JQuery

Class notes

By
Mr . **Narasimha** sir

Sathya Technologies

JQuery

⇒ <u>What is jQuery?</u>

→ JQuery is a Javascript library.

→ It is used to perform all Kinds of client side activities

→ It simplifies javascript programming.

→ It provides built in methods to perform all types of clientside activities.

→ jquery can be intigrated with any serverside technologies like JSP, Asp, Asp.net & Asp.Net mvc.

<u>Note</u>:- jquery was developed & created by "JohnResig".

→ compare with other clientside libraries jquery is most popular Js library.

→ JQuery is "opensource" library. Any one can download & intigrate with their technology.

13/10/15

⇒ <u>How to integrate jquery in webpages?</u>

1) Download jquery library from "www.jquery.com" website.

   filename : jQuery-1.11.3.min.js.

2) Integrate with a webpage by using <script> tag.
   ```
   <script: scr ="jQuery-1.11.3min.Js">
   </script>
   ```

3) Write the client side code. using jquery with another script tag.
   ```
   <script>
      // required code
   <Iscript>
   ```

→ **jQuery features:**

→ jQuery is a javascript library. It provides built-in function to perform clientside operations.

→ These functions will be used based on the required feature that you want to use for the requirement.

→ jQuery provides the following features.

1) **DOM Traversing :-** (Document Object Model)

→ Dom Traversing feature is used to access html elements at runtime

→ So, that we can perform required operations.

→ jQuery provides different operations to implement Dom Traversing.

   Ex:- Selectors, each(), eq()].

2) **DOM Manipulations:-** (Document Object Model)

→ we can update the DOM elements at runtime based on the requirement.

→ Ex:- Adding additional rows to table. Removing items from the list etc;

→ To perform these manipulations jQuery provides pre-defined functions.

   Ex:- append(), remove(), empty().

3) **Event Handling :-**

→ Similar to javascript we can handle all the events in jQuery

→ In jQuery, we can handle the events by using either named functions & unnamed functions.

→ jQuery provides some additional events that are not exists in javascript

   Ex:- ready, success, error

→ Most of the events are same as javascript.

Ex:- click, change, focus, blur. etc.,

<u>M|10|15</u>

4) jQuery Styles :—

→ Dynamic styles in web pages can be implemented using javascript

→ Instead of writing manual javascript code, jQuery provides readymade methods to perform dynamic styles.

→ we can easy to do this task with jQuery methods.

Ex:- css(), addclass( ), removeclass( ).

5) Animations in jQuery

→ jQuery allows to apply animations on any html element

→ Animation in a concept to change the behaviour of an object from one state to another state in the given interval of time.

→ jQuery library provides a method called "animate()" to implement animation.

→ we can apply animation on different styles properties like size, position, etc.,

6) jQuery effects :—

→ In order to apply different types of effects on html elements, jQuery provides pre-defined methods.

→ In order to core jQuery, we can apply sliding effects, fading effects etc.

→ jQuery-UI provides more than 10 effects to apply on html elements.

method : effect().

7) Lightweight Js Library:-

→ jQuery library is available in two different sizes.

a) compressed

b) uncompressed

a) → Compressed one is used to deployment/ hosting purpose so, that browser will easily download the file from server.

→ Comprend file is very less size due to this downloading from server, loading in browser memory & execution will be faster.

file size is: 94KB.

8) CSS3 Complaint :-

→ jquery supports all versions of css properties.

→ It includes old version & latest version of css. (css3)

→ we can use css3 style properties also in jquery methods.

9) Cross Browser :-

→ jquery library is browser independent.

→ It means it will works in all browsers.

→ jquery library will take care execution of the code based on Browser.

→ It supports all browsers : IE, Chrome, firefox, Opera etc.,

10) jquery is Extendable :-

→ jquery is open source to that anyone can use.

→ jquery is more comfortable to extend and add additional features by third party developers.

→ These features are consider as jquery plug-in's.

→ we can find so many jquery plug-in's in internet.

Ex: Image slid show plugin

Zooming plugin

Validation plugin

Java Script

→ Javascript is a scripting language used for client side programming in webpages.

→ Javascript is light weight programming language.

→ Javascript in case sensitive.

→ control structures and operators are same as c/c#/Java languages.

→ Java Script is used to perform following operations:

    → To perform clientside Validation.

    → To perform clientside dynamic operations.

    → clientside calculations.

    → To control the behaviour of html elements

    → To implement Ajax technology.

Javascript Programming:

→ we should understand the following important rules / points to develop Javascript programs.

1) Include Javascript in webpage.

    → we can include the Javascript in webpage either directly (or) indirectly.

Directly :-
```
<script type = "text/javascript">
    // Javascript code
</script>
```

Indirectly (External files) :-
```
<script type = "text/javascript"  src = "test.js">

</script>
```

Note :- <script> tag can be add in any section of html document (either head (or) body).

2) Creating Variables :-

→ Java script Variables are declare with "var" keyword.

→ These type of Variables are called Dynamic Typing Variables.

→ It will change the behaviour based on the value that we stored in it.

Examples:-

```
Var x,y,z;
Var x=100;
Var st="Hello World";
Var ar=[10,20,30];
Var obj={ };
```

3) Numeric conversions :-

→ By default all html controls returns the value in string format.

→ To perform numeric operations we need convert string formatted data into numeric data.

→ Java script contain two builtin functions to do this conversation.

1) parpse Int()

2) parpse Float()

Ex:-
```
Var x,y,z;
x= parpse Int (obj1-Value);      → reference of TextBox 1
y= parpse Int (obj2-Value);      → reference of TextBox 2
z=x+y;
```

4) Accessing reference of html elements :-

→ we can get the reference of any html element by using "document" object.

→ "getElementById" is function of "document" object.

usage:

```
document.getElementById ("Id of html element");
```

Note:- we can get the reference, we can access the details of the element like value, text, attributes etc.,

5) Creating functions in Javascript:-

→ If you write the code in script tag, it will execute only at the time of page load.

→ If you prepare the code using functions, that can execute whenever we need.

Syntax :-

```
function fn_name ([args])
{

}
```

Note:- No need to mentioned return type & arguments type. Just provide required Arguements. Return statement can be include in function if required.

6) Event Handling :-

→ All html elements supports events.

→ we can handle these events by using Javascript functions.

Ex:-

Description	Html event
page load	onload
Button click	onclick
mouse over	on mouseover
mouse out	onmouseout
curper Enter	onfocus
curper Leave	on blur.

→ we can invoke the Javascript functions using these events.
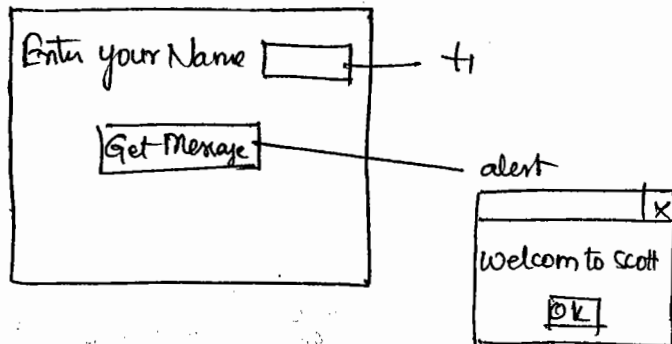
Ex:-

1) `<input type="Button" id="b1" onclick="f1()" />`

2) `<body onload="f1()" />`

Note:-

→ use the following statement to display output message:

   alert(" message");

Example:7:-


```html
<head>
   <script type="text/Javascript">
      function f1()
      {
         Var obj=document.getElementById("t1").Val;
         alert(" welcomet to + " obj.Value);
      }
   </script>
</head>
<body>
   Enter your Name <input type="tent" id="t1"/>
   <input type="button" id="b1" onclick="f1()" Xvalue="GetMessage"/>
</body>
```

16/10/15   **Example:2**
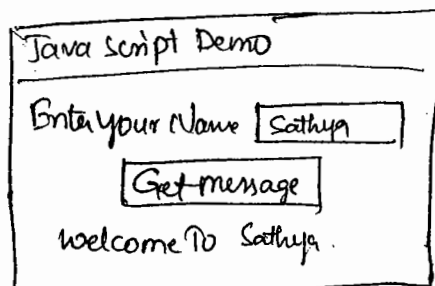
```html
<html>
  <head>
    <script>
        function f1()
        {
            Var obj = document.getElementById("t1");
            Var spobj = document.getElementById("sp1");
            SpObj.InnerHTML = "Welcome To <u>" + obj.Value + "</u>";
        }
    </script>
  </head>
  <body>
    <h1> Java script Demo1 </h1>
      <hr>
    Enter your Name <input type="text" id="t1" /> <br></br>
    <input type="button" id="b1" onclick="f1()" Value="GetMessage" />
      <br></br>
      <span id="sp1" />
  </body>
</html>
```

Example 3 :-


```html
<html>
  <head>
    <script>
        function f()
```

```
    {
        Var obj1 = document.getElementById("t1");
        Var obj2 = document.getElementById("t2");
        Var z = parseInt(obj1.Value) + parseInt(obj2.Value);
        Var spobj = document.getElementById("sp1");
        spobj.innerHTML = "Result is:" + z;
    }
</script>
</head>
<body>
    Enter First Number <input type="text" id="t1" /> <br></br>
    Enter Second Number <input type="text" id="t2" /> <br></br>
    <input type="button" id="b1" Value="Add" onClick="f1()" /> <br></br>
    <span id="sp1" />
</body>
</html>
```

Example : 4:-


```
<html>
    <head>
        <script>
            function fnAdd()
            {
                Var obj1 = document.getElementById("t1");
                Var obj2 = document.getElementById("t2");
```

```javascript
Var z = parseInt(obj1.value) + parseInt(obj2.value);
Var spobj1 = document.getElementById('sp1');
Var spobj2 = document.getElementById('sp2');
spobj1.innerHTML = "operation:" + "Add";
spobj2.innerHTML = "Result is:" + z;
}
function fnsub()
{
Var obj1 = document.getElementById("t1");
Var obj2 = document.getElementById("t2");
Var z = parseInt(obj1.value) - parseInt(obj2.value);
Var spobj1 = document.getElementById("sp1");
Var spobj2 = document.getElementById("sp2");
spobj1.innerHTML = "operation:" + "Subtraction";
spobj2.innerHTML = "Result is:" + z;
}
function fnmul()
{
Var obj1 = document.getElementById("t1");
Var obj2 = document.getElementById("t2");
Var z = parseInt(obj1.value) * parseInt(obj2.value);
Var spobj1 = document.getElementById("sp1");
Var spobj2 = document.getElementById("sp2");
spobj1.innerHTML = "operation:" + "Multiply";
spobj2.innerHTML = "Result is:" + z;
}
```

```html
<body>
Enter First Number <input type="text" id="t1" /> <br></br>
Enter Second Number <input type="text" id="t2" /> <br></br>
<input type="button" id="b1" value="Add" onClick="fnAdd()" /> <br></br>
<input type="button" id="b2" value="sub" onClick="fnsub()" /><br></br>
```

```html
<input type="button" id="b3" value="Multiply" onClick="fnmul()"/>
<br></br>
<input type="button" id="bu" value="Division" onClick="fnDiv()"/>
<br></br>
<span id="sp1"/>
<span id="sp2" />
</body>
</html>
<head>
  <script>
    function fnDiv()
    {
      var obj1 = document.getElementByID("t1");
      var obj2 = document.getElementById("t2");
      var z = parpeInt(obj1.value) / parpeInt(obj2.value);
      var spobj1 = document.getElementByID("sp1");
      var spobj2 = document.getElementById("sp2");
      spobj1.innerHTML = "Operation : Division";
      spobj2.innerHTML = "Result is:" + z;
    }
  </script>
</head>
```

