl in Panel.Controls)
{
if(ctrl is CheckBox)
{
CheckBox chk = (CheckBox) ctrl;
if(chk.Checked)
SelectedOptions += Chk.Text + ",";
}
}
if(selectedOptions != "")
{
selecteted Options = sclectedOptions.Substring(0, selectedOptions.
Length - 1);
lblResult.Text = "sclected options are:" + selectedOptions;
}
```

```
else {
lblResult.Text = "<b style='color:Red'>Please Select Option</b>";
}
}
```

Tue
11/8/15

Select Max 3 Options

☐ Option1 ☐ Option2 ☐ Option3 ☐ Option4 ☐ Option5

HTML

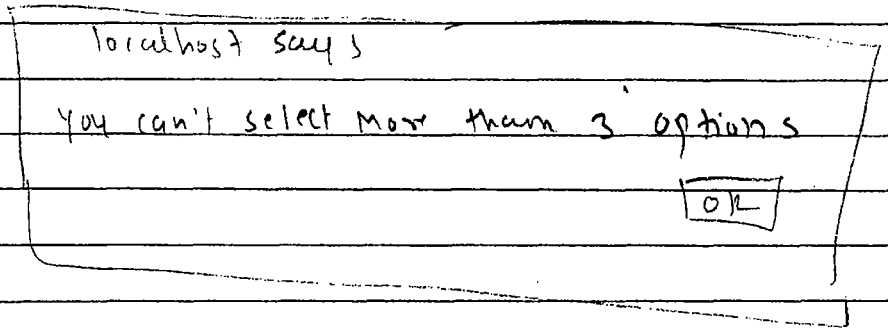
```
<b>Select Max 3 Options:</b>
<br/>
<input type="checkbox" id="chk1" value="Opt1" name="Options" />
<label for="chk1">Option1</label>
<input type="checkbox" id="chk2" value="Opt2" name="Options" />
<label for="chk2">Option2</label>
<input type="checkbox" id="chk3" value="Opt3" name="options" />
<label for="chk3">Option3</label>
<input type="checkbox" id="chk4" value="Opt4" name="options" />
<label for="chk4">Option4</label>
```

Jquery

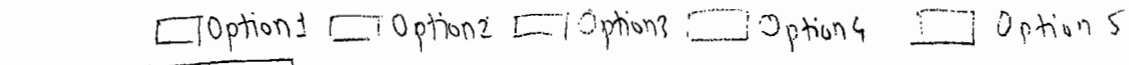
```
<script type = "text/javascript">
$(document).ready(function(){
$("input:checkbox[name=Options]").change(function(){
if($("input checkbox[name=Options]:checked").length>3){
alert("You can't select more than 3 options !!!");
$(this).prop("checked", false);
}
});
});
</script>
```

o/p

☐ Option 1 ☐ Option 2 ☐ Option 3 ☐ Option 4 ☐ Option 5



Select Max 2 & Max 3:



Submit

HTML Source

```
<body>
<form id = "form1" runat = "server">
<b> Select Min 2 & Max 3 Options : </b>
<br />
<asp:Panel ID = "Panel1" runat = "server">
<asp:CheckBox ID = "chk1" runat = "server" Text = "Option1" />
<asp:CheckBox ID = "chk2" runat = "server" Text = "Option 2" />
<asp:CheckBox ID = "chk3" runat = "server" Text = "Option 3" />
<asp:CheckBox ID = "chk4" runat = "server" Text = "Option 4" />
<asp:CheckBox ID = "chk5" runat = "server" Text = "Option5" />
</asp:Panel>

<asp:Button ID = "bthSubmit" runat = "server" Text = "Submit" />
```

Jquery

```
<script type = "text/javascript">
$(document).ready(function () {
$("#<%= Panel1.ClientID %> > input:checkbox:checked").
    length > 3) {

alert("You can't select more then 3 options");
    $(this).prop("Checked", false);
    }
});

$("#<%= bthSubmit.ClientID %>").click(function () {
if ($("#<%= Panel1.ClientID %> > input:checkbox:checked").
    length < 2) {
```

```
        alert(" Please select Min 2 Options")',
            return False;
        }
    });
});
</script>
```

o/p

Select Min 2 & Max 3 options:

☐ option 1 ☐ ☐ ☐ ☐

Submit

local host 5995
Please select Min 2 options

if you want to write this same program with ASP.NET so take ~~check list box~~ Checkbox list rather Panel.

NOTE: Working with a group of checkboxes in ASP.NET, it is recommended to use checkbox list control that represents a list of items in the form of checkbox which are already grouped and they renders depends on repeat layout property. Either in the form of html table, span, unorder list or order list.

WORKING WITH ASP.NET CHECK BOX LIST:

Select Options:

☐ option1
☐ option2
☐ option 3

Submit

```
<b>Select Options:</b>
<asp:CheckBoxList ID="chklist1" runat="server" RepeatLayout="Table">
<asp:ListItem Text="Option1" Value="Opt 1" />
<asp:ListItem Text="Option 2" Value="Opt 2" />
<asp:ListItem Text="option 3" value="Opt 3" />
</asp:CheckBoxList>

<input type="button" id="btnSubmit" value="Submit" />
</b>
</Spand id
Jquery

<script type="text
$(document).ready(function() {
$("#btnSubmit").click(function() {
$("#btnSubmit").click(function() {
if($("#chkList1.ClientID> input:checkbox").is
$("#chkList1.ClientID> input:checkbox:checked").e
(function() { var values="";
var texts="";
$("#chklist1.ClientID> input:checkbox:checked").
each(function() {
values += $(this).val() + ",";
texts += $("label[for='" + $(this).prop("id") + "']").text
() + ",";
});
});
```

```
texts = texts.substring(0, texts.length - 1);

values = values.substring(0, values.length - 1);
$("#span1").html("<b> Selected Options : </b> <br/>
<b> Texts : </b>" + texts + "<br/> <b> Values : </b>"
+ values);
}
else{
$("#span1").html("<b style='color:red'> No Option selected
</b>");
}
});
});
</script>
```

Olp : Select Options :

- [x] Option 1
- [ ] Option 2
- [ ] Option 3

submit

selected Options :
Texts : Option 1
Values : opt1

seleted option :
Text : Option 1
values : opt1

12/8/15

WORKING WITH RADIO BUTTON ELEMENTS:

21) $("input:radio") → It selects all the elements matched by <input> type of radio.

OR

$(":radio") → It selects all elements which are type of radio.

22) $("input:radio[name=Options]") → It selects a group of radio elements that have name attributes value equal to Options.

23) $("input:radio[name=Options]:checked") → It selects a group of radio elements that are checked

24) $("input:radio[name=Options]").not(":checked") → It selects a group of radio elements that are not checked.

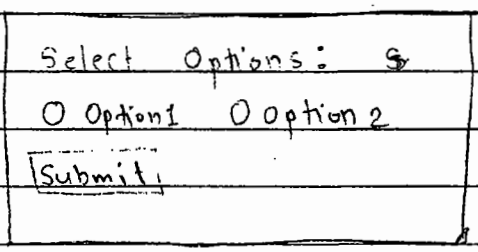
25) $("input:radio[name=Options]".is(":checked") → It checks whether any radio element in a group of radio elements is checked or not. If Yes returns true otherwise False.

26) $("#rdb1").prop("checked", true/false) → It applies checked property true or false to a particular radio element that has an id of rdb1

27) $("#rdb1").prop("checked") → It returns the value of checked property of an elements that has an id pf rdb1

Examples :

UI [.aspx]



HTML source

```
<b> Select Option : </b>
<br/>
<input type = "radio" id = "rdb1" value = "Opt1" name = "Options" />
<label for = "rdb1"> Option1 </label>
<input type = "radio" id = "rdb2" value = "Opt2" name = "Options" />
<label for = "rdb2"> Option 2 </label>
<br/>
<input type = "button" id = "btnSubmit" value = "Submit" />
<br/>
<span id = "spanResult"></span>
```

jQuery

```
<script type = "text/javascript">
$(document).ready(function () {
$("#btnSubmit").click(function () {
  if ($("input:radio[name=Options]").is(":checked")) {
    var rdb = $("input:radio[name=Options]:checked");
```

```
var value = "";
text = "";

value = rdb.val();
text = $("label[for=" + rdb.prop("id") + "]").text();

$("#spanResult").html("Selected Option:" + text + "/" + value);
}
else{
alert("Please Select Any Option");
return false;
}
});
});
</script>
```

o/p: Select Options:

⦿ Option 1 ○ Option 2

[Submit]

selected Option = option1 /opt1

The above code can be also done by follows:

```
<script type="text/javascript">
$(document).ready(function(){
$("#btnSubmit").click(function(){
var rdb = $("input:radio[name=Options]:checked");
if (rdb.val() == undefined) {
var value = ""; // dont put semicolon only comma.
text = "";
```

```
value = rdb.val();
text = $("label[for=" + rdb.prop("id") + "]").text();
$("#spanResult").html("Selected Options:" + text + "/"
+ value);
}
else{
alert("Please Select Any Option");
return false;
}
});
});
</script> // same o/p will generated as before.
```

* Example With ASP.NET Radio Button *

UI

select Option

○ Option1 ○ Option2

Submit

```
<b> select option : </b>
<br/>
<asp:RadioButton ID="rdb1" runat="server" Text="Option1"
GroupName="Options" />
<asp:RadioButton ID="rdb2" runat=
/>
<br/>
<asp:Button ID="btnSubmit" runat="server" Text="Submit" />
```

JQuery

```
<script type
$(document).ready(function(){
$("<%=btnSubmit.ClientID%>").click(function(){
if($("input:radio[name=Options]").is(":checked")==false)
{
alert("Please select Any One Option");
}
```

OR

```
if($("input:radio[name=Options]:checked").length==0){
alert("Please select Any One Option");
return false;
}
});
});
</script>
```

o/p

select option :

O option1 O option2

WORKING WITH ASP.NET RADIO BUTTON LIST [its already grouped]

UI [Radio Button List]

Select Option

O Option1

O Option2

O Option3

[Submit] → HTML Button

HTML Source

```
<b> Select Option </b>
<asp:RadioButtonList ID="rdbList1" runat="server" RepeatLayout="Table">
<asp:ListItem Text="Option1" Value="Opt1" />
<asp:ListItem Text="Option2" Value="Opt2" />
<asp:ListItem Text="Option3" Value="Opt3" />
</asp:RadioButtonList>
<input type="button" id="btnSubmit" value="Submit" />
<br />
<span id="Span1"></span>
```

JQuery

```
<script type="text/javascript">
$(document).ready(function() {
$("#btnSubmit").click(function() {
if ($('#<%=rdbList1.ClientID%> input:radio').is(":checked"))
{
    var text = "";
    value = "";
var rdb = $("#<%=rdbList1.ClientID%> input:radio:checked");
    value = rdb.val();
```

```
text = $("label[For=" + rdb.prop("id") + "]").text();
$("#span1").html("selected Option:" + text + "/" + value);
}
else {
$("#span1").html("<b style='color:red'> Please select
Any One option</b>");
return false;
}
});
});
</script>
```

o/p: select Option:

- ⊙ Option 1
- ○ Option 2
- ○ Option 3

[submit]

selected Option: Option 1/opt 1

From client side it will give output

```
<asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick=
"btnSubmit_Click" />
```

Server Side

Select Option

O

O

O

[Submit]

lblResult

} asp.Net element.

```
<asp:Button ID="btnSubmit" runat="server" Text="Submit"
OnClick="btnSubmit_Click" />
<br />
<asp:Label ID="lblResult" runat="server" />
```

JQuery

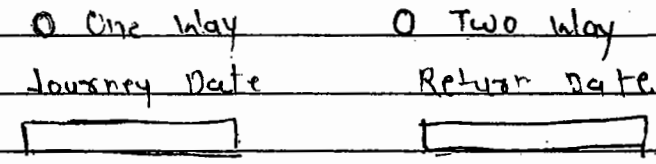
```
<script type="text/javascript">
$(document).ready(function() {
$("#<%=btnSubmit.ClientID%>").click(function() {
if ($("#<%=rdbtlist1.ClientID%> input:radio").is("checked")
== false) {
$("#<%=lblResult.ClientID%>").html("<b style='color:red'>
Please Select Any option!!! *</b>");
return false;
}
});
</script>
```

Server side code [aspx.cs]

```
Protected void btnSubmit_Click(object sender, EventArgs e)
{
ListItem selectedItem = rdblist1.SelectedItem;
if (selectedItem != null)
{
 lblResult.Text = "selected option:" + selectedItem.Text + "/"
  + selectedItem.Value;
 }
else
{
lblResult.Text = "<b style='color:red'> Please select
  Any one Option</b>";
   return;
  }
 }
```

Thursday
13/8/15

UI



HTML

```
<table>
  <tr>
    <td>
      <input type="radio" id="rdbOneWay" value="OneWay"
       name="Ticket" />
      <lbl for="rdbOneWay">One Way</label>
    </td>
    <td>
    <input type="radio" id="rdbTwoWay" value="Two Way"
    name="Ticket" /> // here Ticket is group name
    <lbl for="rdbTwoWay">Two Way</label>
    </td>
  </tr>
  <tr>
    <td>
      <b> Journey Date </b>
      <br/>
      <input type="date" id="txtJourneyDate" />
    </td>
    <td>
      <b> Return Date </b>
      <br/>
      <input type="date" id="txtReturnDate" />
    </td>
  </tr>
</table>
```

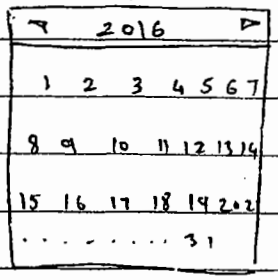
JQuery

```
<script type = "text/javascript>
$(document).ready(function () {
$("#rdbOneWay").prop("checked", true);
$("#rdbReturnDate").prop("disabled", true);

$("input:radio[name=Ticket]").change(function () {
$("#txtReturnDate").prop("disabled", $("#rdbOneWay").
    prop("checked"));
    });
});
</script>
```

output :

◉ One Way ○ Two Way

Journey Date Return Date



WORKING WITH SINGLE RADIO BUTTON AS TOGGLE OPTION

This is possible by ~~custom~~ nature [customization]

○ Show Image

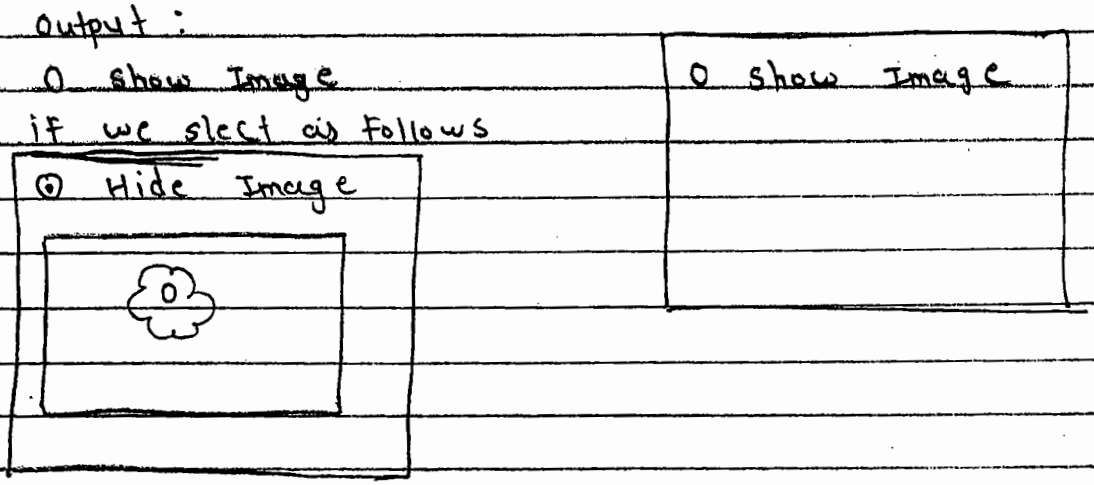
HTML Source

```
<input type="radio" id="rdb1" />
<label for="rdb1">Show Image</label>
<br /> <br />
<img id="img1" src="Images/Ads1.jpg" width="250" height="250" />
```

jQuery

```
<script type="text/javascript">
$(document).ready(function() {
 $("#img1").hide();
  var flag = true;
  $("#rdb1").click(function(){
      if(flag == true) {
     $(this).prop("checked", true);
     $("#img1").show();
     $("label[for=" + $(this).prop("id") + "]").text
         ("Hide Image");
           flag = false;
   }
   else {
   $(this).prop("checked", False);
   $("#img1").hide();
   $("label[for=" + $(this).prop("id") + "]").text("Show Image");
         flag = true;
   }
});
});
</script>
```

Output :



* WORKING WITH DROPDOWN LIST *

28) $("#ddl1 option") → It selects all options of a dropdownlist that has an id of ddl1.

29) $("#ddl1 option:selected") → It selects a selected option of a dropdownlist that has an id of ddl1

30) $("#ddl1 option:selected").val() → It returns the value of selected option in a dropdownlist

OR

$("#ddl1").val() → It returns the value of selected option in dropdownlist.

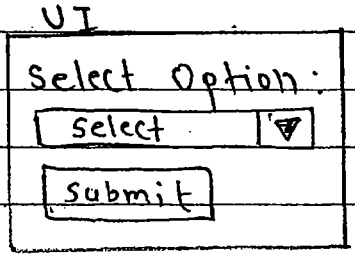
31) $("#ddl1").val("value") → It makes any particular option to be selected that matched with specified value.

32) $("#ddl1 option").not(":selected") → It selects all options that are not selected in a dropdown list.

33) $("#ddl1 option[value=opt1]") → It selects an option that has a value of opt1 in a dropdownlist.

Examples:

UI



HTML Source

```
<input type="dropdownlist
<b> select Option </b>
<select id="ddl1">
  <option value="0"> Select </option>
  <option value="Opt1"> Option 1 </option>
  <option value="Opt2"> Option 2 </option>
  <option value="Opt3"> Option 3 </option>
  <option value="Opt4"> Option 4 </option>
  <option value="Opt5"> Option 5 </option>
</select>
<br/>
<input type="button" id="btnsubmit" value="submit"/>
<br/>
<span id="span1"> </span>
```

JQuery

```
<script type = "text/javascript>
$(document).ready(function(){
$("#btnSubmit").click(function(){
    var opt = $("#ddl option:selected");
    if(opt.val() == "0"){
    $("#span1").html("<b style='color:red'>
Please select Any Option</b>");
    return false;
}
else {
$("#span1").html("<b> selected option:</b>" + opt.text()
    + "/" + opt.val());
}
});
});
</script>
```

Output:

Select Option: [option 4 ▽]

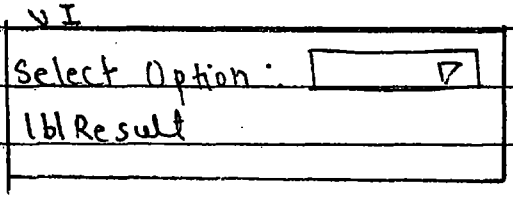
[submit]

Selected Option:

option4/opt4

* WORKING WITH ASP.NET DropDownList :-

UI



{ on select only gives the result [button is not use]

HTML

```
<b> Select Option </b>
<asp:DropDownList ID = "ddl1" runat = "server">
<asp:ListItem Text = "select" Value = "0" />
<asp:ListItem Text = "Option 1" value = "Opt1" />
<asp:ListItem Text = "Option 2" value = "Opt2" />
<asp:ListItem Text = "Option 3" value = "Opt3" />
<asp:ListItem Text = "Option 4" value = "opt4" />
</DropDownList>
<asp:lblResult = <asp:Label ID = "lblResult" runat = "server" />
```

jQuery

```
<input <script type = "text/javascript>
$(document).ready(function(){
  $("#<%= ddl1.ClientID %>").change(function(){

  var opt = $("#<%= ddl1.ClientID %> option' selected");
  if (opt.val() == "0") {
$("#<%= lblResult.ClientID %>").html("<b style=
'color: red' > Please Select Any option </b>");
    return false;
  }
else {
$("#<%= lblResult.ClientID %>").html("<b>
  Selected option: </b>" + opt.text() + " " + opt.val());
```

```
});
});
</script>
```

O/P

Select option: [option 3]
selected option: Option3 | opt3

14/8/15

WORKING WITH ListBox [Multiple Selection]

34> $("#listBox1 option") → It selects all the options of a listbox element that has an id of the listbox1.

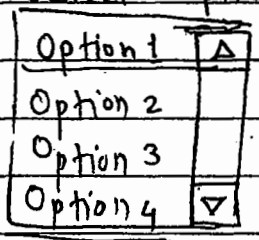
35> $("#listBox1 option:selected") → It selects all selected options of a listbox.

36> $("#listBox1 option").not(":selected") → It selects all options that are not selected of a listbox.

37> $("#listBox1 option").is(":selected") → It checks whether any option is selected or not. It returns true if, at deast one option is selected otherwise False.

Examples :

Select Options:



HTML :- Submit

```
<b> Select Option : </b>
<br/>
<select id = "listBox1" multiple = " multiple">
<option value = "Opt1" > Option 1 </option>
<option value = "Opt2" > Option 2 </option>
<option value = "Opt3" > Option 3 </option>
<option value = "Opt4" > Option 4 </option>
<option value = "Opt5" > Option 5 </option>
</select>
<br/>
<input type = " button" id = "btnSubmit" value = "Submit" />
<br/>
<span id = "span1" > </span>
```

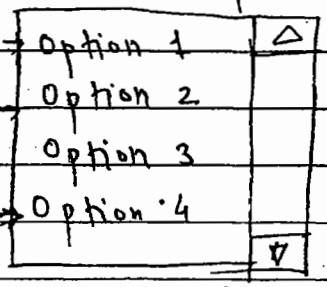
JQuery

```
<script type = " text/javascript>
$ (document). ready ( function () {
$ ("#btnSubmit"). click ( function() {
   if($("#listBox1 option"). is(" : selected")) {
        var texts = "";
        values = "";
$("#listBox1 option: selected "). each ( funtion () {
```

```
        values += $(this).val() + ",";
        texts += $(this).text() + ",";
    });

    values = values.substring(0, values.length - 1);
    texts = texts.substring(0, texts.length - 1);
    $("#span1").html("selected Options: <br/> Texts:" +
    texts + "<br/> Values:" + values);
    }
    else{
    $("#span1").html("<b style = color:red> No option selected
    </b>");
    return;
    }
    });
    });
</script>
```

output:

selected Options:

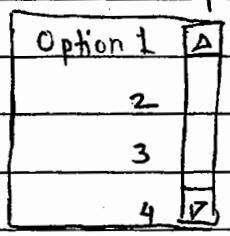


selected Options:

Texts : Option 1, Option 2, Option 4

values: Opt1, Opt2, Opt4

Select Options : } button is nt required, on selection only get the o/p



[lblResult]

HTML

```
<b> Select Options : </b>
<asp:ListBox ID ="ListBox1" runat="server" SelectionMode="
    Multiple">
<asp:ListItem Text ="Option1" Value ="Opt1"/>
<asp:ListItem Text="Option2" Value ="Opt2"/>
<asp:ListItem Text="Option3" Value ="Opt3"/>
<asp:ListItem Text ="Option4" Value ="Opt4"/>
<asp:ListItem Text ="Option 5" value="Opt5"/>
</asp:ListBox>
<br/>
<asp:Label ID="lblResult" runat ="server"/>.
```

JQuery

```
<Script type
$(document).ready(function(){
$("#<%=ListBox1.ClientID%> option").click(function(){
  var texts = "",
  Values ="";
$("#<%=ListBox1.ClientID%> option:selected").each(function
  (){
```

```
values += $(this).val() + ",";
texts += $(this).text() + ",";
});

if ($("#<%=ListBox1.ClientID%> option").is("selected")) {
texts = texts.substring(0, texts.length - 1);
values = values.substring(0, values.length - 1);
$("#<%=lblResult.ClientID%>").html("Selected Options:
<br/>Texts:" + texts + "<br/> Values:" + values);
}
else {
$("#<%=lblResult.ClientID%>").html("", "No option is selected")
}
});
});
```

→ this we can display msg.

</script>

output

Select Options:

→ Option 1

→ Option 2

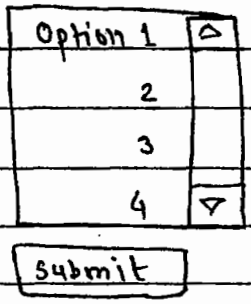
Option 3

Selected Options:

Texts: Option 1, Option 2

Values: Opt1, Opt2

Select Min 2 Options & Max 3 options:



HTML

```
<b> select Min 2 Options & Max 3 Options </b>
<br/>
<select id="listBox1" multiple="multiple">
<option value = "opt1"> Option1 </option>
<option value = "opt 2"> Option 2 </option>
<option value = "opt 3" > Option3 </option>
<option value = "opt 4" > Option4 </option>
<option value = "opt 5" > Option5 </option>
</select>
<br/>
<input type="button" id = "btnSubmit" value = "Submit" />
<br/>
<span id = "span1" </span>
```

JQuery

```
<script type = "text/javascript>
$(document).ready(function() {
$("#listBox1 option").click(function(){
  if($("#listBox1 option:selected").length > 3){
  alert("Please Select Max 3 options only");
  $(this).prop("selected", false);
    return;
}
```

```
});
$("#btnSubmit").click(function(){
if($("#listBox1 option:selected").length < 2)
{
  alert("Please select Min 2 Options");
  return;
}

var texts = "",
  values = "";
$("#listBox1 option:selected").each(function(){
  values += $(this).val() + ",";
  texts += $(this).text() + ",";
});
texts = texts.substring(0, texts.length-1);
values = values.substring(0,values.length-1);

$("#span1").html("Selected Options:-<br/>Texts: "
+texts + "<br/>Values:" + values);
});
});
</script>
```

o/p :

17.8.15

There are the following more selector can be used or supported by Jquery used to selecting down element:

1) $("li:not(.myclass)") : It selects all elements match by li that are not use in class of myclass.

OR

$("li").not(".myclass")

2) $("ul li:first") → It selects first li element which is deceendent of each element matched by ul.

3) $("li > ul") → It selects all the elements matched by ul that are childvens of each element matched by li.

4) $(":empty") – It selects all the elements that have children including text also.

5) $("p:empty") – It selects all the elements matched by p that have no children.

6) $("input[name=myname]") – It selects all elements matched by input that have a name value exactly equal to myname

7) $("input[name^=myname]") – It selects all elements matched by input that have a name value beging with myname.

8> $('input[name*=myname]') - It selects all elements matched by input that have a name value containing myname

9> $("li:even") - It selects all the elements matched by li that have an even index value.

=	→ myname
^=	→ mynameabc
*=	→ abcMyname123

examples.

10> $("tr:odd") - It selects all the elements matched with

tr 1 - index value 0
tr 2 - index value 1
tr 3 - index value 2
tr 4 - index value 3
tr 5 - index value 4
tr 6 - index value 5

11> $("li:first") - It selects all first li elements.

12> $("li:last") - It selects last li element.

13> $("li:visible") - It selects all the element matched by li that are visible.

14> $("li:hidden") - It selects all the element matched by li that are hidden.

15> $("li:eq(2)") - It selects all elements matched ~~equivalent to~~ by li that have an index value equal to 2.

16> $("li:lt(2)") - It selects all the elements matched by li that have an index value less than 2
[it means it selects first two li elements]

17> $("p:

18> $("li:first-child") - It selects all elements matched by li that are the first child of their parent.

19> $("li:last-child") - It selects all elements matched by li that are the last child of their parent.

20> $(":parent") - It selects all element that are the parent of another element, including text.

21> $("p:parent") - It selects all element that matched by p that are parent of other element
ex:
Hello → select because it has children.

22> $("li:contains

Tuesday
18.08.15

DOM Attributes In jQuery.

- jquery provides several methods to handle the attributes of an element or set of matched element(s). These are following :

1> Applying Attribute :

attr("attrName", value) → It is used to apply the specified attribute with their value to set of matched element (s).

Syntax :

```
$(selector).attr("attrName", value);
```

2> Applying Multiple Attributes:

attr({"attrName": value, "attrName": value}) →

It is used to apply multiple attributes with their values to a set of matched elements.

Syntax:

```
$(selector).attr({"attrName": value, "attrName": value});
```

3> Retreiving The Attribute Values:

attr("attrName");

This method returns the specified attribute value of first matched element.

syntax :

```
var value = $(selector).attr("attrName");
```

4) Removing the Attribute(s):

removeAttr("attrName")

This method is used to remove the specified attribute of set of matched element(s).

Syntax:

```
$(selector).removeAttr("attrName");
```

NOTE: Even we can remove multiple attributes also as following:

```
$(selector).removeAttr("attrName1 attrName2 attrName3");
```

- These all above methods can be also handled using following enhanced methods that are ~~available~~ available from jquery 1.6+ version onwards:

1. prop("propName", value)

This method is used to apply a property attribute with their value to a set of matched element(s).

Syntax:

```
$(selector).prop("propName", value)
```

2) Prop("propName": value, "propName": value})

It is used to applies multiple properties attributes with their values to a set of matched element(s).

Syntax:

```
$(selector).prop({"propName": value, "propName": value});
```

3) prop("propName")

It is used to return the value of specified property attribute of first matched element

Syntax:

```
var value = $(selector).prop("propName");
```

4) removeProp("propName")

It is used to remove the specified property of a set of matched element(s)

Syntax:

```
$(selector).removeProp("propName");
```

NOTE: This method can be also used for removing multiple properties attributes as following:

```
$(selector).removeProp("propName1 propName2 propName3");
```

WORKING WITH CSS CLASSES

1. Applying CSS Class:

addClass("className") This method is used to add the class to a set of matched element(s)

Syntax:

```
$(selector).addClass("className");
```

NOTE: Even multiple classes can be also added by specifying with white space seperator as follows:

```
$(selector).addClass("className1 className2 className3");
```

This all classes is added to the set of matched elements.

2. Removing the CSS Class:

removeClass("className") It is used to remove the specified class of a set of matched element(s).

Syntax:

```
$(selector).removeClass("className");
```

NOTE: Even multiple classes can be also remove. by as follows:

```
$(selector).removeClass("className1 className2");
```

3> Toggle Class:

toggleClass("className") → This method is used to toggle the specified class. It means if the

specified class is already added, then it removes otherwise it adds it.

Syntax:

$(selector).toggleClass("className");

4. To Check Whether Class is Present or Not :

hasClass("className") → This method is used to checks whether the specified class is present or not. It returns either true or False

Syntax:

$(selector).hasClass("className")

Example

HTML Source

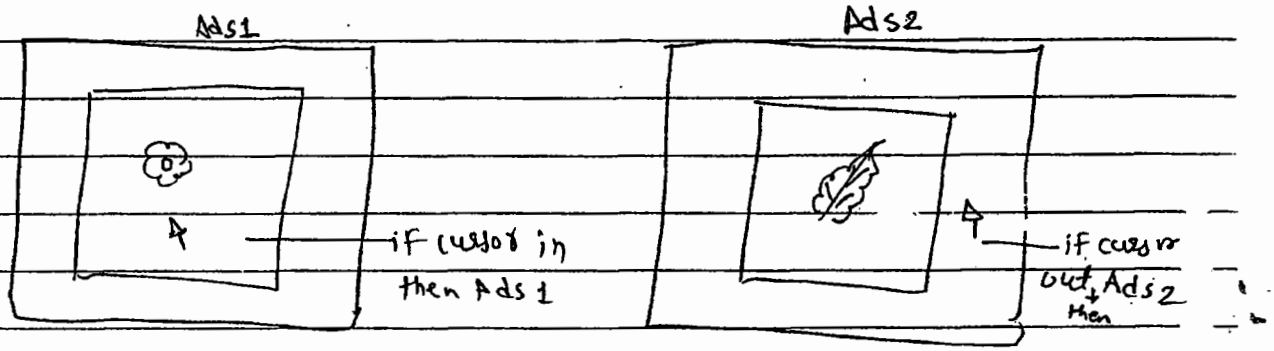
```
<div align="center">
<img id="img1" src="Images/Ads1.jpg" width="300"
height="300"/>
</div>
```

Jquery

```
<script type="text/javascript">
$(document).ready(function(){
var src="";
$("#img1").mouseover(function(){
src=$(this).attr("src");
$(this).attr("src","Images/Ads2.jpg");
```

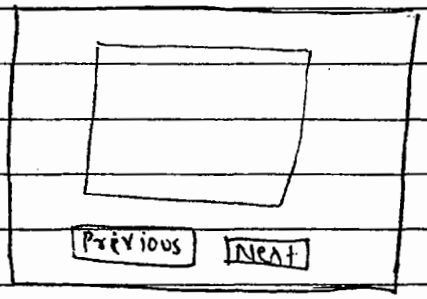
```
});.mouseOut(function(){
$(this).attr("src",src);
});
});
</script>
```

O/p:



HTML UI:

2.



HTML

```
<div align="center">
<img id="img1" src="Images/Ads1.jpg" width="300" height="300" />
<br><br>
<input type="button" id="btnPrev" value="Previous" />
```

```
 
<input type="button" id="btnNext" value="Next" />
</div>

jquery
<script type="text/javascript>
$(document).ready(function() {
$("#btnPrev").attr("disabled", true);
var count = 1;
$("#btnNext").click(function(){
if (count <5) {
count++;
$("#img1").attr("src","Images/Ads" + count + ".jpg");
$("#btnPrev").attr("disabled", false);
if (count == 5)
$("#btnNext").attr("disabled", true);
}
else{
$("#btnNext").attr("disabled", true);
}
});
$("#btnPrev").click(function() {
if (count>1) {
count--;
$("#img1").attr("src", "Images/Ads" + count + ".jpg");
$("#btnNext").attr("disabled", false);
if (count == 1)
$("#btnPrev").attr("disabled", true);
}
```

});
});
</script>

Output

Wednesday 19.08.15

Program for automatic image come [slideshow]

UI [HTML]

