WESTCLIFF UNIVERSITY
Educate. Inspire. Empower.

MODULE 2

Review

Week 1 Day 3

Agenda - Review

- jQuery Basics
- jQuery Selectors
- jQuery Attributes
- jQuery Traversing
- jQuery CSS
- jQuery DOM
- jQuery Event
- jQuery Ajax
- jQuery Effects
- Lab Assignment
- Homework

jQuery DOM Manipulation

jQuery DOM

- provides various methods to add, edit or delete DOM element(s) in the HTML page.
- provides methods to manipulate DOM in efficient way. You do not need to write big code to modify the value of any element's attribute or to extract HTML code from a paragraph or division.
- provides methods such as .attr(), .html(), and .val() which act as getters, retrieving information from DOM elements for later use.

About jQuery

About jQuery

- JavaScript library created by John Resig in 2006.
- Simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development
- Motto: “write less, do more”

- Required Knowledges
  - HTML
  - CSS
  - JavaScript

jQuery

- Core features of jQuery include:
  - DOM manipulation
    - easy to select DOM elements, negotiate them and modifying their content
  - Event handling
    - elegant way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers
  - AJAX Support
    - helps develop a responsive and feature rich site using AJAX technology
  - Animations
    - comes with plenty of built-in animation effects
  - Lightweight
    - very lightweight library - about 19KB in size (Minified and gzipped)
  - Cross Browser Support
    - works well in IE 6.0+, FF 2.0+, Safari 3.0+, Chrome and Opera 9.0+
  - Latest Technology
    - supports CSS3 selectors and basic XPath syntax

How to Use jQuery

How to use jQuery

- Local Installation
  - Download from https://jquery.com/download/
- CDN Based Version
  - <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

jQuery Example with Local Installation

jquery-example.html

fullstack-bootcamp > jquery-example.html > html

```
1   <html>
2     <head>
3       <title>Westcliff Fullstack Bootcamp jQuery Example</title>
4       <script type = "text/javascript" src = "js/jquery-compressed-3.5.1.js"></script>
5
6       <script type = "text/javascript">
7         $(document).ready(function() {
8           document.write("Hello, World!");
9         });
10      </script>
11    </head>
12
13    <body>
14      <h1>Hello</h1>
15    </body>
16  </html>
```

jQuery Example with CDN Based Version

jquery-example.html | jquery-example-cdn-google.html

fullstack-bootcamp > jquery-example-cdn-google.html > html

```
1   <html>
2     <head>
3       <title>Westcliff Fullstack Bootcamp jQuery Example</title>
4       <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
5
6       <script type = "text/javascript">
7         $(document).ready(function() {
8           document.write("Hello, World!");
9         });
10      </script>
11    </head>
12
13    <body>
14      <h1>Hello</h1>
15    </body>
16  </html>
```

jQuery Syntax

- The jQuery syntax is designed for **selecting** HTML elements and performing some type of **action** on the element(s).
- Basic syntax: **$(*selector*).*action*()**
  - **$** sign to define/access jQuery
  - **(*selector*)** to "query (or find)" HTML elements
  - **jQuery *action*()** to be performed on the element(s)

Calling jQuery Functions

jquery-function-example.html

fullstack-bootcamp > jquery-function-example.html > html > head > script

```
Live Demo
<html>
    <head>
        <title>The jQuery Example</title>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

        <script type = "text/javascript" language = "javascript">
            $(document).ready(function() {
                $("div").click(function() {alert("Hello, world!");});
            });
        </script>
    </head>

    <body>
        <div id = "mydiv">
            Click on this to see a dialogue box.
        </div>
    </body>
</html>

```

jQuery Data Types

jQuery Data Types #1

- jQuery uses same data types as Javascript.

- String
  - “A Javascript String”
  - ‘A Javascript String’
  - “A ‘Javascript’ String”
  - ‘A “Javascript” String’
- Numbers
  - 1234
  - 123.456

jQuery Data Types #2

- Boolean
  - Boolean values
    - true
    - false
  - Numeric values
    - 0 means false
    - 1 means true
  - String values
    - “” (empty String) means false
    - “any string” means true

jQuery Data Types #3

- Objects
  - var student = {
    name: “Full Name”,
    major: “Computer Science”
    };
  - Referencing the Object Properties
    - student.name
    - student.major
  - Setting values to Object Properties
    - student.name = “New Name”;
    - student.major = “New Major”;

jQuery Data Types #4

- Arrays
  - var x = [];
  - var y = [1, 2, 3, 4, 5];
  - Example
    - var x = [1, 2, 3, 4, 5];

```
for (var i = 0; i < x.length; i++) {
  // Do something with x[i]
}
```

jQuery Fundamentals

jQuery Functions

- Named Functions
  - function function_name() {
    // do something here
    }
- Anonymous Functions
  - var handler = function () {
    // do some stuff here
    }
  - $(document).ready(function() {
    // do some stuff here
    });

Arguments

- Arguments are passed like an array.
- Ex:
  - function sample_function(x) {
    console.log(typeof x, arguments.length);
    }

```
sample_function();                  //==> "undefined", 0
sample_function(1);                 //==> "number", 1
sample_function("1", "2", "3");     //==> "string", 3
```

Scope

- Global Variable
  - A variable defined for global access within Javascript code.
    - <script>
      var globalVar = “A Global Variable”;
      </script>
- Local Variable
  - A variable defined for local access within a function which is visible only within the function.
  - Arguments(parameters) passed to a function are always local to the function.
    - <script>
      function func() {
      var localVar = “A Local Variable”;
      }
      </script>

Context

- *this* keyword
  - Always refers to current context
  - Ex:
    - $(document).ready(function() {
      $(this).click…. // this refers to window.document
      });
    - $("div").click(function() {
      $(this).hide…. // this refers to a div DOM element
      });
- call()
  - Passes all arguments through as arguments to the function
- apply()
  - Passes an array as the arguments to the function

Callback

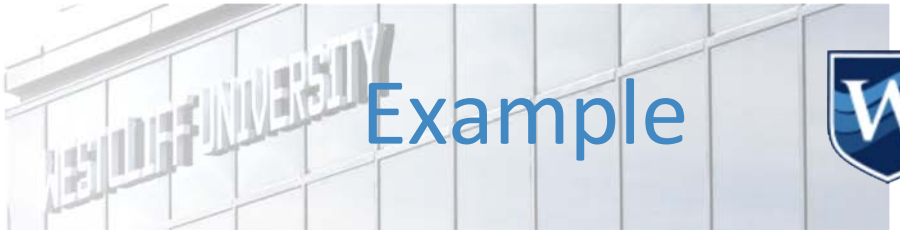
- JavaScript functions passed to some method as an argument or option.
- Some callbacks require return values, others make the return values optional.
- Ex:
  - $("body").click(function(event) {
    console.log("clicked: " + event.target);
    });
  - $("#some-form").submit(function() {
    return false;
    });

jQuery Selectors

jQuery Selectors

- Cascading Style Sheets (CSS) selectors
- A function to find matching elements from a DOM.
- by Tag Name
  - Selects all elements with the given element Name.
    - $('p')
- by Tag ID
  - Selects a single element with the given ID.
    - $(‘#idName')
- by Tag Class
  - Selects all elements with the given Class.
    - $('.className')
- Multiple Elements
  - Selects the combined results of all the specified selectors by tag name, id, class.

jQuery Selectors Example




```html
<html>
<head>
    <title>Westcliff Fullstack Bootcamp jQuery Selector Example</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <script type="text/javascript" language="javascript">
        $(document).ready(function () {
            $("p").css("background-color", "yellow");
        });
    </script>
</head>
<body>
    <div>
        <p class="myclass">This is first paragraph.</p>
        <p id="myid">This is second paragraph.</p>
        <p>This is third paragraph.</p>
    </div>
</body>
</html>
```

jQuery Selectors

- Universal
  - Selects all elements in a DOM.
  - $('*')
    - Selects all elements
  - $("p > *")
    - All elements that are a direct child descendent of <p> element
- Multiple Elements
  - Selects the combined results of all the specified selectors.
  - $("#container p")
    - All <p> elements that are descendent of the element with id=container
  - $("code, em, strong")
    - All these elements will be selected
  - $("ul li:first")
    - Selects the first li element that is the descendent of the ul element

jQuery Attributes

Attributes

- Reference and Manipulate the properties and attributes of DOM elements.
  - Ex: className, tagName, id, href, title, rel, src
- Methods
  - **attr(properties)** // sets or returns attributes and values of the 1st match elements
  - **attr(key, fn)** // sets property and value using a function
  - **removeAttr(name)** // remove an attribute from each of the matched elements
  - **hasClass(class), removeClass(class), toggleClass(class** // Returns true, removes specified class, adds/removes specified class
  - **html(), html(value)** //get html content, set html content of matched elements
  - **text(), text(value)** //get text content, set text content of matched elements
  - **val(), val(value)** //get input value, set value of matched input element

Attributes

Example


```
<html>
<head>
    <title>Westcliff Fullstack Bootcamp jQuery Attribute Example</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <script type="text/javascript" language="javascript">
        $(document).ready(function () {
            var title = $("em").attr("title");
            $("#divid").text(title);
            $("#myimg").attr("src", "/jquery/images/jquery.jpg");
        });
    </script>
</head>
<body>
    <div>
        <em title="Sample Title">This is first paragraph.</em>
        <p id="myid">This is second paragraph.</p>
        <div id="divid"></div>
        <img id="myimg" src="/images/jquery.jpg" alt="Sample Image" />
    </div>
</body>
</html>
```

jQuery Traversing

jQuery DOM Traversing

- Help select elements in a document randomly as well as in sequential method, most without modifying the jQuery object and are used to filter out elements.

- DOM Filter Methods:
  - **eq(index)** //reduce the set of matched elements to a single element
  - **filter(selector)** //exclude elements that do not match the specified selector(s)
  - **filter(fn)** //exclude elements that do not match the specified function
  - **is(selector)** //checks current selection and returns true, if one element of selection fits the given selector
  - **map(callback)** //translate a set of elements in the jQuery object into another set of values in a jQuery array (which may, or may not contain elements)
  - **not(selector)** //removes elements matching the specified selector
  - **slice(start, [end])** //selects a subset of the matched elements

jQuery DOM Traversing

- DOM Traversing Methods
  - **add(selector)** //adds more elements, matched by the given selector, to the set of matched elements
  - **andSelf()** //add the previous selection to the current selection
  - **children([selector])** //get set of elements containing all unique immediate children of matched elements
  - **closest(selector)** //get set of elements containing closest parent element that matches specified selector, including starting element
  - **contents()** //find all child nodes inside matched elements (including text nodes), or the content document if element is iframe
  - **end()** //revert most recent 'destructive' operation, changing set of matched elements to previous state

jQuery DOM Traversing

- DOM Traversing Methods (con't)
  - **find(selector)** //searches descendant elements that match specified selectors
  - **next([selector]), nextAll([selector])** //get set of elements containing unique next siblings of given elements, find all sibling elements after current element
  - **offsetParent()** //returns jQuery collection with positioned parent of first matched element
  - **parent([selector]), parents([selector])** //get direct parent of an element or elements, get elements containing unique ancestors of matched elements (except root element)
  - **prev([selector]), prevAll([selector])** //get elements containing unique previous siblings of matched elements, find all sibling elements in front of current element
  - **siblings([selector])** //get elements containing all unique siblings of matched elements

jQuery DOM Traversing Example


```html
<html>
<head>
    <title>Westcliff Fullstack Bootcamp jQuery Filter Example</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <script type="text/javascript" language="javascript">
        $(document).ready(function () {
            $("li").filter(".middle").addClass("selected");
        });
    </script>
    <style>
        .selected {
            color: red;
        }
    </style>
</head>
<body>
    <div>
        <ul>
            <li class="top">list item 1</li>
            <li class="top">list item 2</li>
            <li class="middle">list item 3</li>
            <li class="middle">list item 4</li>
            <li class="bottom">list item 5</li>
            <li class="bottom">list item 6</li>
        </ul>
    </div>
</body>
</html>
```

jQuery CSS

jQuery CSS

- Supports Cascading Style Sheet (CSS) specifications 1 through 3.
- Can enhance websites without worrying about browsers and their versions as long as the browsers have JavaScript enabled.
- Not modifying the contents of the jQuery object
- Use to apply CSS properties on DOM elements.

jQuery CSS Methods #1

Method	Description
css(name)	Return a style property on the first matched element.
css(name, value)	Set a single style property to a value on all matched elements.
css(properties)	Set a key/value object as style properties to all matched elements.
height(val)	Set the CSS height of every matched element.
height( )	Get the current computed, pixel, height of the first matched element.
innerHeight( )	Gets the inner height (excludes the border and includes the padding) for the first matched element.

jQuery CSS Methods #2

Method	Description
innerWidth( )	Gets the inner width (excludes the border and includes the padding) for the first matched element.
offset( )	Get the current offset of the first matched element, in pixels, relative to the document.
offsetParent( )	Returns a jQuery collection with the positioned parent of the first matched element.
outerHeight([margin])	Gets the outer height (includes the border and padding by default) for the first matched element.
outerWidth([margin])	Get the outer width (includes the border and padding by default) for the first matched element.
position( )	Gets the top and left position of an element relative to its offset parent.

jQuery CSS Methods #3

Method	Description
scrollLeft(val)	When a value is passed in, the scroll left offset is set to that value on all matched elements.
scrollLeft( )	Gets the scroll left offset of the first matched element.
scrollTop(val)	When a value is passed in, the scroll top offset is set to that value on all matched elements.
scrollTop( )	Gets the scroll top offset of the first matched element.
width(val)	Set the CSS width of every matched element.
width( )	Get the current computed, pixel, width of the first matched element.

jQuery CSS Methods Sample

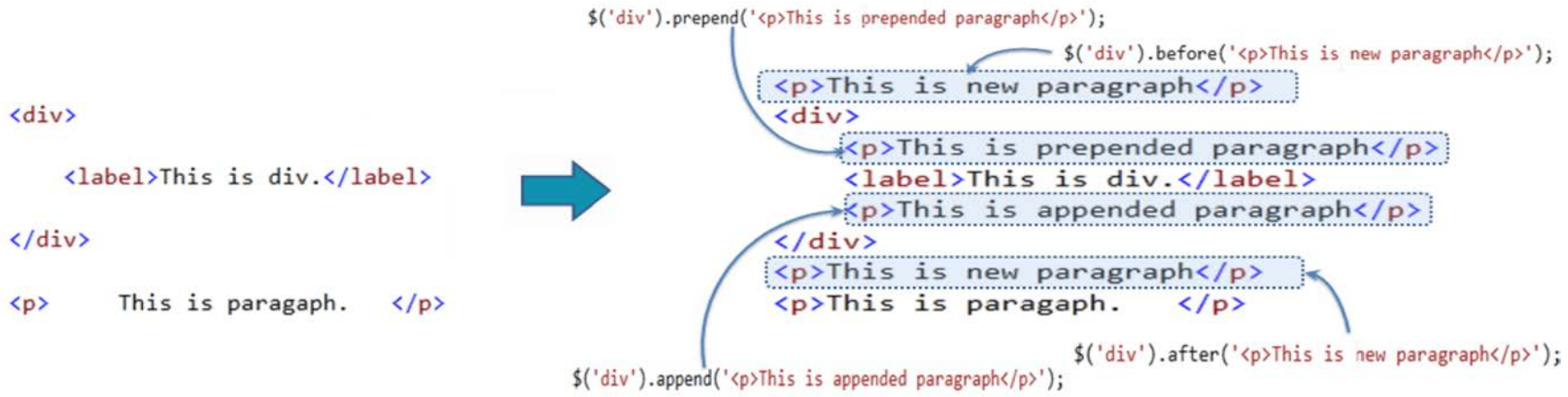

```html
<html>
<head>
    <title>Westcliff Fullstack Bootcamp jQuery CSS Example</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript" language="javascript">
        $(document).ready(function () {
            $("div:first").width(100);
            $("div:first").css("background-color", "blue");
        });
    </script>
    <style>
        div {
            width: 70px;
            height: 50px;
            float: left;
            margin: 5px;
            background: red;
            cursor: pointer;
        }
    </style>
</head>
<body>
    <div>Blue 1</div>
    <div>Red 1</div>
    <div>Red 2</div>
    <div>Red 3</div>
    <div>Red 4</div>
</body>
</html>
```

jQuery DOM

jQuery DOM Manipulation

jQuery DOM

- provides various methods to add, edit or delete DOM element(s) in the HTML page.
- provides methods to manipulate DOM in efficient way. You do not need to write big code to modify the value of any element's attribute or to extract HTML code from a paragraph or division.
- provides methods such as .attr(), .html(), and .val() which act as getters, retrieving information from DOM elements for later use.



jQuery DOM Manipulation #1

Method	Description
after(content)	Insert content after each of the matched elements.
append(content)	Append content to the inside of every matched element.
appendTo(selector)	Append all of the matched elements to another, specified, set of elements.
before(content)	Insert content before each of the matched elements.
clone(bool)	Clone matched DOM Elements, and all their event handlers, and select the clones.
clone( )	Clone matched DOM Elements and select the clones.

jQuery DOM Manipulation #2

Method	Description
empty( )	Remove all child nodes from the set of matched elements.
html(val)	Set the html contents of every matched element.
html( )	Get the html contents (innerHTML) of the first matched element.
insertAfter(selector)	Insert all of the matched elements after another, specified, set of elements.
insertBefore(selector)	Insert all of the matched elements before another, specified, set of elements.
prepend(content)	Prepend content to the inside of every matched element.

jQuery DOM Manipulation #3

Method	Description
prependTo(selector)	Prepend all of the matched elements to another, specified, set of elements.
remove(expr)	Removes all matched elements from the DOM.
replaceAll(selector)	Replaces the elements matched by the specified selector with the matched elements.
replaceWith(content)	Replaces all matched elements with the specified HTML or DOM elements.
text(val)	Set the text contents of all matched elements.
text( )	Get the combined text contents of all matched elements.

jQuery DOM Manipulation #4

Method	Description
wrap(elem)	Wrap each matched element with the specified element.
wrap(html)	Wrap each matched element with the specified HTML content.
wrapAll(elem)	Wrap all the elements in the matched set into a single wrapper element.
wrapAll(html)	Wrap all the elements in the matched set into a single wrapper element.
wrapInner(elem)	Wrap the inner child contents of each matched element (including text nodes) with a DOM element.
wrapInner(html)	Wrap the inner child contents of each matched element (including text nodes) with an HTML structure.

jQuery DOM Manipulation Example

```
<html>
<head>
    <title>Westcliff Fullstack Bootcamp jQuery Dom Manipulation Example</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type = "text/javascript" language = "javascript">
        $(document).ready(function() {
            $("div").click(function () {
                $(this).before('<div class="div"></div>' );
            });
        });
    </script>

    <style>
        .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}
    </style>
</head>
<body>
    <p>Click on any square below:</p>
    <span id = "result"> </span>

    <div class = "div" style = "background-color: blue;"></div>
    <div class = "div" style = "background-color: green;"></div>
    <div class = "div" style = "background-color: red;"></div>
</body>
</html>
```

jQuery Events

jQuery Events

- Actions that can be detected and/or fired by/from your Web Application.
- Examples of Events:
  - A mouse click
  - A web page loading
  - Taking mouse over an element
  - Submitting an HTML form
  - A keystroke on your keyboard, etc.

jQuery Event Attributes #1

Property	Description
altKey	Set to true if the Alt key was pressed when the event was triggered, false if not. The Alt key is labeled Option on most Mac keyboards.
ctrlKey	Set to true if the Ctrl key was pressed when the event was triggered, false if not.
data	The value, if any, passed as the second parameter to the bind() command when the handler was established.
keyCode	For keyup and keydown events, this returns the key that was pressed.
metaKey	Set to true if the Meta key was pressed when the event was triggered, false if not. The Meta key is the Ctrl key on PCs and the Command key on Macs.

jQuery Event Attributes #2

Property	Description
pageX	For mouse events, specifies the horizontal coordinate of the event relative from the page origin.
pageY	For mouse events, specifies the vertical coordinate of the event relative from the page origin.
relatedTarget	For some mouse events, identifies the element that the cursor left or entered when the event was triggered.
screenX	For mouse events, specifies the horizontal coordinate of the event relative from the screen origin.
screenY	For mouse events, specifies the vertical coordinate of the event relative from the screen origin.
shiftKey	Set to true if the Shift key was pressed when the event was triggered, false if not.

jQuery Event Attributes #3

Property	Description
target	Identifies the element for which the event was triggered.
timeStamp	The timestamp (in milliseconds) when the event was created.
type	For all events, specifies the type of event that was triggered (for example, click).
which	For keyboard events, specifies the numeric code for the key that caused the event, and for mouse events, specifies which button was pressed (1 for left, 2 for middle, 3 for right).

jQuery Event Methods

Method	Description
preventDefault()	Prevents the browser from executing the default action.
isDefaultPrevented()	Returns whether event.preventDefault() was ever called on this event object.
stopPropagation()	Stops the bubbling of an event to parent elements, preventing any parent handlers from being notified of the event.
isPropagationStopped()	Returns whether event.stopPropagation() was ever called on this event object.
stopImmediatePropagation()	Stops the rest of the handlers from being executed.
isImmediatePropagationStopped()	Returns whether event.stopImmediatePropagation() was ever called on this event object.

jQuery Event Manipulation Methods #1

Method	Description
bind(type, [data], fn)	Binds a handler to one or more events (like click) for each matched element. Can also bind custom events.
off(events, [selector], [handler(eventObject)])	This does the opposite of live, it removes a bound live event.
hover(over, out)	Simulates hovering for example moving the mouse on, and off, an object.
on(events, [selector], [data], handler)	Binds a handler to an event (like click) for all current − and future − matched element. Can also bind custom events.
one(type, [data], fn)	Binds a handler to one or more events to be executed once for each matched element.

jQuery Event Manipulation Methods #2

Method	Description
ready(fn)	Binds a function to be executed whenever the DOM is ready to be traversed and manipulated.
trigger(event, [data])	Trigger an event on every matched element.
triggerHandler(event, [data])	Triggers all bound event handlers on an element.
unbind([type], [fn])	This does the opposite of bind, it removes bound events from each of the matched elements.

jQuery Event Helper Methods #1

Method	Description
blur( )	Triggers the blur event of each matched element.
blur(fn)	Bind a function to the blur event of each matched element.
change( )	Triggers the change event of each matched element.
change(fn)	Binds a function to the change event of each matched element.
click( )	Triggers the click event of each matched element.
click(fn)	Binds a function to the click event of each matched element.
dblclick( )	Triggers the dblclick event of each matched element.

jQuery Event Helper Methods #2

Method	Description
dblclick(fn)	Binds a function to the dblclick event of each matched element.
error( )	Triggers the error event of each matched element.
error(fn)	Binds a function to the error event of each matched element.
focus( )	Triggers the focus event of each matched element.
focus(fn)	Binds a function to the focus event of each matched element.
keydown( )	Triggers the keydown event of each matched element.
keydown(fn)	Bind a function to the keydown event of each matched element.

jQuery Event Helper Methods #3

Method	Description
keypress( )	Triggers the keypress event of each matched element.
keypress(fn)	Binds a function to the keypress event of each matched element.
keyup( )	Triggers the keyup event of each matched element.
keyup(fn)	Bind a function to the keyup event of each matched element.
load(fn)	Binds a function to the load event of each matched element.
mousedown(fn)	Binds a function to the mousedown event of each matched element.
mouseenter(fn)	Bind a function to the mouseenter event of each matched element.

jQuery Event Helper Methods #4

Method	Description
mouseleave(fn)	Bind a function to the mouseleave event of each matched element.
mousemove(fn)	Bind a function to the mousemove event of each matched element.
mouseout(fn)	Bind a function to the mouseout event of each matched element.
mouseover(fn)	Bind a function to the mouseover event of each matched element.
mouseup(fn)	Bind a function to the mouseup event of each matched element.
resize(fn)	Bind a function to the resize event of each matched element.
scroll(fn)	Bind a function to the scroll event of each matched element.

jQuery Event Helper Methods #5

Method	Description
select( )	Trigger the select event of each matched element.
select(fn)	Bind a function to the select event of each matched element.
submit( )	Trigger the submit event of each matched element.
submit(fn)	Bind a function to the submit event of each matched element.
unload(fn)	Binds a function to the unload event of each matched element.

jQuery Events Sample

```
<html>
<head>
    <title>Westcliff Fullstack Bootcamp jQuery Event Example</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript" language="javascript">
        $(document).ready(function () {
            $('div').bind('click', function (event) {
                alert('Event type is ' + event.type);
                alert('pageX : ' + event.pageX);
                alert('pageY : ' + event.pageY);
                alert('Target : ' + event.target.innerHTML);
            });
        });
    </script>
    <style>
        .div {
            margin: 10px;
            padding: 12px;
            border: 2px solid #666;
            width: 60px;
        }
    </style>
</head>
<body>
    <p>Click on any square below to see the result:</p>

    <div class="div" style="background-color: blue;">ONE</div>
    <div class="div" style="background-color: green;">TWO</div>
    <div class="div" style="background-color: red;">THREE</div>
</body>
</html>
```

jQuery Ajax

jQuery Ajax

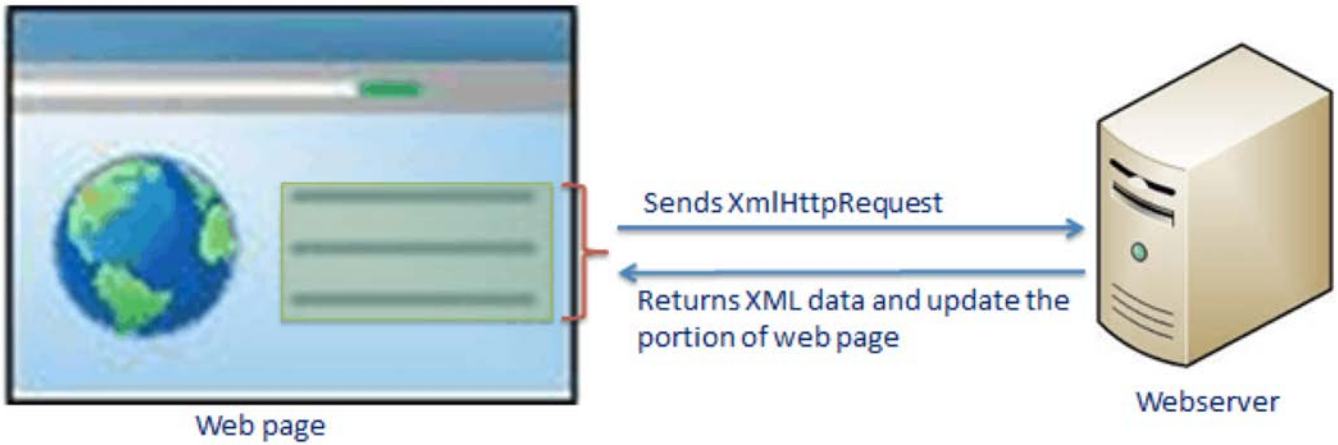
Ajax stands for "***A****synchronous* ***J****avaScript* ***a****nd* ***X****ML*".

JavaScript includes features of sending asynchronous http request using *XMLHttpRequest* object. Ajax is about using this ability of JavaScript to send asynchronous http request and get the xml data as a response (also in other formats) and update the part of a web page (using JavaScript) without reloading or refreshing entire web page.

With the jQuery AJAX methods, you can request text, HTML, XML, or JSON from a remote server using both HTTP Get and HTTP Post - And you can load the external data directly into the selected HTML elements of your web page!

jQuery Ajax

The following figure illustrates the Ajax functionality.



jQuery Ajax

- Here's the Syntax for a method:

```
selector.method( )
```

- For example:
  - selector.html()
  - selector.load()
  - selector.append()
- Script examples:
  - $('div').html('
Name:' + team.name + '
')
  - $('#info').load('results.html')
  - $('p .p1').append(data)

jQuery Ajax

- As an example, let's look at the load() method in a little more detail and understand how it works. The syntax for this method is:

  [selector].load(URL, [data], [callback]);

  - **URL** − The URL of the server-side resource to which the request is sent.
  - **data** − This optional parameter represents an object whose properties are serialized into properly encoded parameters to be passed to the request. If specified, the request is made using the POST method. If omitted, the GET method is used.
  - **callback** − A callback function invoked after the response data has been loaded into the elements of the matched set. The first parameter passed to this function is the response text received from the server and second parameter is the status code.

jQuery Ajax

- So we have the following jQuery script:

```
$(document).ready(function() {
  $("#btn").click(function(event){
    $('#showinfo').load('/jq/result.html');
  });
});
```

- And the following html:

```
<p>Click button to load result.html file</p>

<div id = "showinfo" style = "background-color:cc0;">SHOW DATA</div>

<input type = "button" id = "btn" value = "Load Data" />
```

- Here load() initiates an Ajax request to the specified URL /jq/result.html file. After loading this file, all the content would be populated inside tagged with ID *showinfo*. Assuming, our /jq/result.html file has just one HTML line:

```
<h1>THIS IS RESULT...</h1>
```

jQuery Ajax

- Let's look at another example - the getJSON() method in a little more detail and understand how it works as well. The syntax for this method is:

  [selector].getJSON(URL, [data], [callback]);

  - **URL** − The URL of the server-side resource to which the request is sent.
  - **data** − This optional parameter represents an object whose properties are serialized into properly encoded parameters to be passed to the request. If specified, the request is made using the POST method. If omitted, the GET method is used.
  - **callback** − A callback function invoked after the response data has been loaded into the elements of the matched set. The first parameter passed to this function is the response text received from the server and second parameter is the status code.

jQuery Ajax

- So we have the following jQuery script:

```
$(document).ready(function(){
  $("btn").click(function(event){
    $.getJSON('/jq/result.json', function(jqdata) {
      $('#showinfo').html('<p> Name: ' + jqdata.name + '</p>');
      $('#showinfo').append('<p>Age : ' + jqdata.age+ '</p>');
      $('#showinfo').append('<p> Sex: ' + jqdata.sex+ '</p>');
    });
  });
});
```

- And the following html:

```
<p>Click button to load result.json file</p>
<div id = "showinfo" style = "background-color:#eee;">SHOW DATA</div>
<input type = "button" id = "btn" value = "Load Data" />
```

- Here JQuery utility method getJSON() initiates an Ajax request to the specified URL result.json file. After loading this file, all the content would be passed to the callback function which finally would be populated inside tagged with ID *stage*. Assuming, our result.json file has following json formatted content:

```
{
   "name": "Zara Ali",
   "age" : "67",
   "sex": "female"
}
```

jQuery Ajax Methods #1

Methods	Description
jQuery.ajax(options)	Load a remote page using an HTTP request.
jQuery.ajaxSetup(options)	Setup global settings for AJAX requests.
jQuery.get(url, [data], [callback], [type])	Load a remote page using an HTTP GET request.
jQuery.getJSON(url, [data], [callback])	Load JSON data using an HTTP GET request.
jQuery.getScript(url, [callback])	Loads and executes a JavaScript file using an HTTP GET request.
jQuery.post(url, [data], [callback], [type])	Load a remote page using an HTTP POST request.

jQuery Ajax Methods #2

Methods	Description
load(url, [data], [callback])	Load HTML from a remote file and inject it into the DOM.
serialize( )	Serializes a set of input elements into a string of data.
serializeArray( )	Serializes all forms and form elements like the .serialize() method but returns a JSON data structure for you to work with.

jQuery Ajax Events

Methods	Description
ajaxComplete(callback)	Attach a function to be executed whenever an AJAX request completes.
ajaxStart(callback)	Attach a function to be executed whenever an AJAX request begins and there is none already active.
ajaxError(callback)	Attach a function to be executed whenever an AJAX request fails.
ajaxSend(callback)	Attach a function to be executed before an AJAX request is sent.
ajaxStop(callback)	Attach a function to be executed whenever all AJAX requests have ended.
ajaxSuccess(callback)	Attach a function to be executed whenever an AJAX request completes successfully.

jQuery Ajax Example

```
<html>
    <head>
        <title>Westcliff Fullstack Bootcamp jQuery Ajax Example</title>
        <script type = "text/javascript"
            src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
        </script>

        <script type = "text/javascript" language = "javascript">
            $(document).ready(function() {
                $("#btn").click(function(event){

                    $.getJSON('/jq/result.json', function(jqdata) {
                        $('#showinfo').html('<p> