jQuery

- Brendan Eich Introduced **Mocha -> Live Script -> JavaScript**
- JavaScript was developed for “Netscape Communicator” browser in 1994.
- JavaScript belongs to Netscape and Maintained by “Sun Microsystems”.
- 2002 Netscape stopped JavaScript.
- ECMA took responsibility for JavaScript.
- Internet Exporer, Safari, Opera, etc. using JavaScript.
- Every browser started extending JavaScript by its own.
- **Browser incompatibility issues started.**
- 2003 W3C started maintaining JavaScript with “WHATWG”.
- W3C & WHATWG are extending HTML and JavaScript.
- 2006 “John Resig” introduced a library form JavaScript called “jQuery”.
- It provides pre-defined plugins for browser to reduce compatibility issues.
- Latest jQuery version 3.5

jQuery Features:

- Very Fast
- Extensible, Maintainable, Testable
- Provides Plugins
- **Write less and do more**
- Improves the performance
- Light weight
- Cross platform
- Reduces compatibility issues.
- It can reach broad range of devices. [mobile / browser]
- Support differential loading
- It is a library [Plugins]
- Other jQuery libraries are

- RxJS
- React [Facebook - JSX]
- Glimmer.js
- JQlite

Note: Library can't control the application flow. To control application flow better use a framework. [Angular JS, Angular, Ember JS, Knockout JS etc.].

jQuery Characteristics

- HTML Manipulation
- DOM Manipulation
- Dynamic DOM Manipulation
- CSS Manipulations
- Effects and Animations
- Asynchronous and Modular [Ajax]
- JSON parsing

Setup Environment for jQuery

- You can use any package manager on your computer to download and install jQuery for Project
  - NPM,
  - Yarn,
  - RubyGems
  - NuGet
- Download "Node JS" on your computer for "NPM" package manager
  https://nodejs.org/en/download/
- Test from command prompt
  > npm -v [npm version]
- Go to your project location and open terminal [Ctrl + `]
- Run the command:
  > npm install jquery
  > npm install bootstrap

Note: Your library is maintained in a folder **“node_modules”.**

jQuery Library**:** **Project\node_modules\jquery\dist\jquery.js**

jQuery with HTML

- **Import jQuery library into page**
  <script src="../node_modules/jquery/dist/jquery.js"></script>
- **Entire jQuery library is defined in an anonymous function, you can invoke the library by using “()”.**
- **Implement the library functions and used in your HTML page.**
- $ is a selector for functions to implement in jQuery.

Syntax:

```
<script>
 $(function(){

 })
</script>
```

function(){} is a call back function of jQuery library, which will load entire jQuery factory into browser memory.

Ex: Good for Browser Manipulations. In browser manipulations we have DOM manipulations.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Demo</title>
    <script src="../node_modules/jquery/dist/jquery.js"></script>
    <script type="text/javascript">
```

```
        $(function(){
            document.write("Hello ! JQuery");
        })
    </script>
  </head>
  <body>

  </body>
</html>
```

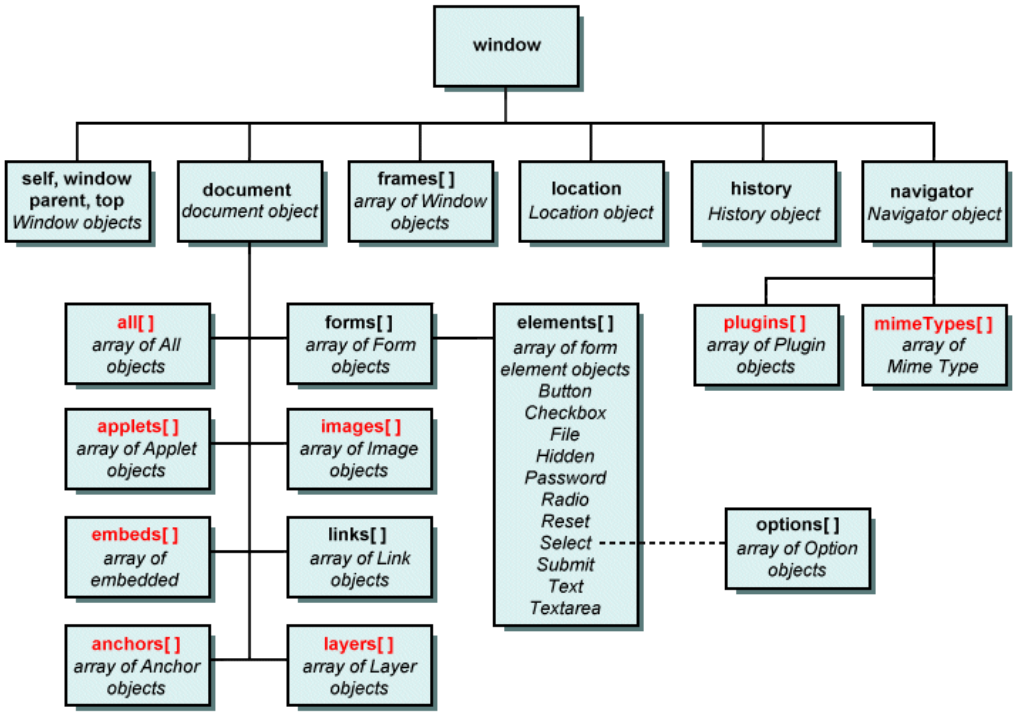
Ex: Good for “DOM” manipulation