```
<div align="center">
<img id="img1" src="Images/Ads1.jpg" width="300" height="300" />
</div>
```

Jquery

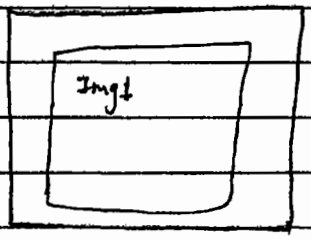
```
<script type
$(document).ready(function(){
    var count = 1;
function showImages(){
    count++;
$("#img1").attr("src","Images/Ads" + count + ".jpg");
    if(count==5)
        count=0;
}
```

window.setInterval(showImages, 1000); // javascript code

});

</script>

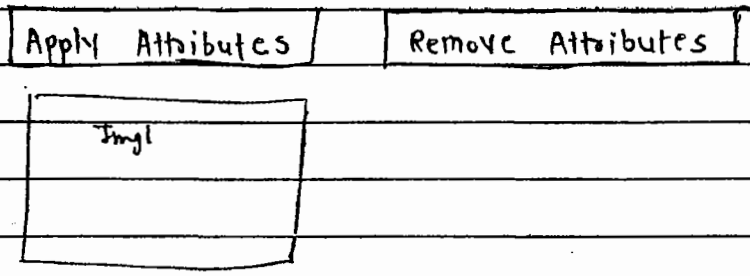
output



→ As o/p image can be change one by one automatically.

Img1, Img2 ..... Img5

UI




```
<input type = "button" id = "btn1" value = "Apply Attributes" />
 
<input type = "button" id = "btn2" value = "Remove Attributes" />
<br /><br />
<img id = "img1" src = "Images/Ads1.jpg" />
```

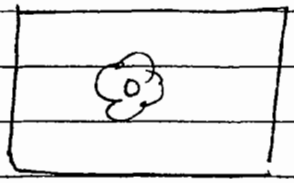
Jquery

```
<script type = "text/javascript">
$(document).ready(function () {
$("#btn1, #btn2").click(function() {
var Id = $(this).attr("id");
if(Id == "btn1") {
```

```
$("#img1").attr({"title":"Image Element", "alt":"Image",
"width":"300","height":"300", "style":"border:10px solid
green; border-radius:30px"});
  }
else if (Id == "btn2"){
$("#img1").removeAttr("style width height");
  }
});
});
</script>
```

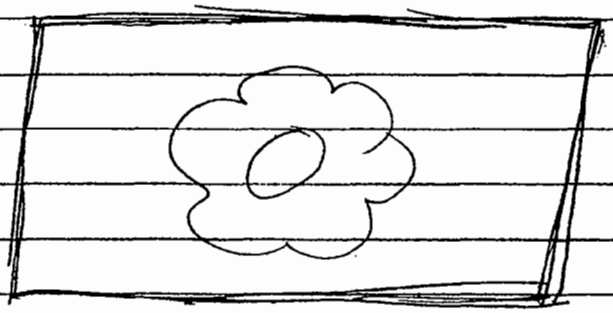
Output :

Apply Attributes | Remove Attributes

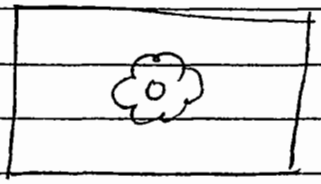


→ by default img1

After Applying Attributes become



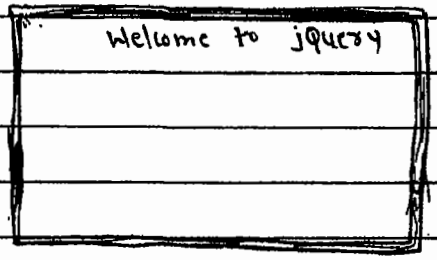
After Removing Attributes become



→ height change, width change, border change.

Examples of CSS classes:

Add Class | Remove Class | Toggle Class




```
<div align="center">
<input type="button" id="btn1" value="Add Class" name="btn"/>
 
<input type="button" id="btn2" value="Remove Class" name="btn"/>
 
<input type="button" id="btn3" value="Toggle class" name="btn"/>
<br/><br/>
<div id="divMain" style="border:10px solid green; width:
300px; height:300px">
Welcome to jQuery
</div>
</div>

<style type="text/css">
.class1{
background-color: yellow;
color: red;
border-radius: 50px;
box-shadow: 30px 15px red;
line-height: 300px;
}
</style>
</head>
```

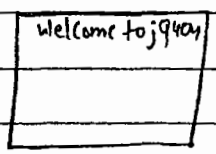
Jquery

~~<input~~ <script type=

```
$(document).ready(function(){
$("input:button[name=btn]").click(function(){
switch($(this).prop("id")){
case "btn1":
$("#divMain").addClass("class1");
break;
case "btn2":
$("#divMain").removeClass("class1");
break;
case "btn3":
$("#divMain").toggleClass("class1");
break;
}
});
});
```

</script>

Add Class Remove Class Toggle Class

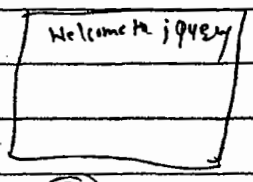


- by default

After Add class

①

After Remove Class



②

Toggle Class Means either ① or ②

H.W.

Welcome to jQuery Library → blinking text after interval.

20.8.15

Welcome to jQuery

HTML

```
<div id="divMain">
Welcome to jQuery
</div>
```

CSS

```
<style type="text/css">
#divMain {
  text-align: center;
  margin: 10px;
  padding: 0 px;
  font-weigth: bold;
  height: 60px;
  line-height: 60px;
}
.class1 {
  border: 10px solid blue;
  border-radius: 20px;
  background-color: red;
  color: yellow;
  font-size: 35 px;
  Font-family: 'Book Antiqua';
  box-shadow: 10 px 15px gray;
}

.class2 {
  border: 15px Solid red;
}
```

```
border-radius: 30px;
background-color: blue;
color: white;
font-size: 50px;
font-family: 'Book Antiqua';
box-shadow: 10px 20px silver;
}
</style>

jquery

<input type="text/javascript">
<script>
$(document).ready(function() {
$("#divMain").addClass("class1");
function blink() {
if ($("#divMain").hasClass("class1")) {
$("#divMain").removeClass("class1").addClass("class2");
}
else {
$("#divMain").removeClass("class2").addClass("class1");
}
}
window.setInterval(blink, 500);
});
</script>
```

DOM TRAVERSAL IN JQUERY [Imp]

- jquery provides a variety of DOM traversal methods to help us select elements in a document randomly as well as in sequential manner.

- Most of the DOM traversal method do not modify the jquery object and they are used to filter out elements from a document based on given conditions

* Finding element by index :

- jQuery provides a method i.e. eq(index), used to find the element with any. the specify index number in a set of matched element.

Syntax:

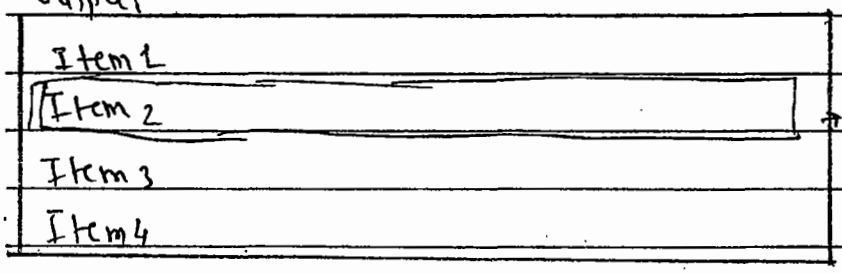
$(Selector).eq(index)

Examples:

```
.aspx
<ul>
  <li> Item1 </li>
  <li> Item 2 </li>
  <li> Item 3 </li>
  <li> Item 4 </li>
</ul>

<style type="text/css">
 .highlight {
   border: 2px solid red;
   color: green;
   font-size: 15px;
 }
</style>
```

```
<script type = "text/javascript">
$(documen
$("li").eq(1).addClass("highlight");
});
<script>
```

Output



→ it will highlight 2nd item.

caz index starts from 0
1, 2, 3. —

FILTERING OUT ELEMENTS

- jquery provides a method i.e- filter(selector), used to filter out all elements from the set of matched element that match the specified selector(s).

Syntax: $(selector).filter(selector)

Example1:

$("li").filter(".class1") → It filters all the elements matched by class1 from a set of element matched by li.

Example2:

```
var chkItems = $("input:checkbox[name=Options]");
```

var chkItems.filter(":checked"); → It filters checked element from the set of matched element which are store in the form of jQuery object.

~~Example 3:~~

Locating Descendent Element:

- jQuery provides the method i.e. find(selector) used to locate all the descendent element of a particular type of elements.

Sytax: $(Selector).find(selector)

Ex. $("p").find("span").addClass("highlight"); → It applies the highlight class to all the descendent elements matched by span which are found in each element matched by p.

21/08/15

is(selector) → This method is used to checks whether anyone one element in a set of matched element matched with specified selector(s) or not. It returns either true or false.

Syntax: $(Selector).is(Selector)

Ex: $("div").is(".class1") → It returns true if atleast any one div element uses a class of class1 in the set of matched elements by div otherwise, it returns false.

$("input:checkbox[name=Options]").is(":checked") It returns true if, atleast any one checkbox is checked in a group of checkboxes, otherwise returns false.

not(selector) - This method is used to selects those elements from the set of matched elements which are not matched with specified selector(s).

Syntax: $(selector).not(selector)

Ex: $("p").not(".class1") - It selects all the elements matched by p that are not uses a class of class1.

$("input:checkbox[name=options]").not(":checked");
It selects a group of checkboxes which are not checked

slice(start,[end]) - This method is used to selects a set of subset elements from the set of matched element.

Sytax: $(selector).slice(start,[end])

Ex: $("li").slice(1,5) - It selects a set of elements matched by li that has starting index 1 to till the ending index (5-1).

$(li).slice(1) - It selects a set of elements matched by li that has starting index1 till the end.

* add(selector) - This method is used to add elements to the set of matched elements and creates a new jquery object.

Syntax: $(selector).add(Selector)

Ex: $("div").add("p").addClass("highlight") - It applies the highlight class to all the elements matched by "p" as well as "div".

.addBack() - This method is used to add privious set of elements on the stack to the current set, optionally filtered by a selector.

Syntax: $(selector).addBack()

$("div").find("p").addBack().addClass("highlight")
It applies the highlight class to all the current element matched by "p" as well as their privious element matched by "div" which are added on the stack to the current set of element.

andSelf() - It is same as addBack() used to add the privious set of elements on the stack Current set of elements.

Children(selector) - This method is used to gets the children of each element in the set of matched elements, optionally filtered by the selector.

Sytax: $(selector).children([selector])

("li").children("ul") – nested ("ul").children("li") – normal.	ul does not contain anything except li. $("li").children("input:radio") – wrong.

$("p").children() – It selects all the children elements of each element matched by "p"

$("div").children("p") – It selects all the children elements matched by "p" ~~that are~~ of each element matched by "div".

- parent([Iselector]) – This method is used to gets the parentes of each element in the current set of matched element, optionally filtered by a selector.

Syntax: $(selector).parent([selector])

Ex: $("p").parent() → It selects all the parent of each element matched by "p".

22/08/15

$("p").parent("div") – It selects all the parent elements of p matched by div of a set of matched elements ~~type~~ of p.

$("p").parent(".class1") – It selects all the parent element that have a class of class1 of a set of elements matched by p

- Parents([selector]) – This method is used to gets the ancestors of each elements in a current set of matched elements, optionally filtered by selectors.

Syntax : $(selector).parents([selector])

Ex : $("p").parents() - It selects all the parents of each element in the current set of matched elements i.e. p till the root element.

$("p").parents("div.class1") - It selects all the ancestors matched by div that have class of class1. of each element matched by p.

*prev([selector]) - This method is used to gets the immediately preceding sibling of each elements in the set of matched element, optionally filtered by a selectors.

Syntax : $(selector).prev(selector)

Ex : $("p").prev → It selects the immediately preceding sibling of each element matched by p

<Form id

 Span1
 Para1
 Para2

 Para3
</form>

is selected.

$("p").prev("div") - It selects immediately preceding sibling matched by div of each element matched by p.

• prevAll([selector]) – This method is used to gets all preceding siblings of each element in the current set of matched elements, optionally filtered by a selector.

Ex. $("p").prevAll(); It selectes all previous sibling elements of each element matched by p.

```
<form id="form1" runat="server">
<div>
<span> span1 </span>
<p> para1 </p>
<p> para2 </p>
</div>
<p> para3 </p>
</form>
```

→ span1
→ para1
→ div

Ex. $("span").prevAll("div") – It selects all previous sibling elements matched by div of each element matched by span

```
<Form id="Form1" runat="server">
<div>
<span> span1 </span>
<p> Para1 </p>
<p Para2 </p>
</div>
<span> Span2 </span>
<p> Para3 </p>
</form>
```

→ div is selected.

.next(selector) - This method is used to gets the immediately following sibling of each element.s. If a selector is provided, it retrives the next sibling only if it matches that selectors.

Syntax: $(selector).next(selector) - It selects just immediately following sibling elements of each element matched by span.

```
<form id="form1" runat="server">
<div>
<span> span1 </span>
<p> para1 </p>
<p> para2 </p>
</div>
<span> span2 </span>
<p> para3 </p>
</form>
```

- Para1 & Para3

$("span").next(".class1") - It selects immediately following sibling that have a class of class1 of each element matched by span

```
<form id="form1" runat="server">
<div>
<span> span1 </span>
<p> para1 </p>
<p> para2 </p>
</div>

<span> span2 </span>
<p class="class1"> para3 </p>
</form>
```

→ Para3 is selected

- nextAll(selector) : ~~It selects all~~ following ~~elements~~ It selects all Following sibling of each elements in the current set of matched elements, optionally filtered by a selector.

siblings(selector) - This method is used to gets the sibling of each element in the set of matched element, optionally filtered by a selector [No matter privious, No matter Next]

Syntax: $(selector).sibling(selector)

$("p.class1").siblings() - It selects all the sibling of each element matched by p and that have a ~~class of class 1.~~

```
<form id = "Form1" runat = "server">
<div>
<span> span1 </span>
<p> para1 </p>
<p> para2 </p>
</div>
<span> span2 </span>
<p>
<p class = "class1"> para3 </p>
<p> para4 </p>
</form>
```

→ div, span2, para 4 is selected

$("p.class1").siblings("span") - It selects all the siblings matched by span of each element matched by p that have class of class1.

