CHAPTER 1
JQUERY

1. Introduction:

jQuery is a lightweight, "write less, do more", JavaScript library. The purpose of jQuery is to make it much easier to use JavaScript on your website. JQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code. jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation. In addition, jQuery has plugins for almost any task out there.

The jQuery library contains the following features:

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities

There are lots of other JavaScript frameworks out there, but jQuery is most popular, and also the most extendable.

Many of the biggest companies on the Web use jQuery, such as:

- Google
- Microsoft
- IBM
- Netflix

Note: Before starting jQuery program download jQuery library from jQuery.com.

2. Selectors:

A very common task when using JavaScript is to read and modify the content of the page. To do this, you need to find the element(s) that you wish to change, and this is where **selector** support in jQuery will help you out. With normal JavaScript, finding elements can be extremely cumbersome, unless you need to find a single element which has a value specified in the ID attribute.
jQuery can help you find elements based on their ID, classes, types, attributes, values of attributes and much, much more. It's based on CSS selectors, it is extremely powerful. W
jQuery selectors start with the dollar sign and parentheses − **$()**.

jQuery lets you select elements based on the following criteria:

- Element name (e.g. 'p', 'a', 'div' etc.)
- Element id
- Element CSS class
- Element attributes
- Element order

2.1. Element Name Selector:
The jQuery element selector selects elements based on the element name

Example:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("p").hide();
    });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>
```

Example: use of "*" to select all elements.

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("*").hide();
```

```
    });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>
```

2.2. Element Id Selector:

id selector take the value from html tag which has id.

It uses # with id.

Example:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#test").css("background-color", "pink");
    });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p id="test">This is the example of id selector.</p>
<p>Without selector.</p>
<button>Click me</button>
</body>
</html>
```

2.3. Element Class Selector:

The jQuery class selector finds elements with a specific class.

To find elements with a specific class, write a period character, followed by the name of the class.

Example:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<title>Insert title here</title>
</head>
<body>
<ul>
      <li class="bold">Test 1</li>
      <li>Test 2</li>
      <li class="bold">Test 3</li>
</ul>
<script type="text/javascript">
$(function()
{
      $(".bold").css("font-weight", "bold");
});
</script>
</body>
</html>
```

2.4. Element Attribute Selector:

This selector finds specific attribute and does action.

Example:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("button").click(function(){
         $("[title]").css("text-decoration", "underline");
 });
});
</script>
</head>
<body>
<span title="Title 1">Example for attribute selector with title attribute</span><br /><br>
<span>Title attribute is not used in this sentence</span> <br /><br>
```

```
<span title="Title 3">Example for attribute selector with title attribute</span><br /><br>
<button>Click me</button>
</body>
</html>
```

2.5. Element Order Selector:

Example:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("tr:even").css("background-color", "yellow");
});
</script>
</head>

<body>
<h1>Welcome to My Web Page</h1>

<table border="1">
  <tr>  <th>Company</th>  <th>Country</th></tr>
  <tr><td>Alfreds Futterkiste</td><td>Germany</td></tr>
  <tr><td>Berglunds snabbköp</td> <td>Sweden</td></tr>
  <tr><td>Centro comercial Moctezuma</td><td>Mexico</td></tr>
  <tr><td>Ernst Handel</td><td>Austria</td></tr>
  <tr><td>Island Trading</td><td>UK</td></tr>
</table>
</body>
</html>
```

3. Effects:

jQuery enables us to add effects on a web page. jQuery provides many methods for effects on a web page. A complete list of jQuery effect methods are given below:

Method	Description
animate()	performs animation.
clearQueue()	It is used to remove all remaining queued functions from the selected elements.

delay()	sets delay execution for all the queued functions on the selected elements.
dequeue()	It is used to remove the next function from the queue, and then execute the function.
fadein()	shows the matched elements by fading it to opaque. In other words, it fades in the selected elements.
fadeout()	shows the matched elements by fading it to transparent. In other words, it fades out the selected elements.
fadeto()	adjusts opacity for the matched element. In other words, it fades in/out the selected elements.
fadetoggle()	shows or hides the matched element. In other words, toggles between the fadeIn() and fadeOut() methods.
finish()	It stops, removes and complete all queued animation for the selected elements.
hide()	hides the matched or selected elements.
queue()	shows or manipulates the queue of methods i.e. to be executed on the selected elements.
show()	displays or shows the selected elements.
slidedown()	shows the matched elements with slide.
slidetoggle()	shows or hides the matched elements with slide. In other words, it is used to toggle between the slideUp() and slideDown() methods.
slideup()	hides the matched elements with slide.
stop()	stops the animation which is running on the matched elements.
toggle()	shows or hides the matched elements. In other words, it toggles between the hide() and show() methods.

Example: hide and show effect:

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
 <!DOCTYPE html>
  <html>
 <head>
   <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
   <script>
   $(document).ready(function(){
         $("#hide").click(function(){
         $("p").hide();
      });
      $("#show").click(function(){
         $("p").show();
      });
```

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

<body>
<p> <b>This is a little poem: </b><br/>
Twinkle, twinkle, little star<br/>
How I wonder what you are<br/>
Up above the world so high<br/>
Like a diamond in the sky<br/>
Twinkle, twinkle little star<br/>
How I wonder what you are
</p>
<button id="hide">Hide</button>
<button id="show">Show</button>
</body>
</html>
```

fadeIn, fadeOut, fadeToggle, fadeTo:

With jQuery you can fade an element in and out of visibility.

jQuery has the following fade methods:

- fadeIn()
- fadeOut()
- fadeToggle()
- fadeTo()

Example: fadeIn

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("button").click(function(){
      $("#div1").fadeIn();
      $("#div2").fadeIn("slow");
      $("#div3").fadeIn(3000);
   });
});
</script>
</head>
<body>
<p>Demonstrate fadeIn() with different parameters.</p>
```

```
<button>Click to fade in boxes</button><br><br>
<div id="div1" style="width:80px;height:80px;display:none;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;display:none;background-
color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;display:none;background-color:blue;"></div>
</body>
</html>
```

Example: FadeOut

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#div1").fadeOut();
        $("#div2").fadeOut("slow");
        $("#div3").fadeOut(3000);
    });
});
</script>
</head>
<body>

<p>Demonstrate fadeOut() with different parameters.</p>
<button>Click to fade out boxes</button><br><br>
<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;background-color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>

</body>
</html>
```

Example: fadeToggle

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
```

```
    $("button").click(function(){
        $("#div1").fadeToggle();
        $("#div2").fadeToggle("slow");
        $("#div3").fadeToggle(3000);
    });
});
</script>
</head>
<body>
<p>Demonstrate fadeToggle() with different speed parameters.</p>
<button>Click to fade in/out boxes</button><br><br>
<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;background-color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>
</body>
</html>
```

Example: fadeTo

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#div1").fadeTo("slow", 0.15);
        $("#div2").fadeTo("slow", 0.4);
        $("#div3").fadeTo("slow", 0.7);
    });
});
</script>
</head>
<body>
<p>Demonstrate fadeTo() with different parameters.</p>
<button>Click to fade boxes</button><br><br>
<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;background-color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>
</body>
</html>
```

Example: Animation

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
```

```
   pageEncoding="UTF-8"%>
<html>
<head>
<title>The jQuery Example</title>
<script type = "text/javascript"
src = "http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
 <script type = "text/javascript" language = "javascript">
 $(document).ready(function() {
$("#out").click(function(){
 $("#block").animate({
                width: "70%",
                opacity: 0.4,
                marginLeft: "0.6in",
                fontSize: "3em",
                borderWidth: "10px"
             }, 1500 );
 $("#in").click(function(){
             $("#block").animate({
                width: "100",
                opacity: 1.0,
                marginLeft: "0in",
                fontSize: "100%",
                borderWidth: "1px"
             }, 1500 );
          });
});
 });
</script>
 <style>
div {background-color:#bca; width:100px; border:1px solid green;}
</style>
</head>
<body>
<p>Click on any of the buttons</p>
<button id = "out"> Animate Out </button>
<button id = "in"> Animate In</button>
<div id = "block">Hello</div>
</body>
</html>
```

Example: sliding

```
 <%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#flip").click(function(){
        $("#panel").slideToggle("slow");
    });
});
</script>
<style>
#panel, #flip {
    padding: 5px;
    text-align: center;
    background-color: #e5eecc;
    border: solid 1px #c3c3c3;
}
#panel {
    padding: 50px;
    display: none;
}
</style>
</head>
<body>
 <div id="flip">Click to slide the panel down or up</div>
<div id="panel">Hello world!</div>
</body>
</html>
```

Example: Start and Stop

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#flip").click(function(){
        $("#panel").slideDown(5000);
    });
    $("#stop").click(function(){
        $("#panel").stop();
    });
});
</script>
 <style>
#panel, #flip {
```

```
    padding: 5px;
    font-size: 18px;
    text-align: center;
    background-color: #555;
    color: white;
    border: solid 1px #666;
    border-radius: 3px;
}
#panel {
    padding: 50px;
    display: none;
}
</style>
</head>
<body>
 <button id="stop">Stop sliding</button>
<div id="flip">Click to slide down panel</div>
<div id="panel">Hello world!</div>
</body>
</html>
```

