

```
});
```

- For loop is value based. The variable contains "value" directly; but not index.
- Forward iterations are only possible. Backward iterations are not possible.
- We can't start the loop from the middle.

### Steps for working with ForOf Loop

---

```
for (variable of array)
```

```
{  
  variable  
}
```

- For loop is value based. The variable contains "value" directly; but not index.
- Forward iterations are only possible. Backward iterations are not possible.
- We can't start the loop from the middle.

### Steps for working with ForIn Loop

---

```
for (variable in array)
```

```
{  
  array[variable]  
}
```

- For loop is index based. The variable contains "index".
- Forward iterations are only possible. Backward iterations are not possible.
- We can't start the loop from the middle.

### Example on For Loop

---

```
<html>  
  <head>  
    <title>for</title>  
  </head>  
  <body>  
    <h1>for</h1>  
    <script>  
      var cities = ["Hyderabad", "Chennai", "Bangalore", "Pune"];  
      for (var i = 0; i < cities.length; i++)  
      {  
        console.log(cities[i]);  
      }  
    </script>  
  </body>  
</html>
```

### Example on ForEach Loop

---

```
<html>
  <head>
    <title>Foreach</title>
  </head>
  <body>
    <h1>Foreach</h1>
    <script>
      var cities = ["Hyderabad", "Chennai", "Bangalore", "Pune"];
      cities.forEach(function (city)
      {
        console.log(city);
      });
    </script>
  </body>
</html>
```

### Example on ForOf Loop

---

```
<html>
  <head>
    <title>ForOf</title>
  </head>
  <body>
    <h1>ForOf</h1>
    <script>
      var cities = ["Hyderabad", "Chennai", "Bangalore", "Pune"];
      for (var city of cities)
      {
        console.log(city);
      }
    </script>
  </body>
</html>
```

### Example on ForIn Loop

---

```
<html>
  <head>
    <title>ForIn</title>
  </head>
  <body>
    <h1>ForIn</h1>
    <script>
      var cities = ["Hyderabad", "Chennai", "Bangalore", "Pune"];
      for (var i in cities)
      {
        console.log(i + ", " + cities[i]);
      }
    </script>
  </body>
</html>
```

HARSHA

# JQUERY

## Fundamentals of jQuery

- jQuery is a JavaScript library, which is a collection of pre-defined functions that are written in JavaScript, which are used to perform DOM manipulations easily.
- jQuery is the shortcut syntax for JavaScript.
- jQuery internally uses JavaScript
- jQuery supports DOM manipulations and AJAX
- jQuery is case sensitive
- jQuery is open source. That means its source code is available for public online for free
- jQuery is cross browser compatible. That means jQuery supports all the browsers such as Google Chrome, Mozilla Firefox, Safari, Microsoft Internet Explorer, Microsoft Edge etc.

### Generations of jQuery

#### 1. Generation 1.x

- It supports all the browsers including IE 6, 7, 8.

#### 2. Generation 2.x

- It supports all the browsers except IE 6, 7, 8.

#### 3. Generation 3.x

- It supports all the browsers except IE 6, 7, 8.

### Formats of jQuery Library file

- “jQuery library file” is available in two formats:
  1. Uncompressed
  2. Compressed (or) Minified

#### 1. Uncompressed

- It contains the code with comments, line breaks and spaces.
- It is understandable.
- The file size is: 260 kb.
- It is recommended in development time.

#### 2. Compressed

- It contains the code with no comments, no line breaks and no spaces.

- It is not understandable.
- The file size is: 88 kb.
- It is recommended, while uploading the project into the server.

### Downloading jQuery

- Go to “<http://jquery.com>”.
- Click on “Download jQuery”.
- Click on “Download the uncompressed, development jQuery 3.2.1”.
- You will get “jquery-3.2.1.js” file.
- Press Ctrl+S to save (download) the file.
- After downloading the file, go to the downloaded location and copy-paste it into the “application folder” (c:\jquery).
- You have to create the html file in the same folder.

### “\$” function

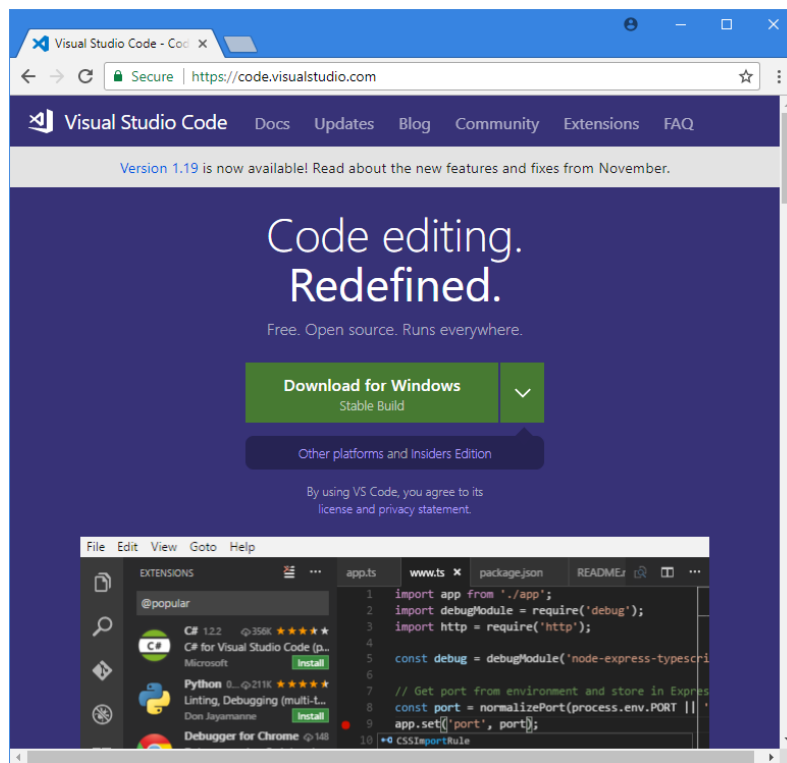
- “\$” is a pre-defined function in jQuery, which is used to select the elements.
- It receives a “selector” as an argument, searches the DOM for the matching elements and returns the matching elements as an array.
- **Syntax:**    \$(“selector”)

## Steps to Prepare First Example in jQuery

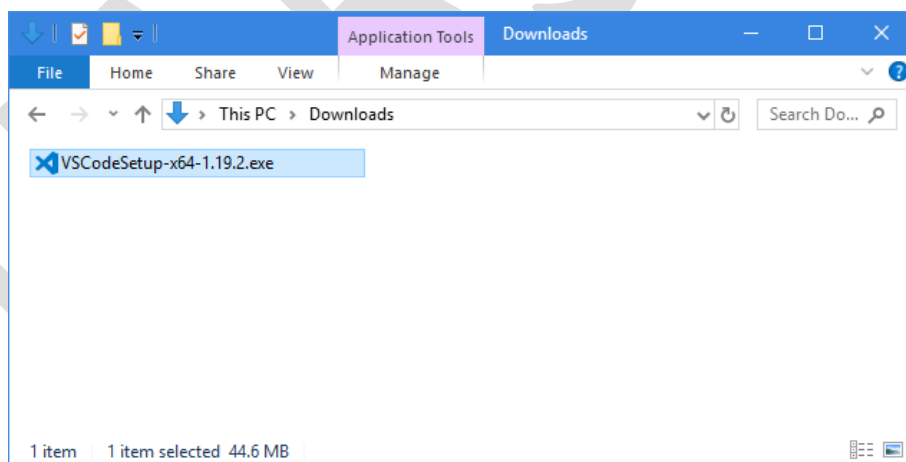
1. Installing Visual Studio Code
2. Creating jQuery Program
3. Executing jQuery Program

### 1. Installing Visual Studio Code

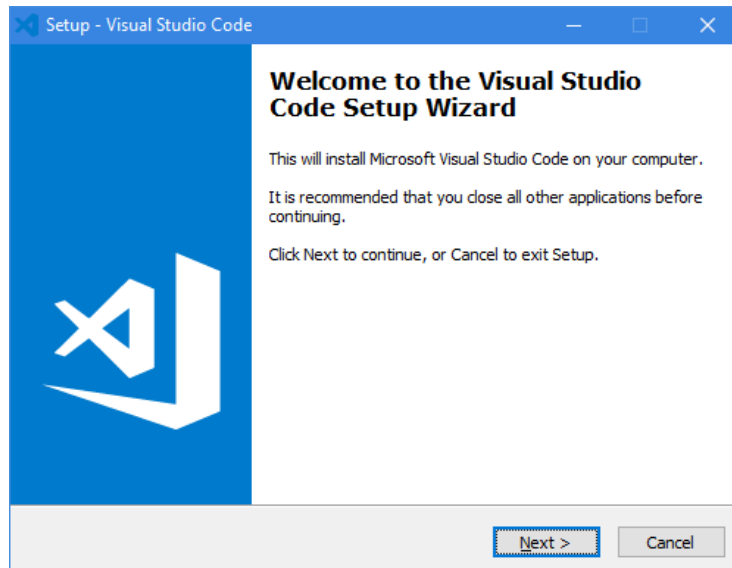
- “Visual Studio Code” is the recommended editor for html, css, javascript, jquery etc.
- Go to <https://code.visualstudio.com>



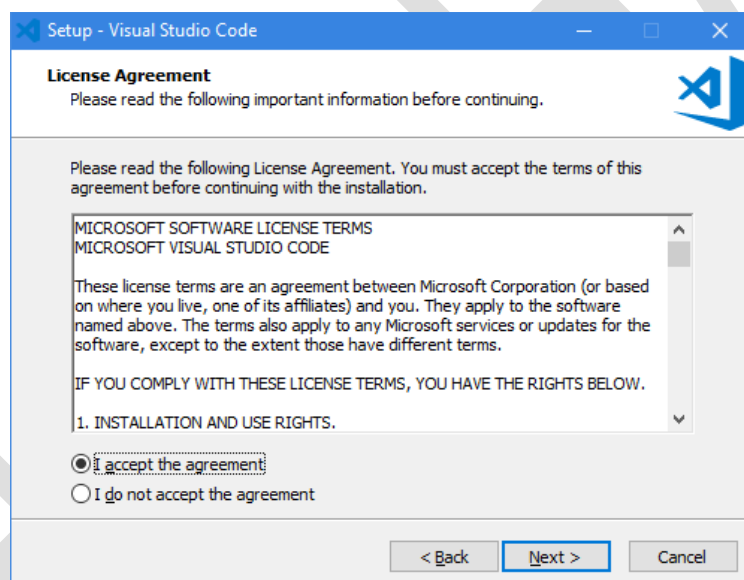
- Click on “Download for Windows”.
- **Note:** The version number may be different at your practice time.
- Go to “Downloads” folder; you can find “VSCodeSetup-x64-1.19.2.exe” file.



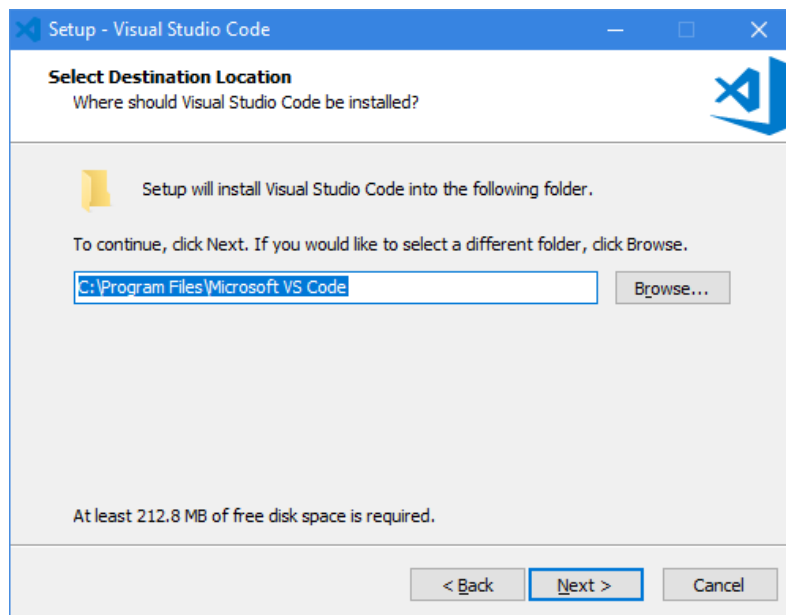
- Double click on “VSCodeSetup-x64-1.19.2.exe” file.
- Click on “Yes”.



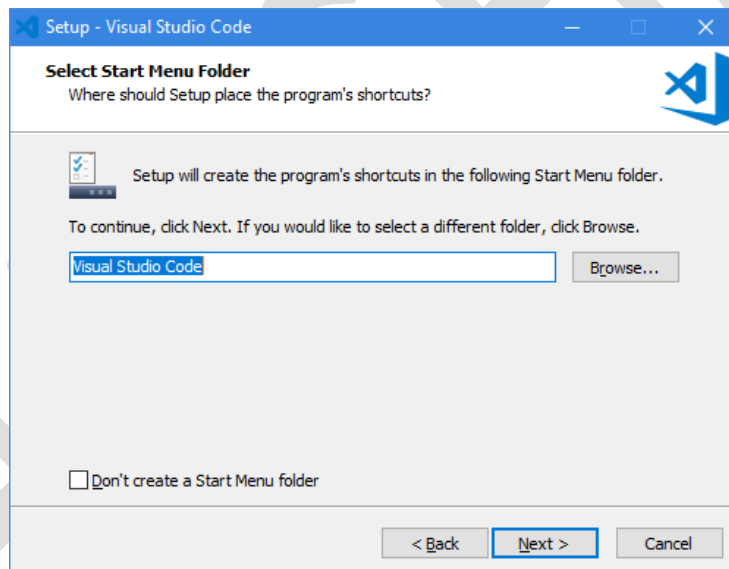
- Click on “Next”.



- Click on “I accept the agreement”.
- Click on “Next”.

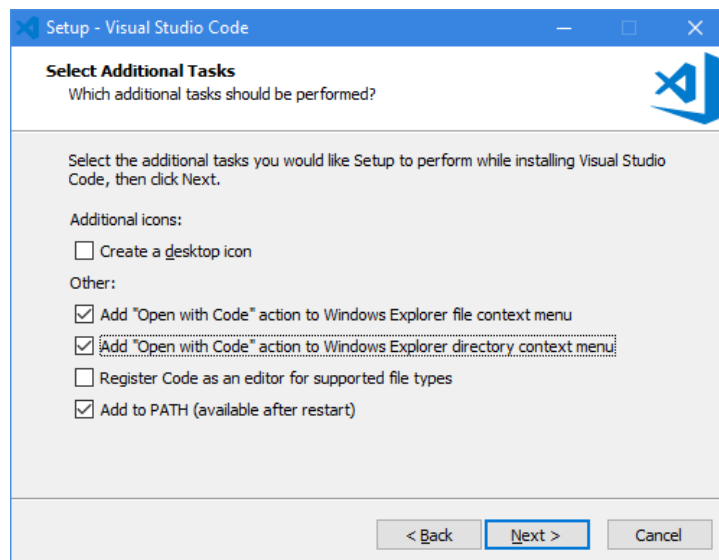


- Click on “Next”.

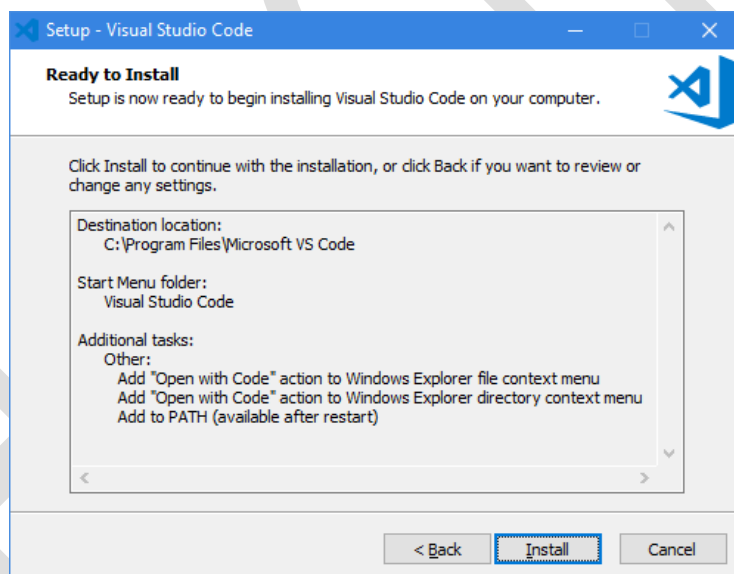


- Click on “Next”.

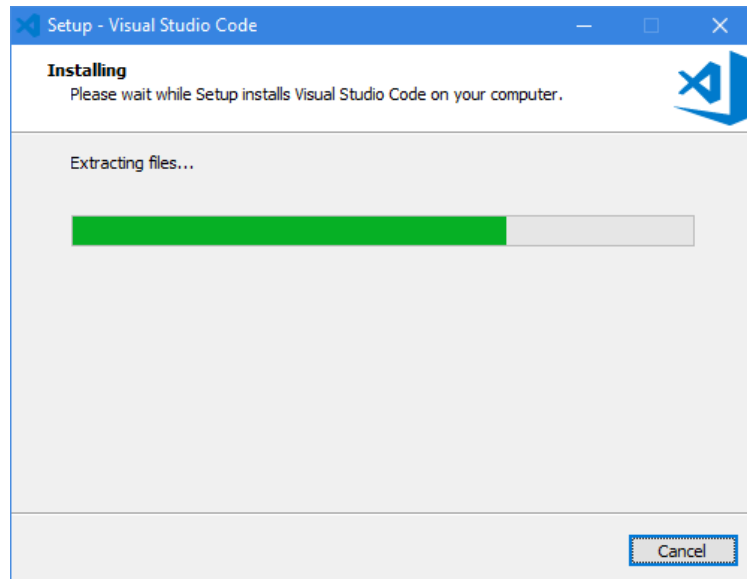




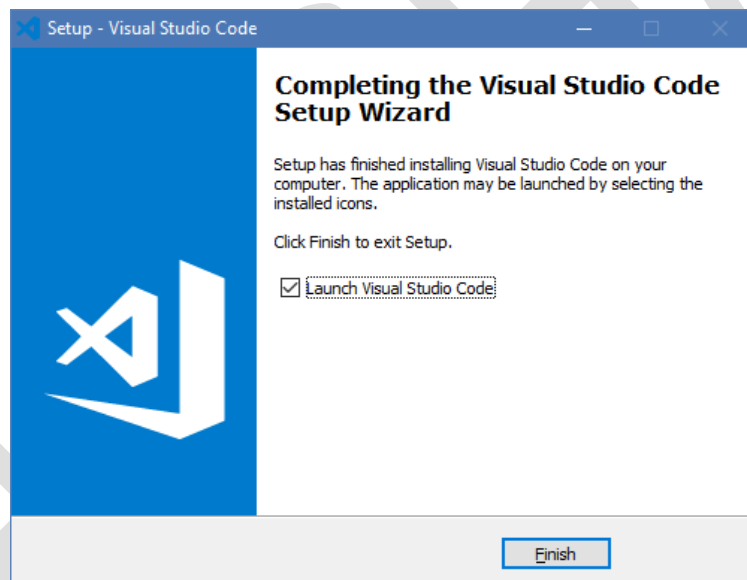
- Check the checkbox “Add Open with Code action to Windows Explorer file context menu”.
- Check the checkbox “Add Open with Code action to Windows Explorer directory context menu”.
- Click on “Next”.



- Click on “Install”.



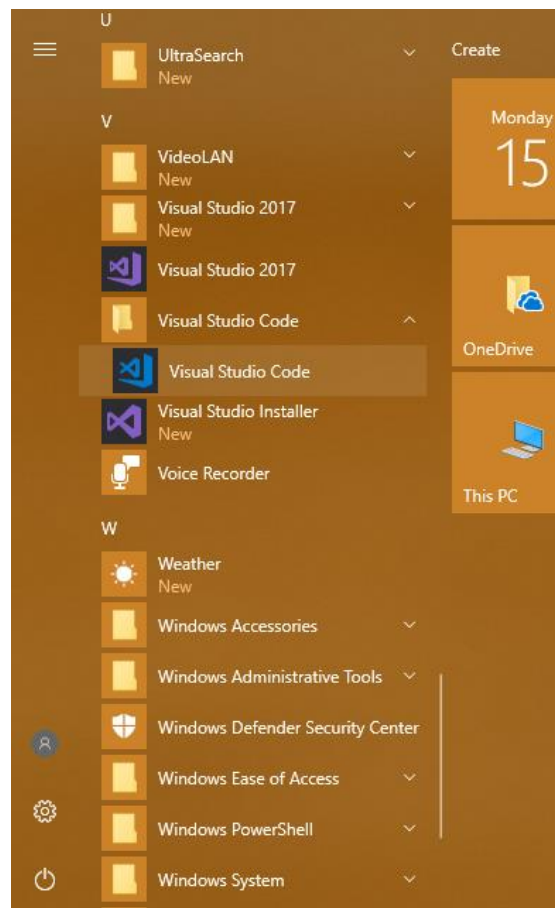
- Installation is going on....

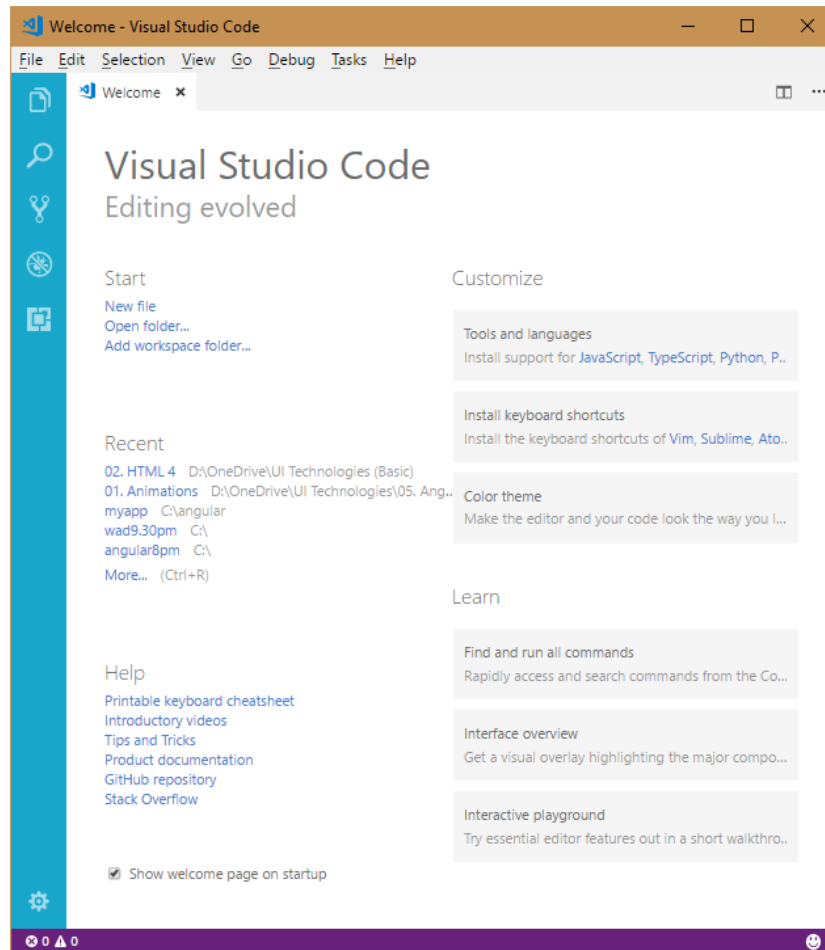


- Click on “Finish”.

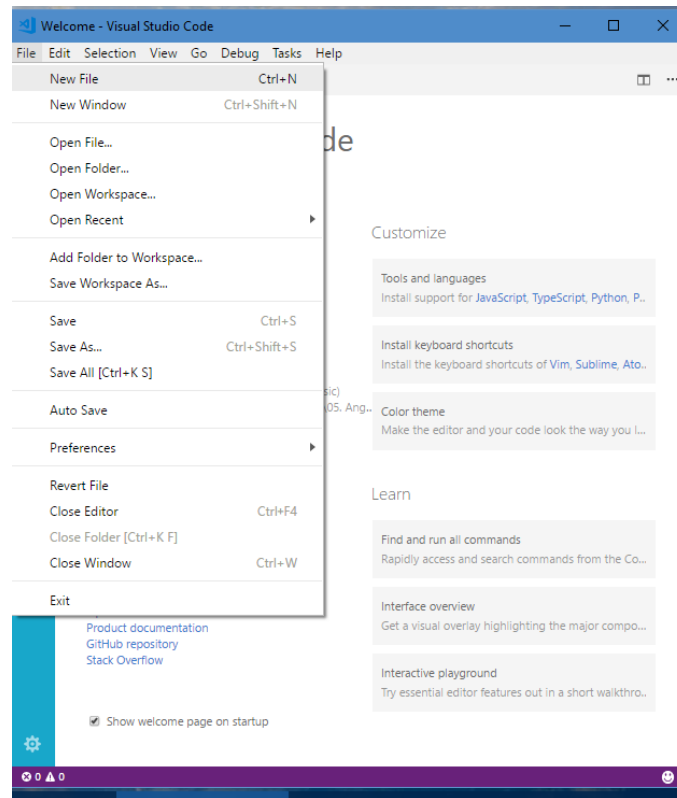
## 2. Create jQuery Program

- Open “Visual Studio Code”, by clicking on “Start” – “Visual Studio Code”.

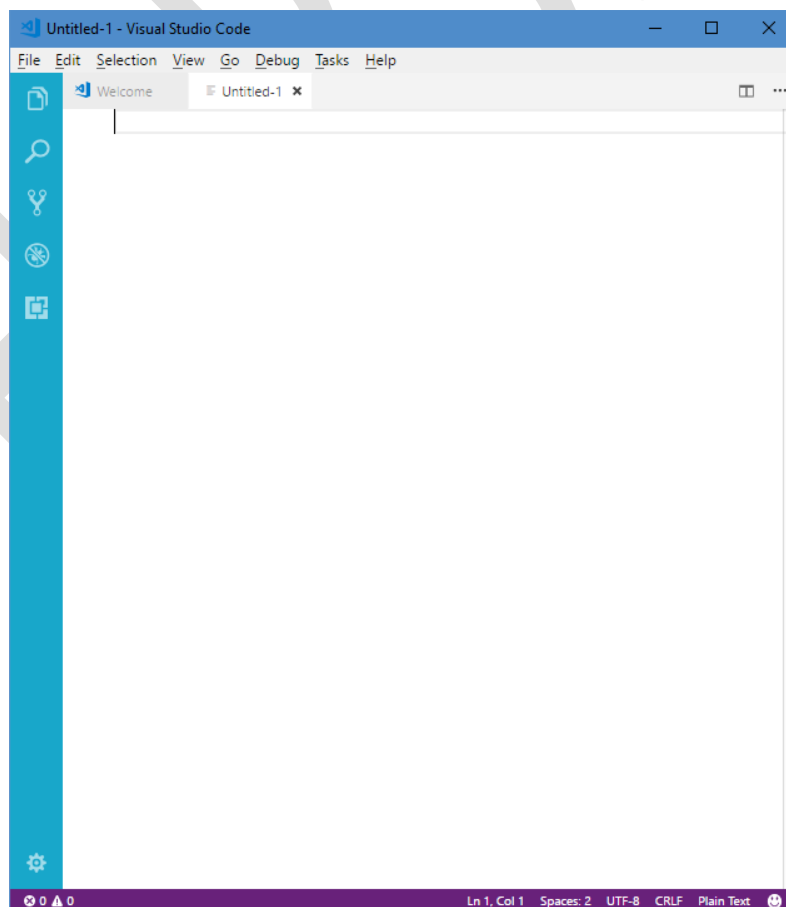




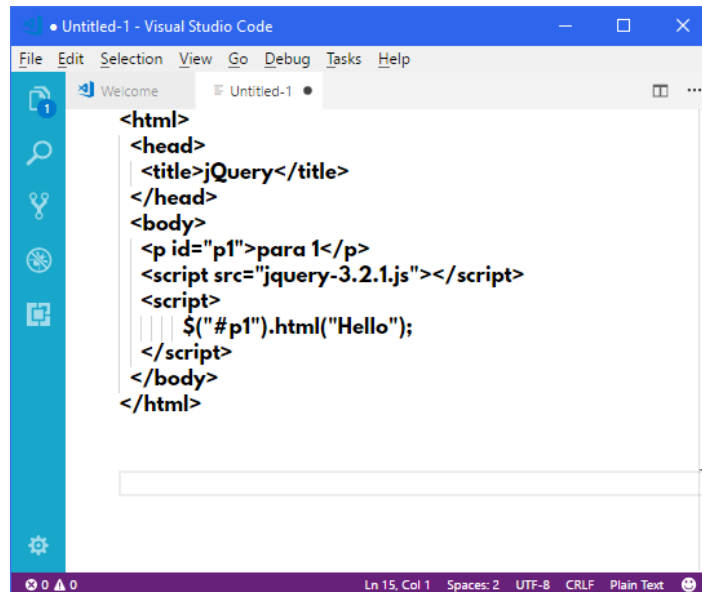
- Visual Studio Code opened.