```
<form id="Form1" runat="server">
<div>
<span> span1 </span>
<p> para1 </p>
<p> para2 </p>
</div>
<span> span2 </span>
<p class="class1"> para3 </p>
<p> para4 </p>
</form>
```

→ span2 is selected

24|8|15

- First() - This method is used to gets the first matched element in the current set of matched element.

Syntax: $(select).first()

Ex: $("p").first().addClass("highlight") - It applies highlight class to a first element matched by p

- Last() - This method is used to gets the last matched element in the current set of matched element.

Syntax: $(selector).last()

Ex: $("p").last().addclass("highlight") - It applies highlight class to last element matched by p.

• has (selector) – This method is used to reduce the set of matched element to those that have a descendant that matched the selector or DOM element(s).

Syntax : $(selector).has(selector)

$(div1.has(class1).addClass("highlight")

```
<form id = "form1" runat = "server">
<div id = "div1">
<p> Para1 in Div1 </p>
<p> Para2 in Div2 </p>
<div id = "div2">
<span> span1 in Div2 </span>
<p> Para3 in Div2 </p>
</div>
</div>
<p class = "class1">para1 with class1 </p>
```

jquery

```
<script type = "text/javascript>
$(document).ready(function(){
$(div).has(".class1).addClass("highlight");
});</script>
```

→ It applies the highlight class to all the elements matched by div that have descendant element with a class of class1.

.each() - This method is used to interate over a jquery object, executing a function for each matched element.

Syntax: $(selector).each(function() {
});

[Note: It returns only single value or single text]

.map() - This method is used to pass ~~the~~ each element in the current matched set through a function, producing a new jquery object containing the return values

Syntax: $(selector).map(function() {
return statement;
});

```
<form id="form1" runat="server">
<b>Select Options:</b>
<br/>
<input type="checkbox" id="chk1" name="Options" value="Option1" />
<label for="chk1">Option1</label>
<input type="chkbox" id="chk2" name="Options" value="Option2" />
<label for="chk2">Option2</label>
<input type="chkbox" id="chk3" name="Options" value="Option3" />
<label for="chk3">Option3</label>
<br/>
<input type="button" id="btnSubmit" value="Submit" />
</form>
```

Output:

Select Options:

☐ Option1 ☐ Option2 ☐ Option3

Submit

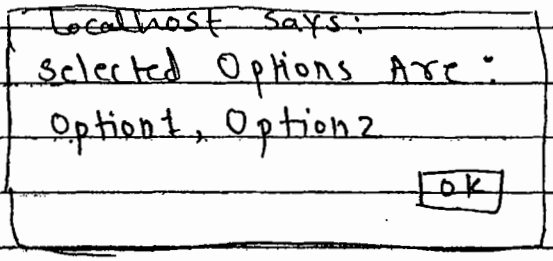
```
<script type="text/javascript">
$(document).ready(function(){
$("#btnSubmit").click(function(){
var options = $("input:checkbox[name=Options]").filter(":
checked").map(function(){
        return $(this).val();
    }).get().join();

if(options.length > 0)
alert("Selected Options Are:\n" + options);
else
alert("No Option Selected");
});
});
</script>
```

O/P

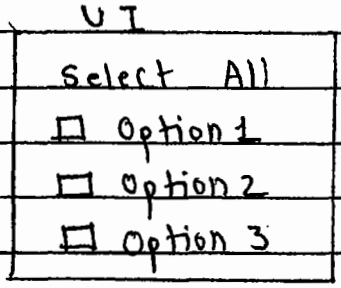
Select options:

☑ Option1 ☑ Option2 ☐ Option3

[Submit]



UI



- HTML Source

```
<input type="checkbox" id="chkselect" />
<label For="chkSelect"> Select All </label>
<br/>
<input type="checkbox" id="chk1" value="Opt1" name="Options" />
<label for="chk1"> Option 1 </label>
<br/>
<input type="Checkbox" id="chk2" value="Opt2" name="Options" />
<label For="chk2"> Option 2 </label>
<br/>
<input type="checkbox" id="chk3" value="Opt3" name="Options" />
<label For="chk3"> Option 3 </label>
<br/>
<input type="button" id="btnSubmit" value="submit" />
<br/>
<span id="spanResult </span>
```

```
<script type = "text/javascript>
$(document).ready(function() {
var chkItems = $("input:checkbox[name:Options]");
var lblChkSelect = $("label[for = chkSelect]");
$("#chkSelect").change(function(){
if($(this).is(":checked")){
chkItems.prop("checked",true);
lblChkselect.text("De-Select All");
}
else{
chkItems.prop("checked", false);
lblChkSelect.text("Select All");
}
});
chkItems.change(function(){
var chkCount = chkItems.length;
var chkCount = chkItems.filter(":checked").length;
if(chkCount == chkdCount){
$("#chkSelect").prop("checked", true);
lblChkSelect.text("De-Select All");
}
else{
$("#chkSelect").prop("checked", false);
lblChkselect.text("Select All");
}
});
$("#btnSubmit").click(function(){
if(chkItems.is(":checked")){
```

```
// Using map function
Var texts = chkItems.Filter(":checked").map(function(){
return $("label[for=" + $(this).prop("id") + "]").text();
}).get().join();
$("#spanResult").html("Selected Option Are:" + texts);
}
else {
$("#spanResult").html("<b style='color:red'>No option
Selected!</b>");
}
});
});
</script>
```

DOM MANIPULATION IN JQUERY

- jQuery provides methods to man DOM in efficient way, we do not need to write big code to modify the value of any elements attributes or to extract html content from a paragraph or div or any other elements of html document

*Content Manipulation :-

- jquery provides following methods to manipulate the contents of HTML DOM :-

1> val() - This method is used to gets the value of first matched element (input element)

Syntax : $(selector).val() - This will return value of first matched element

2> val(value) - This method is used to sets the specified value to a set of matched elements

Syntax : $(selector).val(value)

3> html() - This method is used to gets the html content of the first matched element (Non input element like <p>, , <spam> ... etc) <input element checkbox, texbox...>

Syntax : $(selector).html();

4> html(content) : This method is used to set the specified content to a set of matched elements

Syntax : $(selector).html(content)

Note : The content can be either html content or plain text content.

25/8/15

5> text() – This method is used to gets the combined text of a set of matched elements.

Syntax: $(Selector).text()

6> text(content) - This method is used to sets the specified content to a set of matched elements.

syntax: $(selector).text(content)

Note : Content should be plain text only.

∴ DOM ELEMENT ATTRIBUTE MANIPULATION ∴

1> attr("attr Name")
2> attr("attrName, value)
3> attr({"attrName:value, "attrName": value})
4. removeAttr("attrName")
OR
1- prop("propName")
2. prop("propName", value)
3. prop({"propName": value, "propName": value})
4. removeprop("propName")

DOM ELEMENT REPLACEMENT

- jQuery provides following 2 methods for DOM element replacement :

① .replaceWith(newContent) - This method is used to replace each element in the set of matched element with the provided new content and return the set of element that was removed.

Syntax : $(selector).replaceWith(newContent)

Ex. $("p").replaceWith("<h1>Heading</h1>");

[Note : The new content can be specified either as a plain text or html elem.]

② .replaceAll(target) - This method is used to replace each target element with the set of matched element.

syntax : $(selector).replaceAll(target)

Ex. $("div.class1").replaceAll("p") - It replace all target elements matched by "p" with a set of elements matched by "div" that uses a class of class1

REMOVING DOM ELEMENTS

- There may be a situation when you would like to remove one or more DOM elements from the document. jQuery provides following 2 methods to handle this situation.

1. empty() - This method is used to removes all the children's element including text of a set of matched element.

Syntax : $(selector).empty()

Ex. $("div.class1").empty() - It removes all the children elements of a set of element matched by div that uses a class of class1

2. remove(selector) - This method is used to remove the set of matched element from the DOM

NOTE : The specified selector expression in the method parameter filtres the set of matched element to be removed.

Syntax : $(selector).remove(selector)

Ex1. $("p").remove() - It removes set of element matched by P.

Ex2: $("p").remove(".class1") - It removes a set of elements matched (or $("P.class1").remove()) by P that are filtered with a class of class1

(both are not same.)

variable obj

Ex. var chk = $("input:checkbox[name=Options]");
chk.remove(":checked")

op → In the above example, a group of checkboxes stored in the jQuery variable object and then removing from that object those checkboxes which are checked.

Any html element insert to the runtime use after method.

DOM ELEMENT INSERT

- There may be a situation where we would like to insert new or one more DOM element in our existing document.

- jQuery provides various methods to insert element at various location.

1. after(content) - This method is used to inserts the specified content after each element in the set of matched element.

Syntax: $(selector).after(content)

[Note: The content can be either html content or plain text content.]

Ex. $("#p1").after("");
It inserts the specified content i.e. an image after the one element that has an id of p1.

2. before(content) - This method is used to inserts the specified content before each element in the set of matched element.

Syntax: $(selector).before(content)

Ex. $("#p1").before("");
It inserts the specified content i.e. an image before the one element that has an id of p1.

3. InsertAfter(target) - This method is used to insert every element in the set of matched element after the target.

Syntax: $(selector).insertAfter(target)

Ex. $("h1").insertAfter("#div1") - It inserts all the elements matched by h1 after an element that has an id of div.

4. insertBefore(target) - This method is used to insert every element in the set of matched element before the target.

Sytax: $(selector).insertBefore(target)

Ex. $("h1").insertBefore("div1") ("#div1") - It inserts all the elements matched by h1 before an element that has an id of div.

APPENDING DOM ELEMENT:

- There may be a situation occurs when we would like to Append one or more elements or any new content into the begining or at the end to a set of matched element.

- To do this jquery provides various methods which are following:

1. Append(content) - This method is used to insert content at the end of each element in the set of matched elements.

Syntax: $(selector).append(content)

Ex. $("div").append("<h1>Div Content</h1>"); - It inserts the specified content i.e. h1 to the end of each element matched by div.

2000994008 (NIT)

2. prepend(content) - This method is used to insert content at the begining of each element of in the set of matched element.

Syntax: $(selector).prepend(content)

Ex: $("div").prepend("<h1>Div content</h1>"); - This insert the specified content i.e. h1 to the begining of element matched by Div.

3. appendTo(target) : This method is used to insert the every element in the set of matched element to the end of target.

Syntax: $(selector).appendTo(target)

Ex: $("div").appendTo("#p1") - It inserts all the element matched by div to the end of an element that has an id of p1.

4. prependTo(target) - This method is used to inserts every element in the set of matched element to the begining of the target.

Syntax: $(selector).prependTo(target)

Ex: $("#img1").prependTo("div") - It inserts an element that has an id of img1 to the begining of each element matched by div[target]

26/8/15

CLONING DOM ELEMENTS

- jQuery provides the method [i.e clone()] used to make clone copy of any DOM elements.

Syntax : $(Selector).clone() - It makes a clone copy of the set of matched elements without their event handler.

$(Selector).clone(true) - It makes a clone copy of the set of matched element with their event handler.

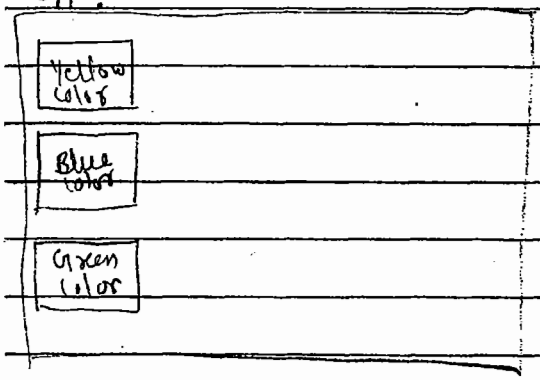
Example :

```
<body>
<div class="class1" style="background-color: Yellow"></div>
<div class="class1" style="background-color: Bule"></div>
<div class="class1" style="background-color: Green"></div>

<style type="text/css">
  .class1 {
    width: 100px;
    height: 100px;
    border: 5px double red;
    margin: 5px;
    cursor: pointer;
  }
</style>
</head>


<script type="text/javascript">
$(document).ready(function() {
  $("div.class1").click(function() {
    $(this).clone(true).insertAfter($(this));
```

of

```
$("div.class").click(function(){
    var obj1 = $(this).clone(true);
    obj1.insertAfter($(this));
});
</script>
```

O/p:



EVENT HANDLING IN jQuery

- We have the ability to create dynamic web pages by using events. Events are actions that can be detected by our web application.

- When events are triggered we can then use a custom function to do pretty much whatever we want with the event. These custom function called event handlers.

NOTE : All the events of javascripts are available in the jQuery as it is same but all the javascript events are defined in the form of event methods in jQuery as well as some additional event methods are defined to handle DOM elements events more easily.

A) Binding Event Handlers: We can established event handler on DOM elements with bind method as following:

Syntax: $(selector).bind(eventType, [event Data], handler)

Parameters:

1) eventType: A string containing a javascript event type, such as click, submit etc.

2) eventData: This is optional parameter is a map of data that will be passed to the event handler.

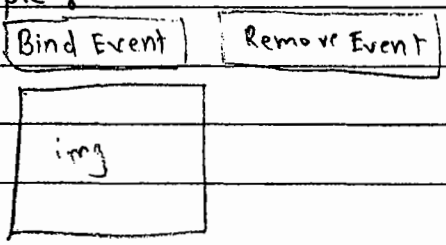
3) handler: It is nothing but a function to execute each time the event is triggered.

B) Removing Event Handlers: Typically, once an event handler is establised, it remains in effect for the rest of the life of the page but there may be a situation when we would like to remove event handler.

- jQuery provides unbind() command to remove an existing event handler at runtime.

Syntax: $(selector).unbind("eventType")

Example:



```
<input type="button" id="btn1" value="Bind Event">
 
<input type="button" id="btn2" value="Remove Event">
<br/><br/>
<img id="img1" src="Images/Ads1.jpg" width="350" height="350"/>

Jquery
<script type="text/javascript">
$(document).ready(function(){
$("btn1").click(function(){
$("#img1").bind("mouseover", function(){
$(this).prop("src","Images/Ads2.jpg");
}).bind("mouseout", function(){
    $(this).prop("src","Images/Ads1.jpg");
});

$(this).prop("disabled", true);
$("#btn2").prop("disabled", false);
});

$("#btn2").click(function(){
$("#img1").unbind("mouseover").unbind("mouseout");
    $(this).prop("disabled", true);
    $("#btn1").prop("disabled", false);
});
});
</script>
```

- jQuery provides Following additional event methods :-

1) hover (HandlerIn, HandlerOut) - This event method is used to bind two event handlers to the matched elements, to the executed when the mouse pointer enters and leaves the elements.

Syntax : $ (selector).hover(function() { F1
},
function() { F2
});

Two handler function in one single method.

Example:

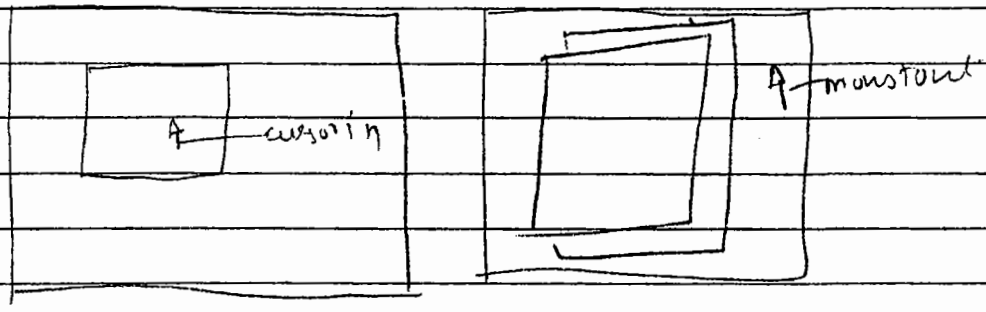
HTML

```
<div align="center">
<div id="div1" class="class1"></div>
</div>

<Style type="text/css">
.class 1 {
    width: 300px;
    Length: 300px;
    border: 5px solid red;
    background-color: Yellow;
}
.class 2 {
width: 500px;
height: 500px;
~~Length: 15px double blue;~~
border: ~~red~~ 15px double blue;
background-color: ~~red~~ red
```

box-shadow: 10px 10px black;
border-radius: 50px;
</style>

jquery
<script
$
$("#div1").hover(function(){
$(this).removeClass("class1").addClass("class2");
},
function(){
$(this).removeClass("class2").addClass("class1");
});
});
</script>

output



www.raybiztech.com
29 Aug 2015
10:00 to 2:pm
040-23114017/18

28/8/15

ready() - This event method is used to specify a function to executer when the DOM is Fully loaded.

Syntax: $(document).ready(function(){
});

.trigger() - This event method is used to execute all handlers and behaviours attached to the matched elements for the given event type.

Syntax: $(selector).trigger(eventType)

Example:

```
<b> Enter Text :</b> <input type="text" id="txt1"/>
 
<span id="span1"></span>
<br/><br/>
<input type="button" id="btnSelect" value="Select Text"/>

<jQuery

<script type="text/javascript">
$(document).ready(function(){
  $("#txt1").select(function(){
  $("#span1").html("<b>Selected Text:</b>" + $(this).val());
  });
$("#btnSelect").click(function(){
  $("#txt1").trigger("select");
});
});
</script>
```

Output:

Enter Text: [Welcome]

[Selecte. Text]

∴ Selected Text: Welcome

if you are entered the text like "Welcome" in textbox then after click on "select Text" button then it should select the Text "Welcome" and show the output.

Focusin() - This event method is used to bind an event handler to the Focusin event.

Syntax: $(Selector).Focusin(function(){

});

Focusout() - This event method is used to bind an event handler to the focusout javascript event.

Syntax: $(selector).Focusout(function(){

});

Example:

```
<b> Enter Name: </b> <input type="text" id="txt1" />

jquery <style type="text/css">
        .style1 {
<script     background-color: yellow;
            border: 2px solid red;
        }
    </style>
```

jquery

```
<script type ="text/javascript">
$(document).ready(function (){
$("#txt1").focusin(function(){
$(this).addClass("style1");
}).focusout(function(){
$(this).removeClass("style");
});
});
</script>
```

Output: Enter Name: [ ]

After comes into textbox (by tab key or cursor) it will become highlight like below:

Enter Name: [ ]

Example:

```
<Form
<b> First Name </b> <asp:TextBox ID ="txtFirstName" runat=
"server"/>
<br/> <br/>
<b> Last Name </b><asp:TextBox ID ="txtLastName" runat=
"server />
<br/> <br/>
<asp:Button ID=" btnSubmit" runat="server" Text =" submit"/>
```

```
<style type="text/css">
 .style1 {
  background-color: yellow;
  border: 2px solid red;
 }
</style>

jQuery
<script type="text/javascript">
$(document).ready(function(){
 var tFName = $("#txtFirstName");
 var tName = $("#txtLastName");
tFName.add(tLName).focusin(function(){
     $(this).addClass("style1");
    }).focusout(function(){
      if($(this).val()!="") {
    $(this).removeClass("style1");
      }
    });
 $("#btnSubmit").click(function(){
    if(tFName.val()=="")
    tFName.addClass("style1");
  if(tLName.val()=="")
    tLName.addClass("style1");

if(tFName.add(tLName).hasClass("style1"))
     return false}
 });
});
</script>
```

Output :

First Name : [ ]

Last Name : [ ]

Submit

→ if we click submit without enter the text then it will highlight.

→ if we click any of textbox without entering text then also empty textbox show highlight textbox.

- If you enter the text value in two textbox like ABC & PQR then after clicking submit it will not highlighted.

31/8/15

CSS HANDLING IN jQuery

- The jQuery library supports nearly all of the selectores included in css specifications 1 through 3, as outlined on the World Wide Web Consortium's site. (W3C)

- Using jquery library developers can inhance their web sites without worrying about browsers and their versions, as long as the browses have javascript enabled

- Most of the jQuery css methods do not modify the content of the jQuery object and they are use to apply css properties on DOM elements.

Applying CSS Properties

- These are the following methods supported by jQuery library to apply css properties as a single property with their value or multiple properties with their value on the selected DOM element(s).

1) .css("propName", "Value") - This method is used to applying a single property with their value to a set of matched element(s).

Syntax : $(selector).css("propertyName","Value")

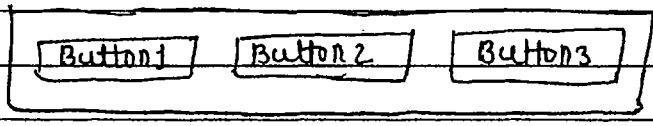
2) .css{"propName":"Value","propName":"Value") - This method is visible used to apply multiple properties with their values in the form of key value pair to a set of matched element(s).

Syntax : $(selector).css({"propName":"Value","propName":"Value"})
[we can write N no of propName & Value]

3) ~~.css~~ * RETRIVING CSS PROPERTY VALUES *

3) css.("propName") - This method is used gets the value of specified css property of First matched element.

Syntax : $(selector).css("propName")

Example. 

} button style is change in runtime

HTML

```
<div align = "center">
<input type = "button" id = "btn1" value = "Button1" />
 
<input type = "button" id = "btn2" value = "Button2" />
 
<input type = "button" id = "btn3" value = "Button3" />
</div>
```

jQuery

```
<script type="text/javascript>
$(document).ready(function(){
  var btns = $("input:button");
  btns.css({"background-color":"red","color":"white",
  "font-size":"15px","font-name":"Tahoma","border":"5px
  solid green",:"border-radius":"5px","box-shadow":
  "5px 7px gray","cursor":"pointer"}).mouseover(function(){
  $(this).css({"background-color":"blue","color":"Yellow",
  "box-shadow":"7px 5px lime"});
  }).mouseout(function(){
$(this).$(this).css({"background-color":"red","color":
  "white","box-shadow":5px 7px gray"});
  }).click(function(){
  btns.css({"background-color":"red","color":"white",
  "box-shadow":"5px 7px gray"});

  btns.bind("mouseout",function(){
  $(this).css({"background-color":"red","color":"white",
  "box-shadow":"5px 7px gray"});
  });

$(this).trigger("mouseover");
$(this).unbind("mouseout");
alert($(this).val()+"Is clicked");
});
});
</script>
```

UI:

Welcome to jQuery Css Handling

```
<div id="divTitle" style="margin-right:10px; padding:
15px; border-radius: 10px; text-align: center">
    Welcome to jQuery Css Handing
</div>

jQuery

<script type="text/javascript>
$
Function style1(){
$("divTitle").css({"border":"10px Solid red","box-shadow":
"10px 10px green","background-color":"blue","color":"yellow",
"font-size":"40px"});
}

Function style2(){
$("#divTitle").css({"border":"10px solid green","box-shadow":
"10px 20px red","background-color":"black";"color":"white",
"font-size":"60px"});
}
style1();
Var flag = true;
Function blink(){
  if (flag){
    style2();
    flag = false;
  }
```

```
else {
style1();
flag = true;
}
}
window.setInterval(blink, 200);
});
</script>
```

Tuesday-
01.09.15

WORKING WITH CSS WIDTH AND HEIGHT METHODS

1) .width() - This method is used is gets the current computed pixel, width of the first matched element.

Syntax: $(selector).width()

2) .width(val) - This method is used to sets the css width of every matched element.

Syntax: $(selector).width(val)

3) .height() - This method is used to gets the current computed pixel, height of the first matched element.

Syntax: $(selector).height()

4) .height(val) - This method is used to sets the css height of every matched element.

Syntax: $(selector).height(val)

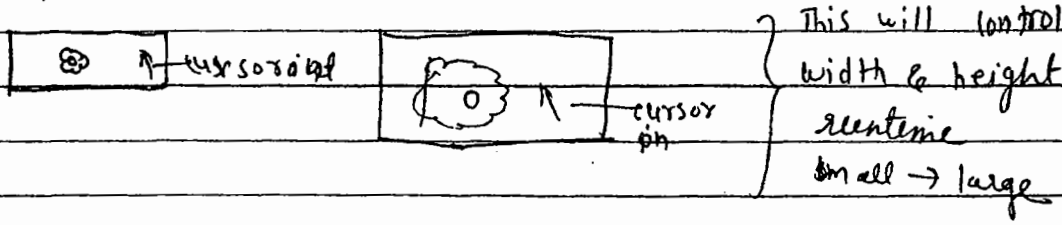
Ex: Controlling the img of runtime.

```
<div align = "center">
<img id = "img1" src = "Images/ads2.jpg" />
</div>
```

jquery

```
<script type = "text/javascript">
$(document).ready(function(){
  var w,
      h;
$("#img1").width(300).height(300).hover(function(){
w = $(this).width();
h = $(this).height();
$(this).width(600).height(500);
  },
function(){
$(this).width(w).height(h);
  });
});
</script>
```

Output:



5> · innerWidth() - This method is used to gets the inner width (excludes the border and includes the padding) for the first matched element.

Syntax : $(selector).innerWidth()

6> · innerHeight() - This method is used to gets the inner height (excludes the border and includes the padding) for the first matched element

Syntax : $(selector).innerHeight()

7> · outerWidth([margin]) - This method is used to gets the outer width (includes the border and padding by default) for the first matched element. It can be also added margin by specifying true in the parameter for margin [It is optional]

Syntax : $(selector).outerWidth(); OR
$(selector).outerWidth(true);

8> · outerHeight([margin]) - This method is used to gets the outer height (includes border & padding by default) for the first matched element. It can be also added margin by specifying true in the parameter for margin[It is optional]

Syntax : $(selector).outerHeight(); OR
$(selector).outerHeight(true);

9) .scrollLeft(val) - When a value is passed in, the scroll left offset is set to that value on all matched elements.

$(selector).scrollLeft(val)

10) .scrollLeft - It is used to gets the scroll left offset of the first matched element.

Syntax - $(selector).scrollLeft()

11) .scrollTop(val) - When a value is passed in, the scroll top offset is set to that value on all matched elements.

Syntax : $(selector).scrollTop(val)

12) .scrollTop() - It is used to get the scroll top offset of the first matched elements.

Syntax : $(selector).scrollTop()

Example:

```
<div id="div1" style="width:500px; height:502px; overFlow:Scroll">
<img id="img1" src="Images/Adz.jpg" />
</div>
<span id="span1"></span>

jquery

<script type="text/javascript">
$(document).ready(function() {
```

```
$("#div1").scrollLeft(200).scrollTop(250);
$("#div1").scroll(function(){
$("#span1").html("left offset Value:"+$(this).scrollLeft()
+"<br/>"+"Top offset Value:"+$(this).scrollTop());
});
});
</script>
```

02/09/15

4.9.15 Welcome to jQuery Animation

```
<style type = text/css">
# divMain {
 text-align: center;
 margin: 0px;
 padding: 10px;
 border: 10 px solid red;
 background-color: black;
 Color: White;
 font-size: 30px;
 height: 70px;
 line-height: 70px;
 }
</style>
<script src = "Scripts/jquery-2.1.4.js" type = "text/javascript">
</script>
<script src="Scripts/jquery-ui-1.11.4.js" type="text/javascript >
</script>
<script type = "text/javascript>
 $(document).ready(function () {
 $("#divMain").animate({
 "backgroundColor": "blue",
 "Color": "Yellow" -
 "font-size": "40px" },
 500, "easeInOutBounce").animate({
 "backgroundColor": "Yellow",
 "color": "Green",
 "Font-size": "50px" },
 500, "easeInOutBounce).animate({
 "background(olor": "aqua",
```

```
"Color": "black",
"fontSize": "70px" },
500,"easeInOutBounce).animate({
"backgroundColor":"Red",
"Color":"White",
"fontSize":"50px"},
500,"easeInOutBounce")
}
windows.setInterval(animate,500);
});
</script>
```

jQuery UI Library Based Effects

- It is a rich effect API and ready to use effects.
- The jQuery UI Library based effects can be used, using (.effect()) method to an element as following:

General Syntax: $(selector).effect((effectName),[options],[duration], [callback])

THE effect() METHOD PARAMETER:

① effectName: String corerosponding to the effectName to use such as ("blind","bounce", etc)

② options: It is an optional object to specify the behaviour of the effect [e.g. "hide" or "show" in options mode]

③ Duration : Duration of the effect in milisecond. It can be also specifies with the fixed value such as "slow" & "fast". [The default duration is 400 miliseconds]

④ Callback : Callback function called for each time in the last item (items corresponding to the selector) when the effect is complete. This is an optional parameter.

THE BLIND EFFECT : The blind effect can hides or display an item making it disappear or appear in the indicated direction

Syntax : $(selector).effect("blind", {Options}, Duration, callback);

→ The options parameters are :

① Mode : It can be specified either "show", "Hide" or "Toggle". Default value is Hide.

② Direction : It can be set to either horizontal or vertical, the default value is vertical.

Ex : HTML

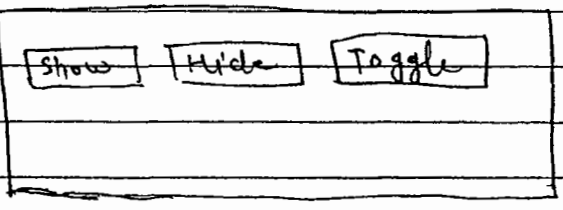
```
<div align="center">
<input type="button" id="btn1" value="show" />
 
<input type="button" id="btn2" value="Hide" />
 
<input type="button" id="btn3" value="Toggle" />
<br /><br />
<img id="img1" src="Images/Ads2.jpg" />
</div>
```

```
<style type = "text/css">
img1 {
width: 400px;
height: 400px;
padding: 20px;
border: 10px solid red;
background-color: yellow;
}
</style>
```

jQuery

```
<script type = "text/javascript">
$(document).ready(function() {
$("#btn1").click(function(){
$("#img1").effect("blind", {mode: "show", direction: "horizontal"},
3000);
});
$("#btn2").click(function() {
$("#img1").effect("blind",{mode:"Hide", direction:"Horizontal"}, 3000);
});
$("#btn3").click(function() {
$("#img1").effect("blind", {mode: "Toggle", direction: "vertical"}, 3000);
});
});
</script>
```

O/P:



* THE BOUNCE EFFECT

- The bounce effect makes the element appear to disappear with bounce of element vertically or Horizontally 'N' times.

Syntax : $(selector).effect("bounce",{options},Duration,Callback);

- The options parameters are :

① Mode : It can be set to either show, hide or Toggle.

② Distance : It defined the distance value in the form of pixels to bounce it.

③ Direction : It can be set to either up, down, left or right.

④ Times : Defined the no. of times value to bounce it.

Ex: All code is same as above only "btn3" is change.

i.e. $("#img1").effect("bounce",

{

mode : "toggle",

direction : "left",

distance : 250,

times : 2}, 3000);

5.9.15 Effects

① The Clip Effect: This effect show or hide element by scrolling horizotally or vertically.

Syntax: $(selector).effect("clip",{options},Duration,callback);

Options parameters are:

① Mode: It can be set to either show, hide or toggle.

② Direction: It can set to either horizontal or vertical.

e.g. $("#img1").effect("clip",
{ mode:"toggle",
Duration:"vertical"}, 3000);

② The Drop effect: This effect is used to show or hide the item by dragging and lowering is opacity.

Syntax: $(selector).effect("drop",{options},Duration,callback);

The option parameters are:

① Mode: It can be set to either show, hide or toggle.

② Duration: It can be set to either up, down, left or right.

③ Distance: Define the distance values from a pixel for drapping it.

OR

$("#btn5").click(function(){
#("#img1").effect("drop",{mode:"toggle",direction:"left",
direction:350},3000);

③ The Explode Effect: This effect is used to next element appear or disappear by spliting it into multiple pices

Syntax: $(selector).effect("explode", {options}, duration, callback);

The options parameters are:

① Mode: It can be set to either show, Hide or Toggle.

② Pices: Define the no. of pices to split.

eg: $("#img1").effect("explode",
{ mode: "toggle",
pices: 16}, 3000);

④ The Fold Effect: This effect is shows & Hide and element of Folding it like a pi of paper.

Syntax: $(selector).effect("fold", {options}, Duration, Callback);

Mode: It can be set to either show, hide or toggle.

horizFirst: It can be set to either true or false, If it is set to true it is start folding first horizontaly then vertically, and if is set to false it does vice versa

$("#img1").effect("Fold",
{
mode: "toggle",
horizFirst: true
}, 3000);

⑤ The Highlight Effect: This effect is show & Hide an element by animating its background color

[means highligh a background with a defined color]

$(selector).effect("highlight", {options}, Duration, Callback)

Mode: It can be set to either show or Hide or toggle.

Color: Defined the any color to highlight to animation.

Example:

HTML

```
<div id="divMain">
jQuery UI Library Based Effects
</div>

<style type="text/css">
#divMain{
text-align: center;
border: 2cpx solid red;
background-color: black;
color: White;
font-size: 60px;
font-weight: bold;
}
</style>

<script type="text/
$(document).ready(function(){
function highlight(){
$("#divMain").effect("highlight",{
mode:"toggle",
color="blue"
3, 250, Function(){
$(this).effect("highlight",{mode:"toggle", color:"green"},300);
```

```
});
}
windows.setInterval(highlight, 500);
});
</script>
```

⑥ The Puff Effect : This effect can be used to show & Hide the element by enlarging or shrinking it and changing its opacity.

Syntax: $(selector).effect("puff", {options}, Duration, callback).

The options parameters are:

① Mode : which can be set to show, hide or toggle option.

② Percent : Defined the percentage value to enlarge or shrink it [default is 150].

```
$("#img1").effect("puff",
    {
        mode: "toggle",
        percent: 250
    }, 3000);
```

⑦ The Pulsate Effect : This effect can be used to show or Hide an element by pulsing it In or Out the opacity multiple times.

Syntax : $(selector).effect("pulsate", {options}, Duration, callback);

Options parameters:

① Mode : can be set to either show and Hide or Toggle.

② Times : Define the no. of times to pulsate it

Example :

```
<div id="divMain">
jQuery UI Library Based Effects
</div>

<style type="text/css">
#divMain {
left-align : center;
border-color : black;
color : white;
font-size : 60px;
Font-weight : bold;
}
</style>

<script type="text/javascript">
$(document).ready(function() {
function Flash() {
$("#divMain").effect("pulsate", {
mode : "toggle",
times : 15.
```

```
}, 500)
}
window.setInterval(Flash, 1000);
});
</script>
```

⑧ The Scale Effect : This effect is used to show or Hide an element by a percentage factor.

Syntax: $(selector).effect("scale", {options}, Duration, Callback)

The option parameters are:

① Mode : It can be set to either show hide or toggle option.

② Direction : Indicates the direction of resizing either horizontal, vertical or both.

③ Percent : Defined the percentage value to enlarge or shrink.

④ Fade : Defined the boolean value true or false indicating whether fade effect apply or not by changing its opacity

⑤ Scale : Defines which areas of the element will resize it can be set to either box, content, or both.

```
$("#btn3").click(function(){
  $("#img1").effect("scale",
  {
    mode:"toggle",
    percent:250,
    fade:true,
    direction:"both",
    scale:"both".
  3, 3000);
```

⑨ The Shake Effect: This effect is used to makes the element to show or Hide by shaking multiple times in the indicated direction.

Syntax: $(selector).effect("shake",{options}, Duration, (allback);

The options parameters:

① Mode: It can be set to show, Hide or Toggle.

② Direction: It can be set to either up, down, lef or Right.

③ Times: Define the no. of times to shake it.

④ Distance: Defines the distance value to be shaked it

```
$("#btn3").click(" function() {
$("#img1").effect("shake",
  {
    mode:"toggle",
    direction:"down",
    times: 3,
    distance: 250
  }, 3000);
```

⑩ The Fade Effect : This effect is used to show or Hide element by fading it.

Syntax : $(selector).effect("Fade", {options}, Duration, Callback);

The options parameters are :

① Mode : It can ~~say~~ set either show or Hide or Toggle.

eg.