Example 5: (Same example)

```html
<html>
  <head>
    <script>
      function fn_demo3(arg)
      {
```

```
Var obj1=document.getElementById ("t1");
Var obj2=document.getElementById ("t2");
Var x = parseInt(obj1.Value);
Var y = parseInt(obj2.Value);
Var z =0;
Switch (arg)
{
    Case "add"   ; z= x+y;
                    break ;
    Case "sub"  : z= x-y;
                    break ;
    Case "mul"  : z= x * y;
                    break ;
    Case "div"  : z = x/y;
                    break ;
}

Var spobj = document.getElementById ("sp1");
Spobj. innerHTML = "Operation:" + arg ;
Spobj. innerHTML +="<br>Result is:' + z ;

}
</Script>
</head>
<body>
    Enter first Number <input type="text" id="t1" />
    Enter second Number <input type="text" id="t2" />
    <input type="button" id="b1" Value="Add" OnClick="fn-demo3 ('add')" />
    <input type="button" id="b2" Value="Subtract" onclick="fn-demo3 ('sub')" />
    <input type="button" id="b3" Value="multiply" onClick="fn-demo3 ('mul')" />
    <input type="button" id="b4" Value="Division" onClick="fn-demo3('Div')" />
    <span id="sp1" />
</body>
</html>
```

Task:-

Product Name [_____]

Quantity [_____]

Unit Price [_____]

[Get Btal]

Total is $
↓
In Green color

17/10/15

⇒ Example 6:—

Image1 ▼

```
<html>
  <head>
    <script>

      function fn-demo4()
      {
         Var obj = document. getElementById (" img1");
         Var obj2 = document. getElementById (" List1");
         Var fname = "Image1" + obj. Value + ".jpg";
         obj.src = fname;

      }
    </script>
  </head>
  <body>
     Select Image : <select id="List1" onChange = "fn-demo4()">
```

```html
<option> Image1 </option>
<option> Image2 </option>
<option> Imag3 </option>
</select>
<br> <lbr>
<img  src=" Images / img1. jpg"  id=" img1".
    height:" 200 px"  width =" 400px"  Boarder = "2" />
</body>
</html>
```

⇒ Example 7 :—
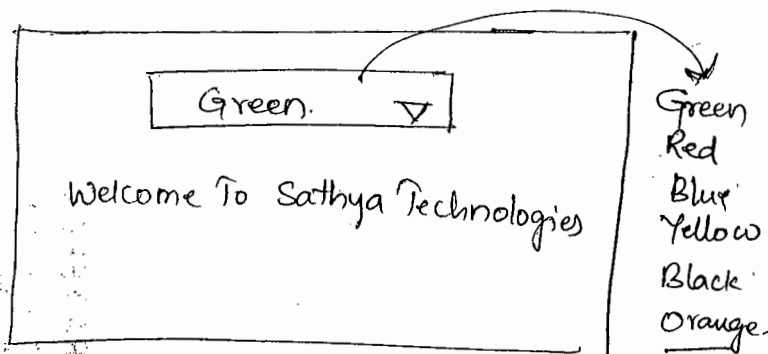

```html
<html>
<head>
    <script>
        function fn-demo5()
        {
            Var obj = document. getElementById (" list1");
            Var obj1 = document. getElementById (" h1");
            // Var obj2 = document. getElementById ("
            obj1. Style. color = obj. Value;
        }
    </script>
</head>
```

```html
<body>

    Select Color: <select id="list" onChange="fn-demo5()">
            <options> Green </options>
            <options> Red </option>
            <options> Blue </option>
            <options> Yellow </options>
            <option> Black </option>

            </select> <br></br>
<h1 id="h1">
        Welcome To Sathya Technologies          </h1>
</body>
</html>
```

⇒ Example : 8 :—
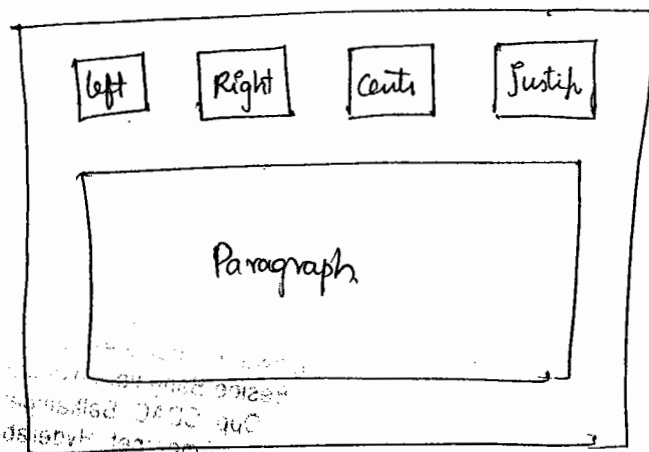
```html
<html>
 <head>
   <style>
     .img
       {
          boarder: 1px solid Gray;
          height: 50px;
          weight: 50px;
       }
      .P
       {
          boarder: 2px solid Blue;
          padding: 5px;
       }
   </style>
```

```html
<script>
    function fn-demo6(str)
    {
        var obj = document.getElementById("p1");
        obj.style.textAlign = str;
    }
</script>
</head>
<body>
    <img onClick="fn-demo6('left')" src="left.png"/>
    <img onClick="fn-demo6('Right')" src="Right.png"/>
    <img onClick="fn-demo6('center')" src="center.png"/>
    <img onClick="fn-demo6('justify')" src="justify.png"/>

    <br><br>

    <p id="p1"> This is text paragraph to change the textAlignment at runtime </p>
</body>
</html>
```

Task! onmouseover.

19/10/15

⟹ Example: 9!—
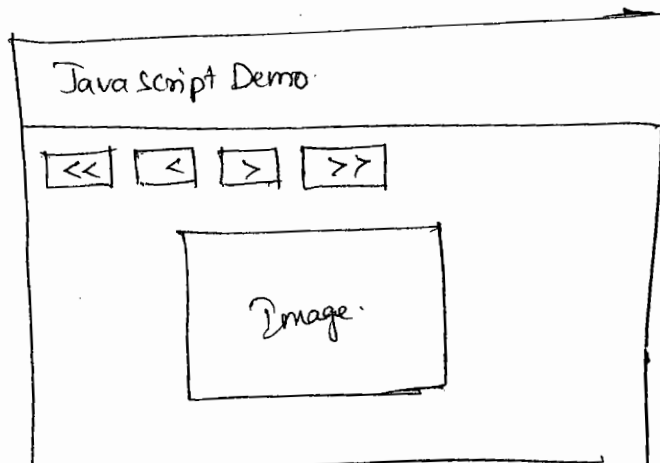
```html
<html>
  <head>
    <script>
      Var n=1
      function f1()    // Next Image
      {
        if(n == 7)
        {
          n = 0;
        }
        n = n + 1;
        Var obj = document.getElementById("img1");
        string str = "images\image" + n + ".jpg";
        obj.src = str;
      }

      function f2()    // prev. Image.
      {
        if(n == 1)
        {
          n = 7;
        }
        n = n - 1;
        Var obj = document.getElementById("img1");
        string str = "images\image" + n + ".jpg";
        obj.src = str;
      }

      function f3()  //first image
      {
        n = 1;
        Var obj = document.getElementById("img1");
        string str = "images\image" + n + ".jpg";
        obj.src = str;
      }
```

```
function f4( ) // Last Image.
{
    n = 7;
    var obj = document. getElementById ("img ");
    string str = "images\image "+ n + "-jpg" ;
    obj.src = str;
}
</script>
</head>
<body>
    <h1> JavaScript Demo </h1>
    <hr1>
    <Button title="firstImage"  onClick = "f3()"  src="images\image1-jpg" > << </Button>
    <Button title="PrevImage"   onClick = "f2()"  src="images\image1-jpg" > < </Button>
    <Button title="NextImage"   onClick = "f$()"> > </Button>
    <Button title="LastImage"   onClick = "f4()"> >> </Button>.
    <br></br>
    <img  id="img1"  src="images\image1-jpg"  height="200px", width="200px"
                                    boarder="2" />
</body>
</html>
```

⇒ Example 10 → Timer Example :-

```
<html>
    <head>
        <script>
            var n=1;
            function fn-update( )
            {
                if( n==7)
                {
                    n == 0;
                }
                n= n+1;
```

```
Var obj = document . get Element By Id ("img1");
string str = "images\image "+ n +".jpg";
obj . src = str;

}

    Var timerobj;

    function fn_start()

    {

       timerobj = setInterval ( fn_update, 1000);

    }

    function fn_stop()

    {

       timerobj = clearInterval ( );

    }

</script>
</head>
<body>

    <h1> Javascript Timer Example </h1>

       <hr />

    <Button  onClick = "fn_start()" > Start </Button>

    <Button  onClick = " fn_stop()"> Stop </Button>

      <br> </br>

    <img  id="img1"   src=" images\image 1.jpg" height="200px" width="200px"
                                             boarder = "2" />
   </body>
</html>
```

Cascading Style Sheets (css) in Web Development

→ By default all the html tags does not support attributes to change its appearance. (Background color, foreground color, etc ...)

Ex:- <h1> tag does not support attributes to change Background. Color.

→ To overcome these practical problems, css concept was introduced.

→ css stands for Cascading Style Sheets.

→ By using css, we can apply any required appearances to all html tags.

→ In css, there is no restriction on applying styles to html elements.

⟹ Implementing of css :—

→ We can implement css concept at three different levels

1. Tag Level   (inline)
2. Page level   (Internal)
3. Project level (External)

1) Tag Level :—

→ In tag level styles implementation, we are using "style" attribute.

→ "style" attribute can be applicable for all html tags.

Ex :-
  <h1 style = "color : Blue' > welcome <1h1>

2) Page Level :—

→ If you want to apply same styles for multiple tags in the page, we can use page level styles.

→ Page level styles are implemented using "<style>" tag

→ \<style> tag is defined within \<head> tag.

Example:-

```
<head>
  <style>
    h1
    {
        Color : Blue;
        Background -Color : Pink;
    }
  </style>
</head>
```

3) Project Level :-

→ If you want to apply same styles for multiple pages in the website, you can use "Project level" styles.

→ Project level styles are implemented by using a seperate file (*.css).

→ To refer the *.css file from html pages, we can use "\<link>" tag

Ex:-

```
<link href="MyStyles.css" rel="Stylesheet"/>
```

Note:- we can use same tag from ASPX/css HTML, *.jsp, *.php etc., files also in order to refer *.css file.

→ Major advantages of Project Level styles:

1) Reusability across multiple web pages.

2) Modifications will be more easier.

⇒ Defining styles:-

→ we can define the styles in css in three ways.

1. Tag name based.

2. Class name based. (.)

3. ID based. (#)

1) Tag Name Based:-

→ By default css will consider as tagname based.

→ In this option we are specify the tagname at the time of defining styles.

    ex:- h1, span, p, table etc.,

2) Class Name Based:-

→ If you want to apply styles for specific set of html elements, then we can use this option.

→ first we have to provide class name for tags using "class" attributes.

    Ex:-
```
<h1  class="c1">
<P   class="c1">
etc.,
```

→ Then we can define the styles based on class name.