- Go to “File” – “New File”.



- Type the program as follows:

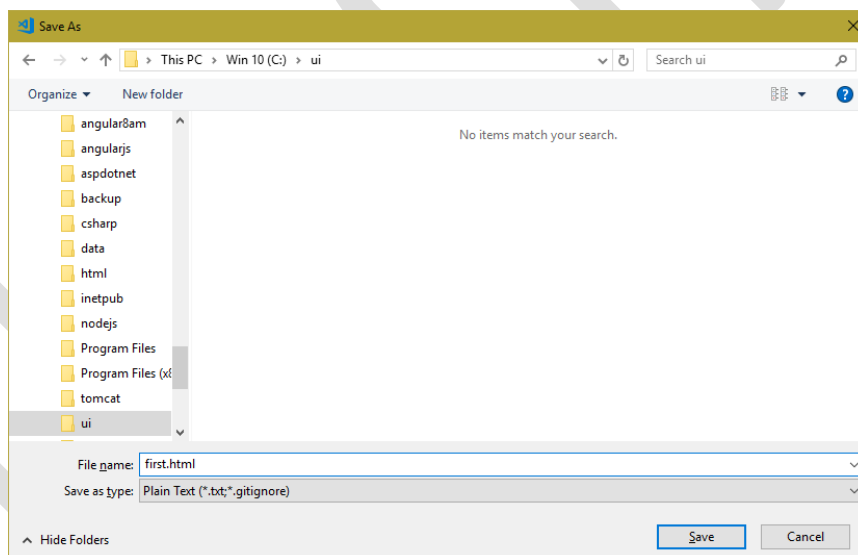


```

<html>
<head>
<title>jQuery</title>
</head>
<body>
<p id="p1">para 1</p>
<script src="jquery-3.2.1.js"></script>
<script>
    $("#p1").html("Hello");
</script>
</body>
</html>

```

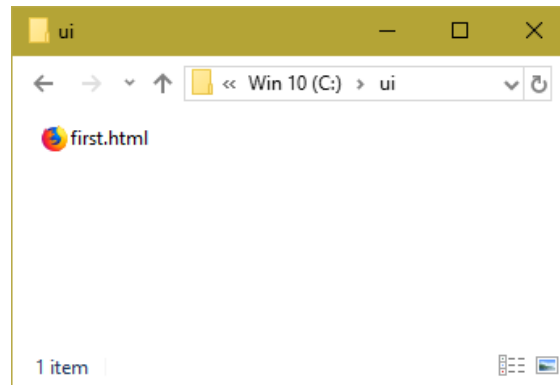
- Go to “File” menu – “Save” (or) Press Ctrl+S.



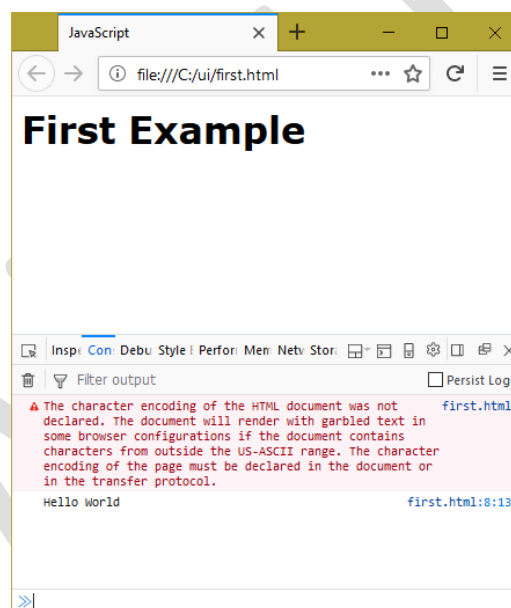
- Go to "c:\\" and click on "New folder". Enter the new folder name as "ui".
- Select “c:\ui” folder and enter the filename as “first.html”.
- Click on “Save”.
- Now the typescript file (c:\ui\first.html) is ready.

### 3. Execute the HTML Program

- Go to “Computer” or “This PC” and go to “c:\html” folder.



- Double click on “first.html” (or) Right click on “first.html” and click on “Open With” – “Mozilla Firefox” / “Open With” – “Google Chrome”.
- In the browser, right click in the web page and click on "Inspect Element" (or) press "F12" function key to open console.



**Output:** Hello World

## Manipulating Content

### Get html( )

- The html( ) gets the current inner html of the tag (including child tags).

**Syntax:**       html( )

**Example:**       \$("#p1").html( )

**Example on get html( )**

---

```
<html>
  <head>
    <title>jQuery - Get Html</title>
    <style type="text/css">
      #div1
      {
        border: 1px solid red;
      }
    </style>
  </head>
  <body>
    <div id="div1">
      Hello, <b>how</b> are you
    </div>
    <input type="button" id="button1" value="get html">
    <script src="jquery-3.2.1.js"></script>

    <script>
      $("#button1").click(fun1);
      function fun1()
      {
        //get html (including child tags)
        var s = $("#div1").html();
        alert(s);
      }
    </script>
  </body>
</html>
```

**Get text( )**

- The text( ) function gets the current inner html of the tag (only plain text, excluding child tags).

**Syntax:**        text( )

**Example:**        \$("#p1").text( )

**Example on get text( )**

---

```
<html>
  <head>
    <title>jQuery - Get Text</title>
    <style type="text/css">
      #div1
      {
        border: 1px solid red;
      }
    </style>
  </head>
  <body>
    <div id="div1">
```



```
    Hello, <b>how</b> are you
</div>
<br>
<input type="button" id="button1" value="get text">
<script src="jquery-3.2.1.js"></script>

<script>
    $("#button1").click(fun1);
    function fun1()
    {
        //get text (excluding child tags)
        var s = $("#div1").text();
        alert(s);
    }
</script>
</body>
</html>
```

### Set html( )

- The html( ) function is used to set (overwrite) the inner html of the tag (including child tags).

**Syntax:**       html("new inner html")

**Example:**       \$("#div1").html("<p>Hello</p>")

### Example on set html( )

```
<html>
<head>
    <title>jQuery - Set Html</title>
    <style type="text/css">
        #div1
        {
            border: 1px solid red;
        }
    </style>
</head>
<body>
    <div id="div1">
        Hello, <b>how</b> are you
    </div>
    <br>
    <input type="button" id="button1" value="set html">
    <script src="jquery-3.2.1.js"></script>

    <script>
        $("#button1").click(fun1);
        function fun1()
        {
            //set html
            $("#div1").html("I am <b>fine</b>");
        }
    </script>
```

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

### Set text( )

- The text( ) function is used to set (overwrite) the inner html of the tag (only plain text).

**Syntax:** text("new inner html" )

**Example:** \$("#div1").text("Hello")

### Example on set text( )

```
<html>
<head>
  <title>jQuery - Set Text</title>
  <style type="text/css">
    #div1
    {
      border: 1px solid red;
    }
  </style>
</head>
<body>
  <div id="div1">
    Hello, <b>how</b> are you
  </div>
  <br>
  <input type="button" id="button1" value="set text">
  <script src="jquery-3.2.1.js"></script>

  <script>
    $("#button1").click(fun1);
    function fun1()
    {
      //set text
      $("#div1").text("I am fine");
    }
  </script>
</body>
</html>
```

### before( )

- This function adds new content before the start tag of the element.

**Syntax:** before("new inner html" )

**Example:** \$("#div1").before("<p>new para</p>")

**Example on before( )**

---

```
<html>
  <head>
    <title>jQuery - Before</title>
    <style type="text/css">
      #div1
      {
        border: 1px solid red;
      }
    </style>
  </head>
  <body>
    <div id="div1">
      <p>para 1</p>
      <p>para 2</p>
      <p>para 3</p>
    </div>
    <br>
    <input type="button" id="button1" value="before">
    <script src="jquery-3.2.1.js"></script>
    <script>
      $("#button1").click(fun1);
      function fun1()
      {
        $("#div1").before("<p>new para</p>");
      }
    </script>
  </body>
</html>
```

**prepend( )**

- This function adds new content after the start tag of the element.

**Syntax:**       prepend("new content" )

**Example:**       \$("#div1").prepend("<p>new para</p>")

**Example on prepend( )**

---

```
<html>
  <head>
    <title>jQuery - Prepend</title>
    <style type="text/css">
      #div1
      {
        border: 1px solid red;
      }
    </style>
  </head>
  <body>
```

```
<div id="div1">
  <p>para 1</p>
  <p>para 2</p>
  <p>para 3</p>
</div>
<br>
<input type="button" id="button1" value="prepend">
<script src="jquery-3.2.1.js"></script>
<script>
  $("#button1").click(fun1);
  function fun1()
  {
    $("#div1").prepend("<p>new para</p>");
  }
</script>
</body>
</html>
```

### append( )

- This function adds new content before the end tag of the element.

**Syntax:**        append("new content" )

**Example:**        \$("#div1").append("<p>new para</p>")

### Example on append( )

```
<html>
<head>
  <title>jQuery - Append</title>
  <style type="text/css">
    #div1
    {
      border: 1px solid red;
    }
  </style>
</head>
<body>
  <div id="div1">
    <p>para 1</p>
    <p>para 2</p>
    <p>para 3</p>
  </div>
  <br>
  <input type="button" id="button1" value="append">
  <script src="jquery-3.2.1.js"></script>
  <script>
    $("#button1").click(fun1);
    function fun1()
    {
```

```

        $("#div1").append("<p>new para</p>");
    }
</script>
</body>
</html>

```

## after( )

- This function adds new content after the closing tag of the element.

**Syntax:** after("new content")

**Example:** \$("#div1").after("<p>new para</p>")

## Example on after( )

```

<html>
<head>
<title>jQuery - After</title>
<style type="text/css">
    #div1
    {
        border: 1px solid red;
    }
</style>
</head>
<body>
<div id="div1">
    <p>para 1</p>
    <p>para 2</p>
    <p>para 3</p>
</div>
<br>
<input type="button" id="button1" value="after">
<script src="jquery-3.2.1.js"></script>
<script>
    $("#button1").click(fun1);
    function fun1()
    {
        $("#div1").after("<p>new para</p>");
    }
</script>
</body>
</html>

```

## insertBefore( )

- This function cuts an existing element and pastes the same before the start tag of destination element.

**Syntax:** insertBefore("destination element")

**Example:** \$("#p1").insertBefore("#div1")

### Example on insertBefore( )

```
<html>
  <head>
    <title>jQuery - InsertBefore</title>
    <style type="text/css">
      #div1
      {
        border: 1px solid red;
      }
    </style>
  </head>
  <body>
    <div id="div1">
      <p>para 1</p>
      <p>para 2</p>
      <p>para 3</p>
    </div>
    <span id="span1">span1</span><br>
    <input type="button" id="button1" value="insert before">
    <script src="jquery-3.2.1.js"></script>
    <script>
      $("#button1").click(fun1);
      function fun1()
      {
        $("#span1").insertBefore("#div1");
      }
    </script>
  </body>
</html>
```

### prependTo( )

- This function cuts an existing element and pastes the same after the start tag of destination element.

**Syntax:** prependTo("destination element")

**Example:** \$("#p1").prependTo("#div1")

### Example on prependTo( )

```
<html>
  <head>
    <title>jQuery - PrependTo</title>
    <style type="text/css">
      #div1
      {
```

```
        border: 1px solid red;
    }
</style>
</head>
<body>
    <span id="span1">span1</span>
    <div id="div1">
        <p>para 1</p>
        <p>para 2</p>
        <p>para 3</p>
    </div><br>
    <input type="button" id="button1" value="prepend to">
    <script src="jquery-3.2.1.js"></script>
    <script>
        $("#button1").click(fun1);
        function fun1()
        {
            $("#span1").prependTo("#div1");
            //$("#span1").clone().prependTo("#div1");
        }
    </script>
</body>
</html>
```

### appendTo( )

- This function cuts an existing element and pastes the same before the end tag of destination element.

**Syntax:**        appendTo("destination element")

**Example:**        \$("#p1").appendTo("#div1")

### Example on appendTo( )

---

```
<html>
<head>
    <title>jQuery - AppendTo</title>
    <style type="text/css">
        #div1
        {
            border: 1px solid red;
        }
    </style>
</head>
<body>
    <span id="span1">span1</span>
    <div id="div1">
        <p>para 1</p>
        <p>para 2</p>
        <p>para 3</p>
    </div>
    <br>
    <input type="button" id="button1" value="append to">
```

```

<script src="jquery-3.2.1.js"></script>
<script>
    $("#button1").click(fun1);
    function fun1()
    {
        //$("#span1").appendTo("#div1");
        $("#span1").clone().appendTo("#div1");
    }
</script>
</body>
</html>

```

### insertAfter( )

- This function cuts an existing element and pastes the same after end tag of destination element.

**Syntax:** insertAfter("destination element")

**Example:** \$("#p1").insertAfter("#div1")

### Example on insertAfter( )

```

<html>
<head>
<title>jQuery - InsertAfter</title>
<style type="text/css">
    #div1
    {
        border: 1px solid red;
    }
</style>
</head>
<body>
<span id="span1">span1</span>
<div id="div1">
    <p>para 1</p>
    <p>para 2</p>
    <p>para 3</p>
</div>
<br><input type="button" id="button1" value="insert after">
<script src="jquery-3.2.1.js"></script>
<script>
    $("#button1").click(fun1);
    function fun1()
    {
        //insert after ending tag
        $("#span1").insertAfter("#div1");
    }
</script>
</body>
</html>

```



**wrap( )**

- This function adds a parent tag for each selected element.

**Syntax:** wrap("<parenttag></parenttag>")

**Example:** \$("p").wrap("<div></div>")

**Example on wrap( )**

---

```
<html>
  <head>
    <title>jQuery - Wrap</title>
    <style type="text/css">
      div
      {
        border: 1px solid red;
      }
    </style>
  </head>
  <body>
    <p>para 1</p>
    <p>para 2</p>
    <p>para 3</p>
    <br>
    <input type="button" id="button1" value="wrap">

    <script src="jquery-3.2.1.js"></script>
    <script>
      $("#button1").click(fun1);
      function fun1()
      {
        $("p").wrap("<div></div>");
      }
    </script>
  </body>
</html>
```

**wrapAll( )**

- This function adds a common parent tag for all the selected elements.

**Syntax:** wrapAll("<parenttag></parenttag>")

**Example:** \$("p").wrapAll("<div></div>")

**Example on wrapAll( )**

---

```
<html>
```

```
<head>
  <title>jQuery - WrapAll</title>
  <style type="text/css">
    div
    {
      border: 1px solid red;
    }
  </style>
</head>
<body>
  <p>para 1</p>
  <p>para 2</p>
  <p>para 3</p>
  <br>
  <input type="button" id="button1" value="wrap all">
  <script src="jquery-3.2.1.js"></script>
  <script>
    $("#button1").click(fun1);
    function fun1()
    {
      $("p").wrapAll("<div></div>");
    }
  </script>
</body>
</html>
```

### empty( )

- This function is used to delete the inner html of the element.

**Syntax:**        empty( )

**Example:**        \$("p").empty( )

### Example on empty( )

---

```
<html>
  <head>
    <title>jQuery - Empty</title>
    <style type="text/css">
      body, input
      {
        font-family: 'Tahoma';
        font-size: 30px;
      }
      #div1
      {
        border: 1px solid red;
      }
    </style>
  </head>
  <body>
```

```
<div id="div1">Hello</div>
<br>
<input type="button" id="button1" value="empty">
<script src="jquery-3.2.1.js"></script>
<script>
    $("#button1").click(fun1);
    function fun1()
    {
        $("#div1").empty();
    }
</script>
</body>
</html>
```

### remove( )

- This function removes an existing element.

**Syntax:**        remove( )

**Example:**        \$("p").remove( )

### Example on remove( )

---

```
<html>
<head>
    <title>jQuery - Remove</title>
    <style type="text/css">
        #div1
        {
            border: 1px solid red;
        }
    </style>
</head>
<body>
    <div id="div1">Hello</div><br>
    <input type="button" id="button1" value="remove">
    <script src="jquery-3.2.1.js"></script>

    <script>
        $("#button1").click(fun1);
        function fun1()
        {
            $("#div1").remove();
        }
    </script>
</body>
</html>
```

**replaceWith( )**

- This function replaces (overwrites) an existing element with another element.

**Syntax:**        replaceWith("new content")

**Example:**        \$("div").replaceWith("<p>new para</p>")

**Example on replaceWith( )**

```
<html>
  <head>
    <title>jQuery - ReplaceWith</title>
    <style type="text/css">
      #div1
      {
        border: 1px solid red;
      }
    </style>
  </head>
  <body>
    <div id="div1">Hello</div>
    <br>
    <input type="button" id="button1" value="replace with">
    <script src="jquery-3.2.1.js"></script>

    <script>
      $("#button1").click(fun1);
      function fun1()
      {
        $("#div1").replaceWith("<p>new para</p>");
      }
    </script>
  </body>
</html>
```

**Event Handling**

- **Event:** A keyboard/ mouse action, performed by the user, at run time.
- jQuery provides a new way to handle the events more easily.
- jQuery supports all the javascript events.

**List of jQuery Functions to handle Events**

1. click( )
2. dblclick( )
3. mouseover( )

4. mouseout()
5. hover()
6. mousemove()
7. focus()
8. blur()
9. keyup()
10. keypress()
11. change()
12. on()
13. off()

### The “click()” function

- The “click()” function handles “click” event.
- The “click” event executes when the user clicks on an element.

#### **Syntax:**

```
$(“selector”).click(functionname);
function functionname()
{
    Code here
}
```

#### **Example on “Click”**

```
<html>
<head>
<title>jQuery - Click</title>
<style type="text/css">
    body,input
    {
        font-family: Tahoma;
        font-size: 30px;
    }
    #div1
    {
        width: 300px;
        height: 200px;
        background-color: #00cccc;
    }
</style>
```

```
</head>
<body>
  <div id="div1">div 1</div>
  <script src="jquery-3.2.1.js"></script>
  <script>
    $("#div1").click(fun1);
    function fun1()
    {
      $("#div1").html("Thanx");
    }
  </script>
</body>
</html>
```

### The “dblclick( )” function

- The “dblclick( )” function handles “dblclick” event.
- The “dblclick” event executes when the user double clicks on an element.

#### **Syntax:**

```
$(“selector”).dblclick(functionname);
function functionname( )
{
  Code here
}
```

#### Example on “Dblclick”

---

```
<html>
<head>
  <title>jQuery - Dblclick</title>
  <style type="text/css">
    body,input
    {
      font-family: Tahoma;
      font-size: 30px;
    }
    #div1
    {
      width: 300px;
      height: 200px;
      background-color: #00cccc;
    }
  </style>
</head>
<body>
  <div id="div1">double click me</div>
```

```
<script src="jquery-3.2.1.js"></script>

<script>
    $("#div1").dblclick(fun1);
    function fun1()
    {
        $("#div1").html("Thanx");
    }
</script>
</body>
</html>
```

## The “mouseover( )” function and “mouseout( )” function

### The mouseover( ) function

- The “mouseover( )” function handles “mouseover” event.
- The “mouseover” event executes when the user places the mouse pointer on the element.

#### **Syntax:**

```
$(“selector”).mouseover(functionname);
function functionname( )
{
    Code here
}
```

### The mouseout( ) function

- The “mouseout( )” function handles “mouseout” event.
- The “mouseout” event executes when the user moves the mouse pointer from inside to outside the element.

#### **Syntax:**

```
$(“selector”).mouseout(functionname);
function functionname( )
{
    Code here
}
```

**Example on “Mouseover” and “mouseout”**

---

```
<html>
  <head>
    <title>jQuery - Mouseover and Mouseout</title>
    <style type="text/css">
      body,input
      {
        font-family: Tahoma;
        font-size: 30px;
      }
      #div1
      {
        width: 300px;
        height: 200px;
        background-color: #00cccc;
      }
    </style>
  </head>
  <body>
    <div id="div1">hover me</div>

    <script src="jquery-3.2.1.js"></script>

    <script>
      $("#div1").mouseover(fun1);
      $("#div1").mouseout(fun2);
      function fun1()
      {
        $("#div1").html("Thanx");
      }
      function fun2()
      {
        $("#div1").html("hover me");
      }
    </script>
  </body>
</html>
```

**The “hover( )” function**

- The “hover( )” function is a shortcut to handle both “mouseover” and “mouseout” events at-a-time.

**Syntax:**

```
$(“selector”).hover(functionname1, functionname2);
function functionname1( )
{
  Code here
}
function functionname2( )
```



```
{  
  Code here  
}
```

### Example on “hover”

---

```
<html>  
  <head>  
    <title>jQuery - Hover</title>  
    <style type="text/css">  
      body,input  
      {  
        font-family:'Tahoma';  
        font-size:30px;  
      }  
    #div1  
    {  
      width:200px;  
      height:200px;  
      background-color:#00ff99;  
    }  
  </style>  
</head>  
<body>  
  <div id="div1">  
    hover me  
  </div>  
  
  <script src="jquery-3.2.1.js"></script>  
  
  <script>  
    $("#div1").hover(fun1, fun2);  
    function fun1()  
    {  
      $(this).html("Thanx");  
    }  
    function fun2()  
    {  
      $(this).html("hover me");  
    }  
  </script>  
</body>  
</html>
```

#### The “mousemove( )” function

- The “mousemove( )” function handles “mousemove” event.
- The “mousemove” event executes when the user moves the pointer from one place to another place within the same element (for every pixel change).

#### Syntax:

```

$(“selector”).mousemove(functionname);
function functionname(event)
{
    event.pageX = X position of mouse pointer
    event.pageY = Y position of mouse pointer
}

```

### Example on “mousemove”

```

<html>
<head>
<title>jQuery - Mousemove</title>
<style type="text/css">
    body,input
    {
        font-family: Tahoma;
        font-size: 30px;
    }
    #div1
    {
        width: 300px;
        height: 200px;
        background-color: #00cccc;
    }
</style>
</head>
<body>
<div id="div1">mouse move me</div>

<script src="jquery-3.2.1.js"></script>

<script>
    $("#div1").mousemove(fun1);
    function fun1(event)
    {
        $("#div1").html(event.pageX + ", " + event.pageY);
    }
</script>
</body>
</html>

```

### The “focus( )” function

- The “focus( )” function handles “focus” event.
- The “focus” event executes when the cursor enters into the element.

#### Syntax:

```

$(“selector”).focus(functionname);

```

```
function functionname( )
{
    Code here
}
```

### The blur( ) function

- The “blur( )” function handles “blur” event.
- The “blur” event executes when the cursor goes out of the element.

#### Syntax:

```
$(“selector”).blur(functionname);
function functionname( )
{
    Code here
}
```

### Example on “focus” and “blur”

```
<html>
<head>
<title>jQuery - Focus and Blur</title>
<style type="text/css">
    body,input
    {
        font-family: Tahoma;
        font-size: 30px;
    }
    #div1
    {
        width: 300px;
        height: 200px;
        background-color: #00cccc;
    }
    #span1
    {
        color: green;
        font-weight: bold;
        display: none;
    }
</style>
</head>
<body>
    Email: <input type="text" id="txt1">
    <span id="span1">Use gmail only.</span>
```

```
<script src="jquery-3.2.1.js"></script>

<script>
    $("#txt1").focus(fun1);
    $("#txt1").blur(fun2);
    function fun1()
    {
        $("#span1").show();
    }
    function fun2()
    {
        $("#span1").hide();
    }
</script>
</body>
</html>
```

### The “keyup( )” function

- The “keyup( )” function handles “keyup” event.
- The “keyup” event executes when the user presses any key on the keyboard, while the cursor is inside the textbox.

#### **Syntax:**

```
$(“selector”).keyup(functionname);
function functionname( )
{
    Code here
}
```

### Example on “keyup”

---

```
<html>
<head>
    <title>jQuery - Keyup</title>
</head>
<body>
    Source text: <input type="text" id="txt1"><br>
    Destination text: <input type="text" id="txt2"><br>

    <script src="jquery-3.2.1.js"></script>

    <script>
        $("#txt1").keyup(fun1);
        function fun1()
        {
            $("#txt2").val( $("#txt1").val() );
        }
    </script>
```

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

### The “keypress( )” function

- The “keypress( )” function handles “keypress” event.
- The “keypress” event executes when the user presses any key on the keyboard, while the cursor is inside the textbox.



### Keypress (vs) Keyup

#### Keypress:

- Executes before placing the character into the textbox.
- We can accept / reject the currently pressed character.

#### Keyup:

- Executes after placing the character into the textbox.
- We can't reject the currently pressed character, because it is already accepted before calling the “keyup” event.

#### Syntax:

```

$("selector").keypress(functionname);
function functionname(event)
{
    event.which = ASCII value of currently pressed character.
    event.preventDefault() = Cancels the currently pressed character.
}
  
```

### Example on “Keypress”

```

<html>
  <head>
    <title>jQuery - Keypress</title>
    <style type="text/css">
      body,input
      {
        font-family: Tahoma;
        font-size: 30px;
      }
    </style>
  </head>
  <body>
  
```

Source text: `<input type="text" id="txt1"><br>`  
 Destination text: `<input type="text" id="txt2"><br>`  
`<script src="jquery-3.2.1.js"></script>`

```
<script>
    $("#txt1").keypress(fun1);
    function fun1()
    {
        $("#txt2").val($("#txt1").val());
    }
</script>
</body>
</html>
```

### The “change ( )” function

- The “change ( )” function handles “change” event.
- The “change” event executes when the user do following:
  1. Modify the value of a textbox and press TAB key
  2. Cheeck / uncheck the checkbox
  3. Select the radio btton.
  4. Select an item in the dropdownlist.