Example: Callback Function

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("p").hide("slow", function(){
            alert("The paragraph is now hidden");
        });
    });
});
</script>
</head>
<body>
<button>Hide</button>
<p>This is a paragraph with little content.</p>
</body>
</html>
```

4. jQUERY HTML:

jQuery contains powerful methods for changing and manipulating HTML elements and attributes.

4.1. jQuery DOM Manipulation:

One very important part of jQuery is the possibility to manipulate the DOM. jQuery comes with a bunch of DOM related methods that make it easy to access and manipulate elements and attributes.

4.2. Get Content - text(), html(), and val():

Three simple, but useful, jQuery methods for DOM manipulation are:

- text() - Sets or returns the text content of selected elements
- html() - Sets or returns the content of selected elements (including HTML markup)
- val() - Sets or returns the value of form fields

Example: text() and html() methods

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#btn1").click(function(){
        alert("Text: " + $("#test").text());
    });
    $("#btn2").click(function(){
        alert("HTML: " + $("#test").html());
    });
});
</script>
</head>
<body>
<p id="test">This is some <b>bold</b> text in a paragraph.</p>
<button id="btn1">Show Text</button>
<button id="btn2">Show HTML</button>
</body>
</html>
```

Example: val() method

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("button").click(function(){
      alert("Value: " + $("#test").val());
   });
});
</script>
</head>
<body>
<p>Name: <input type="text" id="test" value="Enter value"></p>
<button>Show Value</button>
</body>
</html>
```

4.3. Set Content - text(), html(), and val():

Set method used to set the value to the method. It uses same method as the get.

Example:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("#btn1").click(function(){
      $("#test1").text("Welcome in Coder Technology!");
   });
   $("#btn2").click(function(){
      $("#test2").html("<h1>Welcome in Coder Technology!</h1>");
   });
   $("#btn3").click(function(){
      $("#test3").val("Java");
   });
});
</script>
</head>
<body>
<p id="test1">display text in simple format.</p>
<p id="test2">Display text with HTML formatting.</p>
<p>Input field: <input type="text" id="test3" value="enter name"></p>
<button id="btn1">Set Text</button>
<button id="btn2">Set HTML</button>
<button id="btn3">Set Value</button>
```

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

4.4. Add New HTML Content:

We will look at four jQuery methods that are used to add new content:

- **append()** - Inserts content at the end of the selected elements
- **prepend()** - Inserts content at the beginning of the selected elements
- **after()** - Inserts content after the selected elements
- **before()** - Inserts content before the selected elements

Example:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("#btn1").click(function(){
      $("i").append(" <b>The purpose of jQuery is to make it much easier to use JavaScript on your website..</b>.");
   });
      $("#btn2").click(function(){
         $("p").prepend("<b>jQuery is a lightweight</b>. ");
      });

      $("#btn3").click(function(){
         $("img").before("<b>Before</b>");
      });

      $("#btn4").click(function(){
         $("img").after("<i>After</i>");
      });
});
</script>
</head>
<body>
<i></i>
<br><br><br>
<img alt="jquery logo" src="index.png"><br><br>
<p>write less, do more ,JavaScript library.</p><br><br>
<button id="btn1">Append</button>
<button id="btn2">Prepend</button>
<button id="btn3">Insert before</button>
```

```
<button id="btn4">Insert after</button>
</body>
</html>
```

4.5. Remove Elements/Content:

To remove elements and content, there are mainly two jQuery methods:

- remove() - Removes the selected element (and its child elements)
- empty() - Removes the child elements from the selected element

Example:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#btn1").click(function(){
        $("#div1").empty();
    });
    $("#btn2").click(function(){
        $("#div1").remove();
    });
});
</script>
</head>
<body>
<div id="div1" style="height:250px;width:500px;border:2px solid black;background-
color:yellow;">
<img alt="jquery logo" src="index.png"><br><br>
</div>
<br>
<button id="btn1">Empty the div element</button>
<button id="btn2">Remove the div element</button>
</body>
</html>
```

5. Traversing:

Once you've made an initial selection with jQuery, you can traverse deeper into what was just selected. Traversing can be broken down into three basic parts: parents, children, and siblings. jQuery has an abundance of easy-to-use methods for all these parts. Notice that each of these methods can optionally be passed string selectors, and some can also take another jQuery object in order to filter your selection down.