    Ex:-
```
<style>
.c1
{
   color=Red;
}
</style>
```

3) ID Based:-

→ It is used to apply styles for specific element.

→ provide the "id" for html element, then define the style based on id.

    Ex:-
```
<span id="sp1"> -- </span>
<style>
   #sp1
   {
      color=Blue;
   }
</style>
```

→ Example to applying css styls to html webpage

```html
<html>
  <head>
    <style>
      h1
      {
        background - color : Green;
        color : white;
      }
      .RedClass
      {
        border : 2px solid Red ;
        background - color : Pink;
        color : Red;
      }
      .BlueClass
      {
        border : 2px solid Blue;
        background - Color : LightBlue;
        Color : Dark Blue;
      }
      #img1
      {
        border : 2px solid Red;
        height : 180px;
        width : 300px;
        padding : 5px;
        border - radius : 50px;
      }
    </style>
  </head>
```

```html
<body>
    <h1> CSS DEMO </h1>
    <h1>CSS DEMO</h1>
    <hr></hr>
    <P class = "RedClass"> This is test paragraph </P>
    <P class = "BlueClass">This in Test Paragraph </P>
    <br></br>
    <img id = "img1" src = "Image1.JPg" />
    <br></br>
    <span style="color: Blue; border: 2px solid Green; padding: 10px;">
        CSS DEMO web Page . Hello World .
    </span>
</body>
</html>
```

26/10/15

$\Rightarrow$ Developing Webpages using jQuery :-

→ jQuery is a Javascript Library.

→ Using jQuery we can perform all clientside activities.

→ jQuery library can be downloaded from below website.

   www.jQuery.com.

   file: jquery-1.11.3.min.js.

Note :- Download & import the above file in your webpage..
         by using <script> tag.

Ex :- <script src = "jquery-1.11.3.min.js'> </script>

Ex1:
```html
<!-- creating a webpage to display a welcome message
        using jquery -->

<html>
  <head>
    <script src="jquery-1.11.3.min.js"></script>
    <script>
      function f1()
      {
        $("#sp1").text("welcome to jquery programming");
      }
    </script>
  </head>
  <body>
    <h1> jQuery Programming Demo </h1>
    <hr1>
    <input type="button" id="b1" value="Get Message"
              onclick="f1()" />
    <br> </br>
    <span id="sp1"> </span>
  </body>
</html>
```

Ex2:
```html
<!-- creating a webpage to display a message using ready
      event of jquery --->

<html>
  <head>
    <script src="jquery-1.11.3.min.js"></script>
    <script>
      $(document).ready
      (
```

```
function ()
 {
   alert ("welcome to ready event");
   $ ("#sp1"). text ("welcome to ready event in jquery programming");
 }
);
</script>
</head>
<body>
  <h1> jQuery Ready event Demo </h1>
   <hr1>
   <span id = "sp1" > </span>
 </body>
</html>
```

Note :—

1) $ (document). ready();

⇒ $: The $ (doller) jquery function. jquery function is called as factory function. which is used to refer all html elements in clientside programming.

→ Every statement in jquery starts with jquery function.

→ $ is the shortcut notation of jquery function.

> e.g. → $() (or) jQuery()

⇒ document :- It is a predefined object in Javascript. jquery user this object, inorder to execute "ready" event.

⇒ ready() :-

→ It is a jquery event that will be fired, after all the document elements are ready.

→ To subscribe this event, we have to pass a function that should execute, when the event is firing.

→ we can subscribe the events in 2 ways.

1) By using named function

```
$ (document). ready (f1);

function f1()
{
    // function body.
}
```

2) By using un-Named / Anonymous functions

(i)
```
$ (document). ready (
    function ()
    {
        // Body
    }
);
```

(ii) Standard:
```
$ (document). ready (function()
    {
        // Body
    });
```

⇒ $ ("#sp1"). text ()

→ The Above statement in a jquery selector to access the reference of the span tag, based on id.

⇒ text():— It in jquery function / method used to get (or) set text belongs to html tags (except html controls).

27/10/15

⇒ **jQuery Selectors:—**

→ jQuery selectors are used to get the reference of html elements.

→ once we get the reference, we can apply / perform required operations.

→ $("string") is refered as jQuery selector.

→ It provides several options to select html elements based on our requirement.

→ Before applying any operation on html tags, first we need to select with jquery selector.

Usage of jQuery Selectors:-

→ We can select the html tags based on its attributes (or) tag name.

→ jQuery selectors are implemented based on the following options.

1) **Tag based**

→ Get the html elements based on the tag name.

Syntax:- $("tagname")

Ex:- $("h1")

   $("div")

   $("p")

2) **Class Based**

→ Get the html elements based on the class name.

→ "class" is one of the attributes of html tags.

→ "class" attributes is available for all html tags.

→ It is used to group more than one html elements

Syntax: $(".className")

Ex:- $(".class1")

   $(".myClass")

3) **ID Based**

→ If you want to select specific element, you can select based on id-

Syntax :- `$ ("#id")`

Ex:- `$("# txt1")`

`$ ("#div1")`

`$ ("# p1")`

4) **Select all elements**

→ use "*" to select all elements.

Ex:- `$ (" * ")`

5) **Select multiple elements**

→ If you want to select multiple items, specific names by seperating with " , ".

Ex:- `$ (" h1 , P ")`

`$ ("# text1 , " text 2")`

`$ (".class1", ".class 2")`

6) **Selecting document**

→ All jquery codes with placed with in document selector.

Ex:- `$ (document).`

7) **Current Item Selection**

→ To select current element, we can " `$ (this)` " selector.

Ex:- `$(this)`

⇒ Example :-

```
<!-- Create a webpage to perform sum operation using jquery -->

<html>
  <head>
    <script src="jquery-1.11.3.min.js"> </script>
    <script>
      $(document).ready(function()
      {
        $("#b1").click(function()
        {
          var x = $("#t1").val();    // get
          var y = $("#t2").val();    // get
          var z = parseInt(x) + parseInt(y);
          $("#t3").val(z);    // set
        });
      });
    </script>
  </head>
  <body>
    <h1> jquery Selectors </h1>
    <hr></hr>
    Enter first Number : <input type="text" id="t1" /> <br><br>
    Enter Second Number : <input type="text" id="t2" /> <br><br>
    Result <input type="text" id="t3" readonly="true" /> <br><br>
    <input type="button" id="b1" value="sum" />
  </body>
</html>
```

→ Methods to get/set the content of html elements

1) Val()

→ Val() method is used to get/set Value of html controls Like textBox, password, checkBox etc.,

usage:-

```
Var x = $("#text1").Val();
$("#text1").Val("Hello");
```

2) text()

→ This function is used to get/set data from content tags like <P>, <h1>, <span>, etc ...

usage:-

```
Var str = $("P1").text();
$("#P1").text("Hello World");
```

3) html()

→ If you want set/get the text with html tags involvement, then you can use "html()".

usage:-

```
Var str = $("#P1").html();
$("#P1").html("<u> Hello World.</u>");
```

4) attr()

→ By using this function we can get/set any attribute value.
usage:-

```
Var str = $("h1").attr("align");
$("h1").attr("align", "center");
```

Example 1

```html
<!-- Create a webpage to display total amount based on unit price & quantity
     using jquery -->

<html>
  <head>
    <script src='jquery-1.11.3.min.js'></script>
    <script>
      $(document).ready(function()
      {
        $("#b1").click(function()
        {
          var s1 = $("#pname").val();
          $("#sp1").text("Product Name :" +sp1)

          var x = $("#uprice").val();
          var y = $("#qty").val();
          var z = parpeInt(x) * parpeInt(y);
          $("#sp2").html("<u> Result :" + z + "</u>");
        });
      });
    </script>
  </head>
  <body>
    <h1> Product Details </h1>
    <hr></hr>
    Product Name <input type="text" id="pname" /> <br></br>
    UnitPrice <input type="text" id="uprice" /> <br></br>
    Total Quantity <input type="text" id="qty" /> <br></br>
    <button id="b1" > Get Total </button>
    <span id="sp1" ></span><br><br>
    <span id="sp2" ></span>
  </body>
</html>
```

⇒

```
Enter first Number [          ]

Enter Second Number [          ]

  [ + ]   [ - ]   [ * ]   [ / ]
```

```html
<html>
  <head>
    <script src="jquery-1.11.3.min.js"> </script>
    <script>
      $(document).ready(function()
      {
        $(".c1").click(function()
        {
          Var x = $("#t1").Val();
          Var y = $("#t2").Val();
          Var z = 0;  Var op = $(this).Val();
          switch(op)
          {
            case "+":
            z = parpeInt(x) + parpeInt(y);
             break;
            case "-":
            z = parpeInt(x) - parpeInt(y);
             break;
            case "*":
            z = parpeInt(x) * pareInt(y);
             break;
            case "/":
            z = parpInt(x) / parpeInt(y);
          }
          $("#t3").Val(z);
        });
      });
```

```
</script>
<head>
<body>
  <h1> jQuery selectors </h1>
  <hr></hr>
  Enter first Number <input type="text" id="t1" /> <br></br>
  Enter second Number <input type="text" id="t2"/> <br></br>
  Result <input type="text" id="t3"/> <br></br>
              readonly="true"
  <input type="button" class="c1" id="b1" value="+" />
  <input type="button" class="c1" id="b2" value="-" />
  <input type="button" class="c1" id="b3" value="*" />
  <input type="button" class="c1" id="b4" value="/" />
</body>
</html>
```

```
$ ("TagName")
$ ("#Id")
$ (".ClassName")
$ (this)
$ (document)
```

Styles in jQuery:

→ Styles in jQuery+

→ Applying styles to html elements using jquery is one of the frequently required task.

→ jQuery provides special methods to apply styles at runtime

→ These methods supports all versions of css properties.

Methods:

1) css()

css ("property", "value") — Single property.

css ({property : value, property : value}) — Multiple property.

2) addClass ("className")

3) removeClass ("classname")

4) toggleClass ("classname")

1) css() — Single Property

→ By using this method we can get/set the style property values.

→ Get:

Syntax: $(selector). css ("prop");

Eg: var str = $("#P1").css("color");

→ Set:
Syntax: $(selector).css ("prop", "value");

Eg: $("#P1").css("color", "blue");

css() - Multiple properties

→ we can set multiple properties using this method.

Syntax: $ (selector). css ({ "prop":"value", "prop" : "value" }) ;

Ex: $("#div1") . css ({ "color":"blue", "background-color" : "Red" });

Examples

```
<!-- Create a webpage to change the background color based on,
     focus/blur event using jquery -->
<html>
  <head>
    <script src="jquery-1.11.3.min.js"> </script>

    <script>
      $(document).ready( function()
      {
          $(".c1") . focus ( function()
          {
            $(this). css ("background-color", "pink");
          })

          $(".c1"). blur ( function()
          {
            $(this). css ("background-color", "white");
          })
      });
    </script>
  </head>
  <body>
    <h1> Styles in jquery </h1>
    <hr/>
    firstName <input type="text" class="c1"/> <br><br>
    lastName <input type="text" class="c1"/> <br><br>
    <input type="button" value="Registration"/>
  </body>
</html>
```

→ Example :-

```html
<!--- Create a webpage to change the styles based on mouse events
       using jquery --->

<html>
  <head>
    <script src=" jquery-1.11.3.min.js"></script>
    <style>
      .c1
      {
          border : 1px solid Blue;
          background-color : Blue;
          Color : White;
      }
    <script>
      $(document).ready(function()
      {
```
```
          $(this).css()
          Var s1,s2;
          $(".c1").mouseover(function()
          {
             s1 = $(this).css("background-color");
             s2 = $(this).css("Color");
             $(this).css({"background-color": s2, "color": s1});
          });
          $(".c1").mouseout(function()
          {
             $(this).css({"background-color":s1, "color": s2});
          });
      });
    </script>
  </head>
  <body>
    <h1> styles in jquery</h1><hr/>
    <Button class="c1" > Login </Button>
    <Button class="c1"> Register</Button>
  </body>
</html>
```

2) Add Class ();

→ This method is used to apply css class using jquery.
→ It is easy to apply more number of css properties at a time with class.
→ we can reuse exchsting css classes & integrate with jquery programming.
→ To implement, first we need to prepare css class in your webpage (or) *. css file.