#### Syntax:

```
$(“selector”).change(functionname);
function functionname( )
{
}
}
```

### Example on “Change” with TextBox

```
<html>
<head>
    <title>jQuery - Change - TextBox</title>
    <style type="text/css">
        body,input,select
        {
            font-family: Tahoma;
            font-size: 30px;
        }
    </style>
</head>
<body>
    Source text: <input type="text" id="txt1"><br>
    Destination text: <input type="text" id="txt2"><br>
```

```
<script src="jquery-3.2.1.js"></script>

<script>
    $("#txt1").change(fun1);
    function fun1()
    {
        var s = $("#txt1").val();
        $("#txt2").val(s);
    }
</script>
</body>
</html>
```

### Example on “Change” with CheckBox

---

```
<html>
<head>
<title>jQuery - Change - Checkbox</title>
<style type="text/css">
    body,input,select
    {
        font-family: Tahoma;
        font-size: 30px;
    }
    #p1
    {
        font-weight: bold;
        font-size: 40px;
        color: darkblue;
    }
</style>
</head>
<body>
<input type="checkbox" id="checkbox1">
<label for="checkbox1">I accept license agreement</label>
<p id="p1"></p>

<script src="jquery-3.2.1.js"></script>

<script>
    $("#checkbox1").change(fun1);
    function fun1()
    {
        if ($("#checkbox1")[0].checked == true)
        {
            $("#p1").html("Accepted");
        }
        else
        {
            $("#p1").html("Not accepted");
        }
    }
</script>
</body>
```

```
</html>
```

### Example on “Change” with RadioButton

---

```
<html>
  <head>
    <title>jQuery - Change - RadioButton</title>
    <style type="text/css">
      body,input,select
      {
        font-family: Tahoma;
        font-size: 30px;
      }
      #p1
      {
        font-weight: bold;
        font-size: 40px;
        color: darkblue;
      }
    </style>
  </head>
  <body>
    Gender:
    <input type="radio" id="rb1" name="gender">
    <label for="rb1">Male</label>
    <input type="radio" id="rb2" name="gender">
    <label for="rb2">Female</label>
    <p id="p1"></p>

    <script src="jquery-3.2.1.js"></script>

    <script>
      $("#rb1").change(fun1);
      $("#rb2").change(fun1);
      function fun1()
      {
        if ($("#rb1")[0].checked == true)
        {
          $("#p1").html("Male selected");
        }
        else
        {
          $("#p1").html("Female selected");
        }
      }
    </script>
  </body>
</html>
```

### Example on “Change” with DropDownList

---

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



```

<title>jQuery - Change - Dropdownlist</title>
<style type="text/css">
    body,input,select
    {
        font-family: Tahoma;
        font-size: 30px;
    }
    #p1
    {
        font-weight: bold;
        font-size: 40px;
        color: darkblue;
    }
</style>
</head>
<body>
    Country:
    <select id="dropdownlist1">
        <option>Please select</option>
        <option>India</option>
        <option>China</option>
        <option>UK</option>
        <option>USA</option>
    </select>
    <p id="p1">para 1</p>

    <script src="jquery-3.2.1.js"></script>

    <script>
        $("#dropdownlist1").change(fun1);
        function fun1()
        {
            if ($("#dropdownlist1").val() == "Please select")
            {
                $("#p1").html( "You selected none" );
            }
            else
            {
                $("#p1").html( $("#dropdownlist1").val() );
            }
        }
    </script>
</body>
</html>

```

### The “on( )” function

- The “on( )” function handles any event. It attaches a function with an event; so that whenever the event occurs, the function will be called automatically.
- It supports all the events: click, dblclick, mouseover, mouseout, mousemove, focus, blur, keyup, keypress, change, contextmenu, cut, copy, paste etc.

#### • Advantages:

- You can use “off()” to unhandle the event.
- You can handle rare events such as contextmenu, cut, copy, paste etc.

### **Syntax:**

```
$(document).on("event", "selector", functionname);
function functionname( )
{
}
```

### **Example on “On”**

```
<html>
<head>
<title>jQuery - On</title>
<style type="text/css">
    body,input
    {
        font-family: Tahoma;
        font-size: 30px;
    }
    #div1
    {
        width: 300px;
        height: 200px;
        background-color: #00cccc;
    }
</style>
</head>
<body>
<div id="div1">div 1</div>
<script src="jquery-3.2.1.js"></script>
<script>
    $(document).on("click", "#div1", fun1);
    function fun1()
    {
        $("#div1").html("Thanx");
    }
</script>
</body>
</html>
```

### **The “off()” function**

- The “off()” function unhandles any event. It detaches a function from an event; so that whenever the event occurs, the function will not be called.

### **Syntax:**

`$(document).off("event", "selector");`

### Example on “Off”

---

```
<html>
  <head>
    <title>jQuery - Off</title>
    <style type="text/css">
      #div1
      {
        width: 300px;
        height: 200px;
        background-color: #00cccc;
      }
    </style>
  </head>
  <body>
    <div id="div1">div 1</div>

    <script src="jquery-3.2.1.js"></script>

    <script>
      $(document).on("click", "#div1", fun1);
      function fun1()
      {
        $("#div1").html("Thanx at " + new Date().toLocaleTimeString());
        $(document).off("click", "#div1");
      }
    </script>
  </body>
</html>
```

### The “contextmenu” event

- The “contextmenu” event executes when the user right clicks on an element.

#### **Syntax:**

```
$(document).on("contextmenu", document, functionname);

function functionname( )
{
}
}
```

### Example on “Contextmenu”

---

```
<html>
  <head>
    <title>jQuery - Contextmenu</title>
  </head>
```

```
<body>
  <input type="text" id="txt1" >
  <input type="text" id="txt2">
  <input type="text" id="txt3">

  <script src="jquery-3.2.1.js"></script>

  <script>
    $(document).on("contextmenu", document, fun1);
    function fun1(event)
    {
      alert("right click not allowed");
      event.preventDefault();
    }
    //event = browser given information
  </script>
</body>
</html>
```

### The “cut”, “copy”, “paste” events

- The “cut” event executes when the user selects “cut” option with keyboard / mouse.
- The “copy” event executes when the user selects “copy” option with keyboard / mouse.
- The “paste” event executes when the user selects “paste” option with keyboard / mouse.

#### **Syntax:**

```
$(document).on("cut", document, functionname);
function functionname( )
{
}

$(document).on("copy", document, functionname);
function functionname( )
{
}

$(document).on("paste", document, functionname);
function functionname( )
{
}
```

**Example on “Cut, copy, paste”**

---

```
<html>
  <head>
    <title>jQuery - Disabling cut, copy, paste</title>
    <style type="text/css">
      body,input
      {
        font-family: 'Tahoma';
        font-size: 30px;
      }
    </style>
  </head>
  <body>
    <input type="text" id="txt1">
    <input type="text" id="txt2">
    <input type="text" id="txt3">
    <script src="jquery-3.2.1.js"></script>
    <script>
      $(document).on("cut copy paste", "document", fun1);
      function fun1(event)
      {
        alert("cut copy paste not allowed");
        event.preventDefault(); //default functionality will be stopped
      }
      //event = browser given information
    </script>
  </body>
</html>
```

**The “data( )” function**

- The “data( )” function is used to store / retrieve data in the browser memory, temporarily (while the web page is running the browser).

**Syntax to set data:**

\$(“selector”).data(“key”, “value”);

**Syntax to get data:**

\$(“selector”).data(“key”);

**Example on “data( )”**

---

```
<html>
  <head>
    <title>jQuery - Data</title>
    <style type="text/css">
      body,input
      {
```

```

        font-family: Tahoma;
        font-size: 30px;
    }
</style>
</head>
<body>
    Username: <input type="text" id="txt1"><br>
    <input type="button" value="Set data into memory" id="button1">
    <input type="button" value="Get data from memory" id="button2">

    <script src="jquery-3.2.1.js"></script>

    <script>
        $("#button1").click(fun1);
        $("#button2").click(fun2);
        function fun1()
        {
            $("#txt1").data("x", $("#txt1").val());
            $("#txt1").val("");
            alert("Saved");
        }
        function fun2()
        {
            $("#txt1").val($("#txt1").data("x"));
        }
    </script>
</body>
</html>

```

## Effects

- jQuery effects are used to hide / show the elements smoothly.
- It is used for showing menus, validation messages etc.
- jQuery supports 3 types of effects:
  1. fade effect : fade out / fade in
  2. slide effect : slide up / slide down
  3. hide / show effect : hide / show

### 1. fade effect:

fadeOut(milli seconds);

fadeIn(milli seconds);

### 2. slide effect:

slideUp(milli seconds);

slideDown(milli seconds);

### 3. hide / show effect:

hide(milli seconds);

show(milli seconds);

**Note:** 1000 milli seconds = 1 second

### Example on Fade

---

```
<html>
  <head>
    <title>jQuery - Fade</title>
    <style type="text/css">
      body,input
      {
        font-family: 'Tahoma';
        font-size: 30px;
      }
    </style>
    </head>
    <body>
      <div id="div1">Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum is
      simply dummy text of the printing and typesetting industry.</div>
      <input type="button" id="button1" value="fade out">
      <input type="button" id="button2" value="fade in">

      <script src="jquery-3.2.1.js"></script>

      <script>
        $("#button1").click(fun1);
        $("#button2").click(fun2);
        function fun1()
        {
          $("#div1").fadeOut(1000); //1000 milli sec = 1 sec
        }
        function fun2()
        {
          $("#div1").fadeIn(1000);
        }
      </script>
    </body>
  </html>
```

### Example on Slide

---

```
<html>
  <head>
    <title>jQuery - Slide</title>
    <style type="text/css">
      body,input
      {
        font-family: 'Tahoma';
        font-size: 30px;
      }
    </style>
  </head>
  <body>
    <div id="div1">Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum is
    simply dummy text of the printing and typesetting industry.</div>
    <input type="button" id="button1" value="slide up">
    <input type="button" id="button2" value="slide down">
    <script src="jquery-3.2.1.js"></script>

    <script>
      $("#button1").click(fun1);
      $("#button2").click(fun2);
      function fun1()
      {
        $("#div1").slideUp(1000);
      }
      function fun2()
      {
        $("#div1").slideDown(1000);
      }
    </script>
  </body>
</html>
```

### Example on Hide / Show

---

```
<html>
  <head>
    <title>jQuery - Hide or Show</title>
    <style type="text/css">
      body,input
      {
        font-family: 'Tahoma';
        font-size: 30px;
      }
    </style>
  </head>
  <body>
    <div id="div1">
```



```

    {
        width: 400px;
        height: 300px;
        background-color: #ff6699;
        text-align: justify;
    }
</style>
</head>
<body>
    <div id="div1">Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum is
    simply dummy text of the printing and typesetting industry.</div>
    <input type="button" id="button1" value="hide">
    <input type="button" id="button2" value="show">
    <script src="jquery-3.2.1.js"></script>
    <script>
        $("#button1").click(fun1);
        $("#button2").click(fun2);
        function fun1()
        {
            $("#div1").hide(1000);
        }
        function fun2()
        {
            $("#div1").show(1000);
        }
    </script>
</body>
</html>

```

### toggle( )

- toggle( ) function hides / shows the element.
- It hides the element if it is already visible.
- It shows the element if it is already invisible.

fadeToggle(milli seconds) : fade out / fade in  
 slideToggle(milli seconds) : slide up / slide down  
 toggle(milli seconds) : hide / show

### Example on toggle( )

```

<html>
<head>
    <title>jQuery - Toggle</title>
    <style type="text/css">
        body,input
        {
            font-family: 'Tahoma';
            font-size: 30px;

```

```

    }
    #div1
    {
        width: 400px;
        height: 300px;
        background-color: #ff6699;
        text-align: justify;
    }
</style>
</head>
<body>
    <div id="div1">Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum is
    simply dummy text of the printing and typesetting industry.</div>
    <input type="button" id="button1" value="fade toggle">
    <input type="button" id="button2" value="slide toggle">
    <input type="button" id="button3" value="show/hide toggle">

    <script src="jquery-3.2.1.js"></script>

    <script>
        $("#button1").click(fun1);
        function fun1()
        {
            $("#div1").fadeToggle(1000); //fadeOut or fadeIn
        }

        $("#button2").click(fun2);
        function fun2()
        {
            $("#div1").slideToggle(1000); //slideUp or slideDown
        }

        $("#button3").click(fun3);
        function fun3()
        {
            $("#div1").toggle(1000); //hide or show
        }
    </script>
</body>
</html>

```

### “this” keyword

- “this” keyword represents the “current element”, which has raised the current event.
- **Syntax:** \$(this)

### Example on “this” keyword

```

<html>
<head>
    <title>jQuery - Fadeout with Click</title>
    <style type="text/css">

```

```

body,input
{
    font-family: 'Tahoma';
    font-size: 30px;
}
div.class1
{
    width: 140px;
    height: 100px;
    background-color: #00cc99;
    margin: 10px;
    float: left;
    color: #003399;
    font-size: 40px;
    padding: 5px;
    font-weight: bold;
}
</style>
</head>
<body>
<div id="div1" class="class1">1</div>
<div id="div2" class="class1">2</div>
<div id="div3" class="class1">3</div>
<div id="div4" class="class1">4</div>
<div id="div5" class="class1">5</div>
<script src="jquery-3.2.1.js"></script>
<script>
    $("div").click(fun1);
    function fun1()
    {
        $(this).fadeOut(700);
        //this = current element (div), which is currently clicked
    }
</script>
</body>
</html>

```

### fadeTo( )

- fadeTo( ) function is used to change the “opacity” property gradually, based on the given no. of milli seconds.
- **Syntax:** fadeTo(milli seconds, opacity);
- **Example:** fadeTo(2000, 0.6);

### Example on fadeTo( )

```

<html>
<head>
    <title>jQuery - fadeTo</title>
    <style type="text/css">
        body,input

```

```

    {
        font-family: 'Tahoma';
        font-size: 30px;
    }
    div.class1
    {
        width: 150px;
        height: 130px;
        background-color: #00cc99;
        margin: 10px;
        float: left;
        color: #003399;
        font-size: 40px;
        padding: 5px;
        font-weight: bold;
    }
</style>
</head>
<body>
    <div id="div1" class="class1">1</div>
    <div id="div2" class="class1">2</div>
    <div id="div3" class="class1">3</div>
    <div id="div4" class="class1">4</div>
    <div id="div5" class="class1">5</div>
    <script src="jquery-3.2.1.js"></script>
    <script>
        $("div").click(fun1);
        function fun1()
        {
            $(this).fadeOut(700, 0.4); //Syntax: fadeTo(milli sec, opacity)
        }
    </script>
</body>
</html>

```

### Example 2 on fadeTo( )

---

```

<html>
<head>
    <title>jQuery - FadeTo and FadeTo</title>
    <style type="text/css">
        body,input
        {
            font-family: 'Tahoma';
            font-size: 30px;
        }
        div.class1
        {
            width: 150px;
            height: 130px;
            background-color: #00cc99;
            margin: 10px;
            float: left;
            color: #003399;

```

```
        font-size: 40px;
        padding: 5px;
        font-weight: bold;
    }
</style>
</head>
<body>
    <div id="div1" class="class1">1</div>
    <div id="div2" class="class1">2</div>
    <div id="div3" class="class1">3</div>
    <div id="div4" class="class1">4</div>
    <div id="div5" class="class1">5</div>

    <script src="jquery-3.2.1.js"></script>

    <script>
        $("div").click(fun1);
        $("div").data("flag", "0");
        function fun1()
        {
            var n = $(this).data("flag");
            if (n == "0")
            {
                $(this).fadeTo(1000, 0.4); //milli sec, opacity
                $(this).data("flag", "1");
            }
            else
            {
                $(this).fadeTo(1000, 1); //milli sec, opacity
                $(this).data("flag", "0");
            }
        }
    </script>
</body>
</html>
```

## Manipulating Attributes

- We can manipulate attributes by using the following functions in jQuery.

### 1.Set an attribute:

- attr("attribute name", "value")

### 2.Get an attribute:

- attr("attribute name")

### 3.Removing an attribute:

- removeAttr("attribute name")

### 4.Setting multiple attributes:

- attr({ "attribute": "value", "attribute": "value", ... })

### Example on Set Attribute

---

```
<html>
  <head>
    <title>jQuery - Set Attribute</title>
    <style type="text/css">
      body,input
      {
        font-family: 'Tahoma';
        font-size: 30px;
      }
    </style>
  </head>
  <body>
    <br>
    <input type="button" id="button1" value="set attribute">
    <script src="jquery-3.2.1.js"></script>
    <script>
      $("#button1").click(fun1);
      function fun1()
      {
        $("#myImage").attr("src", "img2.jpg");
      }
    </script>
  </body>
</html>
```

Note: Place “img1.jpg” and “img2.jpg” in “c:\jquery” folder.

### Example on Set Multiple Attributes

---

```
<html>
  <head>
    <title>jQuery - Set Multiple Attributes</title>
    <style type="text/css">
      body,input
      {
        font-family: 'Tahoma';
        font-size: 30px;
      }
    </style>
  </head>
  <body>
    <br>
    <input type="button" id="button1" value="set multiple attributes">

    <script src="jquery-3.2.1.js"></script>

    <script>
      $("#button1").click(fun1);
      function fun1()
      {
```

```

        $("#myImage").attr({ "src": "img3.jpg", "title": "this is tooltip", "alt": "this is alternate text" });
    }
</script>
</body>
</html>

```

Note: Place "img1.jpg" and "img3.jpg" in "c:\jquery" folder.

### Example on Remove Attribute

```

<html>
<head>
<title>jQuery - Remove Attr</title>
<style type="text/css">
    body,input
    {
        font-family: 'Tahoma';
        font-size: 30px;
    }
</style>
</head>
<body>
<br>
<input type="button" id="button1" value="remove attribute">
<script src="jquery-3.2.1.js"></script>
<script>
    $("#button1").click(fun1);
    function fun1()
    {
        $("#myImage").removeAttr("title");
    }
</script>
</body>
</html>

```

### Example on Get Attribute

```

<html>
<head>
<title>jQuery - Get Attribute</title>
<style type="text/css">
    body,input
    {
        font-family: 'Tahoma';
        font-size: 30px;
    }
</style>
</head>
<body>
<br>
<input type="button" id="button1" value="get attribute">

<script src="jquery-3.2.1.js"></script>

```

```
<script>
    $("#button1").click(fun1);
    function fun1()
    {
        var s=$("#myImage").attr("src");
        alert(s);
    }
</script>
</body>
</html>
```

Note: Place "img1.jpg" in "c:\jquery" folder.

## Manipulating CSS

- We can manipulate css styles dynamically at run time by using the following jquery functions.

### 1.Adding CSS class to the element:

- `addClass("css class name")`

### 2.Removing CSS class to the element:

- `removeClass("css class name")`

### 3.Toggle (add / remove) CSS class:

- `toggleClass("css class name")`

### 4.Check whether the element has specific css class or not:

- `hasClass("css class name")`

### 5.Setting individual css properties using jquery:

- `css("property", "value")`

### 6.Getting the individual css properties using jquery:

- `css("property")`

### 7.Setting multiple individual css properties:

- `css({ "property": "value", "property": "value", ... })`

### Example on Add Class

---

```
<html>
<head>
<title>jQuery - AddClass</title>
<style type="text/css">
    body,input
    {
        font-family: 'Tahoma';
        font-size: 30px;
    }
    .class1
```



```
{
  background-color: darkred;
  color: cyan;
  font-size: 50px;
}
</style>
</head>
<body>
  <div id="div1">Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum is
  simply dummy text of the printing and typesetting industry.</div>
  <input type="button" id="button1" value="add class">
  <script src="jquery-3.2.1.js"></script>

  <script>
    $("#button1").click(fun1);
    function fun1()
    {
      $("#div1").addClass("class1");
    }
  </script>
</body>
</html>
```

### Example on Remove Class

---

```
<html>
<head>
  <title>jQuery - RemoveClass</title>
  <style type="text/css">
    body,input
    {
      font-family: 'Tahoma';
      font-size: 30px;
    }
    .class1
    {
      background-color: darkred;
      color: cyan;
    }
  </style>
</head>
<body>
  <div id="div1" class="class1">Lorem Ipsum is simply dummy text of the printing and typesetting industry.
  Lorem Ipsum is simply dummy text of the printing and typesetting industry.</div>
  <input type="button" id="button1" value="remove class">

  <script src="jquery-3.2.1.js"></script>

  <script>
    $("#button1").click(fun1);
    function fun1()
    {
      $("#div1").removeClass("class1");
    }
  </script>
```

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

### Example on Toggle Class

---

```
<html>
  <head>
    <title>jQuery - ToggleClass</title>
    <style type="text/css">
      body,input
      {
        font-family: 'Tahoma';
        font-size: 30px;
      }
      .class1
      {
        background-color: darkred;
        color: cyan;
      }
    </style>
  </head>
  <body>
    <div id="div1">Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum is
    simply dummy text of the printing and typesetting industry.</div>
    <input type="button" id="button1" value="toggle class">

    <script src="jquery-3.2.1.js"></script>

    <script>
      $("#button1").click(fun1);
      function fun1()
      {
        $("#div1").toggleClass("class1");
      }
    </script>
  </body>
</html>
```

### Example on Has Class

---

```
<html>
  <head>
    <title>jQuery - HasClass</title>
    <style type="text/css">
      body,input
      {
        font-family: 'Tahoma';
        font-size: 30px;
      }
      .class1
      {
        background-color: darkred;

```

```
        color: cyan;
    }
</style>
</head>
<body>
    <div id="div1" class="class1">Lorem Ipsum is simply dummy text of the printing and typesetting industry.
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.</div>
    <input type="button" id="button1" value="has class">

    <script src="jquery-3.2.1.js"></script>

    <script>
        $("#button1").click(fun1);
        function fun1()
        {
            var b = $("#div1").hasClass("class1");
            alert(b);
        }
    </script>
</body>
</html>
```

### Example on Get CSS

---

```
<html>
<head>
    <title>jQuery - Get CSS</title>
    <style type="text/css">
        body,input
        {
            font-family: 'Tahoma';
            font-size: 30px;
        }
        .class1
        {
            background-color: darkred;
            color: cyan;
        }
    </style>
</head>
<body>
    <div id="div1" class="class1">Lorem Ipsum is simply dummy text of the printing and typesetting industry.
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.</div>
    <input type="button" id="button1" value="get css property">

    <script src="jquery-3.2.1.js"></script>

    <script>
        $("#button1").click(fun1);
        function fun1()
        {
            var item = $("#div1").css("font-size");
            alert(item);
        }
    </script>
```

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

### Example on Set CSS

---

```
<html>
  <head>
    <title>jQuery - Set CSS</title>
    <style type="text/css">
      body,input
      {
        font-family: 'Tahoma';
        font-size: 30px;
      }
    </style>
  </head>
  <body>
    <div id="div1">Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum is
    simply dummy text of the printing and typesetting industry.</div>
    <input type="button" id="button1" value="set a css property">

    <script src="jquery-3.2.1.js"></script>

    <script>
      $("#button1").click(fun1);
      function fun1()
      {
        $("#div1").css("color", "#ff0099");
      }
    </script>
  </body>
</html>
```

### Example on Set Multiple CSS

---

```
<html>
  <head>
    <title>jQuery - Set CSS Multiple Properties</title>
    <style type="text/css">
      body,input
      {
        font-family: 'Tahoma';
        font-size: 30px;
      }
    </style>
  </head>
  <body>
    <div id="div1">Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum is
    simply dummy text of the printing and typesetting industry.</div>
    <input type="button" id="button1" value="set a css property">

    <script src="jquery-3.2.1.js"></script>
```

```
<script>
    $("#button1").click(fun1);
    function fun1()
    {
        $("#div1").css({ "color": "yellow", "background-color": "darkblue", "font-size": "30px", "font-weight":
"bold", "font-family": 'Tahoma' });
    }
</script>
</body>
</html>
```

## Animations

- jQuery Animations are used to change a css property's value gradually, based on the specified milli seconds.
- We can animate any pixels-based and color-based properties.

### Syntax:

animate( { "property": "value", "property": "value", ... }, milli seconds)

### Example on Animations - Width

---

```
<html>
<head>
<title>jQuery - Animations</title>
<style type="text/css">
    body,input
    {
        font-family: 'Tahoma';
        font-size: 30px;
    }
    #div1
    {
        width: 100px;
        height: 100px;
        background-color: #00cc66;
        border: 2px solid red;
        position: relative;
    }
</style>
</head>
<body>
<div id="div1">Hello World</div><br>
<input type="button" id="button1" value="animate width">

<script src="jquery-3.2.1.js"></script>

<script>
    $("#button1").click(fun1);
```

```
        function fun1()
        {
            $("#div1").animate({"width":"200px"},1000);
        }
    </script>
</body>
</html>
```

### Example on Animations - Height

---

```
<html>
<head>
<title>jQuery - Animations - Height</title>
<style type="text/css">
    body,input
    {
        font-family: 'Tahoma';
        font-size: 30px;
    }
    #div1
    {
        width: 100px;
        height: 100px;
        background-color: #00cc66;
        border: 2px solid red;
        position: relative;
    }
</style>
</head>
<body>
<div id="div1">Hello World</div><br>
<input type="button" id="button1" value="animate height">

<script src="jquery-3.2.1.js"></script>

<script>
    $("#button1").click(fun1);
    function fun1()
    {
        $("#div1").animate({"height":"200px"},1000);
    }
</script>
</body>
</html>
```

### Example on Animations – Width and Height

---

```
<html>
<head>
<title>jQuery - Animations - Width and Height</title>
<style type="text/css">
    body,input
    {
```

```
        font-family: 'Tahoma';
        font-size: 30px;
    }
    #div1
    {
        width: 100px;
        height: 100px;
        background-color: #00cc66;
        border: 2px solid red;
        position: relative;
    }
</style>
</head>
<body>
    <div id="div1">Hello World</div><br>
    <input type="button" id="button1" value="animate width and height">

    <script src="jquery-3.2.1.js"></script>
    <script>
        $("#button1").click(fun1);
        function fun1()
        {
            $("#div1").animate({width: "300px", "height": "300px"}, 1000);
        }
    </script>
</body>
</html>
```

### Example on Animations – Width and Then Height

---

```
<html>
<head>
    <title>jQuery - Animations</title>
    <style type="text/css">
        body,input
        {
            font-family: 'Tahoma';
            font-size: 30px;
        }
    </style>
    #div1
    {
        width: 100px;
        height: 100px;
        background-color: #00cc66;
        border: 2px solid red;
        position: relative;
    }
</head>
<body>
    <div id="div1">Hello World</div>
    <br>
    <input type="button" id="button1" value="animate width and then height">
```

```

<script src="jquery-3.2.1.js"></script>

<script>
    $("#button1").click(fun1);
    function fun1()
    {
        //jquery chaining (calling jquery functions in a sequence)
        $("#div1").animate({"width": "300px"}, 1000).animate({"height":
"300px"}, 1000).delay(1000).fadeOut(3000);
    }
</script>
</body>
</html>

```

### Example on Animations – Multiple Properties

---

```

<html>
<head>
<title>jQuery - Animations - Multiple Properties</title>
<style type="text/css">
    body,input
    {
        font-family: 'Tahoma';
        font-size: 30px;
    }
    #div1
    {
        width: 100px;
        height: 100px;
        background-color: #00cc66;
        border: 2px solid red;
        position: relative;
        left: 0px;
        top: 0px;
    }
</style>
</head>
<body>
<div id="div1">Hello World</div>
<br>
<input type="button" id="button1" value="animate multiple properties">
<script src="jquery-3.2.1.js"></script>

<script>
    $("#button1").click(fun1);
    function fun1()
    {
        $("#div1").animate({ "font-size": "50px", "border-width": "10px", "left": "30px", "top": "50px", "width":
"400px", "height": "400px" }, 1000);
    }
</script>
</body>
</html>

```



## Example on Animations – scrollTop

```
<html>
  <head>
    <title>jQuery - Animations - scrollTop</title>
    <style type="text/css">
      body,input
      {
        font-family: Tahoma;
        font-size: 30px;
      }
    </style>
  </head>
  <body>
    <a href="#" id="link1">Go to Heading 7</a>
```

```
    <h1 id="heading1">Heading 1</h1>
```

<p>If you have a Google Account, we may display your Profile name, Profile photo, and actions you take on Google or on third-party applications connected to your Google Account (such as +1's, reviews you write and comments you post) in our Services, including displaying in ads and other commercial contexts. We will respect the choices you make to limit sharing or visibility settings in your Google Account. For example, you can choose your settings so your name and photo do not appear in an ad.</p>

```
    <h1 id="heading2">Heading 2</h1>
```

<p>If you have a Google Account, we may display your Profile name, Profile photo, and actions you take on Google or on third-party applications connected to your Google Account (such as +1's, reviews you write and comments you post) in our Services, including displaying in ads and other commercial contexts. We will respect the choices you make to limit sharing or visibility settings in your Google Account. For example, you can choose your settings so your name and photo do not appear in an ad.</p>

```
    <h1 id="heading3">Heading 3</h1>
```

<p>If you have a Google Account, we may display your Profile name, Profile photo, and actions you take on Google or on third-party applications connected to your Google Account (such as +1's, reviews you write and comments you post) in our Services, including displaying in ads and other commercial contexts. We will respect the choices you make to limit sharing or visibility settings in your Google Account. For example, you can choose your settings so your name and photo do not appear in an ad.</p>

```
    <h1 id="heading4">Heading 4</h1>
```

<p>If you have a Google Account, we may display your Profile name, Profile photo, and actions you take on Google or on third-party applications connected to your Google Account (such as +1's, reviews you write and comments you post) in our Services, including displaying in ads and other commercial contexts. We will respect the choices you make to limit sharing or visibility settings in your Google Account. For example, you can choose your settings so your name and photo do not appear in an ad.</p>

```
    <h1 id="heading5">Heading 5</h1>
```

<p>If you have a Google Account, we may display your Profile name, Profile photo, and actions you take on Google or on third-party applications connected to your Google Account (such as +1's, reviews you write and comments you post) in our Services, including displaying in ads and other commercial contexts. We will respect the choices you make to limit sharing or visibility settings in your Google Account. For example, you can choose your settings so your name and photo do not appear in an ad.</p>

```
    <h1 id="heading6">Heading 6</h1>
```

<p>If you have a Google Account, we may display your Profile name, Profile photo, and actions you take on Google or on third-party applications connected to your Google Account (such as +1's, reviews you write and comments you post) in our Services, including displaying in ads and other commercial contexts. We will respect the choices you make to limit sharing or visibility settings in your Google Account. For example, you can choose your settings so your name and photo do not appear in an ad.</p>

```
    <h1 id="heading7">Heading 7</h1>
```

<p>If you have a Google Account, we may display your Profile name, Profile photo, and actions you take on Google or on third-party applications connected to your Google Account (such as +1's, reviews you write and

comments you post) in our Services, including displaying in ads and other commercial contexts. We will respect the choices you make to limit sharing or visibility settings in your Google Account. For example, you can choose your settings so your name and photo do not appear in an ad.</p>

<h1 id="heading8">Heading 8</h1>

<p>If you have a Google Account, we may display your Profile name, Profile photo, and actions you take on Google or on third-party applications connected to your Google Account (such as +1's, reviews you write and comments you post) in our Services, including displaying in ads and other commercial contexts. We will respect the choices you make to limit sharing or visibility settings in your Google Account. For example, you can choose your settings so your name and photo do not appear in an ad.</p>

<h1 id="heading9">Heading 9</h1>

<p>If you have a Google Account, we may display your Profile name, Profile photo, and actions you take on Google or on third-party applications connected to your Google Account (such as +1's, reviews you write and comments you post) in our Services, including displaying in ads and other commercial contexts. We will respect the choices you make to limit sharing or visibility settings in your Google Account. For example, you can choose your settings so your name and photo do not appear in an ad.</p>

<h1 id="heading10">Heading 10</h1>

<p>If you have a Google Account, we may display your Profile name, Profile photo, and actions you take on Google or on third-party applications connected to your Google Account (such as +1's, reviews you write and comments you post) in our Services, including displaying in ads and other commercial contexts. We will respect the choices you make to limit sharing or visibility settings in your Google Account. For example, you can choose your settings so your name and photo do not appear in an ad.</p>

<script src="jquery-3.2.1.js"></script>

<script>

\$("#link1").click(fun1);

function fun1()

{

\$("#body").animate({ "scrollTop": \$("#heading7").offset().top }, 1000);

}

</script>

</body>

</html>

## document.ready( )

- “document.ready( )” function is used to call a function automatically after the web page loading completed in the browser.

### Syntax 1:

\$(document).ready(functionname);

### Syntax 2:

\$(functionname);

### Syntax 3:

jQuery(functionname);

### Example on Document.Ready( )

```
<html>
  <head>
    <title>jQuery - Document.ready</title>
    <style type="text/css">
      body,input
      {
        font-family: 'Tahoma';
        font-size: 30px;
      }
    </style>

    <script src="jquery-3.2.1.js"></script>

    <script>
      //$(document).ready(fun1);
      $(fun1);
      //jQuery(fun1);
      function fun1()
      {
        alert("Page loaded");
      }
    </script>
  </head>
  <body>
    <h1>Page</h1>
    <div id="div1">div 1</div>
  </body>
</html>
```

## Selectors

- “Selector” is a syntax to select the desired elements in the web page.
- We have to select the elements, and then we can manipulate them.
- We have to pass the selector as argument to the \$( ) function. The \$( ) function receives the selector, searches the web page and returns the matching elements in the web page.

**Syntax:**        \$(“selector”)

**Example:**     \$(“p”)

### List of jQuery Selectors

1. Tag Selector
2. ID Selector
3. Class Selector

4. Compound Selector
5. Grouping Selector
6. Child Selector
7. Direct Child Selector
8. Adjacent Siblings Selector
9. Adjacent One Sibling Selector
10. :first filter
11. :last filter
12. :even filter
13. :odd filter
14. :eq filter
15. :gt filter
16. :lt filter
17. :not filter
18. Attribute Selector
19. Attribute Selector – Not
20. Attribute Selector – Starts With
21. Attribute Selector – Ends With
22. Attribute Selector – Contains
23. Contains
24. Has
25. Empty
26. :first-child filter
27. :last-child filter
28. :nth-child filter
29. :only-child filter
30. parent( )
31. next( )
32. prev( )
33. siblings( )
34. children( )
35. index( )
36. :input filter
37. :text filter

38. :password filter
39. :radio filter
40. :checkbox filter
41. :image filter
42. :file filter
43. :submit filter
44. :reset filter
45. :button filter
46. :text:disabled filter
47. :radio:checked filter
48. :checkbox:checked filter

### Tag Selector

- It selects all the instances of the specified tag.

**Syntax:** \$("tag")

**Example:** \$("p")

### Example on Tag Selector

```
<html>
  <head>
    <title>jQuery - Tag Selector</title>
  </head>
  <body>
    <p>para 1</p>
    <p>para 2</p>
    <p>para 3</p>
    <p>para 4</p>
    <p>para 5</p>
    <input type="button" value="Select" id="button1">

    <script src="jquery-3.2.1.js"></script>

    <script>
      $("#button1").click(fun1);
      function fun1()
      {
        $("p").css("background-color", "lightgreen"); //tag selector
        //It selects all <p> tags
        //Selector = Syntax to select
      }
    </script>
  </body>
```

</html>

## ID Selector

- It selects a single element based on the ID.

**Syntax:**     \$("#id")

**Example:**     \$("#p1")

### Example on ID Selector

---

```
<html>
  <head>
    <title>jQuery - ID Selector</title>
  </head>
  <body>
    <p>para 1</p>
    <p>para 2</p>
    <p id="p3">para 3</p>
    <p>para 4</p>
    <p>para 5</p>
    <input type="button" value="Select" id="button1">

    <script src="jquery-3.2.1.js"></script>

    <script>
      $("#button1").click(fun1);
      function fun1()
      {
        $("#p3").css("background-color", "lightgreen"); //id selector
        //It selects the <p> tag that has id="p3"
      }
    </script>
  </body>
</html>
```

## Class Selector

- It selects all the instances based on the class name.

**Syntax:**     \$(".class")

**Example:**     \$(".c1")

### Example on Class Selector

---

```
<html>
  <head>
    <title>jQuery - Class Selector</title>
  </head>
  <body>
```

```
<p class="c1">para 1</p>
<p>para 2</p>
<p class="c1">para 3</p>
<p class="c1">para 4</p>
<p>para 5</p>
<input type="button" value="Select" id="button1">

<script src="jquery-3.2.1.js"></script>

<script>
    $("#button1").click(fun1);
    function fun1()
    {
        $(".c1").css("background-color", "lightgreen"); //class selector
        //It selects the tags that have class="class1"
    }
</script>
</body>
</html>
```

### Example 2 on Class Selector

---

```
<html>
<head>
    <title>jQuery - Class Selector 2</title>
    <style type="text/css">
        body,input
        {
            font-family: 'Tahoma';
            font-size: 30px;
        }
    </style>
</head>
<body>
    <p class="c1">para 1</p>
    <p>para 2</p>
    <p class="c1">para 3</p>
    <p class="c1">para 4</p>
    <p>para 5</p>
    <div class="c1">div1</div>
    <div>div2</div>
    <div class="c1">div3</div>
    <div>div4</div>
    <input type="button" value="Select" id="button1">
    <script src="jquery-3.2.1.js"></script>
    <script>
        $("#button1").click(fun1);
        function fun1()
        {
            $(".c1").css("background-color", "lightgreen"); //class selector
            //It selects the <p> tags and <div> tags that have class="c1"
        }
    </script>
</body>
```

`</html>`

### Compound Selector

- It selects the element that has specified tag and specified class name.

**Syntax:**     `$("tag.class")`

**Example:**     `$("p.c1")`

### Example on Compound Selector

```
<html>
  <head>
    <title>jQuery - Compound Selector</title>
  </head>
  <body>
    <p class="c1">para 1</p>
    <p>para 2</p>
    <p class="c1">para 3</p>
    <p class="c1">para 4</p>
    <p>para 5</p>
    <div class="c1">div1</div>
    <div>div2</div>
    <div class="c1">div3</div>
    <div>div4</div>
    <input type="button" value="Select" id="button1">
    <script src="jquery-3.2.1.js"></script>

    <script>
      $("#button1").click(fun1);
      function fun1()
      {
        $("p.c1").css("background-color", "lightgreen");
      }
    </script>
  </body>
</html>
```

### Grouping Selector

- It selects all the specified group of tags.

**Syntax:**     `$("tag1,tag2,tag3,...")`

**Example:**     `$("div,p,h2,span")`

### Example on Grouping Selector

```
<html>
  <head>
    <title>jQuery - Grouping Selector</title>
```



```
<style type="text/css">
  body,input
  {
    font-family: 'Tahoma';
    font-size: 30px;
  }
  div
  {
    margin-bottom: 5px;
  }
</style>
</head>
<body>
  <p>para 1</p>
  <p>para 2</p>
  <div>div1</div>
  <div>div2</div>
  <div>div3</div>
  <div>div4</div>
  <span>span</span>
  <span>span</span>
  <span>span</span>
  <br>
  <input type="button" value="Select" id="button1">

  <script src="jquery-3.2.1.js"></script>

  <script>
    $("#button1").click(fun1);
    function fun1()
    {
      $("p,div,span").css("background-color", "lightgreen");//grouping selector
      //It selects all <div> tags, <p> tags and all <span> tags
    }
  </script>
</body>
</html>
```

### Child Selector (or) Descendent Selector

- It selects all the child tags of the specified parent tag.
- It selects all the child tags including grand children.

**Syntax:**     \$("parent child")

**Example:**     \$("div p")

### Example on Child Selector / Descendent Selector

---

```
<html>
  <head>
    <title>jQuery - Descendent Selector</title>
    <style type="text/css">
```

```
body,input
{
  font-family: 'Tahoma';
  font-size: 30px;
}
div
{
  margin-bottom: 5px;
}
</style>
</head>
<body>
  <div>
    <p>para 1</p>
    <p>para 2</p>
    <p>para 3</p>
  </div>
  <p>para 4</p>
  <p>para 5</p>
  <p>para 6</p>
  <input type="button" value="Select" id="button1">

  <script src="jquery-3.2.1.js"></script>

  <script>
    $("#button1").click(fun1);
    function fun1()
    {
      $("div p").css("background-color", "lightgreen"); //Descendent Selector or Child Selector
      //It selects all the <p> tags that are children of <div>
    }
  </script>
</body>
</html>
```

### Example 2 on Child Selector / Descendent Selector

---

```
<html>
  <head>
    <title>jQuery - Descendent Selector 2</title>
    <style type="text/css">
      body,input
      {
        font-family: 'Tahoma';
        font-size: 30px;
      }
      div
      {
        margin-bottom: 5px;
      }
    </style>
  </head>
  <body>
    <div>
```

```

<p>para 1</p>
<p>para 2</p>
<b>
  <p>para 3</p>
  <p>para 4</p>
</b>
</div>
<p>para 5</p>
<p>para 6</p>
<p>para 7</p>
<input type="button" value="Select" id="button1">
<script src="jquery-3.2.1.js"></script>
<script>
  $("#button1").click(fun1);
  function fun1()
  {
    $("div p").css("background-color", "lightgreen"); //Descendent Selector or Child Selector (including
grand children)
    //It selects all the <p> tags that are direct children or grand children of <div>
  }
</script>
</body>
</html>

```

### Direct Child Selector

- It selects all the child tags that are direct children of the specified parent tag.
- It selects the child tags excluding grand children.

**Syntax:** \$("parent>child")

**Example:** \$("div>p")

### Example on Direct Child Selector

```

<html>
<head>
<title>jQuery - Direct Child Selector</title>
<style type="text/css">
  body,input
  {
    font-family: 'Tahoma';
    font-size: 30px;
  }
  div
  {
    margin-bottom: 5px;
  }
</style>
</head>
<body>

```

```

<div>
  <p>para 1</p>
  <p>para 2</p>
  <p>para 3</p>
  <b>
    <p>para 4</p>
    <p>para 5</p>
  </b>
</div>
<p>para 6</p>
<p>para 7</p>
<p>para 7</p>
<input type="button" value="Select" id="button1">
<script src="jquery-3.2.1.js"></script>
<script>
  $("#button1").click(fun1);
  function fun1()
  {
    $("div>p").css("background-color", "lightgreen"); //Direct Child Selector (excluding grand children)
    //It selects all the <p> tags that are direct children of <div>
  }
</script>
</body>
</html>

```

### Adjacent Siblings Selector

- It selects all the sibling tags, which are adjacent to the current tag.

**Syntax:** \$("currenttag~siblingtag")

**Example:** \$("#p2~p")

### Example on Adjacent Siblings Selector

```

<html>
  <head>
    <title>jQuery - Adjacent Siblings Selector</title>
  </head>
  <body>
    <div id="div1">
      <p>para 1</p>
      <p id="p2">para 2</p>
      <p>para 3</p>
      <p>para 4</p>
      <p>para 5</p>
      <p>para 6</p>
    </div>
    <input type="button" value="Select" id="button1">

    <script src="jquery-3.2.1.js"></script>

    <script>

```

```
    $("#button1").click(fun1);  
    function fun1()  
    {  
        $("#p2~p").css("border", "10px solid lightgreen");  
    }  
    </script>  
    </body>  
</html>
```

**Note:** Place “img4.jpg”, “img5.jpg”, “img6.jpg” in the current folder (c:\jquery).

### Adjacent One Sibling Selector

- It selects the only one sibling element, which is immediate adjacent to the current element.

**Syntax:**     \$(“currenttag+siblingtag”)

**Example:**     \$(“#p2+p”)

### Example on Adjacent One Sibling Selector

---

```
<html>  
  <head>  
    <title>jQuery - Adjacent One Sibling Selector</title>  
  </head>  
  <body>  
    <div id="div1">  
      <p>para 1</p>  
      <p id="p2">para 2</p>  
      <p>para 3</p>  
      <p>para 4</p>  
      <p>para 5</p>  
      <p>para 6</p>  
    </div>  
    <input type="button" value="Select" id="button1">  
    <script src="jquery-3.2.1.js"></script>  
  
    <script>  
      $("#button1").click(fun1);  
      function fun1()  
      {  
          $("#p2+p").css("border", "10px solid lightgreen");  
      }  
    </script>  
  </body>  
</html>
```

**Note:** Place “img4.jpg”, “img5.jpg”, “img6.jpg” in the current folder (c:\jquery).

**:first filter**

- It selects the first element.

**Syntax:**     \$("tag:first")

**Example:**     \$("p:first")

**Example on :first filter**

---

```
<html>
  <head>
    <title>jQuery - First</title>
  </head>
  <body>
    <p>para 1</p>
    <p>para 2</p>
    <p>para 3</p>
    <p>para 4</p>
    <p>para 5</p>
    <p>para 6</p>
    <p>para 7</p>
    <p>para 8</p>
    <p>para 9</p>
    <p>para 10</p>
    <input type="button" value="Select" id="button1">
    <script src="jquery-3.2.1.js"></script>

    <script>
      $("#button1").click(fun1);
      function fun1()
      {
        $("p:first").css("background-color", "lightgreen");
      }
    </script>
  </body>
</html>
```

**:last filter**

- It selects the last element.

**Syntax:**     \$("tag:last")

**Example:**     \$("p:last")

**Example on :last filter**

---

```
<html>
  <head>
    <title>jQuery - Last</title>
  </head>
  <body>
```

```
<p>para 1</p>
<p>para 2</p>
<p>para 3</p>
<p>para 4</p>
<p>para 5</p>
<p>para 6</p>
<p>para 7</p>
<p>para 8</p>
<p>para 9</p>
<p>para 10</p>
<input type="button" value="Select" id="button1">
<script src="jquery-3.2.1.js"></script>

<script>
    $("#button1").click(fun1);
    function fun1()
    {
        $("p:last").css("background-color", "lightgreen");
    }
</script>
</body>
</html>
```

### **:even filter**

- It selects all the even elements (0, 2, 4, 6, ...) etc.

**Syntax:**     \$("tag:even")

**Example:**     \$("p:even")

### **Example on :even filter**

---

```
<html>
<head>
    <title>jQuery - Even</title>
</head>
<body>
    <p>para 1</p>
    <p>para 2</p>
    <p>para 3</p>
    <p>para 4</p>
    <p>para 5</p>
    <p>para 6</p>
    <p>para 7</p>
    <p>para 8</p>
    <p>para 9</p>
    <p>para 10</p>
    <input type="button" value="Select" id="button1">
    <script src="jquery-3.2.1.js"></script>

    <script>
        $("#button1").click(fun1);
```

```
function fun1()
{
    $("p:even").css("background-color", "lightgreen");
}
</script>
</body>
</html>
```

### :odd filter

- It selects all the odd elements (1, 3, 5, 7, ...) etc.

**Syntax:**     \$("tag:odd")

**Example:**     \$("p:odd")

### Example on :odd filter

---

```
<html>
<head>
    <title>jQuery - Odd</title>
</head>
<body>
    <p>para 1</p>
    <p>para 2</p>
    <p>para 3</p>
    <p>para 4</p>
    <p>para 5</p>
    <p>para 6</p>
    <p>para 7</p>
    <p>para 8</p>
    <p>para 9</p>
    <p>para 10</p>
    <input type="button" value="Select" id="button1">

    <script src="jquery-3.2.1.js"></script>

    <script>
        $("#button1").click(fun1);
        function fun1()
        {
            $("p:odd").css("background-color", "lightgreen");
        }
    </script>
</body>
</html>
```

### :eq filter

- It selects the single element, based on the specified index.

**Syntax:**     \$("tag:eq(n)")



**Example:**     \$("p:eq(n)")

### Example on :eq filter

```
<html>
  <head>
    <title>jQuery - Eq</title>
    <style type="text/css">
      body,input
      {
        font-family: 'Tahoma';
        font-size: 30px;
      }
    </style>
  </head>
  <body>
    <p>para 1</p>
    <p>para 2</p>
    <p>para 3</p>
    <p>para 4</p>
    <p>para 5</p>
    <p>para 6</p>
    <p>para 7</p>
    <p>para 8</p>
    <p>para 9</p>
    <p>para 10</p>
    <input type="button" value="Select" id="button1">
    <script src="jquery-3.2.1.js"></script>
    <script>
      $("#button1").click(fun1);
      function fun1()
      {
        $("p:eq(3)").css("background-color", "lightgreen"); //equal - filter

        //$("p").eq(3).css("background-color", "lightgreen"); //equal - function

        //It selects the <p> which index position is "3". Index starts from zero (0).
      }
    </script>
  </body>
</html>
```

### :gt filter

- It selects the elements where index is greater than specified index.

**Syntax:**     \$("tag:gt(n)")

**Example:**     \$("p:gt(n)")

**Example on :gt filter**

```

<html>
  <head>
    <title>jQuery - Gt</title>
    <style type="text/css">
      body,input
      {
        font-family: 'Tahoma';
        font-size: 30px;
      }
    </style>
  </head>
  <body>
    <p>para 1</p>
    <p>para 2</p>
    <p>para 3</p>
    <p>para 4</p>
    <p>para 5</p>
    <p>para 6</p>
    <p>para 7</p>
    <p>para 8</p>
    <p>para 9</p>
    <p>para 10</p>
    <input type="button" value="Select" id="button1">

    <script src="jquery-3.2.1.js"></script>

    <script>
      $("#button1").click(fun1);
      function fun1()
      {
        $("p:gt(3)").css("background-color", "lightgreen"); //greater than - filter
        //It selects the <p> which index position is greater than "3". Index starts from zero (0).
      }
    </script>
  </body>
</html>

```

**:lt filter**

- It selects the elements where index is less than specified index.

**Syntax:**     \$("tag:lt(n)")

**Example:**     \$("p:lt(n)")

**Example on :lt filter**

```

<html>
  <head>
    <title>jQuery - Lt</title>
    <style type="text/css">
      body,input

```

```

    {
        font-family: 'Tahoma';
        font-size: 30px;
    }
</style>
</head>
<body>
    <p>para 1</p>
    <p>para 2</p>
    <p>para 3</p>
    <p>para 4</p>
    <p>para 5</p>
    <p>para 6</p>
    <p>para 7</p>
    <p>para 8</p>
    <p>para 9</p>
    <p>para 10</p>
    <input type="button" value="Select" id="button1">

    <script src="jquery-3.2.1.js"></script>

    <script>
        $("#button1").click(fun1);
        function fun1()
        {
            $("p:lt(3)").css("background-color", "lightgreen"); //less than - filter
            //It selects the <p> which index position is less than "3". Index starts from zero (0).
        }
    </script>
</body>
</html>

```

### :not filter

- It selects all the tags, except the specified tags.

**Syntax:** \$("selector:not(another selector)")

**Example:** \$("p:not(#p3)")

### Example on :not filter

```

<html>
<head>
    <title>jQuery - Not</title>
    <style type="text/css">
        body,input
        {
            font-family: 'Tahoma';
            font-size: 30px;
        }
    </style>
</head>

```

```

<body>
  <p>para 1</p>
  <p>para 2</p>
  <p>para 3</p>
  <p>para 4</p>
  <p>para 5</p>
  <p>para 6</p>
  <p>para 7</p>
  <p>para 8</p>
  <p>para 9</p>
  <p>para 10</p>
  <input type="button" value="Select" id="button1">
  <script src="jquery-3.2.1.js"></script>
  <script>
    $("#button1").click(fun1);
    function fun1()
    {
      $("p:not(p:eq(3))").css("background-color", "lightgreen"); //not - filter
      //It selects the <p> which index position is not equal to "3".

      //$("p:eq(3),p:eq(5)").css("background-color", "lightgreen");
      //It selects the <p> which index position is equal to "3" and "5".

      //$("p:not(p:eq(3),p:eq(5))").css("background-color", "lightgreen"); //not - filter
      //It selects the <p> which index position is not equal to "3" and "5".
    }
  </script>
</body>
</html>

```

### Attribute Selector

- It selects all the elements that have specified attribute.

**Syntax:**     \$("tag[attribute='value']")

**Example:**     \$("img[src='img1.jpg']")

### Example on Attribute Selector

```

<html>
  <head>
    <title>jQuery - Attribute Selector</title>
    <style type="text/css">
      body,input
      {
        font-family: 'Tahoma';
        font-size: 30px;
      }
    </style>
  </head>
  <body>
    

```

```





<br>
<input type="button" value="Select" id="button1">

<script src="jquery-3.2.1.js"></script>

<script>
    $("#button1").click(fun1);
    function fun1()
    {
        $("img[width='100px']").css("border", "4px solid red"); //attribute selector
        //It selects the <img> tag that have an attribute called width="100px"
    }
</script>
</body>
</html>
```

**Note:** Place “img1.jpg”, “img2.jpg”, “img3.jpg”, “img4.jpg”, “img5.jpg”, “img6.jpg” in the current folder (c:\jquery).

### Attribute Selector - Not

- It selects all the elements that are having the specified attribute, where the value is not equal to the specified value.

**Syntax:**     \$ (“tag[attribute!=’value’]”)

**Example:**     \$ (“img[src!=’img1.jpg’]”)

### Example on Attribute Selector - Not

---

```
<html>
<head>
    <title>jQuery - Attribute Selector - Not</title>
    <style type="text/css">
        body,input
        {
            font-family: 'Tahoma';
            font-size: 30px;
        }
    </style>
</head>
<body>
    
    
    
    
```

```



<br>
<input type="button" value="Select" id="button1">

<script src="jquery-3.2.1.js"></script>

<script>
    $("#button1").click(fun1);
    function fun1()
    {
        $("img[width!=100px]").css("border", "4px solid red"); //attribute selector - not
        //$("img:not(img[width=100px])").css("border", "4px solid red"); //attribute selector - not
        //It selects the <img> tag that width is not equal to "100px"
    }
</script>
</body>
</html>

```

**Note:** Place “img1.jpg”, “img2.jpg”, “img3.jpg”, “img4.jpg”, “img5.jpg”, “img6.jpg” in the current folder (c:\jquery).

### Attribute Selector – Starts With

- It selects all the elements that have specified attribute, where the value starts with specified value.

**Syntax:**     \$ (“tag[attribute^=‘value’]”)

**Example:**     \$ (“img[src^=‘i’]”)

### Example on Attribute Selector – Starts With

```

<html>
<head>
<title>jQuery - Attribute Selector - Starts With</title>
<style type="text/css">
    body,input
    {
        font-family: 'Tahoma';
        font-size: 30px;
    }
</style>
</head>
<body>
    
    
    
    
    
    
    
    
    <br>

```

```

<input type="button" value="Select" id="button1">

<script src="jquery-3.2.1.js"></script>

<script>
    $("#button1").click(fun1);
    function fun1()
    {
        $("img[src^='wa']").css("border", "4px solid red"); //attribute selector - starts with
        //It selects <img> tag that has "src" attribute that starts with "wa".
    }
</script>
</body>
</html>

```

**Note:** Place “img1.jpg”, “img2.jpg”, “img3.jpg”, “img4.jpg”, “img5.jpg”, “img6.jpg”, “Spring.jpg”, “Water.jpg” in the current folder (c:\jquery).

### Attribute Selector – Ends With

- It selects all the elements that have specified attribute, where the value ends with specified value.

**Syntax:** \$("tag[attribute\$='value']")

**Example:** \$("img[src\$='jpg']")

### Example on Attribute Selector – Ends With

```

<html>
<head>
<title>jQuery - Attribute Selector - Ends With</title>
<style type="text/css">
    body,input
    {
        font-family: 'Tahoma';
        font-size: 30px;
    }
</style>
</head>
<body>










<br>
<input type="button" value="Select" id="button1">

```

```

<script src="jquery-3.2.1.js"></script>

<script>
    $("#button1").click(fun1);
    function fun1()
    {
        $("img[src$='.jpg']").css("border", "4px solid red"); //attribute selector - ends with
        //It selects <img> tag that has "src" attribute that ends with ".jpg".
    }
</script>
</body>
</html>

```

**Note:** Place “img1.jpg”, “img2.jpg”, “img3.jpg”, “img4.jpg”, “img5.jpg”, “img6.jpg”, “Spring.jpg”, “Water.jpg”, “apple.gif”, “earth.gif” in the current folder (c:\jquery).

### Attribute Selector – Contains

- It selects all the elements that have specified attribute, where the value contains specified value.