```
$("#img1").effect("Fade",
  {
    mode:"toggle"
  }, 3000);
```

⑪ The Size Effect : This effect can be used to shows or Hides an element by resizing an element to specify Height and ~~weight~~ width.

Syntax : $(selector).effect("size", {options}, Duration, callback);

The options parameters are :

8.9.15

① Mode : It can be set to either show or Hide or toggle.

② to : #used to sets height and width to resize to. This is an object in the form of { height: ...., width: ....}

③ Scale : It is used to defined which areas of the element will be resize. It can be set to either box, content or both.

e.g. $("#img1").effect("size",
{ mode : "toggle",
to : { width : 700, height : 700}
scale : "both"
}, 3000);

The Slide Effect : This effect is used to shows or Hides the item by sliding it across the screen.

~~Syt~~ Syntax : $(selector).effect("slide", {options}, Duration, callback);

The option parameters are :

① Mode : It can be set to either show, Hide or toggle.

② Direction : Can be set to up, down, left or right.

③ Distance : Defined the distance value in the form of pixel covered by the element for slide it [sliding it]

e.g. $("#img1").effect("slide",
{
mode:"toggle",
direction:"up",
distance:450
}, 3000);

jQuery UI WIDGETS

- The jQuery UI widgets is a full featured UI controls and each has a range of options and it is fully themeable.

① jQuery UI Datepicker : The jQuer UI provides a function i.e. datepicker(), used to select a date from a pop-up or inline calendar.

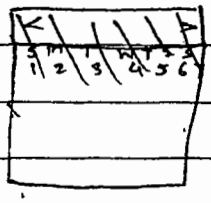
Syntax: $(selector).datepicker()

Ex: [For this add a Folder named as jQueryThemes]

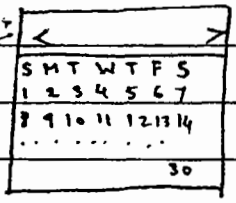
```
<form id="form1" runat="server">
<b>Enter Date:</b>
<asp:TextBox ID="txtDate" runat="server"/>
</form>

<script src="Scripts/jquery-2.1.4.js" type="text/javascript"/>
</script>
<script src="Scripts/jquery-ui-1.11.4.js" type="text/
javascript"></script>
```

```
<script src="Scripts/jquery-2.1.4.js" type="text/
javascript"></script>
<script src="Scripts/jquery-ui-1.11.4.js" type="text/
javascript"></script>

<link href="jQueryThemes/le-frog/jquery-ui.css" rel="stylesheet"/>
<link href="jQueryThemes/le-frog/theme.css" rel="stylesheet" />

<script type="text/javascript">
$(document).ready(function(){
  $("#txtDate").datepicker();
});
</script>
```

Output :



Enter Date:



*jquery UI theme from Microsoft CDN :

```
<link href="http://ajax.aspnetcdn.com/ajax/jquery.ui/
1.11.4/themes/excite-bike/jquery-ui.css" rel="stylesheet"/>
```

Note : You can download it once after that you can use it as many time as you want. [For disconnected environment also it is worked]

-The jQuery UI datepicker supports following options:

(i) showOn:

(A) $("#txtDate").datepicker({
showOn: "button",
buttonText: "Open"
});

O/P: Enter Date: [ ] [Open]

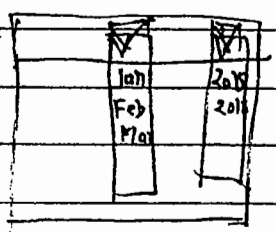
→ After click on "Open" button the calendar will be open.

(B) $("#txtDate").datepicker({
showOn: "both", // button, focus, both
buttonImage: "Images/calendar.gif",
buttonImageOnly: true
});

o/p:
Enter Date: [ ] → button Image

→ Calendar will open in three ways (i) After click on button image, (ii) After click on textbox or (iii) both.

(C) $("#txtDate").datepicker({
changeMonth: true,
changeYear: true
});

o/p: Enter Date: 

→ This will change the month and year.

09.09.15

② yearRangeType : It represent the value in the form of string
Default value is : "c-10:c+10". It can be set either relative to todays year ("-nn:+nn"), relative to the currently selected year ("c-nn:c+nn"), absolute ("nnnn:nnnn"), or combinations of these formats ("nnnn:-nn")

NOTE : This option only affects whats appears in the dropdown, to restrict which dates may be selected use the minDate and maxDate options.

e.g. : $("#txtDate").datepicker({
changeMonth : true,
changeYear : true,
// yearRange : "c-100:c+100"
// yearRange : "-100:+100"
// yearRange : "-100:+0"
// yearRange : "2010:2020"
// yearRange : "-20:2015"
yearRange : "2015:+20"
});

minDate AND maxDate Options : These 2 options can be used to specifies the Date Range. By default for both value is set to "Null", means there is no minimum and no maximum. It can be set with a different types of value such as Date, All Number, All String.

Ⓐ Date : A date object containing the minimum or maximum date.

(B) Number : A number of days from today.
Ex: 2 represents two days from today and -1 represents one day before today.

(C) String : A string in the format defined by the dateformat option. or a relative date. It can be also set value in the string representation such as "+10d" [next 10 days from today) "+1m" (next 1 month from today) "-1w" (previous 1 week from today) "+5y" (next 5 years from today)

- Even though it can be used combination of these all such as : "+1m +7d" represents 1 month and 7 days from today

e.g. $("#txtDate").datepicker({
minDate: new Date("01-01-2015"),
maxDate: new Date()
});

Output: It will show date from 01-01-2015 upto 9.09.15 [9.9.15 is today's date]

(D) $("#txtDate").datepicker({ // Numeric example
minDate: -0,
maxDate: +60
});

Output : It will allow to select date upto next 60 days.

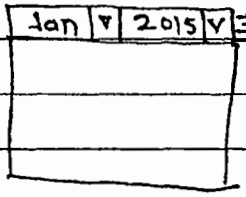
© $("#txtDate").datepicker({ // Date, String & No.
minDate: "-1m",
maxDate: "+1m +1w +10d"
});

output: It will show the date from 1 month 1 week and upto 10 days.

* Create an example of jQuery UI datepicker to be used as Date of Birth.

```
<form id="form1" runat="server">
<b> Enter Date of Birth </b>
<asp:TextBox ID="txtDate" runat="server"/>
</form>
jQuery
$("#txtDate").datepicker({
showOn: "button",
buttonImage: "Images/calendar.gif";
buttonImageOnly: true,
changeMonth: true,
changeYear: true,
yearRange: "-100:+0",
maxDate: 0
});
```

O/P: Enter Date of Birth : Jan 2015



☆ Create an example of jQuery UI datepicker to use as start date as well as end date.

Note: This kind of requirement can be use in any event management website where we need to add events with start event and end event. as well as we can also use in ticket booking website for Dateofjourney & ReturnDateofjourney.

```
<form id="form1" runat="server">
<b>Journey Date:</b>
<asp:TextBox ID="txtJourneyDate" runat="server"/>
   
<b>Return Date:</b>
<asp:TextBox ID="txtReturnDate" runat="server"/>
</form>

jQuery
$(document).ready(function(){
$("#txtJourneyDate, #txtReturnDate").datepicker({
  showOn:"button",
  buttonImage:"Images/calendar.gif",
  buttonImageOnly: true,
  minDate: -0,
  maxDate: +60,
  numberOfMonths:2,
 onClose: function(selectedDate){
 if ($(this).prop("id")=="txtJourneyDate"){
$("#txtReturnDate").datepicker("option","minDate",selectedDate).
 }
```

```
else if ($(this).prop("id") == "txtReturnDate") {
$("#txtJourneyDate").datepicker("option", maxDate", selectedDate);
    }
  }
});
});
```

output :

Journey Date: < 2015 >

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
...						30

Return Date: < 2015 >

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
...						31

10/9/15
THURSDAY

jQuery UI Datepicker With Date Format

- It can defined the date format in the form of string representation to display dates.

- The default value is "mm/dd/yy".

* Initialize the datepicke with the dateFormat option specified :

```
$(selector).datepicker({
  dateFormat: "yy-mm-dd"});
```

* Get or set the dateFormat option, after initialization :

```
// getter
var dateFormat = $(selector).datepicker("option","dateFormat");
```

// setter
$(selector).datepicker("option", "dateFormat", "yy-mm-dd");

These are the following dateFormat options can be used:
- Short - d M, y
- Medium - d MM, y
- Full - DD, d MM, yy

~~Eg.~~ jQuery UI datepicker With Animation
- This can be done by using an option i.e. showAnim

- Default value is set to "show".

- The name of animation use to show and hide the datepicker. set to an empty string to disable animation.

* Initialize the Datepicker with the showAnim Option specified:
$(selector).datepicker({
showAnim: "fold"});

Get or Set the showAnim option, after initialization:
// getter
var showAnim = $(selector).datepicker("option", "showAnim");

// setter
$(selector).datepicker("option", "fold");

[Note: While working with this option we may also set duration option for controlling the speed of animation, it can be set as a string representation such as "slow", "Normal" "fast" or any numeric values that represents a time miliseconds.]

eg. <form id="form1" runat="server">
 Enter the Date :
<asp:TextBox ID="txtDate" runat="server" />
</form>

```
$("#txtDate").datepicker({
duration : 3000,
showAnim : "explode"
});
```

* jQuery UI Datepicker With Other Months and Showing Week Numbers In the Open Calendar :

- This can be done using following 2 options →

① showOtherMonths : true or False

② showWeek : true or False

```
$('#txtDate').datepicker({
showOtherMonths : true,
selectOtherMonths : true, // it will allow to select any date
showWeek : true                of any month
});
```

* jQuery UI Datepicker With Multiple Months Calendar :

- This can be done using Following options

① ~~noOfMonth~~ numberOfMonths : By default it is set to 1 but it can be set to any no. between 1 to 12.

```
$("#txtDate").datepicker({                // 1(Jan) to 12(Dec)
numberOfMonths : 3,
showCurrentAtPos : 1
});
```

② showCurrentAtPos : It can be used to set the position to display the current month, default value is zero (0).

* jQuery UI datepicker also supports Following events :-

① beforeShow & onClose :- This event uses a callback function to be fired when the datepicker is closed whether date is selected or not.

② onSelect : This event also uses the callback function and execute when the datepicker is select.

③ OnChangMonthYear : This event is also uses the callback function and executes when the datepicker moves to a new month and year.

Note : These all above 3 events by default set to null.

* jQuery UI datepicker also supports Following keyboard options to intract with :

1. PAGE UP : Move to the previous month.

2. PAGE DOWN : Move to the next month

3. CTRL + PAGE UP : Move to the previous year

4. CTRL + PAGEDOWN : Mov to next year.

5. CTRL + HOME : Move to the current month.

6. CTRL + LEFT : Move to the previous day.
7. CTRL + RIGHT : Move to the next day.
8. CTRL + UP : Move to the previous week.
9. CTRL + DOWN : Move to the next week.

10. ENTER : Select the focused date.

11. CTRL + END : Close the datepicker and erase the date.

12. ESCAPE : close the datepicker without selection

jQuery UI Accordion

- jQuery UI provides a function i.e. Accordion(), used to provides a set of user interface to be appear in the form of expandable and collapsible.

Syntax: $(selector).accordion();

- The jQuery UI accordion also supports following options
  - active
  - animate
  - Collapsible
  - disabled
  - event
  - header
  - heightStyle
  - icons

- It also supports following events callback function:

① Activate

② beforeActivate

③ create

```
<div id="divMain">
  <h1> Personal Information </h1>
   <div>
     <p> Personal Information Content </p>
  </div>
  <h1> Address Information </h1>
     <div>
       <p> Address Information Content </p>
     </div>
  <h1> Contact Information </h1>
       <div>
         <p> Address Information Content </p>
         <img src="images/Ads1.jpg" />
       </div>
  </div>


$(document).ready(function(){
$("#divMain").accordion({
  heightStyle: "content",  // auto, content
  event: "click",          // mouseover, click
  collapsible: true,       // true/false
  active: 2,               // Index Number
  animate: "easeInOutBounce",  // Any Easing Option
  icons: {"header": "ui-icon-plus", "headerSelected": "ui-icons-minus"}
```

3);

3);

</script>

output

Personal Information

Address Information

Contact Information

11.09.15 → Working with jQuery UI Accordian to displays departmentwise employee data dynamically from a data source in the form of expandable & collapsible [for this we need to create Dept table & Emp table in db]

HTML

```
<div id = "divMain">
<asp:Repeater ID = "Repeater1" runat = "server" OnItemDataBound =
"Repeater1_ItemDataBound">
<ItemTemplate>
<h1> Dept Name : <%# Eval("DeptName") %> <asp:Label ID =
"lblDeptId" runat = "server" Text = '<%# Eval("DeptID") %>'
style = "display:none" /> </h1>
<div>
<asp:GridView ID = "GridView1" runat = "server" EmployeeDataText =
"No Employee Data Available!!!" width = "100%"></asp:GridView>
</div>
</ItemTemplate>
</asp:Repeater>
</div>
```

jQuery :

```
<script type="text/javascript">
$(document).ready(function(){
$("#divMain").accordion({
  heightStyle:"content",
  collapsible: true
  });
});
</script>
```

Server Side Code

```
using System.Data;
                .SqlClient;

public partial class Default57 : System.Web.UI.Page
 {
  SqlConnection cn = null;
  SqlDataAdapter da = null;
  DataSet ds = null;
  String strSqlCommand = string.Empty;

  protected void Page_Load(object sender, EventArgs e)
  {
    cn = new SqlConnection("DataSource=(local); Database=AJAXDB;
    User ID=Sa; Password = "123");

    if(!Page.IsPostBack)
    {
      BindDeptData();
  }
}
```

```
void BindDeptData()
{
strSqlCommand = "Select * from Dept";
da = new SqlDataAdapter(strSqlCommand, cn);
ds = new DataSet();
da.Fill(ds, "Dept");
Repeater1.DataSource = ds.Tables["Dept"];
Repeater1.DataBind();
}

protected void Repeater1_ItemDataBound(object sender, RepeaterItemEventArgs e)
{
if (e.Item.ItemType == ListItemType.Item || e.Item.ItemType
    == ListItemType.AlternatingItem)
  {
    Label lblDeptId = (Label) e.Item.FindControl("lblDeptId");
    if (lblDeptId != null)
    {
      strSqlCommand = "select e.EmpId, e.EmpName, e.EmpJob,
e.EmpSalary, d.DeptName from Emp e, e.DeptId=" +
    lblDeptId.Text;

da = new SqlDataAdapter(strSqlCommand, cn);
ds = new DataSet();
da.Fill(ds, "Emp");

GridView GridView1 = (GridView) e.Item.FindControl("GridView1");
```

```
if (GridView1 != null)
{
    GridView1.DataSource = ds.Tables["Emp"];
    GridView1.DataBind();
}
}
}
}
}
```

o/p:

Dept Name : IT
Dept Name : HR
Dept Name : Sales
Dept Name : Marketing

EmpId EmpName EmpJob EmpSalary DeptName.

jQuery UI tabs:

- jQuery UI provides a method i.e. Tabs, used to displays a set of user interface in the form of tab page

Syntax : $(selector).Tabs();

- jquer UI tabs also supports following options that can be used in a tabs function. and they are similar to jQuery UI Accordian:-

1. Active
2. Collapsible
3. Disabled
4. Event
5. HeightStyle
6. Hide
7. Show

① active - index number
② Collapsible - true or False
③ disable - true or False
④ heightstyle - Auto or Content
⑤ Hide - specifies the any effect
⑥ Show - specifies the any effect.
⑦ Event - Click or Mouseover

- show and hide option are used to apply animation when the tab page content shows or hides.

- These 2 options can be set either in the form of boolean or Number or string or object.

① boolean: true or false [FadeIn / FadeOut]

② number: specifies the duration in the form of milliseconds for controlling the speed of Animations.

③ string: specifies the name of effect such as slideup, slidedown, fold etc.

④ object: It can be used to set any effect as well as duration in the form of an object

- jQuery UI tabs also supports following events callback functions:

① activate [Imp]
② beforeActivate
③ beforeLoad
④ create
⑤ load

e.g:

 Personal Information

 Address Information

 Contact Information

 Personal Information Content

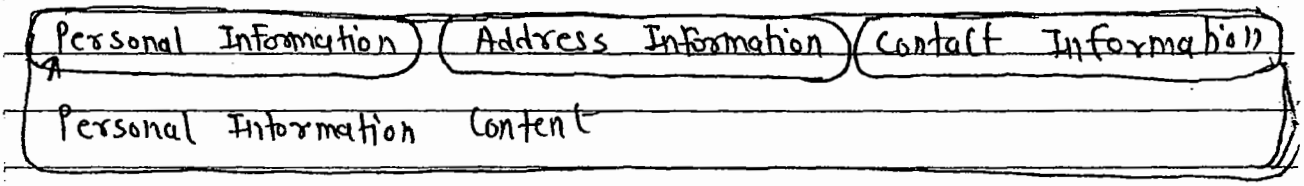
 Address Information Content

 Contact Information Content

jQuery Code

```
<script type = "text/javascript">
$(document).ready(function() {
$("#divMain").tabs({
  heightStyle : "content",
  collapsible : true
  });
});
</script>
```

Output



12.9.15 ✱ Working With jQuery UI tabs to displays departmentwise Employee data dynamically from a DataSource

HTML :

```
<div id = "divMain">
<ul>
<asp:Repeater ID = "Repeater1" runat = "server" OnItemDataBound = "Repeater1_ItemDataBound">
<ItemTemplate>
<li>
<asp:HyperLink ID = "link1" runat = "server" NevigateUrl = '<%# "#" + Eval("DeptId") %>' Text = '<%# Eval("DeptName") %>' />
```

```
<asp:Label ID="lblDeptId" runat="server" Text='<%#
Eval("DeptID") %>' style="display:none" />
</li>
</ItemTemplate>
</asp:Repeater>
</ul>
<asp:PlaceHolder ID="PlaceHolder1" runat="server">
</asp:PlaceHolder>
</div>
```

Server side Code:-

```
using System.Data;
        .Data.SqlClient;

public partial class Default5q : System.Web.UI.Page
{
  SqlConnection cn = null;
  SqlDataAdapter da = null;
  Dataset ds = null;
  String strSqlCommand = string.Empty;

  protected void Page_Load(object sender, EventArgs e)
  {
     cn = new SqlConnection("Data Source=(local);
Database=AJAXDB; User Id=sa; password=123");
     if (!Page.IsPostBack)
     {
        BindDeptData();
     }
  }
}
```

```
ABCDEFGBindDeptData()
IJKLMNOP
QRSTUVWSqlCommand = "Select * From Dept";
    da = new SqlDataAdapter(strSqlCommand, cn);
    ds = new DataSet();
    da.Fill(ds, "Dept");
    Repeater1.DataSource = ds.Tables["Dept"];
    Repeater1.DataBind();
}

protected void Repeater1_ItemDataBound(object sender,
                                       RepeaterItemEventArgs e)
{
    if (e.Item.ItemType == ListItemType.Item || e.Item.
      ItemType == ListItemType.AlternatingItem)
    {
      Label lblDeptId = (Label) e.Item.FindControl
      ("lblDeptId");

      iF(lblDeptId != null)
      {
strSqlCommand = "Select e.EmpId, e.EmpName, e.Emp
  Job, e.Empsalary, d.DeptName From Emp e, Dept d
  where e.DeptId = d.DeptId and e.DeptId =" +
  lblDeptId.Text;

da = new SqlDataAdapter(strSqlCommand, cn);
ds = new DataSet();
da.Fill(ds, "Emp");
```

```
GridView GridView1 = new GridView();
GridView1.EmptyDataText = "No Employee Data Available";
GridView1.width = Unit.Percentage(100);
GridView1.DataSource = ds.Tables["Emp"];
GridView1.DataBind();

Panel panel1 = new Panel();
panel1.ID = lblDeptId.Text;

panel1.Controls.Add(GridView1);
placeholder1.Controls.Add(panel1);
}
}
}
}
```

Client Side

```
<script type = "text/javascript">
$(document).ready(function(){
  $("#divMain").tabs({
   highStyle: "content",
   collapsible: true
  });
});
</script>
```

jQuery UI Dialog

- jQuery UI provides a function i.e. dialog(), used to display any set of user interface in the form of dialog box.

Syntax: $(selector).dialog();

- The jQuery UI dialog also supports following options

1. autoOpen
2. buttons
3. closeOnEscape
4. closeText
5. dialogClass
6. draggable
7. height
8. hide
9. maxHeight
10. maxWidth
11. minHeight
12. minWidth
13. modal
14. position
15. resizable
16. show
17. title
18. width

- The jQuery UI dialog also supports following method:

1. open()
2. close()

- The jQuery UI dialog also supports following events callback functions:

1. open
2. close
3. beforeClose
4. drag
5. resize
6. create

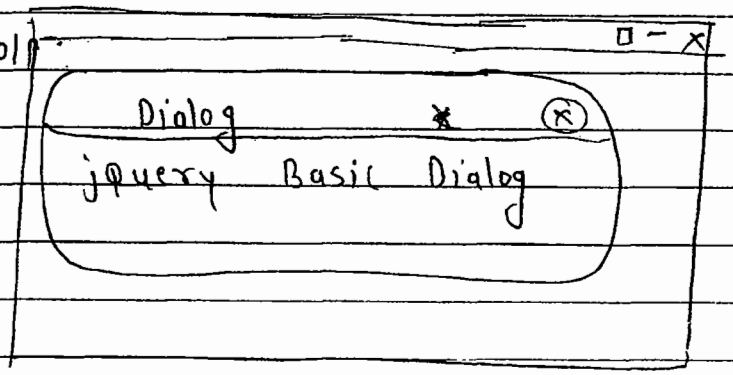
ex: Create an example of jQuery UI dialog to show a set of user interface in the form of dialog as soon as document is ready.

HTML

```
<asp:Panel ID="Panel1" runat="server" ToolTip="Dialog">
  jQuery Basic Dialog
</asp:Panel>
```

jQuery

```
<script type="text/javascript">
$(document).ready(function() {
$("#Panel1").dialog();
});
</script>
```



Ex: Create an example of jQuery UI dialog to display a set of interface in the form of dialog when any link or button is clicked.

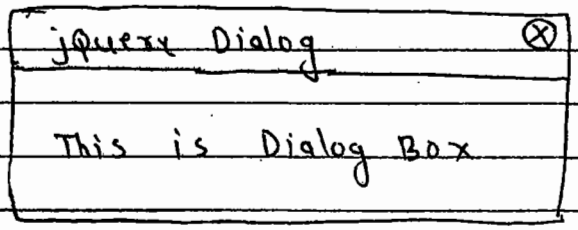
```
<a id="link1" href="#">Click Here</a>  
To open the Dialog Window
<div id="divDialog">This is Dialog Box</div>
```

jQuery

```
<script ="text/javascript">
$(document).ready(function(){
$("#divDialog").dialog({
 autoOpen: false,        // Default value true
 modal: true,            // Defaut value. false
 title: "jQuer Dialog",
 width: 500,             // Default value 300.
 height: 300,            // Default value "auto"
 draggable: false,       // Default value true
 closeText: "Close Popup", // Default value 'close'
closeOnEscape: false,    // Default value true
 maxHeight: 500,         // Default value is false
 maxWidth: 500,          // Default value is false
 resizable: true,        // Default value true
 show: {
  effect: "bounce",
  duration: 20000
  },

 hide: {
 effect: "slide",
 duration: 2500
      }
 });
$("#link1).click(function(){
$("#divDialog").dialog({autoOpen: true});
 });
});
</script>
```

Click Here To Open the Dialog Window

after click



Ex: Create an example of jQuery UI dialog to show the dialog with button as confirmation dialog. in which displays 2 buttons [i.e. ok or Cancle | Yes or No] so if user click ok or Yes button then any business operation can be performed. and if you click cancle or No button the we may close the dialog.

HTML

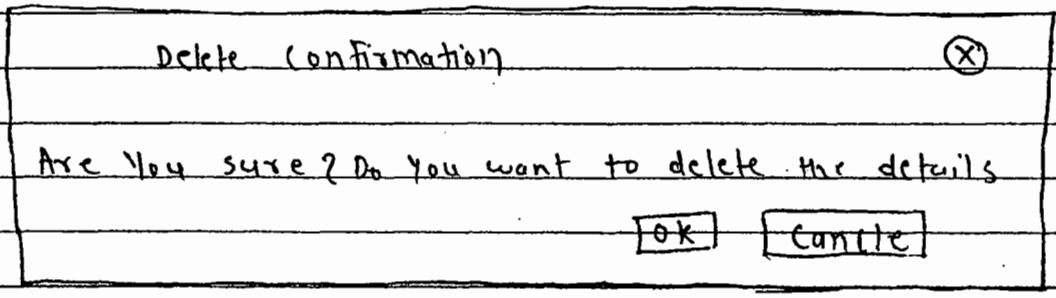
```
<input type="button" id="btnDelete" value="Delete" />
<asp:Panel ID="panel" runat="server">
Are You Sure? Do you want to delete the details?
</asp:panel>
```

jQuery

```
<script="text/javascript">
$(document).ready(function(){
$("#Panel1").dialog({"autoOpen": False});
$("#btnDelete").click(function(){
$("#Panel1").dialog({
"autoOpen": true;
"title": "Delete Confirmation",
"resizable": false,
```

```
"width": 500,
"modal": true,
buttons: {
  OK: function(){
// write business code to Delete the Details via AJAX
   Request only
  },

Cancle: function() {
  $(this).dialog("close");
 }
}
});
});
</script>
```

o/p: [Delete] after click on delete



Working With jQuery UI Dialog To Set Their Position :

- This can be done by using an option i.e. position and the default value is set to :

```
$(selector).dialog({position:{my:"center" at:"center",
                    of:window}})
```

- This option can be set their value in the form of object or string or array.

- Note : The string and array forms are deprecated [Remove]

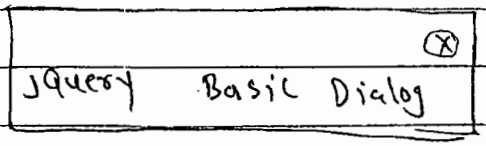
* Multiple types supported :

1. objects : It identifies the position of dialog when opened The off option defaults to the window But we can specify another element to position against.

- For example, initialize the dialog with position

```
<script="text/javascript">
$(document).ready(function(){
$("#Panel1").dialog({position:{my:"left top" at:"left
top", of:window}});
});
</script>
```

o/p :



② string : A string representing the position within the Few more posible values are left, right, ~~left~~. bottom, top, center.

③ Array: An array containg an x, y coordinate pair in pixel effect from the top, left corner (0, 0) of the view board or the name of a possible string value.

14.9.15

jQuery UI Tooltip

- jQuery UI provides a function i.e. tooltip(), used to displays a tooltip text when the mouse pointer overs on a particular element

- The jQuery UI tooltip replaces native tooltips, making them themeable as well as various custmizations:
  - display other content than just the title
  - customizing the positioning to center the tooltip above element.
  - add extra styling to customize the appearance for warening all error fields.

Syntax: $(selector).tooltip();

- The jQuery UI provides Following options:
  1. content
  2. disabled
  3. show
  4. Hide
  5. track
  6. position
  7. tooltipClass

- jQuery UI tooltip also supports following events callback Functions:
  1. Create
  2. Open
  3. close

- jQuery UI tooltip supports by default fade animation to show and hide the tooltip. but it can be changed with other animation using show and hide option.

ex: HTML

```
<b> First Name : </b>
<input type = "text" id = "txtFirstName" title = "Please
  Enter First Name "/>

<b> Last Name </b>
<input type = "text" id = "txtLastName" title = "Please
  Enter Last Name"/>
```

jQuery

```
<script type = "text/javascript">
$(document).ready(function() {
  $(document).tooltip({
  show: {
      effect : "fade",
      delay : 250
      },
  hide: {
      effect : "fade",
      delay : 250
    },
    track true
  });
});
</script>
```

// $("#txt1").tooltip({content: "Please Enter First Name"});
});, </script>

o/p :

Enter First Name: [ ]

(Enter First Name)

Enter Last Name: [ ]

→ it will move ~~error~~ along the cursor pointer.

```
$("#txt1").tooltip({disabled: true});
```

* jQuery UI Tooltip With CSS style %

```
<style type = "text/css">
.ui-tooltip{
padding : 10px;
background : black;
color : yellow;
border-radius:20px;
text-transform: uppercase;
box-shadow: 5px 5px gray;
border-color: blue;
}
</style>


<script type = "text/javascript">
$(document).ready(function(){
$("#txt1").tooltip();
});
```

jQuery UI Tooltip with custom CSS class:

```
<style type ="text/css">
.class1{
    background : red;
    color : white;
    box-shadow : 5px 5px blue;
    Font-weight : bold;
    Font-family : 'Monotype Corsiva'
    }
</style>

<script type ="text/javascript">
$(document).ready(function (){
 $("#txt1").tooltip({tooltipClass:"class1"});
});
```

* jQuery UI Slider [Imp]

- jQuery UI provides a function i.e. slider(), used to provides an user interface to be appear with slider which can be ~~interact~~ dragged left to right & right to left with their dragg handle to perform any operation.

- Or we can say the jQuery UI slider makes selected element into slider. There are various options such as multiple handlers and ranges.

- The handle can be move with the mouse ore the arrow keys.

Syntax :- $(selector).slider();

- The jQuery UI slider also supports following options:
1. animate 4. min 7. step
2. disabled 5. orientation 8. value
3. max 6. range 9. values

- jQuery UI slider also supports following events callback function:
1- Create 4. Change
2- Start 5. Stop
3- Slide *

```
<div id="divSlider1"></div>
<br/><br/>
<span id="span1"></span>
<br/><br/>
<img id="img1" src="Images/Ads2.jpg" alt="Image"/>

<script type="text/javascript">
$(document).ready(function(){
$("#divSlider1").slider({
 animate:"show",
orientation:"horizontal";
min:100,
max:700,
value:300,
```

step : 5,

```
slide : function(event, ui){
    $("#img1").width(ui.value);
    $("#img1").height(ui.value);
},

$("#span1").html("<b>Width :</b>" + $("#img1").
width() + "<br/><b>Height :</b>" + $("#img1").
height());
});

$("#img1").width($("#divSlider1").slider("value"));
$("#img1").height($("#divSlider1").slider("value"));

$("#span1").html("<b>Width :</b>" + $("#img1").width()
+ "<br/><b>Height :</b>" + $("#img1").height());
});
</script>
```

15.9.15

jQuery

- jQuery UI provides a function i.e. spinner(), used to make an input field [textbox] to be appear in the form of numeric up-down control for incrementing and decrementing the numeric value by clicking up and down button with the specified step.

Syntax: $(selector).spinner();

- The jQuery UI spinner also supports following options: similar to jQuery UI slider:

1. min
2. max
3. step

- jQuery UI spinner also supports events callback functions:

1. Create
2. Spin
3. Change
4. stop
5. start

ex:

```
<b> Enter Value: </b>
<input type="text" id="txt1" readonly="readonly" />
<br/>
<span id="span1"></span>
```

jQuery Code:

```
<script type="text/javascript">
$(document).ready(function(){
$("txt1").spinner({
```

```
min : 1,
max : 100,
step : 1,
spin : function (event, ui) {
$("#span1").html ("<b> Current Value : </b>" + ui.value);
}
});
$("#txt1").spinner ("value", 1);
$("#span1").html ("<b>Current Value : </b>" + $("#txt1")
.spinner ("value"));
});
</script>
```

o/p

Enter Value : [74] [▲▼]

Current value : 74

→ current value is changing along with Enter value.

* jQuery UI buttons :

- jQuery UI provides a function i.e. buttons(), used to makes a set of radio button, checkbox, links to be appear in the form of button style [appearance]

Syntax : $(selector).button();

ex :

```
<b> Select Option </b>
<br />
```

```
<input type="radio" id="rdb1" value="Option 1" name="Options" />
<label for="rdb1">Option 1</label>

<input type="radio" id="rdb2" value="Option 2" name="Options2" />
<label for="rdb2">Option 2</label>

jQuery
<sc
$(doc
$("input:radio[name=Options]").button();
});
</script>
```

o/p: Select Option :

~~button 1~~ ~~to~~ } This is Radio buttons but looks like buttons.

[Option 1] [Option 2]

Note: jQuery UI also provides an additional function i.e. buttonset(), used to apply on a container that contains set of elements such as Radiobutton, Checkbox, AnyLink to be appear in a form of button style.

Syntax: $(selector).buttonset();

```
ex: <div id="divLink">
<a href="http://www.google.com" target="_blank">Google</a>
```

```
<a href="http://www.yahoo.com" target="_blank">Yahoo</a>
<a href="http://www.msn.com" target="_blank">MSN</a>
</div>

<script type="text/javascript">
$(document).ready(function() {
$("#divLink").buttonset(); // $("a").buttonset();
});
</script>
```

o/p: [Google | Yahoo | MSN] (Google) (Yahoo) (MSN)

jQuery UI Menu:

- jQuery UI provides a function i.e. menu(), used to displays an user interface in the form of menus and sub menus to navigate to corrosponding link.

Syntax: $(selector).menu();

ex. [Prototype]

- Home
- About Us
  - About Management
  - About Sales
- Services
  - IT Services
  - BPO Services
- Contact Us

```
<ul id="menu1">
 <li>
 <a href="Home.aspx">Home</a>
</li>

 <li>
 <a href="About Us</a>
 <ul>
  <li>
   <a href="AboutMgmt.aspx" target="_blank">
     About Management </a>
 </li>

 <li>
 <a href="AboutSales.aspx">About Sales</a>
 </li>
</ul>
<li>

 <li>
  <a href="#">Serviles</a>
 <ul>
  <li>
  <a href="#">IT Serviles</a>
  </li>