Syntax:-
   $(selector).add Class ("classname");
   Ex:- $("#div 1").add Class ("c1");

3) remove Class ():

→ This method will remove css classes from html element.
→ It means it remove all css properties belongs to that class.

Syntax:- $(selector).remove Class ("classname");
   Ex:- $("#div 1").remove Class ("c1");

4) toggle Class ():

→ If you use this method, it will apply if element in not having style class.
→ It will remove if element in having style class.

   Syntax:- $(selector).toggle Class ("classname");

Example:-
<!-- - create a web page to apply styles using css class with jquery --->
<html>
   <head>
      <style>
         .BlueClass
         {
            border: 1px solid Blue;
            color: Blue;
         }
```

```
. GreenClass
{
    font-family : Ariel;
    color : Green;
</style>
<script src= "jQuery-1-11-3-min-js" > </script>
<script>
    $(document)-ready(function()
    {
        $("#b1") . click(function()
        {
            $("h1").addClass("BlueClass");
            $("P1") . add Class("GreenClass");
        });
        $("#b2") . click(function()
        {
            $("h1") . removeClass("BlueClass");
            $("P1") . removeClass("GreenClass");
        });
        $("#b3") . click(function()
        {
            $("h1") . toggleClass("BlueClass");
            $("P1") . toggleClass("GreenClass");
        });
    });
</script>
</head>
<body>
    <h1> Styles in jquery-css classes </h1>
    <hr>
    <h1> Applying styles with css class </h1>
    <P> This is Test Paragraph ----</P>
```

```
<hr></hr>
<h1> Applying styles with css classes </h1>
<P> This is Test Paragraph --- </P>.
<hr></hr>
<Button id="b1"> Add css class </Button>
    <Button id="b2"> Remove css class </Button>
    <Button id="b3"> Toggle css class </Button>
</body>
</html>
```

29/10/15

→ Event Handling in jQuery :-

→ jQuery library provides special methods in order to handle the events

→ It supports all the events of html elements.

→ we can handle the events in two ways.

    1) Dynamic handling events.

    2) Static handling events.

1) Dynamic handling events :-

→ It means at runtime we can handle (or) unhandled the events.

→ It is reffered as bind / unbinding events programmatically.

→ To implement this, jQuery provides the following methods :

    1) bind()

    2) unbind()

   1) bind() : This method is used to add a function to event.

     Syntax :- $(selector).bind("eventname", function);

usage:-

```
$ ("#div1").bind ("click", function()
    {
        //required code.
    });
```

2) unbind():-

→ This method is used to remove the function from event.

→ After unbind, corresponding html element does not respond to user actions.

Syntax: `$ (selector).unbind ("eventname");`

usage:- `$ ("div1").unbind ("click");`

Example :-

```html
<!-- Create a webpage to add/remove the click event to div tag
using dynamic event handling concept of jquery -->
<html>
  <head>
    <script src="jquery.1-11-3.min.js"></script>
    <script>
        function f1()
        {
            $("div1").bind ("click", function()
            {
                alert ("Message from div click event");
            });
            $("div1").css ("cursor","pointer");
            $("sp1").text ("Click event added to div tag");
        });
```

```
function f2( )
{
    $ ("#div1"). unbind ("click");
    $ ("#div1"). css ("cursor", "auto");
    $ ("sp1"). text ("click event is removed from div tag");

}
</script>
</head>
<body>
    <h1> jQuery Event Handling - Demo </h1>
    </hr>
    < Button onclick = "f1()" > Add Event </Button>
    <Button onclick = "f2()"> Remove Event </Button>.
    <br><lbr>.
    <div id="div1" style = "border: 2px solid blue; padding :5px;
                    background - color : pink;">
        This is Test Paragraph. <br><lbr> This is Test paragraph.

    </div>.
    <span id = "sp1" > </span>
</body>
</html>
```

2) Static Event Handling :-

→ It is a standard approach to process events.

→ In this approach, event binding is fixed before execution.

→ In order to implement, jQuery provides event handler function like:
    click (), mouseover(), mouseout(), ready(), focus(), blur() etc.

→ Most of the examples we are using this approach.

Syntax:-
    $ (selector) . eventname (function ()
        {
        ≡
        });

Example:-
<!-- Create a webpage to change the images dynamically based on mouse events of jquery -->

```html
<html>
  <head>
    <script src="jquery-1-11-3-min-js"> </script>
    <style>
      .c1
      {
        margin : 2px;
        border : 2px solid gray;
        padding : 8px;
      }
    <script>
      $ (document).ready (function ()
      {
        $ (".c1").mouseover (function ()
        {
          var s1= $ (this)-attr ("src");
          $ ("#main") . attr ("src", s1);
          $ (this).css ("border-color", "Gold");
        });
        $ (".c1").mouseout (function ()
        {
          $ (this).css ("border-color", "Gray");
        });
      });
```

```html
<Iscript>
</head>
<body>
<h1> JQuery Events-Demo2 </h1>
<lhr>
<img  id="main"  src="Images/Image1.jpg" border="2" height="350px"
      width="380px"/> <br></br>

<img  class="c1"  src="Images/Image1.jpg" border="2" height="80px"
      width="80px"/>

<img   class="c1"  src="Images/Image2.jpg" border="2" height="80px"
      width="80px"/>

<img  class="c1"  src="Images/Image3.jpg" border="2" height="80px"
      width="80px"/>

<img  class="c1"  src="Images/Image4.jpg" border="2" height="80px"
      width="80px"/>
</body>
</html>
```

⇒ opacity : 1 → default. (full image)

0 → (No image).

Task:
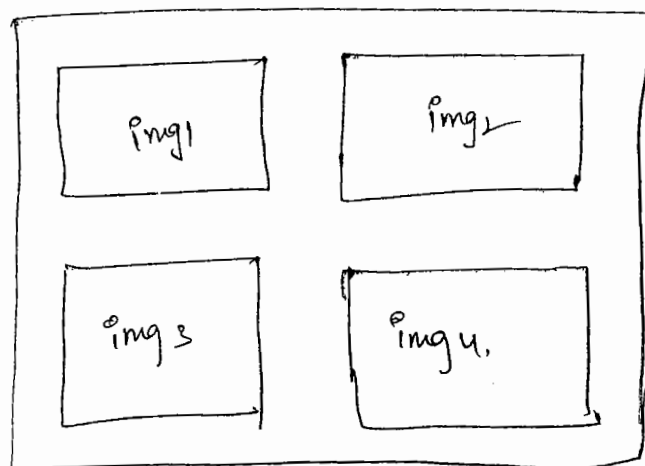
→ Changing the opacity on mouse events using jquery

⇒ DOM Traversing :-

→ jQuery selectors are used to get the elements from the document.

→ Most of the selectors will return all elements that are matching given selector expression.

→ so that these selectors will return collection of objects (html elements).

→ If you want to select specific item from the collection, jQuery provides special functions (or) selector options.

→ By using these options we can easily traverse the items from the collection.

1) Traversing all items :-

→ we can traverse all items from the selector in 2-ways.

1) using each() function
   - item based processing.

2) using eq() function
   - index based processing.

Syntax :-

each: $ (Selector). each (function()
               {
                 $ (this) - - .
                        - - -
               });

eq: for (i=0; i<n; i++)
               {
                 $ (selector)·eq(i);
               }

→ where 'n' is length of the collection.

Ques: Difference b/w "$(this)" and "this" in jquery?

→ Example:-

```html
<!-- Create a webpage to access list items from the UL by using
     eq() & each() -->

<html>
 <head>
  <script src="jquery-1.11.3.min.js"> </script>
  <script>
   $(document).ready(function()
    {
       var str;
       $("#list1").click(function()
        {
          var n = $("#list1 li").length;
           str = "using Eq() : ";
          for(i=0; i<n; i++)
           {
             str = str + $("#list1 li").eq(i).text()+", ";
           }
          $("#sp1").text(str);
        });

       $("#b2").click(function()
        {
           str = "using Each() : ";
          $("#list1 li").each(function()
           {
              str = str + $(this).text()+", ";
           });
           $("#sp1").text(str);
        });
    });
  </script>
 </head>
```

```html
<body>
<h1> jQuery Dom Traversing - Demo 1 </h1>
<hr>
   <ul id="List1">
      <li> jQuery </li>
      <li> MVC </li>
      <li> Angular Js </li>
      <li> HTML </li>
      <li> CSS </li>
   </ul>  <br><br>
<Button id="b1"> Value Get eqsing eq() </Button>
<Button id="b2"> Get using each() </Button>
<hr></hr>
<span id="sp1"> </span>
</body>
</html>
```

usingeq : jquery, mvc, An

→ Example :-

```html
<!-- Create a webpage to create multicolor list using jquery -->
<html>
  <head>
     <script src="jquery-1.11.3.min.js"> </script>
     <style>
       #list1
       {
          border : 2px solid Blue;
          width : 150px;
          height : 400px;
          list-style : none;
          padding : 3px;
       }
       # list li
       {
          border : 2px solid Black;
          margin : 8px;
```

```
        height : 20px
        padding : 3px;
    }
</style>
<script>
$(document).ready(function()
    {
    var colors = ["Pink", "Yellow", "LightBlue", "LightGreen", "LightGray"];
    var n = $("#list1 li").length;
    j=0;
    for(i=0; i<n; i++)
        {
        $("#list1 li").eq(i).css("background-color", colors[j]);

        j++;
        if( j == colors.length)
            {
            j=0;
            }
        }
    });
</script>
</head>
<body>
    <h1>jQuery Dom Traversing - Demo 2</h1>
    <hr><hr>
    <ul id="list1">
        <li> jQuery </li>
        <li> mvc </li>
        <li> Angular Js </li>
        <li> HTML </li>
        <li> css </li>
    </ul>
</body>
</html>
```

Task:

```
<!-- Create a webpage to apply multiple colors for each row in the given
table -->
```

Product Name	unit Price	unit In stock
→ Printer	2500.00	20
→ 4GB-RAM	1200.00	23
→ 1TB Samsung HDD	4500.00	0
wifi Router	1800.00	0
Printer.	2500-00	5

+(- .

if (n%.2 = 0)

{

—

}

2) Traversing alternate items from the Collection:

→ If you want to select alternate items from the Collections, jquery provides special options in selector Expression.

→ It works based on index.

→ Even number based index.
   odd number based index.

usage :-

```
$ ("# list1  li : even")
$ ("#list1  li : odd")
```

Above Example :