**Syntax:**            \$(“tag[attribute\*='value']”)

**Example:**            \$(“img[src\*=j]”)

### Example on Attribute Selector – Contains

```

<html>
<head>
<title>jQuery - Attribute Selector - Contains</title>
<style type="text/css">
    body,input
    {
        font-family: 'Tahoma';
        font-size: 30px;
    }
</style>
</head>
<body>










<br>
<input type="button" value="Select" id="button1">

<script src="jquery-3.2.1.js"></script>

```



```

<script>
    $("#button1").click(fun1);
    function fun1()
    {
        $("img[src*='5']").css("border", "4px solid red"); //attribute selector - contains
        //It selects <img> tag that has "src" attribute that contains "5".

        //$("img[src*='5'],img[src*='2']").css("border", "4px solid red"); //attribute selector - contains
        //It selects <img> tag that has "src" attribute that contains "5" or "2".
    }
</script>
</body>
</html>

```

**Note:** Place “img1.jpg”, “img2.jpg”, “img3.jpg”, “img4.jpg”, “img5.jpg”, “img6.jpg”, “Spring.jpg”, “Water.jpg”, “apple.gif”, “earth.gif” in the current folder (c:\jquery).

### :contains filter

- It selects the elements that have inner html that contains specified value.

**Syntax:**            \$("tag:contains('value')")

**Example:**            \$("p:contains('Services')")

### Example on Contains Filter

```

<html>
<head>
<title>jQuery - Contains</title>
<style type="text/css">
    body,input
    {
        font-family: 'Tahoma';
        font-size: 30px;
    }
</style>
</head>
<body>
<p>Lorem Ipsum is simply dummy text of the printing.</p>
<p>It is a long established fact that a reader will be distracted by the readable content of a page when
looking at its layout.</p>
<p>Contrary to popular belief, Lorem Ipsum is not simply random text.</p>
<p>It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old.</p>
<p>If you are going to use a passage of Lorem Ipsum, you need to be sure there isn't anything embarrassing
hidden in the middle of text.</p>
<p>The generated Lorem Ipsum is therefore always free from repetition, injected humour, or non-
characteristic words etc.</p>
<p>Various versions have over the years, sometimes by accident, sometimes on purpose.</p>
<p>There are many variations of passages of Lorem Ipsum available, but the majority have suffered
alteration in some form, by injected humour, or randomised words which don't look even slightly believable.</p>

```

<p>If you are going to use a passage of Lorem Ipsum, you need to be sure there isn't anything embarrassing hidden in the middle of text.</p>

<p>All the Lorem Ipsum generators on the Internet tend to repeat predefined chunks as necessary, making this the first true generator on the Internet.</p>

<input type="button" value="Select" id="button1">

<script src="jquery-3.2.1.js"></script>

<script>

\$("#button1").click(fun1);

function fun1()

{

\$("#p:contains('you')").css("background-color", "lightgreen"); //contains

//It selects <p> that has content "you"

}

</script>

</body>

</html>

## :has filter

- It selects the elements that have child elements that matches with specified selector.

**Syntax:** \$("tag:has('selector')")

**Example:** \$("p:has('span')")

## Example on Has Filter

<html>

<head>

<title>jQuery - Has</title>

<style type="text/css">

body,input

{

font-family: 'Tahoma';

font-size: 30px;

}

</style>

</head>

<body>

<p>Lorem Ipsum is simply dummy text of the printing.</p>

<p>It is a long established fact that a reader <span>will be distracted</span> by the readable content of a page when looking at its layout.</p>

<p>Contrary to popular belief, Lorem Ipsum is not simply random text.</p>

<p>It has roots in a piece of classical Latin literature from <span>45 BC</span>, making it over 2000 years old.</p>

<p>If you are going to use a passage of <span>Lorem Ipsum</span>, you need to be sure there isn't anything embarrassing hidden in the middle of text.</p>

<p>The generated Lorem Ipsum is therefore always free from repetition, injected humour, or non-characteristic words etc.</p>

<p>Various versions have over the years, sometimes by accident, sometimes on purpose.</p>

<p>There are many variations of passages of <span>Lorem Ipsum available</span>, but the majority have suffered alteration in some form, by injected humour, or randomised words which don't look even slightly believable.</p>

<p>If you are going to use a passage of Lorem Ipsum, you need to be sure there isn't anything embarrassing hidden in the middle of text.</p>

<p>All the Lorem Ipsum generators on the Internet tend to repeat predefined chunks as necessary, making this the first true generator on the Internet.</p>

<input type="button" value="Select" id="button1">

<script src="jquery-3.2.1.js"></script>

<script>

\$("#button1").click(fun1);

function fun1()

{

\$("#p:has('span')").css("background-color", "lightgreen"); //has

//It selects the <p> tags that have a child called <span>

}

</script>

</body>

</html>

## :empty filter

- It selects the elements where inner html is empty.

**Syntax:** \$("tag:empty")

**Example:** \$("p:empty")

## Example on Empty Filter

<html>

<head>

<title>jQuery - Empty</title>

<style type="text/css">

body,input

{

font-family: 'Tahoma';

font-size: 30px;

}

p

{

padding: 5px;

}

</style>

</head>

<body>

<p>Lorem Ipsum is simply dummy text of the printing.</p>

<p>It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout.</p>

<p></p>

<p>It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old.</p>

<p>If you are going to use a passage of Lorem Ipsum, you need to be sure there isn't anything embarrassing hidden in the middle of text.</p>

<p></p>

<p>Various versions have over the years, sometimes by accident, sometimes on purpose.</p>

<p>There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration in some form, by injected humour, or randomised words which don't look even slightly believable.</p>

<p></p>

<p>All the Lorem Ipsum generators on the Internet tend to repeat predefined chunks as necessary, making this the first true generator on the Internet.</p>

<input type="button" value="Select" id="button1">

<script src="jquery-3.2.1.js"></script>

<script>

\$("#button1").click(fun1);

function fun1()

{

\$("#p:empty").css("border", "3px solid red"); //empty - filter

//It selects the empty <p> tags

}

</script>

</body>

</html>

## :first-child filter

- It selects all the elements that are first child of its parent.

**Syntax:** \$("tag:first-child")

**Example:** \$("p:first-child")

## Example on :first-child Filter

```
<html>
<head>
<title>jQuery - First-Child</title>
<style type="text/css">
body,input
{
font-family: 'Tahoma';
font-size: 30px;
}
#div1
{
background-color: #ffccff;
}
#div2
{
background-color: #ff9966;
}
#div3
{
```

```

        background-color: #99ffcc;
    }
</style>
</head>
<body>
    <div id="div1">
        <p>para 1</p>
        <p>para 2</p>
        <p>para 3</p>
        <p>para 4</p>
        <p>para 5</p>
    </div>
    <div id="div2">
        <p>para 6</p>
        <p>para 7</p>
        <p>para 8</p>
        <p>para 9</p>
        <p>para 10</p>
    </div>
    <div id="div3">
        <p>para 11</p>
        <p>para 12</p>
        <p>para 13</p>
        <p>para 14</p>
        <p>para 15</p>
    </div>
    <input type="button" value="Select" id="button1">
    <script src="jquery-3.2.1.js"></script>
    <script>
        $("#button1").click(fun1);
        function fun1()
        {
            $("p:first-child").css("background-color", "lightgreen"); //first-child
            //It selects the <p> tag, which is the first child or its parent.
        }
    </script>
</body>
</html>

```

### :last-child filter

- It selects all the elements that are last child of its parent.

**Syntax:**        \$("tag:last-child")

**Example:**        \$("p:last-child")

### Example on :last-child Filter

```

<html>
<head>
    <title>jQuery - Last-Child</title>
    <style type="text/css">

```

```
body,input
{
    font-family:'Tahoma';
    font-size:30px;
}
#div1
{
    background-color:#ffccff;
}
#div2
{
    background-color:#ff9966;
}
#div3
{
    background-color:#99ffcc;
}
</style>
</head>
<body>
    <div id="div1">
        <p>para 1</p>
        <p>para 2</p>
        <p>para 3</p>
        <p>para 4</p>
        <p>para 5</p>
    </div>
    <div id="div2">
        <p>para 6</p>
        <p>para 7</p>
        <p>para 8</p>
        <p>para 9</p>
        <p>para 10</p>
    </div>
    <div id="div3">
        <p>para 11</p>
        <p>para 12</p>
        <p>para 13</p>
        <p>para 14</p>
        <p>para 15</p>
    </div>
    <input type="button" value="Select" id="button1">

    <script src="jquery-3.2.1.js"></script>

    <script>
        $("#button1").click(fun1);
        function fun1()
        {
            $("p:last-child").css("background-color", "lightgreen"); //last-child
            //It selects the <p> tag, which is the last child or its parent.
        }
    </script>
</body>
</html>
```

**:nth-child filter**

- It selects all the elements that are  $n^{\text{th}}$  child of its parent.

**Syntax:**        `$(“tag:nth-child(n)”)`

**Example:**        `$(“p:nth-child(2)”)`

**Note:** Index starts from ‘1’.

**Example on :nth-child Filter**

---

```
<html>
  <head>
    <title>jQuery - Nth-Child</title>
    <style type="text/css">
      body,input
      {
        font-family: 'Tahoma';
        font-size: 30px;
      }
      #div1
      {
        background-color: #ffccff;
      }
      #div2
      {
        background-color: #ff9966;
      }
      #div3
      {
        background-color: #99ffcc;
      }
    </style>
  </head>
  <body>
    <div id="div1">
      <p>para 1</p>
      <p>para 2</p>
      <p>para 3</p>
      <p>para 4</p>
      <p>para 5</p>
    </div>
    <div id="div2">
      <p>para 6</p>
      <p>para 7</p>
      <p>para 8</p>
      <p>para 9</p>
      <p>para 10</p>
    </div>
    <div id="div3">
```

```

    <p>para 11</p>
    <p>para 12</p>
    <p>para 13</p>
    <p>para 14</p>
    <p>para 15</p>
  </div>
  <input type="button" value="Select" id="button1">

  <script src="jquery-3.2.1.js"></script>

  <script>
    $("#button1").click(fun1);
    function fun1()
    {
      $("p:nth-child(2)").css("background-color", "lightgreen"); //nth-child. Index starts from '1'.
      //It selects the <p> tag, which index position is "2" in its parent.
    }
  </script>
</body>
</html>

```

### :only-child filter

- It selects the child element, which is only one child of its parent.

**Syntax:**        \$(“tag:only-child”)

**Example:**        \$(“p:only-child”)

### Example on :only-child Filter

```

<html>
  <head>
    <title>jQuery - Only-child</title>
    <style type="text/css">
      #div1
      {
        background-color: #ffccff;
      }
      #div2
      {
        background-color: #ff9966;
      }
      #div3
      {
        background-color: #99ffcc;
      }
      #div4
      {
        background-color: #3399cc;
      }
      #div5
      {

```



```

        background-color: #ffff99;
    }
</style>
</head>
<body>
    <div id="div1">
        <p>para 1</p>
        <p>para 2</p>
        <p>para 3</p>
        <p>para 4</p>
    </div>
    <div id="div2">
        <p>para 5</p>
    </div>
    <div id="div3">
        <p>para 6</p>
        <p>para 7</p>
    </div>
    <div id="div4">
        <p>para 8</p>
        <p>para 9</p>
    </div>
    <div id="div5">
        <p>para 10</p>
    </div>
    <div id="div6">
        <p>para 11</p>
        <p>para 12</p>
        <p>para 13</p>
        <p>para 14</p>
        <p>para 15</p>
    </div>
    <input type="button" value="Select" id="button1">
    <script src="jquery-3.2.1.js"></script>
    <script>
        $("#button1").click(fun1);
        function fun1()
        {
            $("p:only-child").css("background-color", "lightgreen"); //only-child
            //It selects the <p> tag that is only one child of its parent.
        }
    </script>
</body>
</html>

```

### parent( )

- It selects the parent element of the current element.

**Syntax:** parent( )

**Example:** \$("#p1").parent( )

**Example on parent( )**

---

```
<html>
  <head>
    <title>jQuery - Parent</title>
  </head>
  <body>
    <div id="div1">
      <p>para 1</p>
      <p id="p2">para 2</p>
      <p>para 3</p>
      <p>para 4</p>
    </div>
    <input type="button" value="Select" id="button1">

    <script src="jquery-3.2.1.js"></script>

    <script>
      $("#button1").click(fun1);
      function fun1()
      {
        $("#p2").parent().css("background-color", "lightgreen");
        //The "parent()" function returns the parent tag of "#p1"
      }
    </script>
  </body>
</html>
```

**next( )**

- It selects the next element of the current element.

**Syntax:**            next( )

**Example:**            \$("#p1").next( )

**Example on next( )**

---

```
<html>
  <head>
    <title>jQuery - Next</title>
  </head>
  <body>
    <div id="div1">
      <p>para 1</p>
      <p id="p2">para 2</p>
      <p>para 3</p>
      <p>para 4</p>
    </div>
    <input type="button" value="Select" id="button1">

    <script src="jquery-3.2.1.js"></script>

    <script>
```

```
$("#button1").click(fun1);
function fun1()
{
    $("#p2").next().css("background-color", "lightgreen");
    //The "next()" function returns next tag, which is present after "#p2".
}
</script>
</body>
</html>
```

### prev( )

- It selects the previous element of the current element.

**Syntax:** prev( )

**Example:** \$("#p2").prev( )

### Example on prev( )

```
<html>
<head>
<title>jQuery - Prev</title>
</head>
<body>
<div id="div1">
<p>para 1</p>
<p>para 2</p>
<p id="p3">para 3</p>
<p>para 4</p>
<p>para 5</p>
</div>
<input type="button" value="Select" id="button1">
<script src="jquery-3.2.1.js"></script>
<script>
    $("#button1").click(fun1);
    function fun1()
    {
        $("#p3").prev().css("background-color", "lightgreen");
        //The "prev()" function returns the previous tag, which is present before "#p3".
    }
</script>
</body>
</html>
```

### siblings( )

- It selects all the sibling elements of the current element.

**Syntax:** siblings( )

**Example:** \$("#p2").siblings( )

### Example on siblings( )

---

```
<html>
  <head>
    <title>jQuery - Siblings</title>
  </head>
  <body>
    <div id="div1">
      <p>para 1</p>
      <p>para 2</p>
      <p id="p3">para 3</p>
      <p>para 4</p>
      <p>para 5</p>
    </div>
    <p>para 6</p>
    <p>para 7</p>
    <p>para 8</p>
    <input type="button" value="Select" id="button1">

    <script src="jquery-3.2.1.js"></script>

    <script>
      $("#button1").click(fun1);
      function fun1()
      {
        $("#p3").siblings().css("background-color", "lightgreen");
        //It selects all tags that are siblings (brothers) of "#p1".
      }
    </script>
  </body>
</html>
```

### children( )

- It selects all the child tags of the specified parent tag.

**Syntax:** children( )

**Example:** \$("#div1").children( )

### Example on children( )

---

```
<html>
  <head>
    <title>jQuery - Children</title>
  </head>
```

```

<body>
  <div id="div1">
    <p>para 1</p>
    <p>para 2</p>
    <p>para 3</p>
    <p>para 4</p>
    <p>para 5</p>
  </div>
  <p>para 6</p>
  <p>para 7</p>
  <p>para 8</p>
  <input type="button" value="Select" id="button1">
  <script src="jquery-3.2.1.js"></script>
  <script>
    $("#button1").click(fun1);
    function fun1()
    {
      $("#div1").children().css("background-color", "lightgreen");
      //It selects all tags that are children of "#div1".
    }
  </script>
</body>
</html>

```

## index( )

- It returns the index of the current element.

**Syntax:**        index( )

**Example:**        \$("#p2").index( )

## Example on index( )

```

<html>
  <head>
    <title>jQuery - Index</title>
  </head>
  <body>
    <p>para 1</p>
    <p>para 2</p>
    <p>para 3</p>
    <p id="p4">para 4</p>
    <p>para 5</p>
    <p>para 6</p>
    <p>para 7</p>
    <p>para 8</p>
    <input type="button" value="Index" id="button1">

    <script src="jquery-3.2.1.js"></script>

```

```
<script>
    $("#button1").click(fun1);
    function fun1()
    {
        var n = $("#p4").index();
        alert(n);
    }
</script>
</body>
</html>
```

## Form Filters

- Form filters are used to select the various form elements, such as textboxes, checkbox, radio buttons, dropdownlists etc.
  - `:input` It selects all `<input>`, `<textarea>` and `<select>` tags.
  - `:text` It selects all `<input type="text">`
  - `:password` It selects all `<input type="password">`
  - `:submit` It selects all `<input type="submit">`
  - `:reset` It selects all `<input type="reset">`
  - `:checkbox` It selects all `<input type="checkbox">`
  - `:radio` It selects all `<input type="radio">`
  - `:file` It selects all `<input type="file">`
  - `:image` It selects all `<input type="image">`
  - `:hidden` It selects all `<input type="hidden">`
  - `:button` It selects all `<input type="button">`
  - `:checkbox:checked` It selects all `<input type="checkbox">` that are currently checked
  - `:radio:checked` It selects all `<input type="radio">` that are currently checked
  - `:text:enabled` It selects all `<input type="text">` that are currently enabled
  - `:text:disabled` It selects all `<input type="text">` that are currently disabled

## Example on Form Filters

```
<html>
<head>
    <title>jQuery - Form Filters</title>
    <style type="text/css">
        body,input,textarea,select,table
        {
            font-family: 'Tahoma';
            font-size: 25px;
        }
    </style>
</head>
</html>
```

```

</style>
</head>
<body>
  <h2>Form Filters</h2>
  <form action="server.aspx" method="post" name="frm">
    <table>
      <tr>
        <td>first name</td>
        <td>:</td>
        <td colspan="4"><input type="text" name="first name" class="color" size="35"></td>
      </tr>
      <tr>
        <td>middle name</td>
        <td>:</td>
        <td colspan="4"><input type="text" name="middle name" class="color" size="35"></td>
      </tr>
      <tr>
        <td>last name</td>
        <td>:</td>
        <td colspan="4"><input type="text" name="last name" class="color" size="35"
disabled="disabled"></td>
      </tr>
      <tr>
        <td>password</td>
        <td>:</td>
        <td colspan="4"><input type="password" name="password" class="color" size="35"></td>
      </tr>
      <tr>
        <td>date of birth</td>
        <td>:</td>
        <td>
          <select class="color" name="month">
            <option value="na">Month</option>
            <option value="1">January</option>
            <option value="2">February</option>
            <option value="3">March</option>
            <option value="4">April</option>
            <option value="5">May</option>
            <option value="6">June</option>
            <option value="7">July</option>
            <option value="8">August</option>
            <option value="9">September</option>
            <option value="10">October</option>
            <option value="11">November</option>
            <option value="12">December</option>
          </select>
        </td>
        <td>
          <select name="day" id="day" class="color">
            <option value="na">Day</option>
          </select>
        </td>
        <td>
          <select name="year" class="color">
            <option value="na">Year</option>
          </select>
        </td>
      </tr>
    </table>
  </form>

```

```

        <option value="2015">2018</option>
    <option value="2015">2017</option>
    <option value="2015">2016</option>
    <option value="2015">2015</option>
        <option value="2014">2014</option>
        <option value="2013">2013</option>
        <option value="2012">2012</option>
        <option value="2011">2011</option>
        <option value="2010">2010</option>
        <option value="2009">2009</option>
        <option value="2008">2008</option>
        <option value="2007">2007</option>
        <option value="2006">2006</option>
        <option value="2005">2005</option>
        <option value="2004">2004</option>
        <option value="2003">2003</option>
        <option value="2002">2002</option>
        <option value="2001">2001</option>
        <option value="2000">2000</option>
        <option value="1999">1999</option>
        <option value="1998">1998</option>
        <option value="1997">1997</option>
        <option value="1996">1996</option>
        <option value="1995">1995</option>
        <option value="1994">1994</option>
        <option value="1993">1993</option>
        <option value="1992">1992</option>
        <option value="1991">1991</option>
        <option value="1990">1990</option>
        <option value="1989">1989</option>
        <option value="1988">1988</option>
        <option value="1987">1987</option>
        <option value="1986">1986</option>
        <option value="1985">1985</option>
        <option value="1984">1984</option>
        <option value="1983">1983</option>
        <option value="1982">1982</option>
        <option value="1981">1981</option>
        <option value="1980">1980</option>
        <option value="1979">1979</option>
        <option value="1978">1978</option>
        <option value="1977">1977</option>
        <option value="1976">1976</option>
        <option value="1975">1975</option>
        <option value="1974">1974</option>
        <option value="1973">1973</option>
        <option value="1972">1972</option>
        <option value="1971">1971</option>
        <option value="1970">1970</option>
        <option value="1969">1969</option>
        <option value="1968">1968</option>
        <option value="1967">1967</option>
        <option value="1966">1966</option>
        <option value="1965">1965</option>
        <option value="1964">1964</option>
    
```



```

<option value="1963">1963</option>
<option value="1962">1962</option>
<option value="1961">1961</option>
<option value="1960">1960</option>
<option value="1959">1959</option>
<option value="1958">1958</option>
<option value="1957">1957</option>
<option value="1956">1956</option>
<option value="1955">1955</option>
<option value="1954">1954</option>
<option value="1953">1953</option>
<option value="1952">1952</option>
<option value="1951">1951</option>
<option value="1950">1950</option>
<option value="1949">1949</option>
<option value="1948">1948</option>
<option value="1947">1947</option>
<option value="1946">1946</option>
<option value="1945">1945</option>
<option value="1944">1944</option>
<option value="1943">1943</option>
<option value="1942">1942</option>
<option value="1941">1941</option>
<option value="1940">1940</option>
<option value="1939">1939</option>
<option value="1938">1938</option>
<option value="1937">1937</option>
<option value="1936">1936</option>
<option value="1935">1935</option>
<option value="1934">1934</option>
<option value="1933">1933</option>
<option value="1932">1932</option>
<option value="1931">1931</option>
<option value="1930">1930</option>
<option value="1929">1929</option>
<option value="1928">1928</option>
<option value="1927">1927</option>
<option value="1926">1926</option>
<option value="1925">1925</option>
<option value="1924">1924</option>
<option value="1923">1923</option>
<option value="1922">1922</option>
<option value="1921">1921</option>
<option value="1920">1920</option>
<option value="1919">1919</option>
<option value="1918">1918</option>
<option value="1917">1917</option>
<option value="1916">1916</option>
<option value="1915">1915</option>
<option value="1914">1914</option>
<option value="1913">1913</option>
<option value="1912">1912</option>
<option value="1911">1911</option>
<option value="1910">1910</option>
<option value="1909">1909</option>

```

```

        </select>
    </td>
</tr>
<tr>
    <td>hidden</td>
    <td>:</td>
    <td colspan="4"><input type="hidden" name="hidden" class="color"></td>
</tr>
<tr>
    <td>gender</td>
    <td>:</td>
    <td colspan="4">
        <input type="radio" name="gender" value="male" class="color" checked="checked">
        male
        <input type="radio" name="gender" value="female" class="color">
        female
    </td>
</tr>
<tr>
    <td>address</td>
    <td>:</td>
    <td colspan="4"><textarea></textarea></td>
</tr>
<tr>
    <td>hobbies</td>
    <td>:</td>
    <td>
        <input type="checkbox" name="hobbies" value="movies" checked="checked" class="color">
        movies<br>
        <input type="checkbox" name="hobbies" value="games" class="color">
        games
    </td>
    <td>
        <p>
            <input type="checkbox" name="hobbies" value="sports" class="color">
            sports <br>
            <input type="checkbox" name="hobbies" value="books" class="color">
            books
        </p>
    </td>
</tr>
<tr>
    <td colspan="4">
        <center>
            <input name="image" type="image" src="submit.png" class="color" value="image">
        </center>
    </td>
</tr>
<tr>
    <td colspan="4">
        <center>
            <input type="file" value="file" class="color" name="file">
        </center>
    </td>

```

```

</tr>
<tr>
  <td colspan="4">
    <center>
      <input type="submit" value="submit" class="color" name="submit">
    </center>
  </td>
</tr>
<tr>
  <td colspan="4">
    <center>
      <input type="reset" value="reset" class="color" name="reset">
    </center>
  </td>
</tr>
<tr>
  <td colspan="4">
    <center>
      <input type="button" value="button" class="color" name="button">
      <br>
      <button>button</button>
    </center>
  </td>
</tr>
</table>
</form>
<input type="button" value="Select" id="button1">

<script src="jquery-3.2.1.js"></script>

<script>
  $("#button1").click(fun1);
  function fun1()
  {
    $(".input").css("background-color", "green"); //input, select, textarea
    $(".text").css("background-color", "green");
    $(".password").css("background-color", "green");
    $(".radio").css("background-color", "green")
    $(".checkbox").css("background-color", "green")
    $(".image").css("background-color", "green")
    $(".file").css("background-color", "green")
    $(".submit").css("background-color", "green")
    $(".reset").css("background-color", "green")
    $(".button").css("background-color", "green");
    $(".text:disabled").css("background-color", "green").val("this is disabled");
    $(".text:enabled").css("background-color", "green");
    $(".radio:checked").css("background-color", "green")
    $(".checkbox:checked").css("background-color", "green")
  }
</script>
</body>
</html>

```

## Table Styles

- It is used to apply styles to the tables.

```
<html>
  <head>
    <title>jQuery - Table Styles</title>
    <style type="text/css">
      body,table
      {
        font-family: 'Tahoma';
        font-size: 30px;
      }
      .class1
      {
        background-color: #ff3399;
      }
      .class2
      {
        background-color: #99ffff;
      }
      .class3
      {
        background-color: #33ccff;
      }
      .class4
      {
        padding: 5px;
      }
      .class5
      {
        background-color: #ffff33;
      }
      .class6
      {
        cursor: pointer;
        background-color: #0099ff;
      }
    </style>
  </head>
  <body>
    <table id="table1">
      <tr>
        <th>Item</th>
        <th>Price</th>
      </tr>
      <tr>
        <td>Milk</td>
        <td>1.99</td>
      </tr>
      <tr>
        <td>Eggs</td>
        <td>2.29</td>
      </tr>
    </table>
  </body>
</html>
```

```
<tr>
  <td>Butter</td>
  <td>3.49</td>
</tr>
<tr>
  <td>Bread</td>
  <td>0.99</td>
</tr>
<tr>
  <td>Pasta</td>
  <td>1.19</td>
</tr>
<tr>
  <td>Honey</td>
  <td>4.39</td>
</tr>
<tr>
  <td>Cookies</td>
  <td>2.99</td>
</tr>
<tr>
  <td>Apples</td>
  <td>0.59</td>
</tr>
<tr>
  <td>Sugar</td>
  <td>1.78</td>
</tr>
<tr>
  <td>Pepper</td>
  <td>1.56</td>
</tr>
</table>

<script src="jquery-3.2.1.js"></script>
<script>
  $("#table1").addClass("class1");
  $("#table1 tr:even").addClass("class2");
  $("#table1 tr:odd").addClass("class3");
  $("#table1 tr td, #table1 tr th").addClass("class4");
  $("#table1 tr:first").addClass("class5");
  $("#table1 tr:gt(0)").hover(fun1, fun2);
  function fun1()
  {
    $(this).addClass("class6");
  }
  function fun2()
  {
    $(this).removeClass("class6");
  }
</script>
</body>
</html>
```

## Append – Advanced Examples

### Append – Advanced Example 1

---

```
<html>
  <head>
    <title>jQuery - Append 1</title>
    <style type="text/css">
      body,select
      {
        font-family:'Tahoma';
        font-size:30px;
      }
    </style>
  </head>
  <body>
    Years:
    <select id="dropdownlist1">

    </select>

    <script src="jquery-3.2.1.js"></script>

    <script>
      for(var i=2017; i<=2000; i--)
      {
        var s = "<option>" + i + "</option>";
        $("#dropdownlist1").append(s);
      }
    </script>
  </body>
</html>
```