```
<!DOCTYPE html>
<html>
  <head>
    <title>Demo</title>
    <script src="../node_modules/jquery/dist/jquery.js"></script>
    <script type="text/javascript">
        $(document).ready(function(){
            document.write("Welcome to JQuery");
        })
    </script>
  </head>
  <body>
  </body>
```

</html>

jQuery DOM Manipulations

- JavaScript can use DOM hierarchy, document methods, name reference to access HTML elements.



- jQuery uses all “CSS” selectors to access HTML elements.
  - Combinators
  - Basic Selector
  - Pseudo Selectors etc.
- **Basic Selectors**
  - Id selector
  - Type Selector
  - Class Selector
- **Type Selector**
  - It uses the tag name as element reference.

    $(“h2”)

    $(“div”)

    $(“td”)

  Ex:

  <!DOCTYPE html>

```
<html>
  <head>
    <title>Selectors</title>
    <script
src="../node_modules/jquery/dist/jquery.js"></script>
    <script>
      $(function(){
        $("h2").text("Welcome to jQuery");
      })
    </script>
  </head>
  <body>
    <div>
      <h2></h2>
    </div>
  </body>
</html>
```

- **ID Selector**
  - It uses ID reference.
  - ID is access with “#”

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Selectors</title>
    <script src="../node_modules/jquery/dist/jquery.js"></script>
    <script>
      $(function(){
        $("#msg").text("Welcome to jQuery");
```

```
        })
    </script>
  </head>
  <body>
    <div>
      <h2 id="msg"></h2>
    </div>
  </body>
</html>
```

- **Class Selector**
  - It refers to a class name.
  - Class is referred using ".className".
  - Every element can be defined with multiple classes.

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Selectors</title>
    <script src="../node_modules/jquery/dist/jquery.js"></script>
    <script>
      $(function(){
        $(".msg").text("Welcome to jQuery");
      })
    </script>
  </head>
```

```
  <body>
    <div>
      <h2 class="msg"></h2>
    </div>
  </body>
</html>
```

jQuery Methods to Manipulate HTML Elements

Method	Description
text()	It is similar to innerText in JavaScript. It is RC data method. Formats not allowed Ex: **$(".msg").text("Welcome to jQuery");**
html()	It is similar to innerHTML in JavaScript. Formats allowed Ex: **$(".msg").html("Welcome to jQuery");**
val()	It is used for element that can display value, like Button, textbox, option, radio etc. It allows to get and set value. Ex: <!DOCTYPE html> <html> <head> <title>Selectors</title> <script src="../node_modules/jquery/dist/jquery.js"></script> <script> $(function(){ $(".msg").html("Welcome to jQuery"); $("#btnSubmit").val("Register");

	}) </script> </head> <body> <input type="button" id="btnSubmit"> </body> </html>
prop()	It is used to access any property of an element and verify its value. You can access any property by using “prop()” Syntax: $(selector).prop(“property”)==value
attr()	It is used to get or set an attribute value of element. Ex: <!DOCTYPE html> <html> <head> <script src="../node_modules/jquery/dist/jquery.js"></script> <script> $(function(){ $("#tbl").attr("height",200); $("#tbl").prop("border",1); }) </script> </head> <body> <table id="tbl"> <tr>

	<td>Name</td> <td>Price</td> </tr> </table> </body> </html>

Ex: List all properties of Element

```
<script>
   function f1(){
     var tbl = document.createElement("table");
     for(var property in tbl){
      document.write(`${property}<br>`);
     }
   }
   f1();
</script>
```

Ex:

```
<!DOCTYPE html>
<html>
   <head>
      <title>Value Demo</title>
      <link rel="stylesheet"
href="../node_modules/bootstrap/dist/css/bootstrap.css">
      <script src="../node_modules/jquery/dist/jquery.js"></script>
      <script>
         $(function(){
```

```
        $("#btnSubmit").click(function(){
            $("#lblName").text($("#txtName").val());
            $("#lblPrice").text($("#txtPrice").val());
            $("#lblCity").text($("#lstCities").val());
            if($("#optStock").prop("checked")==true){
                $("#lblStock").text("Available");
            } else {
                $("#lblStock").text("Out of Stock");
            }
        })
    })
  </script>
</head>
<body class="container-fluid">
  <h2 class="text-primary text-center">Amazon Shopping</h2>
  <div class="row">
    <div class="col-3">
      <div class="form-group">
        <label>Name</label>
        <div>
          <input id="txtName" type="text" class="form-control">
        </div>
      </div>
      <div class="form-group">
```