 <li>
  <a href="#" BPO serviles</a>
 </li>
</ul>
<li>
```

```
<li>
<a href="ContactUs.aspx"> Contact Us </a>
</li>
</ul>
```

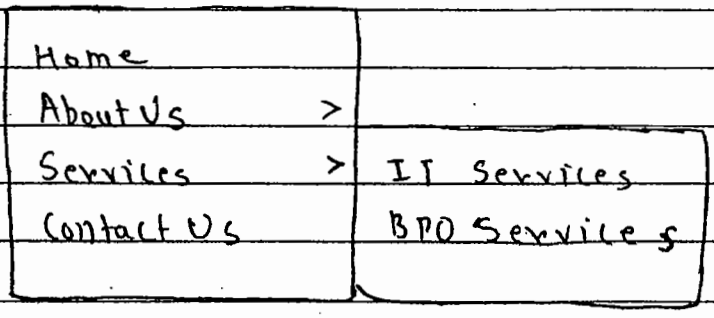
css:

```
<style type="text/css">
.vi-menu {
    width: 200px;
}
</style>
```

jQuery

```
<script type="text/javascript">
$(document).ready(function() {
$("#menu1").menu();
});
</script>
```

o/p:



* jQuery UI AutoComplete [V.V. IMP]

- jQuery UI provies a function i.e. autocomplete(), used to apply on an input field [textbox] to displays a list of matching words from the available list either at the client side or dynamically from the server.

- When uses types initial characters in the textbox [it is useful in searching like google search textbox, IRCTC search textbox for railway station, or bus & flight ticket booking web site search etc.]

Syntax: $(selector).autocomplete();

ex:

HTML

```
<b> Enter Language Name : </b>
<input id = "txtLanguage" type = "text" />
```

jQuery

```
<script
$(doc
// Defining the source object As Javascript Array
(this is static value for real time no use)
var languages = ["C", "C++", "C#", Visual Basic", "Python",
"Java", "Cobol", "Pascal", "Fortran", "VB script", "Java Script"];

$("#txtLanguage").autocomplete({source: languages, minLength: 1,
autoFocus: true});
});
</script>
```

o/p: Enter Language Name: C

C
C
c++
C#
⋮

// This is static

★ jQuery UI Interactions:

- jQuery UI provides mainly following 5 interaction helpers
  1. draggable
  2. droppable
  3. resizable
  4. selectable
  5. sortable

- jQuer UI draggable: The jQuery UI provides a function i.e. draggable(), used to make any element to be drag anyitem[selected] anywhere in the view~~port~~ code of the page. or it can be also drag in the specified direction. or it can be also drag in specified container only.

Syntax: $(selector).draggable();

ex:

```
<div id="div1" style="background-color: yellow"> AXIS:
    Move Anywhere </div>
<div id="div2" style="background-color: azure">
    AXIS-X </div>
<div id="div3" style="background-color: red">
    AXIS-Y </div>
```

```
<div id = "divContainer" style = "width : 500px; height : 300px;
border : 2px solid black">

<div id = "div4" style = "background-color : blue">
  In Container </div>
</div>
```

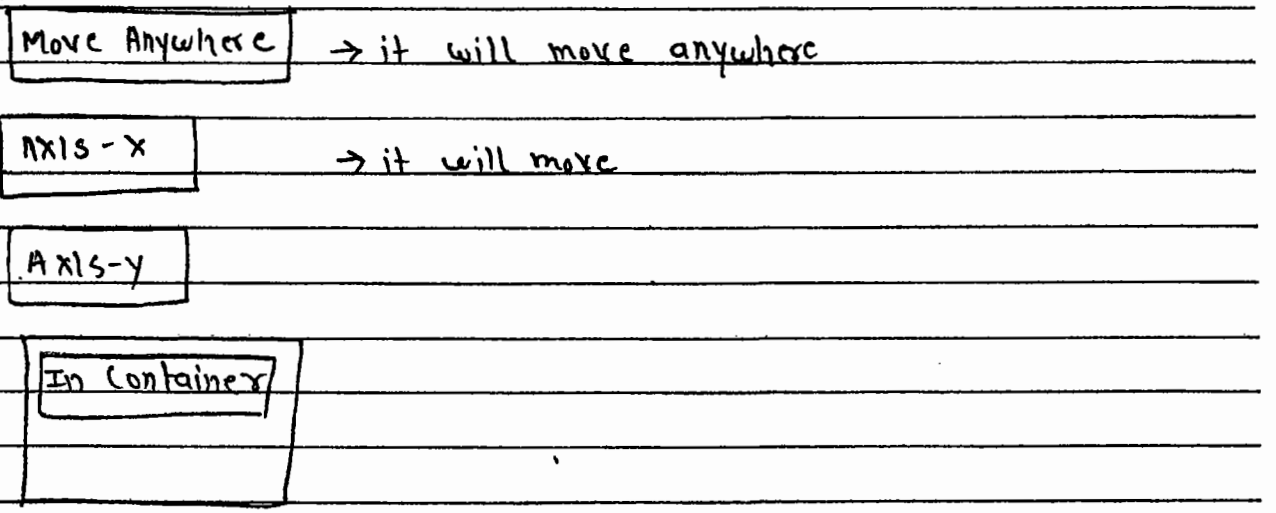
css:

```
<style type = "text/css">
#div1, #div2, #div3, #div4{
  width : 150 px;
  height : 100 px;
  border : 5px solid green;
  padding : 5 px;
  margin : 5 px;
  cursor : move;
}
</style>
```

jQuery

```
<script type = "text/javascript">
 $(document).ready(function(){
$("#div1").draggable();
$("#div2").draggable({axis:"x"});
$("#div3").draggable({axis:"y"});
$("#div4").draggable({containment: "#divContainer"});
  });
</script>
```

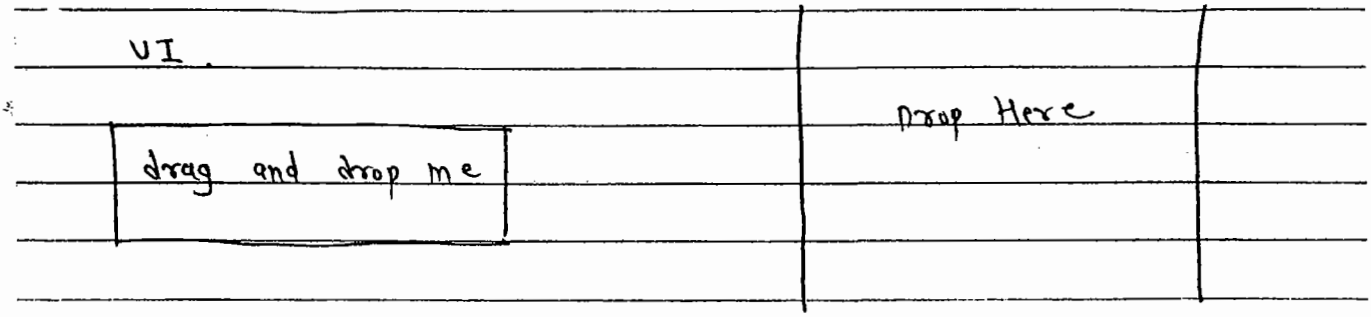
o/p :



* jQuery UI droppable : The jQuery UI provides a function i.e. droppable(), used to makes any selected element to be droped in the specified target location and as soon as it is dropped fire some action to perform any such operation using the drop event callback function.

Syntax: $(selector).drop()

Note: The jQuery UI droppable function is used to make a selected element to be dragged and droped nature.



```
<table width="100%">
 <tr>
  <td width="40%">
  <div id="divDraggable">
    Drag and Drop me
  </div>
 </td>
 <td width="60%" align="right">
 <div id="divDroppable" align="center">
   <p> Drop Here </p>
 </div>
 </td>
 </tr>
</table>
```

css:

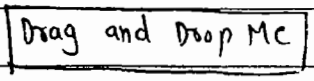
```
<style type="text/css">
#divDraggable{
  width: 150px;
 height: 100px;
 border: 5px solid green;
 padding: 5px;
 margin: 5px;
 background-color: yellow;
  cursor: move
}
#divDroppable{
  width: 300px;
```

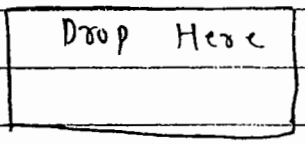
```
height : 200px;
border : 5px solid green;
padding : 5px;
margin : 5px;
background-color : aqua;
}
</style>
```

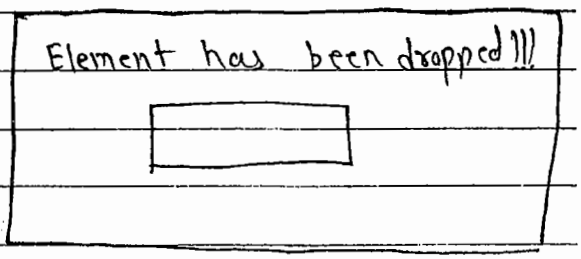
jQuery

```
<script type = "text/javascript">
$(document).ready(function() {
$("#divDraggable").draggable({cursor : "move"});
drop: function(event, ui) {
$(this).css({"background-color":"blue","color":"white",
"border" : "5px solid yellow"}).find("p").html("<b>
Element has been Dropped !!! </b>");
}
});
});
</script>
```

o/p:







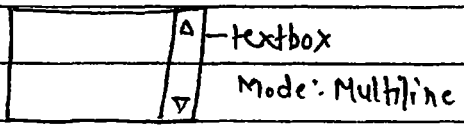
→ after dagging color is change and message displays as above.

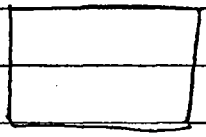
16-9-15

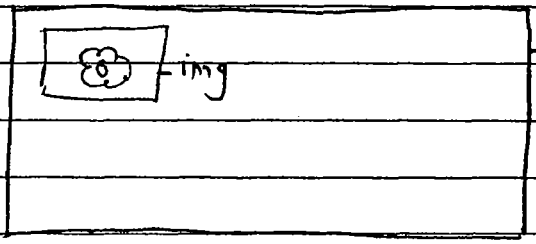
* jQuery UI Resizable : The jQuery UI provides a function i.e. resizable(), used to makes any element to the resize their width and height at the runtime.

Syntax : $(selector).resizable();

ex.








```
<body>
<textarea id = "txt1" rows = "5" cols = "10" ></textarea>

<div id = "div1" style = "width: 100px; height: 100px;
border: 5px solid red; padding: 5px; margin: 5px; ">
Resizable </div>

<div id = "divContainer" style = "width: 400px; height: 300px;
border: 2px solid black; padding: 5px; margin: 5px; ">

<img id = "img1" src = "Images/Ads2.jpg" width = "150" height =
"150" /> </div>
</body>
```

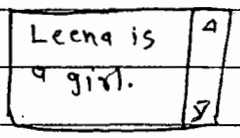
jQuery

```
<script type="text/javascript">
$(document).ready(function(){
$("#txt1").resizable();
$("#div1").resizable({
  minWidth: 100,
  minHeight: 100,
  maxWidth: 700,
  maxHeight: 300,
  animate: true,
  animateDuration: 5000,
  animateEasing: "easeInOutBounce", helper: "ui-resizable-helper"
  });
$("#img1").resizable({
  containment: "#div(container"});
  });
</script>
```

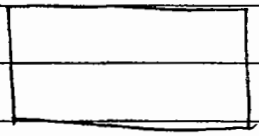
css:

```
<style type="text/css">
.ui-resizable-helper {
  border: 2px dotted red;
  }
</style>
```

output:



This will be resize



- This will also resize

* jQuery UI selectable: jQuery UI provides a functio, i.e. selectable(), used to makes a set of elements to be appear in the form of selection options.

Syntax: $(selector).selectable();

Ex:

```
<ul id="ulSelectable">
    <li>List Item 1</li>
    <li>List Item 2</li>
    <li>List Item 3</li>
    <li>List Item 4</li>
<li>List Item 5</li>
</ul>
```

css

```
<style type="text/css">
#ulSelectable{
 list-style-type: none;
 margine: 5px;
 padding: 5px;
 width: 40%;
}
#ulselectable li{
 margin: 3px;
 padding: 2px;
 font-size: 20px;
 height: 20px;
```

```
background : aqua;
border : 1px solid gray;
cursor : pointer;
}
#ulSelectable. ui-selecting {
background : red;
color : white;
}
#ulSelectable. ui-selected {
background : green;
}
</style>
```

jQuery

```
<script type = "text/javascript">
$(document). ready ( function () {
$ ("#ulSelectable"). selectable();
});
</script>
```

Output :

Item 1
Item 2
Item 3
Item 4
Item 5

→ while selecting the color is red after selected color is red.

* jQuery UI Sortable: jQuery UI provides a function i.e. sortable(), used to makes a set of elements to sortable by adjusting their position or we can say by repositioning them.

Syntax: $(selector).sortable();

Ex:

```
<ul id="ulSortable">
    <li> List Item1 </li>
    <li> List Item 2 </li>
    <li> List Item 3 </li>
    <li> List Item 4 </li>
    <li> List Item 5 </li>
</ul>
```

CSS

```
<script type="text/css">
#ulSortable{
list-style-type: none; // if we remove this prop. bullet symbol
margin: 5px;                will come.
padding: 5px;
width: 60%;
}
#ulSortable li{
margin: 3px;
padding: 2px;
font-size: 20px;
height: 20px;
```

```
background : aqua;
border : 1px solid gray;
cursor : move;
}
</style>
```

jQuery

```
<script type = "text/javascript">
$(document).ready(function() {
$("#ulSortable").sortable();
});
</script>
```

Output:

- List Item 1
- List Item 2
- List Item 3
- List Item 4
- List Item 5

} We can sort it as per our requirement [reposition]

jQuery Ajax

JSON = Javascript Object Notation

22.09.15

jQuery slideshow plugin : [malsup.com/jquery/cycle]

- These are the following popular slide show plugin available to use it

1> jQuery Cycle Plugin

URL's:

1. jQuery Cycle
http://malsup.com/jquery/cycle/

2 → Mouse over images plugin for pulse effect.
http://malsup.com/jquery/hoverpulse/

23/9/15

~~2. jCarousel~~

2. tCycle Plugin
http://jquery.malsup.com/cycle2/tcycle
[t - stands for "tiny" and by "tiny" I means less 1 kilobytes]

3. Cycle2 Plugin
http://jquery.malsup.com/cycle2/

4. jCarousel Plugin
http://sorgalla.com/jcarousel/

5. desoSlide Plugin
http://sylouuu.github.io/desoslide/doc/demo.html

6. jQuery Slides
http://plugins.jquery.com/slides/
http://greenish.github.io/jquery-slides/

7. bxSlider : jQuery Content Slider
http://bxslider.com/

8. SlideJS Plugin
http://www.slidesjs.com/

9. FlexSlider Plugin
http://flexslider.woothemes.com/

10. jQuery-Waterwheel-Carousel
http://www.bkosborne.com/jquery-waterwheel-carousel

11. Wow Slider
http://wowslider.com/

24.9.15 jQuery Validation Plugin

- http://jqueryvalidation.org/

download the jquery core code and validation plugin via manageNuget packages inorder to work with jquery validation.

- Search [jquery validation] // Manage NuGet Package Window

- Create an example of ASP.NET webform with jquery validation.

Validation Example

First Name [ ]
EmaiId [ ]
Mobile No [ ]
Password [ ]
Confirm Password [ ]
[Submit]

jQuery code

```
<script src="Scripts/jquery-2.1.4.js" type="text/javascript">
</script>
<script src="Scripts/jquery.validate.js" type="text/javascript">
</script>
<script type="text/javascript">
$(document).ready(function(){
$("#form1").validate({
rules: {
<%= txtFirstName.UniqueID %>: "required",
<%= txtEmailID.UniqueID %>: {
```

```
        required: true,
        email: true
    },
<%= txtMobileNo, UniqueID %>: {
    required: true,
    digits: true,
    minlength: 10,
    maxlength: 10
    },
<%= txtPassword, UniqueID %>: {
    required: true,
    minlength: 6
    },
<%= txtConfirmPassword.UniqueID %>: {
    required: true,
    equalTo: "#txtPassword"
    }
    },
    messages: {
<%= txtFirstName.UniqueID %>: {
    required: "Please Enter First Name"
    },
<%= txtEmailID.UniqueID %>: {
    required: "Please Enter Email Id",
    email: "Invalid Email Id Format"
    },
<%= txtMobileNo.UniqueID %>: {
    required: "Please Enter Mobile No",
    digits: "Please Enter Digits Only"
    },
```

```
<%=txtPassword.UniqueID%>:{
required:"Please Enter Password"
},
<%=txtConfirmPassword.UniqueID%>:{
required:"Please Enter Confirm Password",
equalTo:"Confirm Password Must be Same As Password"
}
}
});
});
</script>
```

CSS

```
<style type="text/css">
label.error{
color:red;
Font-family:Tahoma;
Font-size:14px;
margin:5px;
}
</style>
```

- When you are working with multiple javascript you can use as follows:

```
<script type = "text/javascript>
  var n$ = jquery.noConflict();          // For creating your own symbol [called as shorthand notation]
  n$(document).ready(function(){

  });
</script>
```

The End
24.9.15