### Append – Advanced Example 2

---

```
<html>
  <head>
    <title>jQuery - Append 2</title>
    <style type="text/css">
      body,input
      {
        font-family:'Tahoma';
        font-size:30px;
      }
      #div1 p
      {
        font-family: Tahoma;
        font-size:35px;
        background-color: orange;
        margin: 2px;
      }
    </style>
```

```
</head>
<body>
  <h3>Friends</h3>
  <div id="div1">
  </div>
  <input type="text" id="txt1">
  <input type="button" id="button1" value="+" title="Add">

  <script src="jquery-3.2.1.js"></script>

  <script>
    $("#button1").click(fun1);
    function fun1()
    {
      var s = "<p>" + $("#txt1").val() + "</p>";
      $("#div1").append(s);
      $("#txt1").val("").focus();
    }
  </script>
</body>
</html>
```

### Append – Advanced Example 3

---

```
<html>
<head>
  <title>jQuery - Append 3</title>
  <style>
    #div1 input[type=text]
    {
      font-family: Calibri;
      font-size: 24px;
      background-color: lightgreen;
      margin: 2px;
    }
  </style>
</head>
<body>
  <h3>Type some numbers</h3>
  <div id="div1">

  </div>
  <input type="button" id="button1" value="Add">
  <span id="span1">Result here</span>

  <script src="jquery-3.2.1.js"></script>

  <script>
    for(var i=1; i <= 10; i++)
    {
      var s = "<input type='text' id='txt' + i + '>'<br>";
      $("#div1").append(s);
    }
  </script>
```

```
var sum = 0;
$("#button1").click(fun1);
function fun1()
{
    var alltextboxes = $("#div1 input[type=text]");
    for (var i = 0; i < alltextboxes.length; i++)
    {
        sum += parseInt( $(alltextboxes[i]).val() ); /*'this' means current text box
    }
    $("#span1").html(sum);
}
</script>
</body>
</html>
```

#### Append – Advanced Example 4

---

```
<html>
<head>
<title>jQuery - Append 4</title>
</head>
<body>
<h3>Enter Employee Details</h3>
<div id="div1">
    Emp ID:<br><input type="text" id="textbox1"><br>
    Emp Name:<br><input type="text" id="textbox2"><br>
    Salary:<br><input type="text" id="textbox3">
</div>
<input type="button" id="button1" value="Add">
<table id="table1" border="1" cellpadding="5px">
<tr style="background-color:gold">
<th>Emp ID</th>
<th>Emp Name</th>
<th>Salary</th>
</tr>
</table>
<script src="jquery-3.2.1.js"></script>
<script>
    $("#button1").click(fun1);
    function fun1()
    {
        var s = "<tr><td>" + $("#textbox1").val() + "</td><td>" + $("#textbox2").val() + "</td><td>" +
        $("#textbox3").val() + "</td></tr>";
        $("#table1").append(s);
        $("#div1 input[type=text]").val("");
        $("#textbox1").focus();
    }
</script>
</body>
</html>
```



## jQuery – JavaScript Objects - Examples

```
<html>
<head>
  <title>jQuery - Object</title>
  <style type="text/css">
    body
    {
      font-family: 'Tahoma';
      font-size: 30px;
    }
    span
    {
      font-weight: bold;
      color: green;
    }
  </style>
</head>
<body>
  <p>
    Emp ID: <span id="span1"></span><br>
    Emp Name: <span id="span2"></span><br>
    Salary: <span id="span3"></span><br>
  </p>

  <script src="jquery-3.2.1.js"></script>

  <script>
    var emp = { "empid": 1, "empname": "Scott", "salary": 5000 };
    $("#span1").html(emp.empid);
    $("#span2").html(emp.empname);
    $("#span3").html(emp.salary);
  </script>
</body>
</html>
```

### jQuery – JavaScript Objects – Example 2

```
<html>
<head>
  <title>jQuery - Object Array - UL</title>
  <style type="text/css">
    body,table
    {
      font-family: 'Tahoma';
      font-size: 30px;
    }
  </style>
</head>
<body>
  <ul id="list1">
  </ul>

  <script src="jquery-3.2.1.js"></script>
```

```
<script>
var employees =
[
  {"empid":1,"empname": "Scott", "salary":5000},
  {"empid":2,"empname": "Allen", "salary":6000},
  {"empid":3,"empname": "John", "salary":7000},
  {"empid":4,"empname": "Jones", "salary":8000},
  {"empid":5,"empname": "Smith", "salary":9000}
];
for(i = 0; i < employees.length; i++)
{
  var employee = employees[i];
  var temp = "<li>" + employee.empname + "</li>";
  $("#list1").append(temp);
}
</script>
</body>
</html>
```

### jQuery – JavaScript Objects – Example 3

---

```
<html>
<head>
<title>jQuery - Object Array - Table</title>
<style type="text/css">
  body,table
  {
    font-family: 'Tahoma';
    font-size: 30px;
  }
</style>
</head>
<body>
<table id="table1" border="1">
<tr style="background-color: gold;">
  <th>Emp ID</th>
  <th>Emp Name</th>
  <th>Salary</th>
</tr>
</table>

<script src="jquery-3.2.1.js"></script>

<script>
var employees =
[
  {"empid":1,"empname": "Scott", "salary":5000},
  {"empid":2,"empname": "Allen", "salary":6000},
  {"empid":3,"empname": "John", "salary":7000},
  {"empid":4,"empname": "Jones", "salary":8000},
  {"empid":5,"empname": "Smith", "salary":9000}
];

for(i = 0; i < employees.length; i++)
```

```

{
    var employee = employees[i];
    var temp = "<tr><td>" + employee.empid + "</td> <td>" + employee.empname + "</td> <td>" +
employee.salary + "</td> </tr>";
    $("#table1").append(temp);
}

$("#table1").on("mouseover", "tr", fun1);
$("#table1").on("mouseout", "tr", fun2);
function fun1()
{
    $(this).css("background-color", "darkgray");
}
function fun2()
{
    $(this).css("background-color", "white");
}
</script>
</body>
</html>

```

#### jQuery – JavaScript Objects – Example 4

---

```

<html>
<head>
<title>jQuery - Object - User Data</title>
<style type="text/css">
    body,input
    {
        font-family: 'Tahoma';
        font-size: 30px;
    }
</style>
</head>
<body>
<h3>Type Person Details</h3>
<div id="div1">
    First Name: <input type="text" id="textbox1"><br>
    Last Name: <input type="text" id="textbox2"><br>
    Age: <input type="text" id="textbox3"><br>
</div>
<input type="button" id="button1" value="Add">

<script src="jquery-3.2.1.js"></script>

<script>
    var Persons = [];

    $("#button1").click(fun1);
    function fun1()
    {
        var p = { "FirstName": $("#textbox1").val(), "LastName": $("#textbox2").val(), "Age": $("#textbox3").val() };
        Persons.push(p);
        alert(JSON.stringify(Persons)); //convert json to string.
    }

```

```
        $("#div1 input[type=text]").val("");
        $("#textbox1").focus();
    };
</script>
</body>
</html>
```

### jQuery – JavaScript Objects – Example 5

---

```
<html>
<head>
<title>jQuery - Complex Object Example</title>
<style type="text/css">
    body,input,table
    {
        font-family: 'Tahoma';
        font-size: 30px;
    }
</style>
</head>
<body>
<h1>Complex json example</h1>
<div id="div1"></div>

<script src="jquery-3.2.1.js"></script>

<script>
var Departments =
[
    { DeptNo: 10, DeptName: "Accounting", Loc: "New York",
      Employees:
      [
          { EmpID: 1, EmpName: "ab" },
          { EmpID: 2, EmpName: "cd" },
          { EmpID: 3, EmpName: "ef" }
      ]
    },
    { DeptNo: 20, DeptName: "Sales", Loc: "New Delhi",
      Employees:
      [
          { EmpID: 4, EmpName: "gh" },
          { EmpID: 5, EmpName: "ij" }
      ]
    },
    { DeptNo: 30, DeptName: "R&D", Loc: "New Mumbai",
      Employees:
      [
          { EmpID: 6, EmpName: "kl" },
          { EmpID: 7, EmpName: "mn" },
          { EmpID: 8, EmpName: "op" },
          { EmpID: 9, EmpName: "qr" }
      ]
    }
]
```

```

    }

    ];

    var temp = "";
    $(fun1);
    function fun1()
    {
        for (i=0; i<Departments.length; i++)
        {
            temp += "<p>" + Departments[i].DeptNo + ", " + Departments[i].DeptName + ", " + Departments[i].Loc
+ "</p>";

            temp += "<table border='1' cellpadding='4px'>";
            for (j=0; j<Departments[i].Employees.length; j++)
            {
                temp += "<tr>";
                temp += "<td>" + Departments[i].Employees[j].EmpID + "</td>";
                temp += "<td>" + Departments[i].Employees[j].EmpName + "</td>";
                temp += "</tr>";
            }
            temp += "</table>";
        }
        $("#div1").html(temp);
    }
</script>
</body>
</html>

```

## CDN

- CDN stands for “Content Delivery Network”.
- The big companies such as Google, Microsoft etc., maintain all versions of jQuery files on their servers, which you can access directly from the html page, without manually downloading the jquery file and placing it in the current folder.

### Google CDN

```

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.js">
</script>

```

### Microsoft CDN

```

<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.2.1.js">
</script>

```

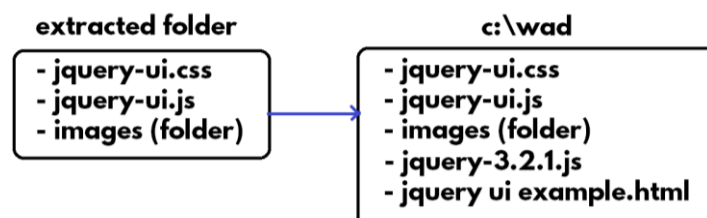
## jQuery UI

### Introduction to jQuery UI

- “jQuery UI” is a set of “plugins”, developed based on “jQuery”.
- “Plugin” is a “ready made component”.
  - Ex: date picker, auto complete, progress bar.

### How to download jQuery UI

- Go to “<https://jqueryui.com>”.
- Click on “Download”.
- Select the version. Ex: 1.12.1
- Select the necessary components (or) select all the components (checkboxes).
- Select the theme. Ex: base
- Click on “Download” button.
- Download the file “jquery-ui-1.12.1.custom.zip”.
- Go to the downloaded folder.
- Right click on “jquery-ui-1.12.1.custom.zip” and click on “Extract All”.
- Go to the extracted folder “jquery-ui-1.12.1.custom”.
- Copy and paste the following files from “extracted folder” into “c:\wad” folder.



### Import jQuery UI:

```
<link href="jquery-ui.css" rel="stylesheet">
<script src="jquery-3.2.1.js"></script>
<script src="jquery-ui.js"></script>
```

### datepicker( )

- This function is used to display a popup calendar for the textbox.
- **Syntax:** datepicker( )
- **Example:** datepicker( )

### Example on datePicker( )

---

```
<html>
<head>
  <title>jQuery UI - DatePicker</title>
  <style type="text/css">
    body, input
    {
      font-family: 'Tahoma';
      font-size: 25px;
    }

    #ui-datepicker-div
    {
      font-size: 70%;
    }
  </style>
</head>
<body>
  Date: <input type="text" id="txt1">

  <script src="jquery-3.2.1.js"></script>
  <script src="jquery-ui.js"></script>
  <link href="jquery-ui.css" rel="stylesheet">

  <script>
    $("#txt1").datepicker();
  </script>
</body>
</html>
```

### spinner( )

- This function is used to display “increment” and “decrement” buttons for a number in the textbox.
- **Syntax:** spinner( )
- **Example:** spinner( )

### Example on spinner( )

---

```
<html>
<head>
  <title>jQuery UI - Spinner</title>
  <style type="text/css">
    body, input
    {
      font-family: 'Tahoma';
      font-size: 20px;
    }
  </style>
</head>
<body>
  <input type="text" value="100">
  <input type="button" value="Increment" />
  <input type="button" value="Decrement" />
</body>
</html>
```

```

</style>
</head>
<body>
    Amount: <input type="text" id="txt1">

    <script src="jquery-3.1.0.js"></script>
    <script src="jquery-ui.js"></script>
    <link href="jquery-ui.css" rel="stylesheet">

    <script>
        $("#txt1").spinner();
    </script>
</body>
</html>

```

### autocomplete( )

- This function is used to display suggestions while typing some text in the textbox.
- **Syntax:**           autocomplete( { source: arrayname } )
- **Example:**           autocomplete( { source: myarray } )

### Example on autocomplete( )

```

<html>
<head>
    <title>jQuery UI - AutoComplete</title>
    <style type="text/css">
        body, input
        {
            font-family: 'Tahoma';
            font-size: 25px;
        }
    </style>
</head>
<body>
    <label for="txt1">Search: </label>
    <input type="text" id="txt1">

    <script src="jquery-3.1.0.js"></script>
    <script src="jquery-ui.js"></script>
    <link href="jquery-ui.css" rel="stylesheet">

    <script>
        var myarray = [ "ActionScript", "AppleScript", "Asp", "BASIC", "C", "C++", "Clojure", "COBOL", "ColdFusion",
        "Erlang", "Fortran", "Groovy", "Haskell", "Java", "JavaScript", "Lisp", "Perl", "PHP", "Python", "Ruby", "Scala", "Scheme"
        ];

        $("#txt1").autocomplete( { source: myarray } );
    </script>
</body>
</html>

```



**tooltip( )**

- This function displays the tooltip attractively.
- **Syntax:** tooltip( )
- **Example:** tooltip( )

**Example on tooltip( )**

```
<html>
<head>
  <title>jQuery UI - Tooltip</title>
  <style type="text/css">
    #div1
    {
      background-color: #33ccff;
    }
    #p1
    {
      background-color: #00cc99;
    }
  </style>
</head>
<body>
  <div id="div1" title="this is div 1">div 1</div>
  <p id="p1" title="this is para 1">para 1</p>
  Name: <input type="text" id="txt1" title="Alphabets only">
  <script src="jquery-3.1.0.js"></script>
  <script src="jquery-ui.js"></script>
  <link href="jquery-ui.css" rel="stylesheet">
  <script>
    $(document).tooltip();
  </script>
</body>
</html>
```

**resizable( )**

- This function is used to make the <div> tag as resizable.
- **Syntax:** resizable( )
- **Example:** resizable( )

**Example on resizable( )**

```
<html>
<head>
  <title>jQuery UI - Resizable</title>
  <style type="text/css">
    body, input
```

```

{
    font-family: 'Segoe UI';
    font-size: 25px;
}
#div1
{
    width: 200px;
    height: 200px;
    background-color: #99ffcc;
    cursor: pointer;
}
</style>
</head>
<body>
<div id="div1">div 1</div>

<script src="jquery-3.1.0.js"></script>
<script src="jquery-ui.js"></script>
<link href="jquery-ui.css" rel="stylesheet">
<script>
    $("#div1").resizable();
</script>
</body>
</html>

```

### draggable( )

- This function is used to make the <div> tag draggable across the page.
- **Syntax:**           draggable( )
- **Example:**           draggable( )

### Example on draggable( )

---

```

<html>
<head>
<title>jQuery UI - Draggable</title>
<style type="text/css">
    #div1
    {
        width: 200px;
        height: 150px;
        background-color: #ccffff;
        cursor: pointer;
    }
</style>
</head>
<body>
<div id="div1">
    <p>Drag me around</p>
</div>
<script src="jquery-3.1.0.js"></script>

```

```
<script src="jquery-ui.js"></script>
<link href="jquery-ui.css" rel="stylesheet">

<script>
    $("#div1").draggable();
</script>
</body>
</html>
```

### droppable( )

- This function is used to make an element allow other draggable elements, to be dropped inside it.
- **Syntax:**        droppable( { drop: functionname } )
- **Example:**        droppable( { drop: fun1 } )
- When the user drops some element into the droppable element, it calls the “fun1” automatically.

### Example on droppable( )

---

```
<html>
<head>
<title>jQuery UI - Droppable</title>
<style type="text/css">
    body, input
    {
        font-family: 'Tahoma';
        font-size: 25px;
    }
    #div1
    {
        width: 200px;
        height: 150px;
        background-color: #ccffff;
        cursor: pointer;
        float: left;
        margin-right: 10px;
    }
    #div2
    {
        width: 400px;
        height: 350px;
        background-color: #99ffcc;
        cursor: pointer;
        float: left;
    }
</style>
</head>
<body>
<div id="div1">div1</div>
<div id="div2">div2</div>
```

```

<script src="jquery-3.1.0.js"></script>
<script src="jquery-ui.js"></script>
<link href="jquery-ui.css" rel="stylesheet">

<script>
    $("#div1").draggable();
    $("#div2").droppable({ drop: fun1 });
    function fun1()
    {
        $(this).html("Dropped!").css("background-color", "#009999");
        //this = div2
    }
</script>
</body>
</html>

```

### selectable( )

- This function is used to make <li> tags selectable.
- **Syntax:**           selectable( )
- **Example:**           selectable( )

### Example on selectable( )

```

<html>
<head>
<title>jQuery UI - Selectable</title>
<style type="text/css">
    body, input
    {
        font-family: 'Segoe UI';
        font-size: 25px;
    }
    .class1 { background-color: #0099ff; }
    #list1 { list-style-type: none; padding: 0; }
    #list1 li { margin: 3px; }
    #list1 .ui-selecting { background-color: #0099cc; }
    #list1 .ui-selected { background-color: darkblue; color: white; }
</style>
</head>
<body>
<ol id="list1">
<li class="class1">Item 1</li>
<li class="class1">Item 2</li>
<li class="class1">Item 3</li>
<li class="class1">Item 4</li>
<li class="class1">Item 5</li>
<li class="class1">Item 6</li>
<li class="class1">Item 7</li>
</ol>

```

```

<script src="jquery-3.1.0.js"></script>
<script src="jquery-ui.js"></script>
<link href="jquery-ui.css" rel="stylesheet">

<script>
    $("#list1").selectable();
</script>
</body>
</html>

```

### sortable( )

- This function is used to allow the user to sort (re-order) the <li> tags of <ul> tag.
- **Syntax:**           sortable( )
- **Example:**           sortable( )

### Example on sortable( )

```

<html>
<head>
<title>jQuery UI - Sortable</title>
<style type="text/css">
    body, input
    {
        font-family: 'Tahoma';
        font-size: 25px;
    }
    .class1 { background-color: #0099ff; }
    #list1 { list-style-type: none; padding: 0; }
    #list1 li { margin: 5px; }
</style>
</head>
<body>
<ul id="list1">
<li class="class1">Item 1</li>
<li class="class1">Item 2</li>
<li class="class1">Item 3</li>
<li class="class1">Item 4</li>
<li class="class1">Item 5</li>
<li class="class1">Item 6</li>
<li class="class1">Item 7</li>
</ul>

<script src="jquery-3.1.0.js"></script>
<script src="jquery-ui.js"></script>
<link href="jquery-ui.css" rel="stylesheet">

<script>
    $("#list1").sortable();
</script>
</body>

```

</html>

### accordion( )

- This function is used to allow the user to view any one content among few.
- **Syntax:** accordion( )
- **Example:** accordion( )

### Example on accordion( )

---

```
<html>
  <head>
    <title>jQuery UI - Accordion</title>
    <style type="text/css">
      body, input
      {
        font-family: 'Tahoma';
        font-size: 25px;
      }
    </style>
  </head>
  <body>
    <div id="div1">
      <h3>Section 1</h3>
      <div>Mauris mauris ante, blandit et, ultrices a, suscipit eget, quam. Integer ut neque. Vivamus nisi metus, molestie vel, gravida in, condimentum sit amet, nunc. Nam a nibh. Donec suscipit eros. Nam mi. Proin viverra leo ut odio. Curabitur malesuada. Vestibulum a velit eu ante scelerisque vulputate.</div>
      <h3>Section 2</h3>
      <div>Sed non urna. Donec et ante. Phasellus eu ligula. Vestibulum sit amet purus. Vivamus hendrerit, dolor at aliquet laoreet, mauris turpis porttitor velit, faucibus interdum tellus libero ac justo. Vivamus non quam. In suscipit faucibus urna.</div>
      <h3>Section 3</h3>
      <div>Nam enim risus, molestie et, porta ac, aliquam ac, risus. Quisque lobortis. Phasellus pellentesque purus in massa. Aenean in pede. Phasellus ac libero ac tellus pellentesque semper. Sed ac felis. Sed commodo, magna quis lacinia ornare, quam ante aliquam nisi, eu iaculis leo purus venenatis dui.</div>
      <h3>Section 4</h3>
      <div>Cras dictum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean lacinia mauris vel est.</div>
    </div>

    <script src="jquery-3.1.0.js"></script>
    <script src="jquery-ui.js"></script>
    <link href="jquery-ui.css" rel="stylesheet">

    <script>
      $("#div1").accordion();
    </script>
  </body>
</html>
```

## tabs( )

- This function is used to display tabs in the web page.
- **Syntax:** tabs( )
- **Example:** tabs( )

### Example on tabs( )

```
<html>
  <head>
    <title>jQuery UI - Tabs</title>
    <style type="text/css">
      body, input
      {
        font-family: 'Tahoma';
        font-size: 20px;
      }
      .ui-tabs, .ui-tabs-anchor
      {
        font-size: 75%;
      }
    </style>
  </head>
  <body>
    <div id="div1">
      <ul>
        <li><a href="#tab1">Home</a></li>
        <li><a href="#tab2">About</a></li>
        <li><a href="#tab3">Contact</a></li>
      </ul>
      <div id="tab1">Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus. Curabitur nec arcu.
      Donec sollicitudin mi sit amet mauris. Nam elementum quam ullamcorper ante. Etiam aliquet massa et lorem. Mauris
      dapibus lacus auctor risus. Aenean tempor ullamcorper leo. Vivamus sed magna quis ligula eleifend adipiscing. Duis
      orci. Aliquam sodales tortor vitae ipsum. Aliquam nulla. Duis aliquam molestie erat. Ut et mauris vel pede varius
      sollicitudin. Sed ut dolor nec orci tincidunt interdum. Phasellus ipsum. Nunc tristique tempus lectus.</div>
      <div id="tab2">Morbi tincidunt, dui sit amet facilisis feugiat, odio metus gravida ante, ut pharetra massa
      metus id nunc. Duis scelerisque molestie turpis. Sed fringilla, massa eget luctus malesuada, metus eros molestie
      lectus, ut tempus eros massa ut dolor. Aenean aliquet fringilla sem. Suspendisse sed ligula in ligula suscipit aliquam.
      Praesent in eros vestibulum mi adipiscing adipiscing. Morbi facilisis. Curabitur ornare consequat nunc. Aenean vel
      metus. Ut posuere viverra nulla. Aliquam erat volutpat. Pellentesque convallis. Maecenas feugiat, tellus pellentesque
      pretium posuere, felis lorem euismod felis, eu ornare leo nisi vel felis. Mauris consectetur tortor et purus.</div>
      <div id="tab3">Mauris eleifend est et turpis. Duis id erat. Suspendisse potenti. Aliquam vulputate, pede vel
      vehicula accumsan, mi neque rutrum erat, eu congue orci lorem eget lorem. Vestibulum non ante. Class aptent taciti
      sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Fusce sodales. Quisque eu urna vel enim
      commodo pellentesque. Praesent eu risus hendrerit ligula tempus pretium. Curabitur lorem enim, pretium nec,
      feugiat nec, luctus a, lacus.</div>
    </div>

    <script src="jquery-3.1.0.js"></script>
    <script src="jquery-ui.js"></script>
    <link href="jquery-ui.css" rel="stylesheet">
```

```

<script>
    $("#div1").tabs();
</script>
</body>
</html>

```

## dialog( )

- This function is used to display modal dialog box (popup box) in the web page.
- Syntax:** dialog( { modal: true, width: pixels, height: pixels, buttons: { "button1": functionname, "button2": functionname } } )

## Example on dialog( )

```

<html>
<head>
<title>jQuery UI - Dialog</title>
<style type="text/css">
    body, input
    {
        font-family: 'Tahoma';
        font-size: 20px;
    }
    #span1
    {
        font-size: 30px;
    }
    .ui-dialog, .ui-dialog-title, .ui-button-text, .ui-dialog p
    {
        font-size: 70%;
    }
</style>
</head>
<body>
<input type="button" id="button1" value="Show dialog">
<br>
<span id="span1">Output here</span>
<div id="div1" title="Title here" style="display:none">
    <p>Hello, how are you!</p>
</div>

<script src="jquery-2.2.0.js"></script>
<script src="jquery-ui.js"></script>
<link href="jquery-ui.css" rel="stylesheet">

<script>
    $("#button1").click(fun1);
    function fun1()
    {
        $("#div1").dialog({ modal:true, width:300, height: 200, buttons: { "OK": fun2, "Cancel": fun3 } });
    }

```



```

function fun2()
{
    $("#span1").html("OK clicked");
    $(this).dialog("close");
}
function fun3()
{
    $("#span1").html("Cancel clicked");
    $(this).dialog("close");
}
</script>
</body>
</html>

```

### buttonset( )

- It is used to display “radio buttons” as “buttons”.
- **Syntax:** buttonset( )

### Example on buttonset( )

```

<html>
<head>
<title>jQuery UI - ButtonSet</title>
<style type="text/css">
    body, input
    {
        font-family: 'Tahoma';
        font-size: 25px;
    }
</style>
</head>
<body>
<div id="div1">
    <input type="radio" id="radio1" name="radio">
    <label for="radio1">Debit Card</label>
    <input type="radio" id="radio2" name="radio">
    <label for="radio2">Credit Card</label>
    <input type="radio" id="radio3" name="radio">
    <label for="radio3">Net Banking</label>
</div>

<script src="jquery-3.1.0.js"></script>
<script src="jquery-ui.js"></script>
<link href="jquery-ui.css" rel="stylesheet">

<script>
    $("#div1").buttonset();
</script>
</body>
</html>

```

**menu( )**

- This function is used to display a menu (along with sub menus) in the web page.
- **Syntax:**    menu( )

**Example on menu( )**

---

```
<html>
  <head>
    <title>jQuery UI - Menu</title>
    <style type="text/css">
      body, input
      {
        font-family: 'Tahoma';
        font-size: 20px;
      }
      .ui-menu
      {
        width: 300px;
      }
    </style>
  </head>
  <body>
    <ul id="list1">
      <li>Home</li>
      <li>About</li>
      <li>Contact</li>
      <li>Download</li>
      <li>Express Download</li>
      <li>Product Info
        <ul>
          <li>Order</li>
          <li>Features</li>
          <li>Installation</li>
          <li>Supported Browsers
            <ul>
              <li>Google Chrome</li>
              <li>Mozilla Firefox</li>
              <li>Internet Explorer</li>
              <li>Safari</li>
              <li>Opera</li>
            </ul>
          </li>
        </ul>
      </li>
      <li>Purchase</li>
      <li>Demo / Installation</li>
    </ul>

    <script src="jquery-3.1.0.js"></script>
```

```
<script src="jquery-ui.js"></script>
<link href="jquery-ui.css" rel="stylesheet">

<script>
    $("#list1").menu();
</script>

</body>
</html>
```

### progressbar( )

- This function is used to display progress bar within the web page.
- **Syntax:** progressbar( { value: n } )
- **Example:** progressbar( { value: 60 } )

### Example on progressbar( )

```
<html>
<head>
    <title>jQuery UI - Progressbar</title>
    <style type="text/css">
        body, input
        {
            font-family: 'Tahoma';
            font-size: 20px;
        }
        #div1
        {
            height: 20px;
        }
    </style>
</head>
<body>
    <div id="div1"></div>

    <script src="jquery-3.1.0.js"></script>
    <script src="jquery-ui.js"></script>
    <link href="jquery-ui.css" rel="stylesheet">

    <script>
        $("#div1").progressbar( { value: 60 } );
    </script>
</body>
</html>
```

### slider( )

- This function is used to display a slider in the web page. Ex: price range

- **Syntax:** slider( { range: true, min: minimum, max: maximum, values: [ value1, value2 ], slide: functionname } );

### Example on slider( )

```
<html>
<head>
  <title>jQuery UI - Slider</title>
</head>
<body>
  <p>
    Price range:
    <span id="span1" style="color:#f6931f; font-weight:bold;"></span>
  </p>
  <div id="div1"></div>

  <script src="jquery-3.1.0.js"></script>
  <script src="jquery-ui.js"></script>
  <link href="jquery-ui.css" rel="stylesheet">

  <script>
    $("#div1").slider( { range: true, min: 0, max: 500, values: [ 200, 300 ], slide: fun1 } );
    function fun1(event, ui)
    {
      $("#span1").html( "Rs. " + ui.values[ 0 ] + " - Rs. " + ui.values[ 1 ] );
    }
    $("#span1").html( "Rs. " + $("#div1").slider( "values", 0 ) + " - Rs. " + $("#div1").slider( "values", 1 ) );
  </script>
</body>
</html>
```

### Color Animation

- “jQuery UI” supports “color animation”, which can animate (change) the color based properties such as “background-color”, “color”, “border-color” etc.
- **Syntax:** animate( { “property”: “value” }, milli seconds );

### Example on Color Animation

```
<html>
<head>
  <title>jQuery UI - Color Animation</title>
  <style type="text/css">
    body, input
    {
      font-family: 'Tahoma';
      font-size: 20px;
    }
  </div>
```

```

{
    background-color: darkred;
    color: cyan;
    font-size: 30px;
    font-family: 'Tahoma';
}
</style>
</head>
<body>
    <div id="div1">Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum is
    simply dummy text of the printing and typesetting industry.</div>
    <input type="button" id="button1" value="color animation">

    <script src="jquery-3.1.0.js"></script>
    <script src="jquery-ui.js"></script>
    <link href="jquery-ui.css" rel="stylesheet">

    <script>
        $("#button1").click(fun1);
        function fun1()
        {
            $("#div1").animate({ "background-color": "green", "color": "yellow" }, 2000);
        }
    </script>
</body>
</html>

```

## Easing Effects

- “Easing Effects” are used to make the animation more attractively.
- **Syntax:** animate( { “property”: “value” }, milli seconds, “easing effect” );
- **List of easing effects:** linear, swing, easeInQuad, easeOutQuad, easeInOutQuad, easeInCubic, easeOutCubic, easeInOutCubic, easeInQuart, easeOutQuart, easeInOutQuart, easeInQuint, easeOutQuint, easeInOutQuint, easeInExpo, easeOutExpo, easeInOutExpo, easeInSine, easeOutSine, easeInOutSine, easeInCirc, easeOutCirc, easeInOutCirc, easeInElastic, easeOutElastic, easeInOutElastic, easeInBack, easeOutBack, easeInOutBack, easeInBounce, easeOutBounce, easeInOutBounce,

## Example on Easing Effects

```

<html>
<head>
    <title>jQuery UI - Easing Effects</title>
    <style type="text/css">
        body, input
        {
            font-family: 'Tahoma';
            font-size: 20px;
        }
    </style>

```

```
#div1
{
    background-color: darkred;
    color: cyan;
    font-size: 30px;
    position: relative;
    width: 300px;
    padding: 5px;
    border: 2px solid green;
}
</style>
</head>
<body>
    <input type="button" id="button1" value="easing effect animation">
    <div id="div1">Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum is
    simply dummy text of the printing and typesetting industry.</div>

    <script src="jquery-3.1.0.js"></script>
    <script src="jquery-ui.js"></script>
    <link href="jquery-ui.css" rel="stylesheet">

    <script>
        $("#button1").click(fun1);
        function fun1()
        {
            $("#div1").animate( { "font-size": "40px", "border-width": "10px", "left": "30px", "width": "700px",
            "padding": "15px" }, 2000, "easeOutQuart");
        }
    </script>
</body>
</html>
```

### addClass( ) with animation

- addClass( ) function supports time duration (milli seconds), in jQuery UI.
- **Syntax:** addClass( "class name", milli seconds );
- **Syntax:** addClass( "class1", 1000 );

### Example on addClass( ) with animation

```
<html>
<head>
    <title>jQuery UI - addClass</title>
    <style type="text/css">
        body, input
        {
            font-family: 'Tahoma';
            font-size: 20px;
        }
        .class1
        {
```

```
        background-color: darkred;
        color: cyan;
        font-size: 30px;
        font-family: 'Tahoma';
        width: 800px;
        padding: 15px;
    }
</style>
</head>
<body>
    <div id="div1">Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum is
simply dummy text of the printing and typesetting industry.</div>
    <input type="button" id="button1" value="add class">

    <script src="jquery-3.1.0.js"></script>
    <script src="jquery-ui.js"></script>
    <link href="jquery-ui.css" rel="stylesheet">

    <script>
        $("#button1").click(fun1);
        function fun1()
        {
            $("#div1").addClass("class1", 1000);
        }
    </script>
</body>
</html>
```

### removeClass( ) with animation

- removeClass( ) function supports time duration (milli seconds), in jQuery UI.
- **Syntax:** removeClass( "class name", milli seconds );
- **Syntax:** removeClass( "class1", 1000 );

### Example on removeClass( ) with animation

---

```
<html>
<head>
    <title>jQuery UI - removeClass</title>
    <style type="text/css">
        body, input
        {
            font-family: 'Tahoma';
            font-size: 20px;
        }
        .class1
        {
            background-color: darkred;
            color: cyan;
            font-size: 30px;
            font-family: 'Tahoma';
        }
    </style>
</head>
<body>
    <div id="div1">Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum is
simply dummy text of the printing and typesetting industry.</div>
    <input type="button" id="button1" value="add class">

    <script src="jquery-3.1.0.js"></script>
    <script src="jquery-ui.js"></script>
    <link href="jquery-ui.css" rel="stylesheet">

    <script>
        $("#button1").click(fun1);
        function fun1()
        {
            $("#div1").removeClass("class1", 1000);
        }
    </script>
</body>
</html>
```

```

        width: 800px;
        padding: 15px;
    }
</style>
</head>
<body>
    <div id="div1" class="class1">Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem
    Ipsum is simply dummy text of the printing and typesetting industry.</div>
    <input type="button" id="button1" value="remove class">
    <script src="jquery-3.1.0.js"></script>
    <script src="jquery-ui.js"></script>
    <link href="jquery-ui.css" rel="stylesheet">

    <script>
        $("#button1").click(fun1);
        function fun1()
        {
            $("#div1").removeClass("class1", 1000);
        }
    </script>
</body>
</html>

```

## Effects

- jQuery UI supports the following advanced effects to hide / show the content attractively.
- **List of effects:** blind | bounce | clip | drop | explode | fold | highlight | puff | pulsate | scale | shake | size | slide
- **Show:** show( "effect", { options }, milli seconds, callback );
- **Hide:** show( "effect", { options }, milli seconds, callback );
- **Note:** The callback function will be called automatically, after completion of animation. It is optional.

## Example on Effects

```

<html>
<head>
    <title>jQuery UI - Effects</title>
    <style type="text/css">
        body, input
        {
            font-family: 'Tahoma';
            font-size: 20px;
        }
        #div1
        {
            background-color: darkred;
            color: cyan;
            font-size: 30px;

```



```
        width: 800px;
        padding: 15px;
    }
</style>
</head>
<body>
    <input type="button" id="button1" value="Hide">
    <input type="button" id="button2" value="Show">
    <div id="div1">Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum is
simply dummy text of the printing and typesetting industry.</div>

    <script src="jquery-3.1.0.js"></script>
    <script src="jquery-ui.js"></script>
    <link href="jquery-ui.css" rel="stylesheet">

    <script>
        $("#button1").click(fun1);
        function fun1()
        {
            $("#div1").hide("bounce", {}, 1000, callback);
        }
        function callback()
        {
            alert("Done");
        }
        $("#button2").click(fun2);
        function fun2()
        {
            $("#div1").show("bounce", {}, 1000, callback);
        }
    </script>
</body>
</html>
```

## jQuery Validations

- “jQuery Validation Plugin” is used to create validations in jQuery.
- Download “jquery-3.2.1.js” file from “https://jquery.com” and place it in “c:\jquery” folder.
- Download the jquery validation plugin file and place it in “c:\jquery” folder:

<http://ajax.aspnetcdn.com/ajax/jquery.validate/1.15.0/jquery.validate.js>

- Create “validation.html” file in “c:\jquery” folder.

### List of files

---

c:\jquery

- jquery-3.2.1.js
- jquery.validate.js
- validation.html

### Code for “Validation.html”

---

```
<html>
  <head>
    <title>jQuery Validation Plugin</title>
  </head>
  <body>
    <h1>Registration Form</h1>
    <form action="http://localhost/someaddress" id="form1">

      <table>
        <tr>
          <td><label for="t1">Username:</label></td>
          <td><input type="text" id="t1" name="txt1"></td>
        </tr>

        <tr>
          <td><label for="t2">Password:</label></td>
          <td><input type="password" id="t2" name="txt2"></td>
        </tr>

        <tr>
          <td><label for="t3">E-Mail:</label></td>
          <td><input type="text" id="t3" name="txt3"></td>
        </tr>

        <tr>
          <td><label for="t4">Re-type E-Mail:</label></td>
          <td><input type="text" id="t4" name="txt4"></td>
        </tr>

        <tr>
```

```

        <td><label for="t5">Phone:</label></td>
        <td><input type="text" id="t5" name="txt5"></td>
    </tr>

    <tr>
        <td><label for="t6">Age:</label></td>
        <td><input type="text" id="t6" name="txt6" maxlength="2"></td>
    </tr>

    <tr>
        <td colspan="2"><input type="submit" value="Register"></td>
    </tr>
</table>
</form>

<script src="jquery-3.2.1.js"></script>
<script src="jquery.validate.js"></script>

<script>
    $("#form1").validate({
        rules:
        {
            txt1: { required: true },
            txt2: { required: true, minlength: 6 },
            txt3: { required: true, regexp: /^[a-zA-Z0-9._-]+@[a-zA-Z0-9-]+\.[a-zA-Z]{2,5}$/i },
            txt4: { required: true, equalTo: '#t3' },
            txt5: { required: true },
            txt6: { required: true, range: [18,70] }
        },
        messages:
        {
            txt1: { required: "Username can't be blank" },
            txt2: { required: "Password can't be blank", minlength: "Minimum length is 6" },
            txt3: { required: "Email can't be blank", regexp: "Enter a valid E-Mail" },
            txt4: { required: "Re-type email can't be blank", equalTo: "Email and re-type email must be same"
        },
            txt5: { required: "Phone can't be blank" },
            txt6: { required: "Age can't be blank", range: "Age must be in between 18 and 70" }
        }
    });

    //for regular expressions
    $.validator.addMethod("regexp", function(value, element, param) {
        return this.optional(element) || param.test(value); // Compare with regular expression
    });
</script>

<style type="text/css">
    body, input, select, textarea, table { font-family: 'Tahoma'; font-size: 28px; }
    label.error { color: red; } /*style for validation error messages */

```

```
        input.error { border: 1px solid #foo; background-color: #fee; } /* style for validation error text boxes */
    </style>

</body>
</html>
```

## jQuery AJAX

### jQuery AJAX

- “jQuery” provides a method called “\$.ajax( )”, to send ajax request to server and get ajax response from server, more easily.

#### Syntax of \$.ajax( )

\$.ajax( { type: “GET | POST | PUT | DELETE”, url: “address”, success: callbackfunction, error: callbackfunction } );

url: Represents the server address, to which you want to send request.

type: Represents type of the request: GET | POST | PUT | DELETE

success: Represents the callback function, which will be called automatically after successfully receiving the response from server.

error: Represents the callback function, which will be called automatically if the server returns an error (exception) as response to the browser.

### jQuery AJAX – NodeJS – Simple - Example

#### Creating the application

- Create “c:\ajax” folder.
- Place “jquery-3.2.1.js” file in “c:\ajax” folder.
- Create “httpserver.js” and “index.html” files in the “c:\ajax” folder.

#### c:\ajax

- httpserver.js
- index.html

- jquery-3.2.1.js

### c:\ajax\httpserver.js

---

```
var http = require("http");
var fs = require("fs");
var server = http.createServer(engine);
server.listen(8080, startup);

function startup()
{
    console.log("Server started at port 8080");
}

function engine(request, response)
{
    if (request.url == "/" || request.url == "/index.html")
    {
        fs.readFile("index.html", "utf8", fun1);
        function fun1(error, data)
        {
            if (error)
            {
                response.setHeader("content-type", "text/html");
                response.writeHead(404);
                response.write("<h1 style='color:red'>Page not found</h1>");
                response.end();
            }
            else
            {
                response.setHeader("content-type", "text/html");
                response.writeHead(200);
                response.write(data);
                response.end();
            }
        }
    }
    else if (request.url == "/jquery-3.2.1.js")
    {
        fs.readFile("jquery-3.2.1.js", "utf8", fun1);
        function fun1(error, data)
        {
            if (error)
            {
                response.setHeader("content-type", "text/html");
                response.writeHead(404);
                response.write("<h1 style='color:red'>Page not found</h1>");
                response.end();
            }
            else
            {
                response.setHeader("content-type", "text/javascript");
                response.writeHead(200);
            }
        }
    }
}
```

```
        response.write(data);
        response.end();
    }
}
else if (request.url.startsWith("/getdata"))
{
    console.log("request received at " + new Date().toLocaleTimeString());
    response.setHeader("content-type", "text/html");
    response.writeHead(200);
    response.write("Response from server at " + new Date().toLocaleTimeString());
    response.end();
}
}
```

c:\ajax\index.html

---

```
<html>
<head>
  <title>jQuery AJAX - Simple</title>
</head>
<body>
  <h1>jQuery AJAX - Simple</h1>
  <input type="button" id="button1" value="GET data from server">
  <div id="div1"></div>

  <script src="jquery-3.2.1.js"></script>
  <script>
    $("#button1").click(fun1);
    function fun1()
    {
      $.ajax({ url:"GET", url:"/getdata", success: fun2 });
    }
    function fun2(response)
    {
      $("#div1").html(response);
    }
  </script>
</body>
</html>
```

### Execution:

---

- Download and install “nodejs” from “<https://nodejs.org>”.
- Open “Command Prompt” and run the following commands:  
cd c:\ajax  
node httpserver.js
- Open browser and enter the url: <http://localhost:8080/index.html>

## jQuery AJAX – NodeJS – Get - Example

### Creating the application

- Create “c:\ajax” folder.
- Place “jquery-3.2.1.js” file in “c:\ajax” folder.
- Create “httpserver.js” and “index.html” files in the “c:\ajax” folder.

#### c:\ajax

- httpserver.js
- index.html
- jquery-3.2.1.js

#### c:\ajax\httpserver.js

```
var http = require("http");
var fs = require("fs");
var server = http.createServer(engine);
server.listen(8080, startup);

function startup()
{
    console.log("Server started at port 8080");
}

function engine(request, response)
{
    if (request.url == "/" || request.url == "/index.html")
    {
        fs.readFile("index.html", "utf8", fun1);
        function fun1(error, data)
        {
            if (error)
            {
                response.setHeader("content-type", "text/html");
                response.writeHead(404);
                response.write("<h1 style='color:red'>Page not found</h1>");
                response.end();
            }
            else
            {
                response.setHeader("content-type", "text/html");
                response.writeHead(200);
                response.write(data);
                response.end();
            }
        }
    }
    else if (request.url == "/jquery-3.2.1.js")
```

```
{
  fs.readFile("jquery-3.2.1.js", "utf8", fun1);
  function fun1(error, data)
  {
    if (error)
    {
      response.setHeader("content-type", "text/html");
      response.writeHead(404);
      response.write("<h1 style='color:red'>Page not found</h1>");
      response.end();
    }
    else
    {
      response.setHeader("content-type", "text/javascript");
      response.writeHead(200);
      response.write(data);
      response.end();
    }
  }
}
else if (request.url.startsWith("/getemployees"))
{
  console.log("request received at " + new Date().toLocaleTimeString());
  response.setHeader("content-type", "application/json");
  response.writeHead(200);
  response.write(`
    { "empid": 1, "empname": "Scott", "salary": 4000 },
    { "empid": 2, "empname": "Allen", "salary": 7500 },
    { "empid": 3, "empname": "Jones", "salary": 9200 },
    { "empid": 4, "empname": "James", "salary": 8400 },
    { "empid": 5, "empname": "Smith", "salary": 5600 }
  `);
  response.end();
}
}
```

c:\ajax\index.html

---

```
<!DOCTYPE html>
<html>
<head>
  <title>NodeJS - jQuery AJAX - GET</title>
</head>
<body>
  <h1>NodeJS - jQuery AJAX - GET</h1>
  <form>
    <input type="button" id="btn1" value="Get Data">
    <table id="table1" border="1">
      <tr> <th>Emp ID</th> <th>Emp Name</th> <th>Salary</th> </tr>

    </table>
  </form>

  <script src="/jquery-3.2.1.js"></script>
```



```
<script>
$("#btn1").click(fun1);
function fun1(event)
{
    event.preventDefault();
    $.ajax({ type: "GET", url: "/getemployees", success: fun2, error: fun3 });
}

function fun2(response)
{
    $("#table1 tr:gt(0)").remove();
    for (var i = 0; i < response.length; i++)
    {
        $("#table1").append("<tr> <td>" + response[i].empid + "</td> <td>" + response[i].empname + "</td> <td>" +
response[i].salary + "</td> </tr>");
    }
}

function fun3(error)
{
    alert(error);
}
</script>
</body>
</html>
```

#### Execution:

- Download and install “nodejs” from “<https://nodejs.org>”.
- Open “Command Prompt” and run the following commands:  
cd c:\ajax  
node httpserver.js
- Open browser and enter the url: <http://localhost:8080/index.html>

### jQuery AJAX – NodeJS – Search - Example

#### Creating the application

- Create “c:\ajax” folder.
- Place “jquery-3.2.1.js” file in “c:\ajax” folder.
- Create “httpserver.js” and “index.html” files in the “c:\ajax” folder.

##### c:\ajax

- httpserver.js
- index.html

- jquery-3.2.1.js

### c:\ajax\httpserver.js

---

```
var http = require("http");
var fs = require("fs");
var querystring = require("querystring");
var server = http.createServer(engine);
server.listen(8080, startup);

function startup()
{
    console.log("Server started at port 8080");
}

function engine(request, response)
{
    if (request.url == "/" || request.url == "/index.html")
    {
        fs.readFile("index.html", "utf8", fun1);
        function fun1(error, data)
        {
            if (error)
            {
                response.setHeader("content-type", "text/html");
                response.writeHead(404);
                response.write("<h1 style='color:red'>Page not found</h1>");
                response.end();
            }
            else
            {
                response.setHeader("content-type", "text/html");
                response.writeHead(200);
                response.write(data);
                response.end();
            }
        }
    }
    else if (request.url == "/jquery-3.2.1.js")
    {
        fs.readFile("jquery-3.2.1.js", "utf8", fun1);
        function fun1(error, data)
        {
            if (error)
            {
                response.setHeader("content-type", "text/html");
                response.writeHead(404);
                response.write("<h1 style='color:red'>Page not found</h1>");
                response.end();
            }
            else
            {
                response.setHeader("content-type", "text/javascript");
            }
        }
    }
}
```

```

        response.writeHead(200);
        response.write(data);
        response.end();
    }
}
else if (request.url.startsWith("/searchemployees"))
{
    console.log("request received at " + new Date().toLocaleTimeString());
    response.setHeader("content-type", "application/json");
    response.writeHead(200);
    var employees = [
        { "empid": 1, "empname": "Scott", "salary": 4000 },
        { "empid": 2, "empname": "Allen", "salary": 7500 },
        { "empid": 3, "empname": "Jones", "salary": 9200 },
        { "empid": 4, "empname": "James", "salary": 8400 },
        { "empid": 5, "empname": "Smith", "salary": 5600 }
    ];
    var q = querystring.parse((request.url.split("?")[1]));
    var employees2 = [];
    for (var i = 0; i < employees.length; i++)
    {
        if (employees[i].empname.indexOf(q.searchstr) >= 0)
        {
            employees2.push(employees[i]);
        }
    }
    response.write(JSON.stringify(employees2));
    response.end();
}
}

```

c:\ajax\index.html

```

<!DOCTYPE html>
<html>
<head>
    <title>NodeJS - jQuery AJAX - Search</title>
</head>
<body>
    <h1>NodeJS - jQuery AJAX - Search</h1>
    <form>
        Search employees: <input type="text" id="txt1">
        <input type="submit" id="button1" value="Search">
        <table id="table1" border="1">
            <tr>
                <th>Emp ID</th>
                <th>Emp Name</th>
                <th>Salary</th>
            </tr>
        </table>
    </form>

    <script src="/jquery-3.2.1.js"></script>

    <script>
        $("#button1").click(fun1);
    </script>

```

```
function fun1(event)
{
    event.preventDefault();
    $.ajax({ type: "GET", url: "/searchemployees?searchstr=" + $("#txt1").val(), success: fun2, error: fun3 });
}

function fun2(response)
{
    $("#table1 tr:gt(0)").remove();
    for (var i = 0; i < response.length; i++)
    {
        $("#table1").append("<tr><td>" + response[i].empid + "</td><td>" + response[i].empname + "</td><td>" +
response[i].salary + "</td></tr>");
    }
}

function fun3(error)
{
    alert(error);
}
</script>
</body>
</html>
```

---

**Execution:**

- Download and install “nodejs” from “<https://nodejs.org>”.
- Open “Command Prompt” and run the following commands:  
    cd c:\ajax  
    node httpserver.js
- Open browser and enter the url: <http://localhost:8080/index.html>

**jQuery AJAX – NodeJS – Post - Example**

---

**Creating the application**

- Create “c:\ajax” folder.
- Place “jquery-3.2.1.js” file in “c:\ajax” folder.
- Create “httpserver.js” and “index.html” files in the “c:\ajax” folder.

**c:\ajax**

- httpserver.js
- index.html
- jquery-3.2.1.js

**c:\ajax\httpserver.js**

```
var http = require("http");
var fs = require("fs");
var querystring = require("querystring");
var server = http.createServer(engine);
server.listen(8080, startup);

function startup()
{
    console.log("Server started at port 8080");
}

function engine(request, response)
{
    if (request.url == "/" || request.url == "/index.html")
    {
        fs.readFile("index.html", "utf8", fun1);
        function fun1(error, data)
        {
            if (error)
            {
                response.setHeader("content-type", "text/html");
                response.writeHead(404);
                response.write("<h1 style='color:red'>Page not found</h1>");
                response.end();
            }
            else
            {
                response.setHeader("content-type", "text/html");
                response.writeHead(200);
                response.write(data);
                response.end();
            }
        }
    }
    else if (request.url == "/jquery-3.2.1.js")
    {
        fs.readFile("jquery-3.2.1.js", "utf8", fun1);
        function fun1(error, data)
        {
            if (error)
            {
                response.setHeader("content-type", "text/html");
                response.writeHead(404);
                response.write("<h1 style='color:red'>Page not found</h1>");
                response.end();
            }
            else
            {
                response.setHeader("content-type", "text/javascript");
                response.writeHead(200);
                response.write(data);
            }
        }
    }
}
```

```
        response.end();
    }
}
else if (request.url.startsWith("/insertemployee"))
{
    console.log("request received at " + new Date().toLocaleTimeString());
    response.setHeader("content-type", "text/html");
    response.write("Successfully Inserted");
    response.end();
}
}
```

c:\ajax\index.html

```
<!DOCTYPE html>
<html>
<head>
<title>NodeJS - jQuery AJAX - Post</title>
<style>
    body, input
    {
        font-family: Tahoma;
        font-size: 24px;
    }
</style>
</head>
<body>
<h1>NodeJS - jQuery AJAX - Post</h1>
<form>
    Emp ID: <input type="text" id="txt1"><br>
    Emp Name: <input type="text" id="txt2"><br>
    Salary: <input type="text" id="txt3"><br>
    <input type="submit" id="btn1" value="Insert">
    <div id="div1"></div>
</form>

<script src="jquery-3.2.1.js"></script>

<script>
    $("#btn1").click(fun1);
    function fun1(event)
    {
        event.preventDefault();
        var mydata = { "empid": $("#txt1").val(), "empname": $("#txt2").val(), "salary": $("#txt3").val() };
        $.ajax({ type: "POST", url: "/insertemployee", success: fun2, data: mydata });
    }

    function fun2(response)
    {
        $("#div1").html(response);
    }
</script>
</body>
```

`</html>`**Execution:**

- Download and install “nodejs” from “<https://nodejs.org>”.
- Open “Command Prompt” and run the following commands:  
    cd c:\ajax  
    node httpserver.js
- Open browser and enter the url: <http://localhost:8080/index.html>

**jQuery AJAX – NodeJS – Put - Example****Creating the application**

- Create “c:\ajax” folder.
- Place “jquery-3.2.1.js” file in “c:\ajax” folder.
- Create “httpserver.js” and “index.html” files in the “c:\ajax” folder.

**c:\ajax**

- httpserver.js
- index.html
- jquery-3.2.1.js

**c:\ajax\httpserver.js**

```
var http = require("http");
var fs = require("fs");
var querystring = require("querystring");
var server = http.createServer(engine);
server.listen(8080, startup);

function startup()
{
    console.log("Server started at port 8080");
}

function engine(request, response)
{
    if (request.url == "/" || request.url == "/index.html")
    {
        fs.readFile("index.html", "utf8", fun1);
        function fun1(error, data)
        {
            if (error)
```

```

    {
        response.setHeader("content-type", "text/html");
        response.writeHead(404);
        response.write("<h1 style='color:red'>Page not found</h1>");
        response.end();
    }
    else
    {
        response.setHeader("content-type", "text/html");
        response.writeHead(200);
        response.write(data);
        response.end();
    }
}
else if (request.url == "/jquery-3.2.1.js")
{
    fs.readFile("jquery-3.2.1.js", "utf8", fun1);
    function fun1(error, data)
    {
        if (error)
        {
            response.setHeader("content-type", "text/html");
            response.writeHead(404);
            response.write("<h1 style='color:red'>Page not found</h1>");
            response.end();
        }
        else
        {
            response.setHeader("content-type", "text/javascript");
            response.writeHead(200);
            response.write(data);
            response.end();
        }
    }
}
else if (request.url.startsWith("/updateemployee"))
{
    console.log("request received at " + new Date().toLocaleTimeString());
    response.setHeader("content-type", "text/html");
    response.write("Successfully Updated");
    response.end();
}
}

```

c:\ajax\index.html

```

<!DOCTYPE html>
<html>
<head>
    <title>NodeJS - jQuery AJAX - Put</title>
    <style>
        body, input

```



```
{
  font-family: Tahoma;
  font-size: 24px;
}
</style>
</head>
<body>
<h1>NodeJS - jQuery AJAX - Put</h1>
<form>
  Existing Emp ID: <input type="text" id="txt1"><br>
  Emp Name: <input type="text" id="txt2"><br>
  Salary: <input type="text" id="txt3"><br>
  <input type="submit" id="btn1" value="Update">
<div id="div1"></div>
</form>

<script src="jquery-3.2.1.js"></script>

<script>
$( "#btn1" ).click(fun1);
function fun1(event)
{
  event.preventDefault();
  var mydata = { "empid": $("#txt1").val(), "empname": $("#txt2").val(), "salary": $("#txt3").val() };
  $.ajax({ type: "PUT", url: "/updateemployee", data: mydata, success: fun2 });
}

function fun2(response)
{
  $("#div1").html(response);
}
</script>
</body>
</html>
```

### Execution:

---

- Download and install “nodejs” from “<https://nodejs.org>”.
- Open “Command Prompt” and run the following commands:  
cd c:\ajax  
node httpserver.js
- Open browser and enter the url: <http://localhost:8080/index.html>

## jQuery AJAX – NodeJS – Delete - Example

### Creating the application

---

- Create “c:\ajax” folder.

- Place “jquery-3.2.1.js” file in “c:\ajax” folder.
- Create “httpserver.js” and “index.html” files in the “c:\ajax” folder.

### c:\ajax

- httpserver.js
- index.html
- jquery-3.2.1.js

### c:\ajax\httpserver.js

```
var http = require("http");
var fs = require("fs");
var querystring = require("querystring");
var server = http.createServer(engine);
server.listen(8080, startup);

function startup()
{
    console.log("Server started at port 8080");
}

function engine(request, response)
{
    if (request.url == "/" || request.url == "/index.html")
    {
        fs.readFile("index.html", "utf8", fun1);
        function fun1(error, data)
        {
            if (error)
            {
                response.setHeader("content-type", "text/html");
                response.writeHead(404);
                response.write("<h1 style='color:red'>Page not found</h1>");
                response.end();
            }
            else
            {
                response.setHeader("content-type", "text/html");
                response.writeHead(200);
                response.write(data);
                response.end();
            }
        }
    }
    else if (request.url == "/jquery-3.2.1.js")
    {
        fs.readFile("jquery-3.2.1.js", "utf8", fun1);
        function fun1(error, data)
        {
            if (error)
```

```

    {
        response.setHeader("content-type", "text/html");
        response.writeHead(404);
        response.write("<h1 style='color:red'>Page not found</h1>");
        response.end();
    }
    else
    {
        response.setHeader("content-type", "text/javascript");
        response.writeHead(200);
        response.write(data);
        response.end();
    }
}
}
else if (request.url.startsWith("/deleteemployee"))
{
    console.log("request received at " + new Date().toLocaleTimeString());
    response.setHeader("content-type", "text/html");
    response.write("Successfully Deleted");
    response.end();
}
}

```

c:\ajax\index.html

---

```

<!DOCTYPE html>
<html>
<head>
<title>NodeJS - jQuery AJAX - Delete</title>
<style>
    body, input
    {
        font-family: Tahoma;
        font-size: 24px;
    }
</style>
</head>
<body>
<h1>NodeJS - jQuery AJAX -Delete</h1>
<form>
    Existing Emp ID: <input type="text" id="txt1"><br>
    <input type="submit" id="btn1" value="Delete">
    <div id="div1"></div>
</form>

<script src="jquery-3.2.1.js"></script>

<script>
    $("#btn1").click(fun1);
    function fun1(event)
    {
        event.preventDefault();
    }

```

```

$.ajax({ type: "DELETE", url: "/deleteemployee?empid=" + $("#txt1").val(), success: funz });
}

function funz(response)
{
    $("#div1").html(response);
}
</script>
</body>
</html>

```

### Execution:

- Download and install “nodejs” from “https://nodejs.org”.
- Open “Command Prompt” and run the following commands:  
 cd c:\ajax  
 node httpserver.js
- Open browser and enter the url: http://localhost:8080/index.html

## jQuery AJAX – .NET – Simple - Example

### Creating the application

- Create “c:\ajax” folder.
- Open Visual Studio 2017.
- Click on “File” menu – “New” – “Project”.
- Select “.NET Framework 4.6”. Select “Visual C#”.
- Select “ASP.NET Web Application (.NET Framework)”.
- Name: AjaxSimple
- Location: c:\ajax
- Solution name: AjaxSimple
- Click on OK.
- Click on “Empty”. Check the check boxes “MVC” and “Web API”.
- Click on OK.
- Open Solution Explorer.
- Place “jquery-3.2.1.js” file in the project.
- Right click on the “Controllers” folder and click on “Add” – “Controller”. Select “MVC 5 Controller – Empty”. Click on “Add”. Enter the controller name as “HomeController”. Click on “Add”.

- Right click on “Views\Home” folder and click on “Add” – “View”. Enter the view name as “Index”. Select the template as “Empty (without model)”. Uncheck the checkbox “Use a layout page”. Click on “Add”.

### **AjaxSimple**

- jquery-3.2.1.js
- Controllers
  - HomeController.cs
- Views
  - Home
    - Index.cshtml

### **Controllers\HomeController.cs**

---

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace AjaxSimple.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }
        public ActionResult GetData()
        {
            return Content("Hello from Server at " + DateTime.Now);
        }
    }
}
```

### **Views\Home\Index.cshtml**

---

```
<!DOCTYPE html>
<html>
<head>
    <title>jQuery AJAX - Simple</title>
    <style type="text/css">
        body, input
        {
            font-family: Tahoma;
            font-size: 24px;
        }
    </style>
```

```
</head>
<body>
  <h1>jQuery AJAX - Simple</h1>
  <input type="button" id="btn1" value="Get data from server">
  <div id="div1"></div>

  <script src="~/jquery-3.2.1.js"></script>

  <script>
    $("#btn1").click(fun1);
    function fun1(event)
    {
      event.preventDefault();
      $.ajax({ type: "GET", url: "/home/getdata", success: fun2 });
    }
    function fun2(response)
    {
      $("#div1").html(response);
    }
  </script>
</body>
</html>
```

**Execution:**

- Go to “Debug” menu – “Start Debugging” in Visual Studio.

**jQuery AJAX – .NET – Get - Example**

**Creating the application**

- Create “c:\ajax” folder.
- Open Visual Studio 2017.
- Click on “File” menu – “New” – “Project”.
- Select “.NET Framework 4.6”.
- Select “Visual C#”.
- Select “ASP.NET Web Application (.NET Framework)”.
- Name: AjaxGet
- Location: c:\ajax
- Solution name: AjaxGet
- Click on OK.
- Click on “Empty”.
- Check the check boxes “MVC” and “Web API”.
- Click on OK.

- Open Solution Explorer.
- Place “jquery-3.2.1.js” file in the project.
- Right click on “Models” and click on “Add” – “New Item”. Click on “Visual C#” – “Class”. Enter the name as “Employee.cs”. Click on “Add”.
- Right click on the “Controllers” folder and click on “Add” – “Controller”. Select “MVC 5 Controller – Empty”. Click on “Add”. Enter the controller name as “HomeController”. Click on “Add”.
- Right click on “Views\Home” folder and click on “Add” – “View”. Enter the view name as “Index”. Select the template as “Empty (without model)”. Uncheck the checkbox “Use a layout page”. Click on “Add”.

### **AjaxGet**

- jquery-3.2.1.js
  - Models
    - Employee.cs
  - Controllers
    - HomeController.cs
  - Views
    - Home
      - Index.cshtml

### **Models\Employee.cs**

---

**using** System;

**namespace** AjaxGet.Models

```
{  
    public class Employee  
    {  
        public int empid { get; set; }  
        public string empname { get; set; }  
        public double salary { get; set; }  
    }  
}
```

### **Controllers\HomeController.cs**

---

**using** System;

**using** System.Collections.Generic;

**using** System.Linq;

**using** System.Web;

**using** System.Web.Mvc;

**using** AjaxGet.Models;

```
namespace AjaxGet.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }
        public ActionResult GetData()
        {
            List<Employee> emps = new List<Employee>()
            {
                new Employee() { empid = 1, empname = "Scott", salary = 4000 },
                new Employee() { empid = 2, empname = "Allen", salary = 7500 },
                new Employee() { empid = 3, empname = "Jones", salary = 9200 },
                new Employee() { empid = 4, empname = "James", salary = 8400 },
                new Employee() { empid = 5, empname = "Smith", salary = 5600 }
            };
            return Json(emps, JsonRequestBehavior.AllowGet);
        }
    }
}
```

#### Views\Home\Index.cshtml

```
<!DOCTYPE html>
<html>
<head>
    <title>jQuery AJAX - Get</title>
    <style>
        body, input, table
        {
            font-family: Tahoma;
            font-size: 24px;
        }
    </style>
</head>
<body>
    <h1>jQuery AJAX - Get</h1>
    <form>
        <input type="submit" id="button1" value="Get Data from Database">
        <table id="table1" border="1">
            <tr> <th>Emp ID</th> <th>Emp Name</th> <th>Salary</th> </tr>

            </table>
        </form>

        <script src="~/jquery-3.2.1.js"></script>

        <script>
            $("#button1").click(fun1);
            function fun1(event)
            {
                event.preventDefault();
            }
        </script>
    </body>
</html>
```



```
$.ajax({ type: "GET", url: "/home/getdata", success: fun2 });
}

function fun2(response)
{
    $("#table1 tr:gt(0)").remove();
    for (var i = 0; i < response.length; i++)
    {
        $("#table1").append("<tr><td>" + response[i].empid + "</td><td>" + response[i].empname + "</td><td>" +
response[i].salary + "</td></tr>");
    }
}
</script>
</body>
</html>
```

**Execution:**

- Go to “Debug” menu – “Start Debugging” in Visual Studio.

**jQuery AJAX – .NET – Search - Example**

**Creating the application**

- Create “c:\ajax” folder.
- Open Visual Studio 2017.
- Click on “File” menu – “New” – “Project”.
- Select “.NET Framework 4.6”.
- Select “Visual C#”.
- Select “ASP.NET Web Application (.NET Framework)”.
- Name: AjaxSearch
- Location: c:\ajax
- Solution name: AjaxSearch
- Click on OK.
- Click on “Empty”.
- Check the check boxes “MVC” and “Web API”.
- Click on OK.
- Open Solution Explorer.
- Place “jquery-3.2.1.js” file in the project.
- Right click on “Models” and click on “Add” – “New Item”. Click on “Visual C#” – “Class”. Enter the name as “Employee.cs”. Click on “Add”.

- Right click on the “Controllers” folder and click on “Add” – “Controller”. Select “MVC 5 Controller – Empty”. Click on “Add”. Enter the controller name as “HomeController”. Click on “Add”.
- Right click on “Views\Home” folder and click on “Add” – “View”. Enter the view name as “Index”. Select the template as “Empty (without model)”. Uncheck the checkbox “Use a layout page”. Click on “Add”.

### **AjaxSearch**

- jquery-3.2.1.js

- Models
  - Employee.cs
- Controllers
  - HomeController.cs
- Views
  - Home
    - Index.cshtml

#### **Models\Employee.cs**

---

**using** System;

**namespace** AjaxSearch.Models

```
{
    public class Employee
    {
        public int empid { get; set; }
        public string empname { get; set; }
        public double salary { get; set; }
    }
}
```

#### **Controllers\HomeController.cs**

---

**using** System;

**using** System.Collections.Generic;

**using** System.Linq;

**using** System.Web.Mvc;

**using** AjaxSearch.Models;

**namespace** AjaxSearch.Controllers

```
{
    public class HomeController : Controller
    {
        //GET: Home/Index
        public ActionResult Index()
        {

```

```
        return View();
    }

    //GET: Home/searchemployees
    public ActionResult searchemployees(string searchstr)
    {
        List<Employee> emps = new List<Employee>()
        {
            new Employee() { empid = 1, empname = "Scott", salary = 4000 },
            new Employee() { empid = 2, empname = "Allen", salary = 7500 },
            new Employee() { empid = 3, empname = "Jones", salary = 9200 },
            new Employee() { empid = 4, empname = "James", salary = 8400 },
            new Employee() { empid = 5, empname = "Smith", salary = 5600 }
        };

        List<Employee> emps2 = new List<Employee>();
        for (int i = 0; i < emps.Count; i++)
        {
            if (emps[i].empname.Contains(searchstr))
            {
                emps2.Add(emps[i]);
            }
        }
        return Json(emps2, JsonRequestBehavior.AllowGet);
    }
}
```

#### Views\Home\Index.cshtml

---

```
<!DOCTYPE html>
<html>
<head>
    <title>jQuery AJAX - Search</title>
    <style>
        body, input, table
        {
            font-family: Tahoma;
            font-size: 24px;
        }
    </style>
</head>
<body>
    <h1>jQuery AJAX - Search</h1>
    <form>
        Search employees: <input type="text" id="txt1">
        <input type="submit" id="btn1" value="Search">
        <table id="table1" border="1">
            <tr>
                <th>Emp ID</th>
                <th>Emp Name</th>
                <th>Salary</th>
            </tr>

            </table>
    </form>
```

```
<script src="~/jquery-3.2.1.js"></script>

<script>
    $("#btn1").click(fun1);
    function fun1(event)
    {
        event.preventDefault();
        $.ajax({ type: "GET", url: "/home/searchemployees?searchstr=" + $("#txt1").val(), success: fun2 });
    }

    function fun2(response)
    {
        $("#table1 tr:gt(0)").remove();
        for (var i = 0; i < response.length; i++)
        {
            $("#table1").append("<tr><td>" + response[i].empid + "</td><td>" + response[i].empname + "</td><td>" +
response[i].salary + "</td></tr>");
        }
    }
</script>
</body>
</html>
```

**Execution:**

- Go to “Debug” menu – “Start Debugging” in Visual Studio.

## jQuery AJAX – .NET – Post - Example

**Creating the application**

- Create “c:\ajax” folder.
- Open Visual Studio 2017.
- Click on “File” menu – “New” – “Project”.
- Select “.NET Framework 4.6”.
- Select “Visual C#”.
- Select “ASP.NET Web Application (.NET Framework)”.
- Name: AjaxPost
- Location: c:\ajax
- Solution name: AjaxPost
- Click on OK.
- Click on “Empty”.
- Check the check boxes “MVC” and “Web API”.
- Click on OK.

- Open Solution Explorer.
- Place “jquery-3.2.1.js” file in the project.
- Right click on “Models” and click on “Add” – “New Item”. Click on “Visual C#” – “Class”. Enter the name as “Employee.cs”. Click on “Add”.
- Right click on the “Controllers” folder and click on “Add” – “Controller”. Select “MVC 5 Controller – Empty”. Click on “Add”. Enter the controller name as “HomeController”. Click on “Add”.
- Right click on “Views\Home” folder and click on “Add” – “View”. Enter the view name as “Index”. Select the template as “Empty (without model)”. Uncheck the checkbox “Use a layout page”. Click on “Add”.

## AjaxPost

- jquery-3.2.1.js

- Models
  - Employee.cs
- Controllers
  - HomeController.cs
- Views
  - Home
    - Index.cshtml

### Models\Employee.cs

`using System;`

`namespace AjaxPost.Models`

```
{
    public class Employee
    {
        public int empid { get; set; }
        public string empname { get; set; }
        public double salary { get; set; }
    }
}
```

### Controllers\HomeController.cs

`using System;`

`using System.Collections.Generic;`

`using System.Linq;`

`using System.Web.Mvc;`

`using AjaxPost.Models;`

`namespace AjaxPost.Controllers`

```
{
```

```
public class HomeController : Controller
{
    //GET: Home/Index
    public ActionResult Index()
    {
        return View();
    }

    //GET: Home/InsertEmployee
    public ActionResult InsertEmployee(Employee emp)
    {
        return Content("Successfully Inserted");
    }
}
}
```

### Views\Home\Index.cshtml

---

```
<!DOCTYPE html>
<html>
<head>
<title>jQuery AJAX - Post</title>
<style>
    body, input
    {
        font-family: Tahoma;
        font-size: 24px;
    }
</style>
</head>
<body>
<h1>jQuery AJAX - Post</h1>
<form>
    Emp ID: <input type="text" id="txt1"><br>
    Emp Name: <input type="text" id="txt2"><br>
    Salary: <input type="text" id="txt3"><br>
    <input type="submit" id="button1" value="Insert">
    <div id="div1"></div>
</form>

<script src="~/jquery-3.2.1.js"></script>

<script>
    $("#button1").click(fun1);
    function fun1(event)
    {
        event.preventDefault();
        var mydata = { "empid": $("#txt1").val(), "empname": $("#txt2").val(), "salary": $("#txt3").val() };
        $.ajax({ type: "POST", url: "/home/insertemployee", success: fun2, data: mydata });
    }

    function fun2(response)
    {

```

```
    $("#div1").html(response);  
}  
</script>  
</body>  
</html>
```

**Execution:**

- Go to “Debug” menu – “Start Debugging” in Visual Studio.

**jQuery AJAX – .NET – Put - Example**

**Creating the application**

- Create “c:\ajax” folder.
- Open Visual Studio 2017.
- Click on “File” menu – “New” – “Project”.
- Select “.NET Framework 4.6”.
- Select “Visual C#”.
- Select “ASP.NET Web Application (.NET Framework)”.
- Name: AjaxPut
- Location: c:\ajax
- Solution name: AjaxPut
- Click on OK.
- Click on “Empty”.
- Check the check boxes “MVC” and “Web API”.
- Click on OK.
- Open Solution Explorer.
- Place “jquery-3.2.1.js” file in the project.
- Right click on “Models” and click on “Add” – “New Item”. Click on “Visual C#” – “Class”. Enter the name as “Employee.cs”. Click on “Add”.
- Right click on the “Controllers” folder and click on “Add” – “Controller”. Select “MVC 5 Controller – Empty”. Click on “Add”. Enter the controller name as “HomeController”. Click on “Add”.
- Right click on “Views\Home” folder and click on “Add” – “View”. Enter the view name as “Index”. Select the template as “Empty (without model)”. Uncheck the checkbox “Use a layout page”. Click on “Add”.

**AjaxPut**

- jquery-3.2.1.js
  - Models
    - Employee.cs
  - Controllers
    - HomeController.cs
  - Views
    - Home
      - Index.cshtml

### Models\Employee.cs

---

using System;

```
namespace AjaxPut.Models
{
    public class Employee
    {
        public int empid { get; set; }
        public string empname { get; set; }
        public double salary { get; set; }
    }
}
```

### Controllers\HomeController.cs

---

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web.Mvc;
using AjaxPut.Models;
namespace AjaxPut.Controllers
{
    public class HomeController : Controller
    {
        // GET: Home/Index
        public ActionResult Index()
        {
            return View();
        }

        // GET: Home/UpdateEmployee
        public ActionResult UpdateEmployee(Employee emp)
        {
            return Content("Successfully Updated");
        }
    }
}
```



**Views\Home\Index.cshtml**

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery AJAX - Put</title>
  <style>
    body, input
    {
      font-family: Tahoma;
      font-size: 24px;
    }
  </style>
</head>
<body>
  <h1>jQuery AJAX - Put</h1>
  <form>
    Existing Emp ID: <input type="text" id="txt1"><br>
    Emp Name: <input type="text" id="txt2"><br>
    Salary: <input type="text" id="txt3"><br>
    <input type="submit" id="button1" value="Update">
    <div id="div1"></div>
  </form>

  <script src="~/jquery-3.2.1.js"></script>
  <script>
    $("#button1").click(fun1);
    function fun1(event)
    {
      event.preventDefault();
      var mydata = { "empid": $("#txt1").val(), "empname": $("#txt2").val(), "salary": $("#txt3").val() };
      $.ajax({ type: "PUT", url: "/home/updateemployee", data: mydata, success: fun2 });
    }

    function fun2(response)
    {
      $("#div1").html(response);
    }
  </script>
</body>
</html>
```

**Execution:**

- Go to “Debug” menu – “Start Debugging” in Visual Studio.

**jQuery AJAX – .NET – Delete - Example**

**Creating the application**

- Create “c:\ajax” folder.

- Open Visual Studio 2017.
- Click on “File” menu – “New” – “Project”.
- Select “.NET Framework 4.6”.
- Select “Visual C#”.
- Select “ASP.NET Web Application (.NET Framework)”.
- Name: AjaxDelete
- Location: c:\ajax
- Solution name: AjaxDelete
- Click on OK.
- Click on “Empty”.
- Check the check boxes “MVC” and “Web API”.
- Click on OK.
- Open Solution Explorer.
- Place “jquery-3.2.1.js” file in the project.
- Right click on the “Controllers” folder and click on “Add” – “Controller”. Select “MVC 5 Controller – Empty”. Click on “Add”. Enter the controller name as “HomeController”. Click on “Add”.
- Right click on “Views\Home” folder and click on “Add” – “View”. Enter the view name as “Index”. Select the template as “Empty (without model)”. Uncheck the checkbox “Use a layout page”. Click on “Add”.

### AjaxDelete

- jquery-3.2.1.js
  - Controllers
    - HomeController.cs
  - Views
    - Home
      - Index.cshtml

### Controllers\HomeController.cs

---

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web.Mvc;

namespace AjaxDelete.Controllers
{
    public class HomeController : Controller
    {
```

```
//GET: Home/Index
public ActionResult Index()
{
    return View();
}

//GET: Home/DeleteEmployee
public ActionResult DeleteEmployee(int empid)
{
    return Content("Successfully Deleted");
}
}
```

#### Views\Home\Index.cshtml

---

```
<!DOCTYPE html>
<html>
<head>
<title>jQuery AJAX - Delete</title>
<style type="text/css">
    body, input
    {
        font-family: Tahoma;
        font-size: 24px;
    }
</style>
</head>
<body>
<h1>jQuery AJAX - Delete</h1>
<form>
    Existing Emp ID: <input type="text" id="txt1"><br>
    <input type="button" id="button1" value="Delete">
    <div id="div1"></div>
</form>

<script src="~/jquery-3.2.1.js"></script>

<script>
    $("#button1").click(fun1);
    function fun1()
    {
        $.ajax({ type: "DELETE", url: "/home/deleteemployee?empid=" + $("#txt1").val(), success: fun2 });
    }
    function fun2(response)
    {
        $("#div1").html(response);
    }
</script>
</body>
</html>
```

**Execution:**

- Go to “Debug” menu – “Start Debugging” in Visual Studio.

**jQuery AJAX – .NET – Grid - Example**

**Creating the application**

- Create “c:\ajax” folder.
- Open Visual Studio 2017.
- Click on “File” menu – “New” – “Project”.
- Select “.NET Framework 4.6”.
- Select “Visual C#”.
- Select “ASP.NET Web Application (.NET Framework)”.
- Name: AjaxGrid
- Location: c:\ajax
- Solution name: AjaxGrid
- Click on OK.
- Click on “Empty”.
- Check the check boxes “MVC” and “Web API”.
- Click on OK.
- Open Solution Explorer.
- Place “jquery-3.2.1.js” file in the project.
- Right click on “Models” and click on “Add” – “New Item”. Click on “Visual C#” – “Class”. Enter the name as “Employee.cs”. Click on “Add”.
- Right click on the “Controllers” folder and click on “Add” – “Controller”. Select “MVC 5 Controller – Empty”. Click on “Add”. Enter the controller name as “HomeController”. Click on “Add”.
- Right click on “Views\Home” folder and click on “Add” – “View”. Enter the view name as “Index”. Select the template as “Empty (without model)”. Uncheck the checkbox “Use a layout page”. Click on “Add”.

**AjaxGrid**

- jquery-3.2.1.js
  - Models
    - Employee.cs
  - Controllers
    - HomeController.cs
  - Views

- Home
  - Index.cshtml

### Models\Employee.cs

---

```
using System;
```

```
namespace AjaxGrid.Models
{
    public class Employee
    {
        public int empid { get; set; }
        public string empname { get; set; }
        public double salary { get; set; }
    }
}
```

### Controllers\HomeController.cs

---

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web.Mvc;
using AjaxGrid.Models;
```

```
namespace AjaxGrid.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }
        public ActionResult GetData()
        {
            List<Employee> emps = new List<Employee>()
            {
                new Employee() { empid = 1, empname = "Scott", salary = 4000 },
                new Employee() { empid = 2, empname = "Allen", salary = 7500 },
                new Employee() { empid = 3, empname = "Jones", salary = 9200 },
                new Employee() { empid = 4, empname = "James", salary = 8400 },
                new Employee() { empid = 5, empname = "Smith", salary = 5600 }
            };
            return Json(emps, JsonRequestBehavior.AllowGet);
        }

        public ActionResult InsertEmployee(Employee emp)
        {
            return Content("Successfully Inserted");
        }
    }
}
```

```

public ActionResult UpdateEmployee(Employee emp)
{
    return Content("Successfully Updated");
}

public ActionResult DeleteEmployee(int empid)
{
    return Content("Successfully Deleted");
}
}

```

#### Views\Home\Index.cshtml

---

```

<!DOCTYPE html>
<html>
<head>
<title>jQuery AJAX - Grid</title>
<style>
    body, input, table
    {
        font-family: Tahoma;
        font-size: 24px;
    }
    #table1 td
    {
        padding: 5px;
    }
</style>
</head>
<body>
<h1>jQuery AJAX - Grid</h1>
<form>
    <table id="table1" border="1">
        <tr> <th>Emp ID</th> <th>Emp Name</th> <th>Salary</th> <th></th></tr>

        </table>
    </form>

    <script src="~/jquery-3.2.1.js"></script>

    <script>
        //get
        fun1();
        function fun1()
        {
            $.ajax({ type: "GET", url: "/home/getdata", success: fun2 });
        }
        function fun2(response)
        {
            $("#table1 tr:gt(0)").remove();
            for (var i = 0; i < response.length; i++)
            {

```

```

        $("#table1").append("<tr><td>" + response[i].empid + "</td><td>" + response[i].empname + "</td><td>" +
        response[i].salary + "</td><td><input type='button' value='Edit' class='class1'> <input type='button' value='Delete'
        class='class2'></td></tr>");
    }
    $("#table1").append("<tr><td><input type='text' class='class3' placeholder='Emp ID'></td><td><input type='text'
    class='class3' placeholder='Emp Name'></td><td><input type='text' class='class3' placeholder='Salary'></td>
    <td><input type='button' value='Insert' class='class4'></td></tr>");
}

//post
$(document).on("click", ".class4", fun3);
function fun3()
{
    var d={ "empid": $(".class3:eq(0)").val(), "empname": $(".class3:eq(1)").val(), "salary": $(".class3:eq(2)").val() };
    $.ajax({ type: "POST", url: "/home/insertemployee", data: d, success: fun4 });
}
function fun4(response)
{
    $("#table1 tr:last").before("<tr><td>" + $(".class3:eq(0)").val() + "</td><td>" + $(".class3:eq(1)").val() + "</td><td>"
    + $(".class3:eq(2)").val() + "</td><td><input type='button' value='Edit' class='class1'> <input type='button'
    value='Delete' class='class2'></td></tr>");
    $(".class3").val("");
}

//put
var currentrow = null;
$(document).on("click", ".class1", fun5);
function fun5()
{
    currentrow = $(this).parent().parent();
    currentrow.html("<td><input type='text' class='class5' placeholder='Emp ID' value='' +
    currentrow.find("<td:eq(0)>").text() + "></td><td><input type='text' class='class5' placeholder='Emp Name' value='' +
    currentrow.find("<td:eq(1)>").text() + "></td><td><input type='text' class='class5' placeholder='Salary' value='' +
    currentrow.find("<td:eq(2)>").text() + "></td><td><input type='button' value='Update' class='class6'></td>");
    currentrow.find(".class5:eq(0)").focus();
}
$(document).on("click", ".class6", fun6);
function fun6()
{
    var d={ "empid": $(".class5:eq(0)").val(), "empname": $(".class5:eq(1)").val(), "salary": $(".class5:eq(2)").val() };
    $.ajax({ type: "PUT", url: "/home/updateemployee", data: d, success: fun7 });
}
function fun7(response)
{
    currentrow.html("<td>" + $(".class5:eq(0)").val() + "</td><td>" + $(".class5:eq(1)").val() + "</td><td>" +
    $(".class5:eq(2)").val() + "</td><td><input type='button' value='Edit' class='class1'> <input type='button' value='Delete'
    class='class2'></td>");
}

//delete
var currentrow2 = null;
$(document).on("click", ".class2", fun8);
function fun8()
{
    currentrow2 = $(this).parent().parent();

```

```
var empid = currentrow2.find("td:eq(0)").text();
var empname = currentrow2.find("td:eq(1)").text();
if (confirm("Are you sure to delete " + empname + "?"))
{
    $.ajax({ type: "DELETE", url: "/home/deleteemployee?empid=" + empid, success: fun9 });
}
}
function fun9(response)
{
    currentrow2.remove();
}
</script>
</body>
</html>
```

### Execution:

- Go to “Debug” menu – “Start Debugging” in Visual Studio.