Example of traversing:

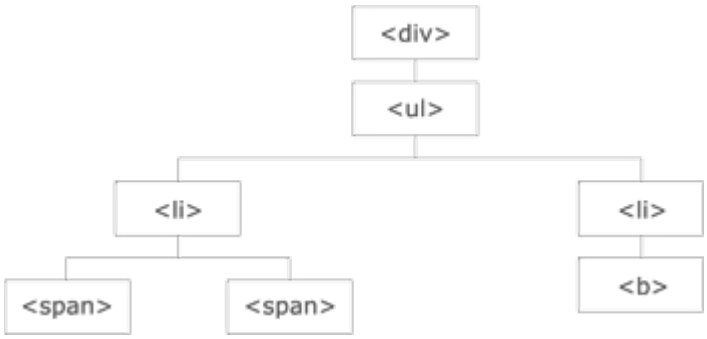

Illustration explained:

- The element is the **parent** of <ul>, and an **ancestor** of everything inside of it
- The <ul> element is the **parent** of both <li> elements, and a **child** of
- The left <li> element is the **parent** of , **child** of <ul> and a **descendant** of
- The element is a **child** of the left <li> and a **descendant** of <ul> and
- The two <li> elements are **siblings** (they share the same parent)
- The right <li> element is the **parent** of <b>, **child** of <ul> and a **descendant** of
- The <b> element is a **child** of the right <li> and a **descendant** of <ul> and

jquery traversing methods used on the DOM tree:

5.1. jQuery Traversing – Ancestors:

An ancestor is a parent, grandparent, great-grandparent, and so on. With jQuery you can traverse up the DOM tree to find ancestors of an element.

Three useful jQuery methods for traversing up the DOM tree are:

- **parent()** - The parent() method returns the direct parent element of the selected element. This method only traverse a single level up the DOM tree.
- **parents()** - The parents() method returns all ancestor elements of the selected element, all the way up to the document's root element (<html>).
- **parentsUntil**() - The parentsUntil() method returns all ancestor elements between two given arguments.

Example:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
```

```
   pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<style>
.ancestors * {
   display: block;
   border: 2px solid lightgrey;
   color: lightgrey;
   padding: 5px;
   margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
        $("#btn1").click(function(){
   $("span").parents().css({"color": "red", "border": "2px solid red"});
        });

        $("#btn2").click(function(){
                $("span").parent().css({"color" :"green", "border":"4px solid green",})
        });

        $("#btn3").click(function(){
                $("span").parentsUntil("#innerdiv").css({"color" :"purple", "border":"3px solid
purple",})
        });

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

5.2. jQuery Traversing – Descendants:

A descendant is a child, grandchild, great-grandchild, and so on. With jQuery you can traverse down the DOM tree to find descendants of an element.

Two useful jQuery methods for traversing down the DOM tree are:

- **children()** - The children() method returns all direct children of the selected element. This method only traverse a single level down the DOM tree.
- **find()** - The find() method returns descendant elements of the selected element, all the way down to the last descendant.

Example:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<style>
.descendants * {
    display: block;
    border: 2px solid lightgrey;
    color: lightgrey;
    padding: 5px;
    margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
        $("#btn1").click(function(){
    $("div").children().css({"color": "purple", "border": "3px solid purple"});
});
        $("#btn2").click(function(){
            $("div").children("p.first").css({"color":"red","border":"3px solid red"});
        });
        $("#btn3").click(function(){
            $("div").find("*").css({"color": "green", "border": "3px solid green"});
        });
        $("#btn4").click(function(){
            $("div").find("span").css({"color":"DarkBlue ","border":"3px solid DarkBlue"})
        });

});
</script>
</head>
<body>
<div class="descendants" style="width:500px;"> div (current element)
  <p class="first">p (child)
  <span>span (grandchild)</span>
  </p>

  <p class="second">p (child)
    <span>span (grandchild)</span>
  </p>
</div>
<button id="btn1">find child</button>
<button id="btn2">find specific child</button>
<button id="btn3">find all descendant</button>
<button id="btn4">find particular descendant </button>
</body>
```

```
</html>
```

5.3. jQuery Traversing – Siblings:

Siblings share the same parent. With jQuery you can traverse sideways in the DOM tree to find siblings of an element.

There are many useful jQuery methods for traversing sideways in the DOM tree:

- **siblings()** - The siblings() method returns all sibling elements of the selected element.
- **next()** - The next() method returns the next sibling element of the selected element.
- **nextAll()** - The nextAll() method returns all next sibling elements of the selected element.
- **nextUntil()** - The nextUntil() method returns all next sibling elements between two given arguments.
- **prev()** - Get a set of elements containing the unique previous siblings of each of the matched set of elements.
- **prevAll()** - Find all sibling elements in front of the current element.
- **prevUntil()** - The prevUntil() method returns all previous sibling elements between two given arguments.

Example:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<style>
.siblings * {
    display: block;
    border: 2px solid lightgrey;
    color: lightgrey;
    padding: 5px;
    margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("h2").siblings("p").css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body class="siblings">
```

```
<div>div (parent)
  <p>p</p>
  <span>span</span>
  <h2>h2</h2>
  <h3>h3</h3>
  <p>p</p>
</div>
</body>
</html>
```

5.4. jQuery Traversing – Filtering:

The three most basic filtering methods are first(), last() and eq(), which allow you to select a specific element based on its position in a group of elements.

Methods:

- **first()** - The first() method returns the first element of the selected elements.
- **last()** - The last() method returns the last element of the selected elements.
- **eq()** - The eq() method returns an element with a specific index number of the selected elements.

Example:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("p").eq(1).css("background-color", "yellow");
});
</script>
</head>
<body>
<h1>Welcome to My Homepage</h1>
<p>My name is Donald (index 0).</p>
<p>Donald Duck (index 1).</p>
<p>I live in Duckburg (index 2).</p>
<p>My best friend is Mickey (index 3).</p>
</body>
</html>
```

6. AJAX:

Traditionally webpages required reloading to update their content. For web-based email this meant that users had to manually reload their inbox to check and see if they had new mail. This had huge

drawbacks: it was slow and it required user input. When the user reloaded their inbox, the server had to reconstruct the entire web page and resend all of the HTML, CSS, JavaScript, as well as the user's email. This was hugely inefficient. Ideally, the server should only have to send the user's new messages, not the entire page. By 2003, all the major browsers solved this issue by adopting the XMLHttpRequest (XHR) object, allowing browsers to communicate with the server without requiring a page reload.

The XMLHttpRequest object is part of a technology called Ajax (Asynchronous JavaScript and XML). Using Ajax, data could then be passed between the browser and the server, using the XMLHttpRequest API, without having to reload the web page. With the widespread adoption of the XMLHttpRequest object it quickly became possible to build web applications like Google Maps, and Gmail that used XMLHttpRequest to get new map tiles, or new email without having to reload the entire page.

Ajax requests are triggered by JavaScript code; your code sends a request to a URL, and when it receives a response, a callback function can be triggered to handle the response. Because the request is asynchronous, the rest of your code continues to execute while the request is being processed, so it's imperative that a callback be used to handle the response.

Unfortunately, different browsers implement the Ajax API differently. Typically this meant that developers would have to account for all the different browsers to ensure that Ajax would work universally. Fortunately, jQuery provides Ajax support that abstracts away painful browser differences. It offers both a full-featured $.ajax() method, and simple convenience methods such as $.get(), $.getScript(), $.getJSON(), $.post(), and $().load().

6.1. AJAX Methods:

The following are the ajax methods:

- load()
- get() and post()

1. load(): The jQuery load() method is a simple, but powerful AJAX method. The load() method loads data from a server and puts the returned data into the selected element.

Syntax:

$(*selector*).load(*URL,data,callback*);

- The required URL parameter specifies the URL you wish to load.
- The optional data parameter specifies a set of querystring key/value pairs to send along with the request.

  The optional callback parameter is the name of a function to be executed after the load() method is completed.

Example:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
  pageEncoding="UTF-8"%>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script type="text/javascript">
$(function(){
        $("#btn1").click(function(){
      $("#divTestArea1").load("test");
      });
});
</script>
<br><br>
<div id="divTestArea1"><b>This text will be changed after clicking button.</b> </div><br><br>
<button id="btn1">Load Content</button>
</body>
</html>
```

6.2 get(): The $.get() method requests data from the server with an HTTP GET request.