```
$(document).ready (function c)
{
    $("#list1 li:even").css ("background-color', "Blue');
    $("#list1 li:odd").css (" background-Color", "Green");
}
```

31/10/15

⇒ Traversing filtered Method:-

→ jQuery selector will return all ~~methods~~ matched elements.

→ filter() function will filter out elements that are match given selector

↪ Syntax:- $(selector).filter(selector)

Ex:- $("#List1 li").filter(".class1");
$("#tabel1 tr").filter(".class2");

Traversing not filterd elements:-

→ filter() function will filter all matched elements from the given collection.

→ not() function in simply negation of filter().

→ not() will return items that are not matching given selector expression.

Syntax:- $(selector).not(selector);

usage:- $("#tabel1 'tr").not(".header");
$("#tabel1 tr").not(".outofStock clan");

Example:-

<!-- creat a webpage to apply styles to table rows by using jQuery filter options ---->

```
<html>
  <head>
    <script src="jquery-1.11.3.min.js"> <Iscript>
    <script>
      $(document).ready (function()
      {
        $("#tabel1 tr").filter(".outofStock").css("background-color",
                    "Bule');
```

```
$("#tabel1 tr").not(".outof stock").css("background-color", "Blue");

$("#tabel1 tr").first().css("background-color", "Green");

$("#tabel1 tr").css("color", "white");

});
</script>
<head>
<body>
  <h1> jquery Dom Traversing </h2>
  <hr>
    <table align="center" id="tabel1" border="2" width="400px">

    <tr>
        <th> ProductName </th>
        <th> Unit Price </th>
        <th> units in stock </th>
    </tr>

    <tr>
        <td> Printer </td>
        <td> 2500.00 </td>
        <td>  20 </td>
    </tr>
    <tr>
        <td>  4GB-RAm </td>
        <td> 1200.00 </td>
        <td>  23 </td>
    </tr>
    <tr class="outof stock">
        <td>1TB Samsund HDD </td>
        <td>   4500.00 </td>
        <td>      0      </td>
    </tr>
    <tr>
        <td>  32-GB kingston USB </td>
        <td> 800 </td>
        <td>  35 </td>
    </tr>
    <table>
<body>
<html>
```

Note :-

~~first~~ jquery library provides the following methods to get
first & last ~~item~~ from the selected collection.
            elements

1) $("selector") · first()   } Syntax.
2) $("selector") · Last()   }

✳ DOM Maniputations

| Dom Manipulations |

→ After accessing DOM elements, we need modify then at runtime
   to perform clientside activities.

→ These DOM manipulatioms are categorized into two types.

    1) content Based
    2) structure Based.

1) content Based manipulatioms :-

   → The following methods of jquery can be used to modify the
      content of the HTML elements.

       1) text()
       2) html()
       3) Val()
       4) attr()
       5) prop()

2) structure Based manipulatioms :

       1) append()
       2) after()
       3) before()
       4) empty()
       5) remove()

```
        ┌─────────────────────┐
        │  Dom manipulation   │
        └─────────────────────┘
           ╱              ╲
┌──────────────┐    ┌──────────────────┐
│ Content Based│    │ Structure Based  │
└──────────────┘    └──────────────────┘
        │                    │
┌──────────────┐    ┌──────────────────┐
│ 1) Val()     │    │ 1) append()      │
│ 2) text()    │    │ 2) after()       │
│ 3) html()    │    │ 3) before()      │
│ 4) attr()    │    │ 4) remove()      │
│ 5) prop()    │    │ 5) empty()       │
└──────────────┘    └──────────────────┘
```

Structure of Dom manipulation

Example:-

```
<!-- create a webpage to demonstrate usage of Dom Manipulations -->

<html>
  <head>
    <script src="jquery-1.11.3-min.js"></script>
    <script>
        var str = "<tr bgcolor='yellow'>";
            str = str + "<td> wifi Router </td>";
            str = str + "<td> 1500.00 </td>";
            str = str + "<td>5</td>";
            str = str + "</tr>";
        function fn-append()
        {
            $('#tabel1').append(str);
        }
        function fn-after()
        {
            $("#tabel1 tr").first().after(str);
        }
        function fn-before()
        {
            $("#tabel1 tr").last().before(str);
        }
```

```
function fn-empty()
{
    $("#table1").empty();
}
function fn-remove()
{
    $("#table1").remove();
}
</script>
<head>
<body>
    <h1> Jquery DOM Manipulations </h1>
    <hr>
    <button onclick="fn-append()"> Append </button>
    <button onclick="fn-after()"> After </button>
    <button onclick="fn-before()"> Before </Button>
    <button onclick="fn-empty()"> Empty </button>
    <button onclick="fn-remove()"> Remove </button>
    <br><br>
        <table id="table1" border="5" width="400px">
        <tr>
            <th> Product Name </th>
            <th> Product Cost </th>
            <th> Quantity </th>
        </tr>
        <tr>
            <td> Printer </td>
            <td> 2500.00 </td>
            <td> 20 </td>
        </tr>
        <tr>
            <td> 4-GBRAM </td>
            <td> 1200.00 </td>
            <td> 23 </td>
        </tr>
        </table>
        </body>
    </html>
```
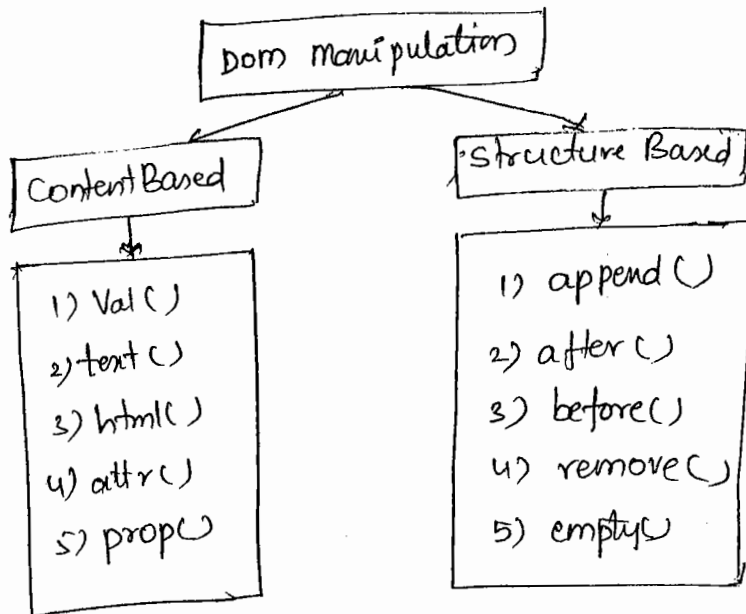
Note:-

→ Empty the table except header row.

$("#table1 tr").not(".header").empty();

2/11/15

⇒ Jquery Effects & Animations

→ In order to make the html elements show/hide at runtime, jquery provides built-in methods.
→ These methods will come with predefined effects.
→ we can perform same task without these methods, but we did not get the any visual effect.
→ But jquery effects methods provides visual effects, so, that it will add more richness to your Application.

Methods.

1) show()
2) Hide()
3) Toggle()
4) fadeIn()
5) fadeout()
6) fadeToggle()
7) slideDown()
8) slideUp()
9) slideToggle()
10) Aanimate()

usage:-

Syntax: $(selector).effectMethod(speed);

Ex: $("#img1").show(2000);

```
$ ("#img1") . show (fast);
```

Note :-

→ Duration can be specified with a number that represents in milliseconds.

→ It also accepts string that take only following options.
  → normal , slow, speed.

Example :-

```
<!-- create a webpage to apply effects on Image using jquery
    function &effects methods -->

<html>
  <head>
    <script src="jquery.1-11.3.min.js"> </script>
    <script>
      $(document) . ready (function()
      {
          $ ("#b1") . click (function()
          {
              $ ("#img1") .show (2000);
          });
          $ ("#b2") . click (function()
          {
              $ ("#img1") . hide (2000);
          });
          $ ("#b3") . click (function()
          {
              $ ("#img1") . toggle (2000);
          });
      });
    </script>
  </head>
```

```html
<body>
  <h1> jQuery styles </h1>
  <hr1>

  <button id="b1"> Show </button>
  <button id="b2"> Hide </button>
  <button id="b3"> Toggle </button>
  <br></br>
  <img id="img1" src="Images/img1.jpg" height="300" width="550"/>

</body>
</html>
```

Example:-

```html
<!-- Create a webpage to apply fade effects on images using
     jquery effects -->

<html>
  <head>
    <script src="jquery-1-11-3-min.js"> </script>
    <style>
      .c1
      {
        height = 150px;
        width = 230px;
        border = 2px solid DarkBlue;
        border-radius : 50px;
        padding : 4px;
        margin : 5px;
      }
    <script>
      $(document).ready(function()
      {
          $(".c1").mouseover(function()
          {
              $(this).fadeIn(1000);
          }); }); $(this).fadeOut(1000);
```

```
</script>
<Ihead>
 <body>        Effects &
   <h1> jQuery styles </h1>
   <hr1>

   <img ⊠ 🖾 class="c1" src="Images/img1.jpg" 1>
   <img   class="c1"   src="Images/img1.jpg" 1>
   <img   class="c1"   src="Images/img1.jpg" 1>

 <Ibody>
<Ihtml>

Example :-

<!-- create a webpage to develop a Image Gallery with tiles and
   apply effects using jquery Effects -->

<html>
  <head>
  <script src="jquery.1-11.3.min.js"></script>
  <style>
    td
     {
      height :150px;
      width : 200px;
      text-align: center;
      background -color: Light Yellow;
     }
    img
     {
      height : 130 px;
      width : 180 px;
      border : 1px solid DarkBlue;
      padding: 2px;
      border-radius: 10px;
     }
  <Istyle>
```

```html
<script>
  $(document).ready (function()
    {
      $(".c1").hide();
      setInterval (f1, 1000);
    });

    function f1()
    {
      $("#.c2").toggle (2000);
      $(".c1").toggle (2000);
    }
</script>
</head>
<body>
  <h1> jquery effects </h1>
  <hr/>
  <table id="table1" border="2">
    <tr>
      <td> <img class="c1" src="Images/img1.jpg'/> </td>
      <td><img class="c2" src="Images/img2.jpg'/> </td>
      <td><img class="c1" src=Images/img3.jpg'/> </td>
      <td> <img class="c2" src=Images/img4.jpg"/> </td>
    </tr>
    <tr>
      <td> <img class="c2' src=Images/img5.jpg'/> </td>
      <td> <img class="c1" src=Images/img6.jpg'/> </td>
      <td> <img class="c2" src="Images/img7.jpg'/> </td>
      <td> <img class="c1" src="Images/img8.jpg'/> </td>
    </tr>
  </table>
</body>
</html>
```

3/11/15

→ <!-- Create a webpage to apply slide effects to display product information -->

```html
<html>
<head>
<script src="jquery.1-113min.js"></script>
<style>
#div1
{
  border : 4px solid Green;
  width : 200px;
  padding : 5px;
  text-align : center;
}
#P1
{
  border : 4px solid Green;
  width : 200px;
  padding : 5px;
  text-align : center;
  background-color : LightYellow;
  cursor : Pointer; margin : 0px;
}
</style>
<script>
  $(document).ready(function()
  {
      $("#div1").hide();
      $("#P1").click(function()
      {
       $("#div1").slideToggle(2000);
      });
  });
</script>
</head>
```

```html
<body>
<h1> jquery Effects - Sliding Effect </h1>
<hr>
<p id="p1"> show/Hide Discription </p>
<div id="div1">
    <h3> Product Title </h3>
    <hr>
    <p> This Description About this product </p> <br>
    <img src="Images/img1-jpg" height="100px" width="90px"
            border="1" />
</div>
</body>
</html>
```

➡ Animate() :—

→ Animate method will apply the animation based on the given ~~class~~ css
   properties.

→ It will apply the styles in given duration.

→ This method will accept 2 arguments.
   1) css properties
   2) duration.

→ we can provide multiple properties.

Syntax:—

$(selector).animate({ "prop":"value", "prop":"value", __ }, duration);

Example:—