```
            <label>Price</label>
            <div>
                <input id="txtPrice" type="text" class="form-control">
            </div>
        </div>
        <div class="form-group">
            <label>Shipped To</label>
            <div>
                <select id="lstCities" class="form-control">
                <option value="Delhi">Delhi</option>
                <option value="Hyd">Hyd</option>
                <option value="Mumbai">Mumbai</option>
                </select>
            </div>
        </div>
        <div class="form-group">
            <label>In Stock</label>
            <div>
                <input id="optStock" type="checkbox"> Yes
            </div>
        </div>
        <div class="form-group">
            <button id="btnSubmit" class="btn btn-primary btn-
block">Submit</button>
```

```
            </div>
        </div>
        <div class="col-9">
            <table class="table table-hover">
                <tr>
                    <td>Name</td>
                    <td id="lblName"></td>
                </tr>
                <tr>
                    <td>Price</td>
                    <td id="lblPrice"></td>
                </tr>
                <tr>
                    <td>City</td>
                    <td id="lblCity"></td>
                </tr>
                <tr>
                    <td>Stock Status</td>
                    <td id="lblStock"></td>
                </tr>
            </table>
        </div>
    </div>
</body>
```

</html>

FAQ: What is difference between attribute and property?

- In HTML tag what ever you write is an attribute.
- Attributes are Immutable.
- Attributes are used to configure element statically.
- **Properties are defined dynamically.**
- **Whatever you use dynamically it is a property.**

<img src=”path”> src is an attribute for “img” tag.

var pic = new Image();

pic.src = “path”; **src is a property of Image element.**

- **HTML attribute may not match with properties every time.**

  <div id=”msg” **class**=”text-primary”>

  <img id=”pic” **src**=”pic.jpg”>

  document.getElementById(“msg”).**className** = “text-primary”;

  document.getElementById(“pic”).**src** = “pic.jpg”;

  Note: className property is mapping to **class** attribute.

- **HTM Attribute may not have relative property every time**
  Ex: Table tag have “height” attribute but it doesn’t have “height” property.

FAQ: How can we apply height for table dynamically?

A. By using styles or CSS classes dynamically.

Method	**Description**

css()	It allows to configure styles for element dynamically. Syntax: $(selector).css(“PropertyName”); **Access the css property** $(selector).css(“Property”, “value”); **Set the CSS property** $(selector).css({Property:value, Property:value}) **multiple**
addClass()	Add a new CSS class to any element dynamically.
hasClass()	It verifies the availability of class in element.
toggleClass()	It is used to switch between the classes dynamically.

Ex:

```
<!DOCTYPE html>
<html>
  <head>
   <script src="../node_modules/jquery/dist/jquery.js"></script>
   <script>
      $(function(){
         $("#tbl").attr("height",200);
         $("#tbl").prop("border",1);
         $("button").click(function(){
            $("#tbl").css("background-color","lightgreen");
         })
      })
   </script>
  </head>
  <body>
```

```
      <div>
         <button>Apply Effects</button>
      </div>
      <table id="tbl">
         <tr>
            <td>Name</td>
            <td>Price</td>
         </tr>
      </table>
   </body>
</html>
```

Ex: Multiple Styles

```
<!DOCTYPE html>
<html>
   <head>
     <script src="../node_modules/jquery/dist/jquery.js"></script>
     <script>
        $(function(){
           $("#tbl").attr("height",200);
           $("#tbl").prop("border",1);
           $("button").click(function(){
              $("#tbl").css({"background-
color":"lightcyan","border":"2px dotted red"});
           })
```

```
        })
    </script>
  </head>
  <body>
    <div>
      <button>Apply Effects</button>
    </div>
    <table id="tbl">
      <tr>
        <td>Name</td>
        <td>Price</td>
      </tr>
    </table>
  </body>
</html>
```

Ex: Dropdown List Change

```
<!DOCTYPE html>
<html>
  <head>
    <script src="../node_modules/jquery/dist/jquery.js"></script>
    <script>
      $(function(){
        $("#tbl").attr("height",100);
```

```
            $("#tbl").prop("border",1);
            $("button").click(function(){
               $("#tbl").css({"background-
color":$("#lstColors").val(),"border":"2px dotted red"});
            })
            $("#lstColors").change(function(){
              $("#tbl").css({"background-
color":$("#lstColors").val(),"border":"2px dotted red"});
            })
        })
    </script>
  </head>
  <body>

      <fieldset>
         <legend>Select Colors</legend>
         <dl>
            <dt>Background Color</dt>
            <dd>
               <select id="lstColors">
                  <option value="red">Red</option>
                  <option value="green">Green</option>
                  <option value="yellow">Yellow</option>
               </select>
```

```
            </dd>
        </dl>
        <div>
            <button>Apply Effects</button>
        </div>
    </fieldset>
    <br><br>
    <table id="tbl">
        <tr>
            <td>Name</td>
            <td>Price</td>
        </tr>
    </table>
  </body>
</html>
```

insertAfter() or after()	Insert content after the element.
before()	Insert content before the element.
prepend()	Insert content left or above hand side. [prefix] Inside the specified element.
append()	Insert content right hand side. [suffix] To the existing content add new content. Inside the specified element. It defines the target element to source.
appendTo()	It defines the source to target.
remove()	It removes element from DOM hierarchy. It is removed from page.
detach()	It removes the selected elements and

	keeps only data and event.
empty()	It keeps the element and remove the content and child nodes.

$.each()	It is an iterator. It reads the elements from a collection in sequential order.
	
	Syntax:
	$.each(collection, function(key, value){
	  console.log(key / value);
	})

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="../node_modules/jquery/dist/jquery.js"></script>
    <script>
      $(function(){
        var products = ["TV", "Mobile", "Shoe"];
        $.each(products,function(key,value){
          alert(value + " " + key);
        })
      })
    </script>
  </head>
  <body>
```

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

Ex:**Append**

```
<!DOCTYPE html>
<html>
    <head>
        <script src="../node_modules/jquery/dist/jquery.js"></script>
        <script>
            $(function(){
                var categories = ["Electornics", "Footwear", "Fashion"];
                $.each(categories, function(key,value){
                    $("ol").append(`<li>${value}</li>`)
                })
            })
        </script>
    </head>
    <body>
        <ol>

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

Ex: AppendTo

```
<!DOCTYPE html>
<html>
  <head>
    <script src="../node_modules/jquery/dist/jquery.js"></script>
    <script>
      $(function(){
        var categories = ["Electornics", "Footwear", "Fashion"];
        $.each(categories, function(key,value){
          $(`<li>${value}</li>`).appendTo("ol");
        })
      })
    </script>
  </head>
  <body>
    <ol>

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

Ex: Adding Dropdown List

```
<!DOCTYPE html>
<html>
  <head>
```

```
    <script src="../node_modules/jquery/dist/jquery.js"></script>
    <script>
      $(function(){
        var categories = ["Electornics", "Footwear", "Fashion"];
        $.each(categories, function(key,value){
          $(`<option>${value}</option>`).appendTo("select");
        })
      })
    </script>
  </head>
  <body>

    <fieldset>
      <legend>Select a Category</legend>
      <select></select>
    </fieldset>
  </body>
</html>
```

Ex: **JavaScript Append Nested**

```
<head>
  <script>
    function bodyload(){
```

```
        var data = [{Category:"Electronics",
Products:["Mobile","Speaker"]},{Category:"Footwear",
Products:["Nike Casuals","Lee Boot"]}];
        for(var item of data)
        {
           var parentli = document.createElement("li");
           parentli.innerHTML = item.Category;
           for(var product of item.Products){
              var ul = document.createElement("ul");
              var childli = document.createElement("li");
              childli.innerHTML = product;
              ul.appendChild(childli);
              parentli.appendChild(ul);
           }
           document.getElementById("lst").appendChild(parentli);
        }
     }
  </script>
</head>
<body onload="bodyload()">
  <ol id="lst">

  </ol>
</body>
```

JQuery:

Create a Table Dynamically and Add rows collecting from Array

[

{ Name: “TV”, Price: 45000, Photo: “Images/tv.jpg” }

{ Name: “mobile”, Price: 15000, Photo: “Images/mobile.jpg” }

]

```
<table>
<tr>
  <td> TV </td>
  <td> 3500.55 </td>
   <td> <img src=”tv.jpg”> </td>
</tr>
</table>
```

Ex: Task – Nested Iterations

```
<!DOCTYPE html>
<html>
   <head>
      <script src="../node_modules/jquery/dist/jquery.js"></script>
      <script>
         $(function(){
            var data = [{Category:"Electronics",
Products:["Mobile","Speaker"]},{Category:"Footwear",
Products:["Nike Casuals","Lee Boot"]}];
```

```
            $.each(data, function(key, value){
              $(`<li>${value.Category}</li>`).appendTo("ol");
              $.each(value.Products,function(innerKey, innerValue){
                $(`<ul><li>${innerValue}</li></ul>`).appendTo("ol");
              })
            })
          })
      </script>
    </head>
    <body>

      <fieldset>
        <legend>Select a Category</legend>
        <ol>

        </ol>
      </fieldset>
    </body>
</html>
```

jQuery Events

- jQuery uses all JavaScript browser events
    - Mouse Events
        - mouseover()
        - mouseout()

        - mousedown()
        - mouseup()
        - mousemove()
    - Key Events
        - keyup()
        - keypress()
        - keydown()
    - Timer Events
        - setTimeout()
        - clearTimeout()
        - setInterval()
        - clearInterval()
    - Miscellaneous events
        - click()
        - dblclick()
        - contextmenu()
        - cut()
        - copy()
        - paste()
        - change()
        - load()
        - select()
        - submit() etc.

Syntax:

```
$("button").click(function(event){
})
```

this,event – JavaScript Event arguments

this – sends information about the current object. [value, class, id]

event – sends information about the current event. [clientX, clientY, shiftKey, ctrlKey, altKey]

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet"
href="../node_modules/bootstrap/dist/css/bootstrap.css">
    <link rel="stylesheet" href="../node_modules/fonts/css/all.css">
    <script src="../node_modules/jquery/dist/jquery.js"></script>
    <script>
      $(function(){
        var categories = ['Select a Category', 'Electronics',
'Footwear', 'Fashion'];
        var electronics = ['Select Electronic Product', 'JBL Speaker',
'Earpods'];
        var footwear = ['Select Footwear', 'Nike Casuals', 'Lee
Cooper Boot'];
        var fashion = ['Select Fashion Product', 'Shirt', 'Jeans'];
        var products = [];
        var data = [
          {Name: 'JBL Speaker', Price: 4500.55, Photo:
'../Images/jblspeaker.jpg'},
          {Name: 'Earpods', Price: 3500.55, Photo:
'../Images/earpods.jpg'},
          {Name: 'Nike Casuals', Price: 5500.55, Photo:
'../Images/shoe.jpg'},
```

```
            {Name: 'Lee Cooper Boot', Price: 2500.55, Photo:
'../Images/shoe1.jpg'},
            {Name: 'Shirt', Price: 1500.55, Photo:
'../Images/shirt.jpg'},
            {Name: 'Jeans', Price: 2500.55, Photo:
'../Images/jeans.jpg'},
        ];
        var cartItems = [];


        $.each(categories, function(key, value){
$(`<option>${value}</option>`).appendTo("#lstCategories");
        })

        function LoadProducts(){
           $("#lstProducts").html("");
           $.each(products, function(key, value){
$(`<option>${value}</option>`).appendTo("#lstProducts");
          })
         }


        $("#lstCategories").change(function(){

           switch($("#lstCategories").val())
```

```
    {
        case 'Electronics':
        products = electronics;
        LoadProducts();
        break;
        case 'Footwear':
        products = footwear;
        LoadProducts();
        break;
        case 'Fashion':
        products = fashion;
        LoadProducts();
        break;
        default:
        products = ['Select a Category'];
        LoadProducts();
    }
})
var searchedProduct;
$("#lstProducts").change(function(){
    var productName = $("#lstProducts").val();
    searchedProduct = data.find(x=>x.Name==productName);
    $("#prodName").html(searchedProduct.Name);
```

```
            $("#prodPrice").html("&#8377;" +
searchedProduct.Price);
            $("#prodImg").attr("src",searchedProduct.Photo);
          })

          function GetCount(){
            $("#itemCount").html(cartItems.length);
          }
          $("#addToCart").click(function(){
            cartItems.push(searchedProduct);
            alert("Item Added to Cart");
            GetCount();
          })
        })
    </script>
  </head>
  <body class="container-fluid">
    <header>
      <h1 class="bg-danger text-white text-center p-3"> <span
class="fa fa-shopping-cart"></span> Amazon Shopping</h1>
    </header>
    <section>
      <div class="row">
       <div class="col-3">
```

```
<div class="form-group">
   <label>Select a Category</label>
   <div>
      <select id="lstCategories" class="form-control">


      </select>
   </div>
</div>
<div class="form-group">
   <label>Select a Product</label>
   <div>
      <select id="lstProducts" class="form-control">


      </select>
   </div>
</div>
<div class="form-group">
   <label>Product Details</label>
   <div class="card">
      <div class="card-header">
         <h4 id="prodName"></h4>
      </div>
      <div class="card-body text-center">
         <img id="prodImg" width="200" height="200">
```

```
                    <h4 id="prodPrice"></h4>
                </div>
                <div class="card-body">
                    <button id="addToCart" class="btn btn-danger btn-block"> <span class="fa fa-shopping-cart"></span> Add to Cart</button>
                </div>
            </div>
        </div>
    </div>
    <div class="col-7">

    </div>
    <div class="col-2">
      <button class="btn btn-outline-danger">
        <span class="fa fa-shopping-bag"></span>
        [<span id="itemCount"></span>] Your Cart Items
      </button>
    </div>
   </div>
  </section>
 </body>
</html>
```

jQuery Effects

- jQuery provides a set of functions to handle various effects dynamically.
    - hide()
    - show()
    - toggle()
    - fadeIn()
    - fadeout()
    - fadeToggle()
    - fadeTo()
    - slideDown()
    - slideUp()
    - slideToggle()
    - animate()
    - delay()

Syntax:

```
$(selector).toggle();
$(selector).toggle(speed);
```

Syntax:

```
$("#displayCart").click(function(){
        $("#cart").fadeToggle(3000);
})
```

Syntax:

```
$(Selector).animate({left:'200px'})
```

Ex:

```
<!DOCTYPE html>
<html>
```

```
<head>
    <title>Animate</title>
    <style>
        div {
            position:absolute;
        }
    </style>
    <script src="../node_modules/jquery/dist/jquery.js"></script>
    <script>
        $(function(){
            $("button").click(function(){
                $("img").animate({left:'200px',top:'100px', width:'300px',
height:'300px'},3000);
            })
        })
    </script>
</head>
<body>
    <button>Click</button>
    <div>
        <img src="../Images/shoe.jpg">
    </div>
</body>
</html>
```

Delay:

$(“selector”).delay(‘slow/12000’).slideIn()

jQuery UI

- It provides a set of jQuery templates that you can use in your application.
- Template comprises of HTML, Styles and Script.
- jQuery provides UI components, which you can customize according to your requirements.
- It provides lot of interactions, widgets, effects, Utilities.

Interactions:

- Draggable
- Droppable
- Resizable
- Selectable
- Sortable

Widgets

- Accordion
- Autocomplete
- Button
- Datepicker
- Dialog
- Menu
- Slider
- Tabs
- Spinner etc.

Effects

- Easing

- Hide
- Show
- Toggle
- Fade

Ex: Interactions

```
<!DOCTYPE html>
<html>
  <head>
    <title>Interactions</title>
    <link rel="stylesheet" href="../node_modules/jqueryUI/jquery-ui.css">
    <script src="../node_modules/jquery/dist/jquery.js"></script>
    <script src="../node_modules/jqueryUI/jquery-ui.js"></script>
    <script>
      $(function(){
        $("#pic").resizable();
        $("#topics").sortable();
      })
    </script>
    <style>
      ol li {
        width:200px
      }
    </style>
  </head>
```

```
  <body>
    <h2>Resizable</h2>
    <img src="../Images/shoe.jpg" id="pic" height="100"
width="100">
    <h2>Sortable</h2>
    <ol id="topics">
      <li class="ui-state-default"> <span class="ui-icon ui-icon-
arrowthick-2-n-s"></span>HTML</li>
      <li class="ui-state-default"><span class="ui-icon ui-icon-
arrowthick-2-n-s"></span>JQuery</li>
      <li class="ui-state-default"><span class="ui-icon ui-icon-
arrowthick-2-n-s"></span>Bootstrap</li>
      <li class="ui-state-default"><span class="ui-icon ui-icon-
arrowthick-2-n-s"></span>CSS</li>
      <li class="ui-state-default"><span class="ui-icon ui-icon-
arrowthick-2-n-s"></span>Styles</li>
    </ol>
  </body>
</html>
```

Ex: Widgets

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="../node_modules/jqueryUI/jquery-
ui.css">
```

```
    <script src="../node_modules/jquery/dist/jquery.js"></script>
    <script src="../node_modules/jqueryUI/jquery-ui.js"></script>
    <script>
      $(function(){
        $("#dob").datepicker();
        $("#topics").accordion();
        $("#toolbar").tabs();
      })
    </script>
  </head>
  <body>
    <h2>Date Picker</h2>
    Date of Birth:
    <input type="text" id="dob">
    <h2>Accordion</h2>
    <div id="topics">
      <h3>HTML</h3>
      <div>
```

<p>Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the

retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

<h3>CSS</h3>

Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

<h3>JavaScript</h3>

Depending on how you obtained the Windows software, this is a license agreement between (i) you and the device manufacturer or software installer that distributes the software with your device; or (ii) you and Microsoft Corporation (or, based on where you live or, if a business, where your principal place of

business is located, one of its affiliates) if you acquired the software from a retailer. Microsoft is the device manufacturer for devices produced by Microsoft or one of its affiliates, and Microsoft is the retailer if you acquired the software directly from Microsoft. Note that if you are a volume license customer, use of this software is subject to your volume license agreement rather than this agreement.

```
            </p>
        </div>
    </div>
    <h2>Tabs</h2>
    <div id="toolbar">
        <ul>
            <li><a href="#html">HTML</a></li>
            <li><a href="#css">CSS</a></li>
            <li><a href="#js">JavaScript</a></li>
        </ul>
        <div id="html">
            <p>Something about HTML</p>
        </div>
        <div id="css">
            <p>Something about CSS</p>
        </div>
        <div id="js">
            <p>Something about JavaScript</p>
```

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

JQuery Ajax

- Ajax is Asynchronous JavaScript and XML.
- Async is a technique used to handle multiple tasks simultaneously at the same time.
- Sync technique uses locking or blocking mechanism.
- Async technique uses unblocking mechanism.
- In Synchronous technique other tasks are locked when any one task is being performed.
- In asynchronous allows to perform all at the same time.
- jQuery uses all Ajax options of JavaScript from "XmlHttpRequest" object.
- jQuery provides a set of Ajax functions for handling asynchronous operations.
- Ajax introduces partial post-back.
- Without reloading the complete page, it loads the new content into page.
- Only specific portion of page is loaded or submitted.
- jQuery Ajax methods
    - ajaxComplete()
    - ajaxError()
    - ajaxSend()
    - ajaxStart()
    - ajaxStop()
    - ajaxSuccess()
    - jquery.get()

- jquery.getJson()
- jquery.post()
- load() etc.

.getJson(): Loads JSON-encode data from the server using HTTP GET request.

Syntax:

```
.getJson(url, function(data){
}
```

```
$.getJSON()
    .done(function(){})
    .fail(function(){})
    .always(function(){})
```

You can get status of your request and issues in the request by using the pipes.

done() : Executes when request is success.

fail() : Executes when request failed

always() : Executes every time.

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ajax</title>
```

```
    <link rel="stylesheet"
href="../node_modules/bootstrap/dist/css/bootstrap.css">
    <script src="../node_modules/jquery/dist/jquery.js"></script>
    <script>
      $(function(){
        $("button").click(function(){
          var products=[];
          $.getJSON("../data/products.json")
          .done(function(data){
            $.each(data, function(key, value){
$(`<tr><td>${value.Name}</td><td>${value.Price}</td><td>${(value.
InStock)?"Available":"Out of
Stock"}</td></tr>`).appendTo("#tbody");
            })
          })
          .fail(function(){
            alert("Something went Wrong");
          })
          .always(function(){
            alert("End of Ajax Request");
          })
        })
      })
    </script>
  </head>
```

```
    <body class="container-fluid">
        <h2>Products Data</h2>
        <div class="form-group">
            <button class="btn btn-primary">Show Data</button>
        </div>
        <div class="form-group">
            <table class="table table-hover">
                <thead>
                    <tr>
                        <th>Name</th>
                        <th>Price</th>
                        <th>Stock</th>
                    </tr>
                </thead>
                <tbody id="tbody">

                </tbody>
            </table>
        </div>
    </body>
</html>
```

Products.json

```
[
    {
```

```
    "Name": "Samsung TV",
    "Price": 34000.55,
    "InStock": true
  },
  {
    "Name": "Nike Casuals",
    "Price": 2400.43,
    "InStock":false
  }
]
```


Ajax Request Status:

- Ajax request have a specific life cycle.
- A set of events preformed to handle ajax request implicitly.
- It has flow of request and response.
- The status of Ajax is shown by using following methods
    - .ajaxComplete()
    - .ajaxError()
    - .ajaxStart()
    - .ajaxStop()
    - .ajaxSuccess()
    - .ajaxError()

Syntax:

```
$(function(){
})
.ajaxComplete(function(){})
.ajaxSuccess(function(){})
```

Ex:

```
<!DOCTYPE html>
<html>
   <head>
      <title>Ajax</title>
      <link rel="stylesheet"
href="../node_modules/bootstrap/dist/css/bootstrap.css">
      <script src="../node_modules/jquery/dist/jquery.js"></script>
      <script>
         $(function(){
            $("button").click(function(){
               var products=[];
               $.getJSON("../data/products.json")
               .done(function(data){
                  $.each(data, function(key, value){
$(`<tr><td>${value.Name}</td><td>${value.Price}</td><td>${(value.
InStock)?"Available":"Out of
Stock"}</td></tr>`).appendTo("#tbody");
                  })
               })
               .fail(function(){
                  alert("Something went Wrong");
               })
            })
```

```
        })
        .ajaxStart(function(){
            alert("Ajax Request Started");
        })
        .ajaxSuccess(function(){
            alert("Ajax Processed Successfully..");
        })
        .ajaxComplete(function(){
            alert("Ajax Request Completed");
        })
    </script>
</head>
<body class="container-fluid">
    <h2>Products Data</h2>
    <div class="form-group">
        <button class="btn btn-primary">Show Data</button>
    </div>
    <div class="form-group">
        <table class="table table-hover">
            <thead>
                <tr>
                    <th>Name</th>
                    <th>Price</th>
                    <th>Stock</th>
```

```
            </tr>
          </thead>
          <tbody id="tbody">

          </tbody>
        </table>
      </div>
  </body>
</html>
```

Ajax Status function have event arguments

function(event, xhr, options, exc)

event: contains the event related properties.

Xhr : contains XmlHttpRequest object

Options: contains options used for ajax request.

Exc: contains the JavaScript exceptions.

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ajax</title>
    <link rel="stylesheet"
href="../node_modules/bootstrap/dist/css/bootstrap.css">
    <script src="../node_modules/jquery/dist/jquery.js"></script>
    <script>
```

```
$(function(){
    $("button").click(function(){
        var products=[];
        $.getJSON("../data/products.json")
        .done(function(data){
            $.each(data, function(key, value){
$(`<tr><td>${value.Name}</td><td>${value.Price}</td><td>${(value.
InStock)?"Available":"Out of
Stock"}</td></tr>`).appendTo("#tbody");
            })
        })
        .fail(function(){
            alert("Something went Wrong");
        })
    })
})
.ajaxStart(function(){
    alert("Ajax Request Started");
})
.ajaxSuccess(function(){
    alert("Ajax Processed Successfully..");
})
.ajaxComplete(function(){
    alert("Ajax Request Completed");
```

```
        })
        .ajaxError(function(e, xhr, opt){
            alert(xhr.statusText + " " + xhr.status);
        })
    </script>
</head>
<body class="container-fluid">
    <h2>Products Data</h2>
    <div class="form-group">
        <button class="btn btn-primary">Show Data</button>
    </div>
    <div class="form-group">
        <table class="table table-hover">
            <thead>
                <tr>
                    <th>Name</th>
                    <th>Price</th>
                    <th>Stock</th>
                </tr>
            </thead>
            <tbody id="tbody">

            </tbody>
        </table>
```

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

Ajax with End to End Integration

- **Database MongoDB**
- **Server Side Node.js**
- **Middleware Express**
- **Client-Side jQuery Ajax**

Database for Application

Install MongoDB database Server

- https://www.mongodb.com/try/download/community

Open MongoDB Client

- Client is the terminal to connect with database server.
- Data is in database server.
- Web connect to database server using a client tool
- Open MongoDB Client after installing and create database, tables

C:\Program Files\MongoDB\Server\4.0\bin>mongo.exe

- View Database
  - show db
- View active database
  - db
- Create or switch to database
  - use database
  - use productsdb
- Create a table [collection]
  - db.createCollection("tblproducts")
- Add records into table

    - db.tblproducts.insert([{ProductId:1, Name: "Samsung TV", Price: 34000.44}, {ProductId:2, Name:"Nike Casuals", Price: 4500.33}])
- To view records from table
    - db.tblproducts.find().pretty()

Server-Side Business Logic to Create API

- Node JS Server-side
- Express Middleware

Install Node JS on your computer

https://nodejs.org/en/download/

Create an API in Node.js

- **Create a new folder server**
- **Install the following packages for your project**
    - npm install mongodb
    - npm install express
    - npm install body-parser
- **Go to "Server" folder add a new script [**Node JS server side programs ".js"]

Api.js code

```
var mongoClient = require("mongodb").MongoClient;

var express = require("express");

var bodyParser = require("body-parser");


var mongourl = "mongodb://127.0.0.1:27017";
```

```
var app = express();


app.get("/getproducts", function(req, res){
    mongoClient.connect(mongourl, function(err,clientObj){
       if(!err){
          var database = clientObj.db("productsdb");
          database.collection("tblproducts").find().toArray(function(err,
records){
              if(!err) {
                 res.send(records);
              }
          })
       }
    })
})
app.post("/addproducts", function(){
   // your post logic
})
app.listen(8080);
console.log("Server Started: http://127.0.0.1:8080");
```

Home.html page

```
<!DOCTYPE html>
```

```
<html>
  <head>
    <title>API</title>
    <link rel="stylesheet"
href="node_modules/bootstrap/dist/css/bootstrap.css">
    <script src="node_modules/jquery/dist/jquery.js"></script>
    <script>
      $(function(){
        $("#btnFetch").click(function(){
          $.ajax({
            url: "http://127.0.0.1:8080/getproducts"
          })
          .done(function(data){
            $.each(data, function(key, value){
$(`<tr><td>${value.ProductId}</td><td>${value.Name}</td><td>${va
lue.Price}</tr>`).appendTo("#tbody");
            })
          })
        })
        $("#btnSearch").click(function(){
          var id = $("#txtId").val();
          var product;
          apiurl = "http://127.0.0.1:8080/getproducts";
          var products = [];
```

```
    $.ajax({
        url: apiurl
    })
    .done(function(data){
        products = data;
        product = products.find(x=>x.ProductId==id);
        $("#lblName").html(product.Name);
        $("#lblPrice").html(product.Price);
    })

})
$("#btnAdd").click(function(){
    var newProduct = {
        ProductId: parseInt($("#txtId").val()),
        Name: $("#txtName").val(),
        Price: parseFloat($("#txtPrice").val())
    };
    $.ajax({
        method: "POST",
        url: "http://127.0.0.1:8080/addproducts",
        data: newProduct
    })
    alert("Record Inserted");
})
```

```
        })
    </script>
</head>
<body>
    <div class="container-fluid">
        <h2>Data from API</h2>
        <div class="form-group">
            <button id="btnFetch" class="btn btn-primary">Fetch Data</button>
            <input id="txtId" type="text"> <button class="btn btn-success" id="btnSearch">Search</button>
            <br>
            <div>
                <dl>
                    <dt>Name</dt>
                    <dd id="lblName"></dd>
                    <dt>Price</dt>
                    <dd id="lblPrice"></dd>
                </dl>
            </div>
        </div>
        <div class="form-group">
            <div class="form-group">
                <h2>Add Record</h2>
```

```
            <dl>
                <dt>Product Id</dt>
                <dd><input id="txtId" type="number"></dd>
                <dt>Name</dt>
                <dd><input id="txtName" type="text"></dd>
                <dt>Price</dt>
                <dd><input id="txtPrice" type="text"></dd>
            </dl>
            <button id="btnAdd" class="btn btn-danger">Add
Product</button>
        </div>
        <table class="table table-hover">
            <thead>
                <tr>
                    <th>Product Id</th>
                    <th>Name</th>
                    <th>Price</th>
                </tr>
            </thead>
            <tbody id="tbody">

            </tbody>
        </table>
    </div>
```

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