Syntax:

$.get(*URL,callback*);

- The required URL parameter specifies the URL you wish to request.
- The optional callback parameter is the name of a function to be executed if the request succeeds.

Example:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("button").click(function(){
      $.get("test", function(data, status){
```

```
        alert("Data: " + data + "\nStatus: " + status);
      });
   });
});
</script>
</head>
<body>
<button>Send an HTTP GET request to a page and get the result back</button>
</body>
</html>
```

The first parameter of $.get() is the URL we wish to request ("test").
The second parameter is a callback function. The first callback parameter holds the content of the page requested, and the second callback parameter holds the status of the request.

6.3. post(): The $.post() method requests data from the server using an HTTP POST request.
Syntax:

$.post(*URL,data,callback*);

- The required URL parameter specifies the URL you wish to request.
- The optional data parameter specifies some data to send along with the request.
- The optional callback parameter is the name of a function to be executed if the request succeeds.

Example:
Demo.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("button").click(function(){
      $.post("reference.jsp",
      {
        name: "yuga",
        },
      function(data,status){
         alert("Data: " + data + "\nStatus: " + status);
```

```
      });
   });
});
</script>
</head>
<body>
<br><br>
<button>Send an HTTP POST request to a page and get the result back</button>
</body>
</html>
```

Reference.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<%
String name=request.getParameter("name");
out.println("Welcome"+name+"This is example of AJAX post() method");
%>
```

Steps:

- The first parameter of $.post() is the URL we wish to request ("reference.jsp").
- Then we pass in some data to send along with the request (name).
- The JSP script in "reference.jsp" reads the parameters, processes them, and returns a result.
- The third parameter is a callback function. The first callback parameter holds the content of the page requested, and the second callback parameter holds the status of the request.

ASSIGNMENT

1. Find all h1 elements that are children of a div element and apply a backgroud to them.

```
<body>
<h1>abc</h1>
<div>
<h1>div-1</h1>
<h1>div-2</h1>
</div>
<h1>xyz</h1>
 </body>
```

2. Hide all the input elements within a form.

```
<body>
```

```
<form  name='demo_form'>
First name: <input type="text" name="fname"><br>
Last name: <input type="text" name="lname"><br>
<input type="submit" value="Submit">
</form>
</body>
```

3. Find the specific option tag text value of a selected option.

```
<body>
<select id="myselect">
 <option value="1">Option-1</option>
 <option value="2">Option-2</option>
 <option value="3">Option-3</option>
 </select>
 </body>
```

4. Write jQuery code to append a div element (and all of its contents) dynamically to the body element.

Insert the following within HTML <body> tag :

```
<div><h1>JQuery Core</h1></div>
```

5. Write a jQuery Code to get a single element from a selection.

```
<body>
<ul>
<li>Html Tutorial</li>
<li>Mongodb Tutorial</li>
<li>Python Tutorial</li>
</ul>
</body>
```

6. Write jQuery Code to change the hyperlink and the text of a existing link.

```
<a href="www.w3resource.com/sqlite/" id='tut'>SQLite Tutorial</a>
```

7. Toggle a specified class when an element is clicked.

HTML:

```
<body>
<p></p>
</body>
```

CSS:

```
<style>
p {
   margin-top:20px;
```

```
    margin-left: 10px;
    padding: 5px;
    border: 2px solid #666;
  }
</style>
```

8. Double click on paragraph to toggle background color.

HTML Code:

```
 <body>
 <p>Double-click here to change the background color.</p>
 </body>
```

CSS Code:

```
p {
    background: blue;
    color: white;
     }
p.dbl {
    background: yellow;
    color: black;
  }
```

9. Change the background color of the element of the following code on clicking the button.

```
<body>
<div style="background-color:yellow">
<p>Click the button to change the background color of this paragraph.</p>
<button>Click me!</button>
</div>
</body>
```

10. Stop people from writing in first text input box (user ID)

```
<body>
<p>User ID : <input type="text" id='field1'></p>
<p>Password : <input type="password" id='field2'>
</body>
```

11. Display the tag name of the click element.

```
<body>
  <p></p>
  <h2>This is a heading 2</h2>
  <div>
  First name: <input type="text"><br>
```

```
Last name: <input type="text">
</div>
</body>
```

12. Animate all hidden elements to show slowly.