```html
<!-- create a webpage to apply animation on image to change the
    position using jquery Animation -->
<html>
  <head>
    <script src="jquery-1-11-3.min.js">
    </script>
```

```html
<style>
  #img1
  {
    width : 80px;
    height : 80px;
    position : relative;
    left : 5px;
    top : 5px;
  }
  #div1
  {
    border : 5px double Green;
    width : 600px;
    height : 300px;
    padding : 5px;
  }
</style>
<script>
  $(document).ready(function()
  {
    $("#b1").click(function()
    {
      $("#img1").animate({"left": "520px", "top": "5px"},1000);
      $("#img1").animate({"left": "520px", "top": "220px"},1000);
      $("#img1").animate({"left": "5px", "top": "220px"},1000);
      $("#img1").animate({"left": "5px", "top": "5px"},1000);
    });
  });
</script>
<head>
  <h1> jQuery Animation </h1>
  <br>
  <button id="b1"> Start Animation </button>
  <br><br>
```

```
<div id="div1">
   <img id="img1" src="ball.jpg" />
<ldiv>
<lbody>
<lhtml>
```

→ jqueryui.com → custom downloade → stable.

1) Images
2) jquery-1.11.3.min.js
3) jquery-ui.css
3) jquery-ui.js

4/4/15

⟶ jQuery UI:-

→ jquery UI is a special library developed by jquery team.

→ It provides several function which makes your webpage is more richness related features to UI.

→ It works based on core jquery library.

→ To implement jquery-ui, you need to include following files:

1. jquery.1-11.3.min.js
2. jquery.ui.js
3. jquery.ui.css
4. Required images.

→ The Above files can be download from www.jqueryui.com.

→ The downloaded *.zip file contains all the required files.

Primary features of jquery-UI :-

→ All the features are devided into 4 categories.

    1) Interactions.
    2) Effects
    3) widgets
    4) Themes.

⇒ Jquery UI Interactions :-

1) Draggable :-

→ jquery Draggable function of jquery UI allows the user can drag and drop an html element

→ By default end user cannot move the items from one place to another pace.

→ This feature allows to position the items with in the page.

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

    Ex :- $("#img1").draggable();

2) Resizable :-

→ Resizable function of jquery UI allows the user can resize the html elements.

→ usually end-user cannot resize the item on the webpage.

→ This feature you can apply for any html elements like img, div, p etc;

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

                $("#img1").resizable();

Example

```
<!-- Create the webpage to allow the user to drag & resize the elements
     using jquery UI ---->

<html>
  <head>
    <script src="jQuery.1-11.3.min.js"></script>
    <script src="jQuery-ui.js"></script>
    <link href="jQuery-ui.css" rel="stylesheet"/>
    <style>
      #div1
      {
        border: 5px solid Red;
      }

      .c1
      {
        border: 2px solid Blue;
        height: 150px;
        width: 200px;
        margin: 5px;
      }
    </style>
    <script>
      $(document).ready(function()
      {
        $(".c1").draggable();
        $("#div1").resizable();
        $("#img1").resizable();
      });
    </script>
  </head>
  <body>
    <h1> jQuery-UI Interactions </h1>
    <hr/>
```

```html
<img class="c1" src="img1.jpg" />
<img class="c1" src="img2.jpg" />
<img class="c1" src="img3.jpg" /> <br><1br>
<div id="div1"> Sample content in div tag that allows resizable.
   by user...
</div> <br><1br>
<img border="2" height="200px" width="300px" src="img4.jpg"
      id="img1" />
</body>
</html>
```

folder structure

E:\ jquery-UI

    ├─ [images] ←   [jquery-UI built in Images]
    ├─ jquery.1-11.3.min.js
    ├─ jquery.UI.js
    ├─ jquery.UI.css
    ├─ Demo.html
    ├─ Image1.jpg
    ├─ Image2.jpg

3) Sortable :—

→ Sortable function allows the user to adjust the items within the
   lists.

→ In this option, items will adjust the positions automatically.

→ Generally this feature we are applying for lists related tags /
   Collection items.

      Syntax :— $(selector).sortable();

      Ex :— $("#list1").sortable();

Example

```html
<!-- create a html page to create number puzzle using jquery-UI
     Sortable() ---->

<html>
  <head>
    <script src="jQuery-1-11-3.min.js"></script>
    <script src="jquery-ui.js"></script>
    <link href="jquery-ui.css" rel="stylesheet"/>
    <script>
      $(document).ready(function()
      {
          $("#list1").Sortable();
      });
    </script>
    <style>
      ul
      {
          border: 2px solid Red;
          width: 90px;
          height: 80px;
          list-style: none;
          padding: 2px;
      }
      ul li
      {
          background-color: Blue;
          color: white;
          float: left;
          padding: 1px;
          margin: 4px;
          border-radius: 50px;
          cursor: pointer;
          width: 15px;
          height: 15px;
          text-align: center;
      }
    </style>
  </head>
```

```
<body>
    <h1> Number puzzle </h1>
    <hr/>
    <ul id=" List1" >
        <li> 5 </li>
        <li> 8 </li>
        <li> 4 </li>
        <li> 1 </li>
        <li> 9 </li>
        <li> 7 </li>
        <li> 3 </li>
        <li> 2 </li>
        <li> 6 </li>
    </ul>
</body>
</html>
```

5/11/15

4) Selectable() :—

→ By default List tags does not allow to select items in the list.

→ selectable() function of jquery will allows the user to select any item from list.

Note :-

→ To implement this ~~function~~ feature you need to provide style for selected item.

→ It will be based on css class : "ui-selected".

→ It also provides a event called "selected" to add any logic at the time of selection.

Syntax:— $(selector).selectable();

Ex:- $("#List1").selectable();

Example:

```html
<!-- create a webpage to allow the user to select course name from
     unordered list using jquery-ui interactions -->
<html>
  <head>
    <script src="jquery-1-11.min.js"></script>
    <script src="jquery.ui.js"></script>
    <link href="jquery-ui.css" rel="stylesheet">
    <style>
    ul ui-selected
    {
        background-color: Blue;
        color: White;
    }
    #List1
    {
      border: 2px solid Blue;
      width: 150px;
      height: 200px;
      list-style: none;
      padding: 3px;
    }
    ul li
    {
      border: 2px solid Black;
      margin: 8px;
      height: 20px;
      padding: 3px;
      background-color: pink;
      cursor: pointer;
    }
    </style>
```

```html
<script>
    $(document).ready(function()
    {
        $("#List1").selectable(selected : function()
        {
            Var str = $(".ui-selected").text();
            $("#sp1").html("selected course :" + str);
        }
    });
</script>
</head>
<body>
    <h1> jQuery UI - Interactions </h1>
    <hr>
        <span id="sp1"> </span> <br/>
        <ul id="List1">
            <li> jQuery </li>
            <li> AngularJs </li>
            <li> HTML </li>
            <li> css </li>
            <li> Javascript </li>
        </ul>
</body>
</html>
```

⟹ **JQuery UI- Widgets:**

→ jquery-ui Library provides another module called "widgets".

→ This module having several functions that you can apply div, list, buttons, inputs etc.,

→ By using this module we can apply additional UI related features like tabs, menus, collapsible panels etc.,

→ widgets also provides new controls like calender for date selection.

Note:-

→ All jquery-ui widgets are themable i.e., If you change the ui theme all widgets will get affected with that theme.

1) button()

→ It is provide jquery-ui button styles.

→ It will change the styles in mouseover & mouseout events.

Syntax: $(selector). button();

Ex: $('#b1'). button();

2) Accordion Panels

→ Accordion Panels are also called collapsable panels.

→ It will looks like horizontal tabs.

→ In the same place we can organize multiple contents based on the selected section.

→ If you select one section, remaining sections will be minimized (collapse).

Implementation
```
<div id="main">
    <h3> heading </h3>
    <div> Content --- </div>
    - - -
</div>
```

→ Apply Acuordion function to "main" division.

```
$("#main").accordion();
```

Example:—

```html
<!-- create a webpage to implement jquery-ui widgets button & acuordion
     panel -->

<html>
 <head>
   <sccript
   <script
   <link
   <style>
     .c1
     {
        height : 50px;
     }
     #main
     {
        width : 500px;
     }
   </style>
   <script>
     $(document).ready(function()
     {
         $("input:button").button();
         $("#main").accordion();
     }
   </script>
 </head>
 <body>
  <h1> jQuery-UI widgets </h1>
   <hr>
```

```html
<input type="button" value="Login"/>|
<input type="button" value="Register"/>|
<input type="button" value="update"/> <br></br>
<div id="main">
    <h3> jQuery </h3>
    <div class="c1"> This panel contains content related to jquery.. </div>
    <h3> JSP </h3>
    <div class="c1"> This panel contains content related to JSP - </div>
    <h3> HTML </h3>
    <div class="c1"> This panel contains content related to HTML - </div>
</div>
</body>
</html>
```

Note:-

→ To set default panel :

$("#main").accordion ({activate : 2});

By default activate panel is set to 0.

→ To change the default event :

$("#main").accordion({event : "mouseover"});

→ By default it is set to "click" event.

6/11/15

3) Toot Tip :

→ In Html we can apply Tooltip by using "Title" Attribute.

※ Syntax:-   <tag title="tooltip text" --> <input type="button" title="userlogin"/>

→ By using jquery ui we can provide more richness to these tooltips.

→ This you can apply at document level, so that all Html elements
   this option will be applicable.

usage: $(document).tooltip();

$("input.button").tooltip();

4) Tabs:-

→ Tabs() function of jquery allows us to organize the content into multiple tabs.

→ So that in the same area we can display different contents based on the selected tab.

usage: $(selector).tabs();

$("#maindiv").tabs();

Example:-

```html
<! create a webpage to implement tooltip and tabs of jquery-ui widgets -->

<html>
  <head>
    <script src="jquery.1-11.3.min.js"></script>
    <script src="jquery-ui.js"></script>
    <link href="jquery-ui.css" rel="stylesheets" />
    <style>
    #main
    {
       width: 500px;
       height: 300px;
    }
    img
    {
       width: 200px;
       height: 120px;
       border: 2px solid Red;
    }
    </style>
```

```html
<script>
    $(document).ready(function()
    {
        $("input:button").tooltip();
        $("#main").tabs({active:1, event:mouseover});
    });
</script>
</head>
<body>
    <h1> jquery-u2 widgets </h1> <hr>
    <input type="button" title="userLogin" value="login"/>
    <input type="button" title="user Registration" Value="user Registration"/>
    <input type="button" title="update userdetails" Value="update"/>
    <br><br>
    <div id="main">
        <ul>
            <li> <a href="#div1">jquery</a> </li>
            <li> <a href="#div2"> Jsp </a> </li>
            <li> <a href="#div3"> ASP.NET </a> </li>
        </ul>

        <div id="div1">
            This panel contains content related to jquery.
            <img src="jquery.jpg"/>
        </div>
        <div id="div2">
            This panel contains content related to Jsp.
            <img src="Jsp.jpg"/>
        </div>
        <div id="div3">
            This panel contains content related to ASP.NET.
            <img src="asp.net.jpg"/>
        </div>
    </div>
</body>
</html>
```

5) dialog() :-

→ jQuery UI "dialog()" method used to display the content as dialog box.

→ we can apply this method to any tag like div, tables, etc;

→ In any jQuery events we can display the content as dialog box.

   Ex:- button click, hyperlink. click, etc...

   usage:- $(selector). dialog( );
              $("#div1"). dialog();

Note :-

1) useful options :-

   auto Open : false,
   height : "430"
   width : "600"
   modal : true,
   resizable : false,
   draggable : false.

2) Show dialog / close dialog.

   $(selector). dialog ("open");
   $(selector). dialog ("close");

Example :-

```
<!-- create a webpage to implement image gallery using dialog() method
     of jquery- ui widgets -->

<html>
 <head>
   <script src="jQuery-1.11.3. min.js"> </script>
   <script src="jquery-ui.js"> </script>
   <link href="jquery-ui.css" rel="stylesheet"/>
 </head>
```

```
<style>
  .c1
  {
      height : 150px;
      width : 200px;
      border : 2px solid Blue;
      padding : 3px;
      margin : 3px;
  }
  #main
  {
      height : 250px;
      width : 300px;
      margin : 5px;
  }
</style>
<script>
  $(document).ready(function()
  {
      $("#main").hide();
      $("#main").dialog(
      {
          autoOpen : false,
          height : "430",
          width : "600",
          model : true,
          resizable : false,
          draggable : false
      });

      $(".c1").click(function()
      {
          var str= $(this).attr("src");
          $("#main").attr("src", str);
          $("#main").dialog("open");
      });
  });
</script>
</head>
```

show: { effect : "explode" (fading / blind), duration : 2000 }.

hide: { effect : "explode" (fading / blind), duration : 2000 }

```html
<body>
    <h1> jquery-ui widgets </h1> <hr1>
    <img class="c1" src="img1.jpg"/>
    <img class="c1" src="img2.jpg"/>
    <img class="c1" src="img3.jpg"/>
    <img class="c1" src="img4.jpg"/>
    <img class="c1" src="img5.jpg"/> <br><br>
    <img id="main" src="img1.jpg" />
</body>
</html>
```

Note :-

3) Dialog with effects :

show : { effect : "explode", duration : 1000}

hide : { effect : "explode", duration : 1000 } .

4) Custom Buttons :-

custom Buttons to dialog Box :

```
buttons : [ { text : "cancel",
            click : function()
                    {
                        Code()
                    }
           },
           { text : "OK",
             click : function()
                     {
                         Code()
                     }
           }];
```

```
$("# main") . dialog (
{ buttons : [ { text : "Get Image Name",
            click : function()
                {
                var str = $("# main") . attr ("src');
                alert ("current Image Name :" + str);
                }
            },
            { text = " cancel"
              click = function()
                {
                $("# main") dialog (" close");
                }
            }]; 
});
```

7/11/15

4) Date picker :—

→ Date picker function will provide moth, calender control so, that user can ~~early~~ early select the date.

→ Selected date will display in the specified textbox.

→ In this control we can early navigate b/w months, dates, gears.

usage :—
```
$(selector) . datepicker ();
$("ti") . datepicker();
```

Note :—

→ use the following options to customise datepicker.

1) change year : true

2) change month : true

3) min Date : 0
                " -10 D"

4) Max Date : 0
                " 10 D"
                "3 m"

5) Auto complete :-

→ Auto complete function of jquery will show list of options suitable to the data that you type in the textBox.

→ It search the options based on the characters enter by user.

→ It will display all possible options from the given source (array)

→ Source array need to prepare as string array contains all possible options.

```
Usage:-   $(selector).autocomplete(
              {
                 Source: arrayname
              });
```

```html
<!-- create html webpage to demonstrate usage of auto complete.
and Datepicker methods of j-Query-UI -->
<html>
  <head>
    <script
    <script
    <link

    <script>
    $(document).ready(function()
    {
      var courses= ["php", "ASP.NET", "ASP.NET mvc", "mvc", "c#"];
      $("#t1").autocomplete(
      {
        Source: Cources
      });
      $("#t1").Datepicker(
      {
        minDate: "-10D";
        maxDate: "10D";
      });
    </script>
```

```html
<body>
    <h1> jQuery  AutoComplete & Date Picker </h1> <br>
    Course Name :< input type = "text" id=" t1" /> <br>
    Course Start Date ; <input type = "text" id="t2" />
</body>
</html>
```

6) Menu :—

→ jQuery UI provides another useful function that makes menu preperations. is more comfortable.

→ In order to implement menus in jquery we need to prepare all the required options by using <ul> tag.

→ Apply " menu()" on <ul> tag.

    Syntax :— $ (selector) . menu ();

      Ex :- $ ('#menu list') . menu();

```html
<!-- jQuery -UI Menu -->

<html>
  <head>
    <style>
      ul
      {
        width :150px;
      }
    </style>
    <script>
      $(document). ready (function()
      {  $("# list1"). menu ();
      });
    </script>
  </head>
```

```
<body>
    <h1>jQuery-UI Menu </h1> </hr>
    <ul> <li> li st </li>
            <li> <ul>
    <ul id = "list1">
        <li> <a href=" jQuery-html'> jquery <1a> </li>
        <li> Java
            <ul>
                <li> Core java </li>
                <li> Adv·java </li>
            </ul>
        </li>
        <li> PHP </li>
        <li> Net
            <ul>
                <li> ASP.NEt
                    <ul>
                        <li> ASP.NET mvc </li>
                    </ul>
                </li>
            </ul>
        </li>
    </ul>
</body>
</html>
```

⇒ jQuery-UI Effects :-

→ jQuery-UI library provides new set of methods related to applying effects to html effects.

→ Most of the methods are override the existing method of core jquery.

→ These methods will provide additional behaviour to our html elements.

→ It can add more richness to your webpages by providing more visual effects.

usage Effects() function

→ jquery-ui library provides a function called "effect()" which is used to apply different types of visual effects.

→ we can apply this function on any html elements.

→ It will apply at the time of show/Hide an html elements.

Syntax:
```
$ (selector). effect (
        "effect Name",
         Option,
         duration,
         call back- function);
```

Ex:
```
$("#img1"). effect ("slide", {direction:"Left"}, 2000, function ()
    {
        // Code that will execute effect completion of applying effect.
    });
```

Details about effect() parameters :—

1) effect Name:

→ It is a string parameter specifies type of effect that you want to apply.

- blind
- clip
- explode
- fold
- bounce
- drop
- fade
- highlight
- puff
- ~~pulsate~~

- shake
- scale
- pulsate
- slide
- size

2) options:-

→ we can specify additional options if required

Ex:- direction, mode --

3) duration:-

— Possible Values.

Normal
slow
fast
Number (Milliseconds)

4) Callback function

→ If you want to execute any code after completion of the effect, then we can use this function.

9/11/15

```html
<!-- Create HTML page to demonstrate usage of jquery ui effects -->
<html>
 <head>
  <script src="jquery-1.11.3.min.js"></script>
  <script src="jquery-ui.js"></script>
  <link href="jquery-ui.css" rel="stylesheet" />
  <script>
   $(document).ready(function()
   {
     $("#list1").change(function()
     {
       var str = $("#list1").val();
       var obj = {};
       if(str == "size")
       {
          obj = {to: {height:400, width: 700}};
       }
```

```
if(str == "scale")
{
    obj = { percent : "150"};
}
if( str == "highlight")
{
    $("#head1").effect("highlight", obj, 3000);
    return;
}
$("#img1").effect(str, obj, 2000,function()
{
    $(this).show();
});
});
});
</script>
<head>
<body>
<h1 id="head1"> jQuery-UI effects </h1>
<hr>
  <select id="List1">
    <option> blind </option>
    <option> clip </option>
    <option>bounce </option>
    <option> drop </option>
    <option> explode </option>
    <option> fade </option>
    <option> fold </option>
    <option> highlight </option>
    <option> scale </option>
    <option> size </option>
  </select>  <br><br>

  <img id="img1" src="img1.jpg" border="2" height="200px" width="350px"/>
<body>
</head>
</html>
```

```html
<!-- Create a webpage of html to make image slide show using jquery
     UI effects -->

<html>
<head>
  <script src="jquery-1-11.min.js"></script>
  <script src="jquery-ui.js"></script>
  <link href="jquery-ui.css" rel="stylesheet"/>
  <script>
    Var n=1;
    $(document).ready(function()
    {
      $("#img1").hide();
      fn_show();
    });

    function fn_show()
    {
      $("#img1").effect("slide",{direction:"right"},2000,function()
      {
        setTimeout("fn_hide()",2000);
      });
    }

    function fn_hide()
    {
      $("#img1").effect("slide",{direction:"left"},mode:"hide"},
                2000,function()
      {
        n=n+1;
        if(n==8)
        {
          n=1;
        }
```

```
    $("#img1").attr("src", "Images2/Image"+n+".png");

    fn_show();

  });

}
</script>
</head>
<body background="Bg-Road.jpg">
  <br><br>
  <img id="img1" src="Images2/Image1.png" height="250px" width="550px"/>
</body>
</html>
```

Jquery-UI Themes

→ jquery-ui provides some different types of themes to make jquery-ui items more colorful.

→ These themes we can download from jquery-ui theme roller section.

> http://jqueryui.com/themeroller

→ Download the theme files and placed required files in your project folder.

→ Create seperate folder for each theme and place following files.

1) jquery-ui.css
2) [Images] folder.

Example:-

     Creat html page to change the theme dynamically by using jquery-ui themes.

```
    — Theme1
        — [Images]
        — jquery.ui.css

    — Theme 2
        — [Images]
        — jquery.ui.css

— jquery-1.11.3.min.js
— jquery.ui.js
— ThemeDemo.html.
```

10|11|15

```html
<!-- Create a webpage to apply jquery-UI themes at runtime -->

<html>
  <head>
    <script src="jquery-1.11.3.min.js"></script>
    <script src="jquery-ui.js"><\script>
    <link href="Theme1/jquery-ui.css" rel="stylesheet"/>
    <style>
      .c1
      {
        height: 50px;
      }
      #main
      {
        width: 500px;
      }
    </style>
    <script>
      $(document).ready(function()
      {
        $("#List1").change(function()
        {
          var str = $(this).val() + "/jquery-ui.css";
          $("#link1").attr("href",str);
        });
```

```
    $("#div2").datepicker();
    $("input:button").button();
    $('#main').accordion({ active :2, event:"mouseover" });
  });
</script>
</head>
<body>
  <h1> jquery-UI Themes </h1> <hr/>

  Select Theme :<select id="list1">
                  <option> Theme1 </option>
                  <option> Theme 2 </option>
                  <option> Theme 3 </option>
                  <option> Theme 4 </option>
              </select>
      <br><br>
      <div id="div2"></div> <br>

      <div id="main">
        <h3> jquery </h3>
        <div class="ci">This panel contains content related to jquery -
        </div>

        <h3> JSP </h3>
        <div class="ci">This panel contains content related to JSP. </div>

        <h3> PHP </h3>
        <div class="ci">This panel contains content related to PHP. </div>

        <h3> ASP.NET </h3>
        <div class="ci"> This panel contains content related to ASP.NET...
        </div>
      </div>
  </body>
</html>
```

JQuery - Plugin's.

→ Additional libraries that are developed based on Jquery Library. are called "jquery- Plug-ins".

→ Jquery team is provides some plugins like jquery-ui.

→ similarly there are other jquery plug-ins available in internet.

→ If you want to use any jquery plug-ins, you need to download from corresponding files from that website.

→ we need the following files in most of the jquery plug-ins:

   1) *.js file related to plugin

   2) *.css file

   3) required images

   4) jquery Library.

Examples of jquery plugin websites

1) http://bxSlider.com/

2) http://wowSlider.com/

3) http://sliderjs.com/   etc.,

```
<!-- create a webpage to implement image slide show using bxSlider
     jquery-plug-in -->

<html>
  <head>
    <script src="jquery-1.4.min.js"></script>
    <script src="jquery.bxSlider.js"></script>
    <link id="link1" href="jquery.bxslider.css" rel="stylesheet"/>
  </head>
```

```html
<style>
    img
    {
        height : 300px;
        width : 100%;
    }
</style>
<script>
  $(document).ready(function()
    {
        $(".bxslider").bxSlider(
            {
            auto : true;
            autoControls : true;
            });
    });
</script>
</head>
<body>
    <h1>jquery bxslider Plug-in </h1> <hr>

    <ul class ="bxslider">
        <li> <img src="img1.jpg"/> </li>
        <li> <img src="img2.jpg"/> </li>
        <li> <img src="img3.jpg"/> </li>
    </ul>

</body>
</html>
```

$\Rightarrow$ jquery validation Plug-in :-

→ Basic jquery library does not provide any methods / properties to perform validations.

→ performing validations is one of the primary activity in client side programming.

→ To implement jquery based validations, you need to download special validation jquery plug-in from the below website.

```
http:// jquery validation.org /
```

Required files :-

1) jQuery - 1.11.3. min. js
2) jquery.validate. js.

<!-- create a html page to implement validations using jquery validation plugin -->

```html
<html>
 <head>
  <script src="jquery-1.11.3. min.js"> </script>
  <script src="jquery.validate.js" > </script>

  <style>
   .error
   {
     color :Red;
   }
  </style>
  <script>
  $(document).ready (function()
  {
```

```
$("#form1").validate(
{
    rules : { txtUname : { required : true },
              txtPass1 : { required : true },
              txtPass2 : { equalTo : "#txtPass1" },
              txtEmail : { required : true, email : true },
              txtAge : { min : 20, max : 40 },
              txtCity : { minlength : 3, maxlength : 10 }
    },

    messages : { txtUname : "User name is required",
                 txtEmail : { required : "Email id is required",
                              email : "Email id should in proper format" }
    }

});
});
</script>
</head>
<body>
<h1> jQuery Validation Plug-in </h1> </hr>

    <form id="form1">

    UserName : <input type="text" id="txtUname" name="txtUname" /> <br/>

    Password : <input type="password" id="txtPass1" name="txtPass1" /> <br/>

    Retype Password : <input type="password" id="txtPass2" name="txtPass2" /> <br/>

    Email-id: <input type="text" id="txtEmail" name="txtEmail" /> <br/>

    Age: <input type="text" id="txtAge" name="txtAge" /><br/>

    City: <input type="text" id="txtCity" name="txtCity" /> <br/>

    <input type="submit" value="Register" id="sb1" name="sb1"/>

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

12/11/15

AJAX programming in JQuery.

What is AJAX?

→ AJAX- Asynchronous JavaScript And XML.

→ AJAX is a client/side programming technique used to communicate with server Asynchronously.

Note:-

→ This technique is first identified and used by Google team.

Advantages of AJAX :-

1) Quick Response.

2) Reduced burden on the server.

3) Reduce network traffic.

4) Partial page updates.

Implementing AJAX programming :-

→ AJAX can implementined using JavaScript.

→ If we use only JavaScript, it will be difficult & time consuming programming

→ In order to make AJAX programming easy, there are several Javascript libraries available.

  Ex: 1) jQuery
        2) Angular Js
        3) DoJo
        4) Moo Tools. etc.

Implementing AJAX programming in jquery:

→ Jquery provides a built-in function called "ajax()" to implement AJAX programming.

→ In AJAX programming, your Javascript code in responsible to send the request, recieve the response & display in the webpage.

AJAX function :-

Syntax:
```
$.ajax({ url : "    ",
         type : "POST",
         data : "    ",
         contentType : "application/JSON;",
         success : function(response)
         {
             // response handling logic
         }
});
```

`<!-- Create a webpage to display Dept data from server using AJAX.-->`

1) ServerData.txt   (ASP.Net)
2) AJAX Demo.html

1) Server Data.txt

```
[ { "Deptno" : 10, "Dname" : "Accounts", "Loc" : "Hyd" },
  { "Deptno" : 20, "Dname" : "Sales", "Loc" : "Hyd" },
  { "Deptno" : 30, "Dname" : "Marketing", "Loc" : "Hyd" } ]
```

2) AJAX Demo1.html :-

```html
<html>
  <head>
    <title></title>
    <script src="Scripts/jquery-1.11.3.min.js"></script>
    <script>
      $(document).ready(function()
      {
          $("#b1").click(function()
          {
```

```
$.ajax( url = "ServerData.txt",
        type = "GET";
        data = " ";
        contentType = "application/JSON";
        success: function (response)
                {
                  var obj = $.parseJSON (response);
                  for (i =0; i < obj.length ; i++)
                  {
                    var str = "<tr>";
                    str = str + "<td>" + obj[i].Deptno + "</td>";
                    str = str + "<td>" + obj[i].Dname + "</td>";
                    str = str + "<td>" + obj[i].Loc + "</td>";
                    str = str + "</tr>";

                    $("table1").append(str);
                  }
              });
        });
   });
</script> </head>
<body>
<h1> AJAX Demo by using jquery </h1> <br1>
<button id="b1" > GetData </button>
<br> <br>
<table id="table1" border="2" width="400">
   <tr>
      <th> Deptno </th>
      <th> Dname </th>
      <th> Loc </th>
   </tr>
<table> </body>
</html>
```

```
<!--Create a webpage to display the dept details by reading data from
database using AJAX with jquery -->     (ASP.NET)
```

1) WebService1.asmx.cs

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

namespace webApplication1
{
// [webservice]
 [ScriptService]

 public class webService1 : WebService
 {
    [webmethod]
    public string GetDept()
    {
      string str=" ";
      string connStr="server=. ; database=' Sathya'; Trusted_connection =
                                                        true;";
      string cmdText="select * from Dept";
      SqlDataAdapter da=new SqlDataAdapter (cmdText, connStr);
      DataTable dt= new DataTable();
      da.fill(dt);
      foreach(DataRow item in dt.Rows)
      {
         if(str=="")
            str=str+"[";
         else
            str=str+",";
```

```
str = str + "{\"Deptno\" : " item[0].Tostring() + ",";
str = str + "{\"Dname\" : " item[1].Tostring() + "\"*";
str = str + "{\"Loc\" : " item[2].Tostring() + "\"}";

}

str = str + "]";
return str;

}

}
```

2) <u>Ajax Demo.html</u>

```html
<html>
  <head>
    <title></title>
    <script src="Scripts/jquery-1.11.3.min.js"></script>
    <script>
      $(document).ready(function()
      {
        $("#b1").click(function()
        {
          $.ajax(
          {
            url : "WebService1.asmx/GetDept",
            type : "POST",
            data = "",
            contentType : "application/JSON",
            success : function(response)
            {
              var obj = $.parseJSON(response.d);
              for(i=0; i< obj.length ; i++)
              {
                var str = "<tr>";
                str = str + "<td>" + obj[i].Deptno + "<td>"
```

```
        str = str + "<td>" + obj[i].Dname + "</td>";
        str = str + "<td>" + obj[i].Loc + "</td>";
        str = str + "<tr>";
        $("#table1").append(str);
      }
    }
  });
});
});
</script>
</head>

<body>
  <h1> AJAX Demo  <lh1> </hr>
    <button id = "b1" > GetDept </button>
    <br></br>
    <table id ="table1"  border ="2" >
      <hr>
        <th> Deptno </th>
        <th> Dname </th>
        <th> Loc </th>
      </hr>
    <Hable>
  </body>
  </html>
```

Note:-

mail id | sir.narasimha@yahoo.com |

⇒ JSON :—

→ JSON stands for JavaScript Object Notation.

→ JSON concept is used to transfer the data from client to server (or) server to client in ajax calls.

→ It is a object representation technique in JavaScript.

→ Before JSON, we need depends on the XML.

→ If we use XML, we need to serialize/de-serialize the XML data into objects.

→ But JSON can access directly in Javascript.

Advantages of JSON :—

1) Easy to respresent, access and read.
2) Light weight objects
3) No need to perform serialize/deserialize.
4) All Javascript Libraries supports
5) Easy to map programming objects to JS objects due to its property respresentation.

Interview Questions.

1) what is Jquery ?
2) what are Advantages of Jquery?
3) How JavaScript & jquery are different?
4) Is jquery replacement of Javascript?
5) Is jquery library for client scripting (or) server scripting.?
6) why there are 2 different versions of jquery library?
7) what is CDN?
8) what are the examples of CDN?
9) How to load jquery from CDN ?

AngularJs jobs | jobs@enterpi·com | 040-66034 111

⇒ <u>CDN:-</u>

→ CDN stands for "Content Distribution Network".

and. "Content Delivery Network"

→ CDN is a global location ~~form~~ which we can refer the files to integrate with other applications.

Examples :

1) Google
2) Microsoft
3) jQuery.

① `<script src = " http ://ajax·googleapis·com/ajax/libs/jquery/1·9·1/`
`jquery·min js">`

`</script>`

② `<script src=" http://ajax·microsoft·com/ajax/jquery/jquery-1·9·1·min.js}`
`<script>`

③ `<script src=" http://code·jquery·com/jquery-1·1·9·1.min·js">`
`</script>`

10) what does dollar sign ($) means in jquery? ($ - jquery factory function)

11) Can we use multiple Java script libraries in same web page?

A) Yes, we can use more than one Js libraries.

12) what is jquery. no conflict() function?

```
jquery. noConflict();
jquery (document). ready(function()
{
    jquery(" div"). hide();
});
```

13) Can we use our own specific characters in the place of $ sign in jquery?

```
var $n = jquery. noConflict();
$n(document). ready (function()
{
    $n("div). hide();
});
```

14) which is the starting point of code execution in jquery?

15) Can we have multiple document. ready() function on same page? — Yes

16) Is there any difference b/w window. load() & document. ready() function?   Window. load()

→ window. load() event occurs when all the html elements are loaded in the memory with content.

→ we can write only one window. Load() event handling code.

document. ready()

→ we can write multiple document. ready() event handling code.

→ It will occurs when the elements are ready (structure) in memory. No need to wait for the content to be load.

17) can we write jquery code without document.ready()?

    – Yes, By using normal Java Script code.

18) what are selectors in jquery?

19) what does $("div.Hello") will select?

    → It access the tag which is having Hello class name with in the div tag.

20) In jquery   fastest Selection (#id)

                Slowest selector is (.class)

21) How to select all textBox in page?

    – $("input : text").

22) Diff b/w '$(this)' and 'this' in jquery.

    this.innerText → It is javascript keyword. which can acces only
                                     JS features.

    $(this).text() → $(this) is a jquery selector which can acces
                            all the jquery methods.

23) How to Iterate each item from the result of Selector?

    – eq() & each()

24) what is the use of jquery. each() function?

25) what is chaining in jquery?

```
$(document).ready(function()
{
    $("#div1").addClass('class1')
            .css('color', 'red')
            .fadeIn('slow');
});
```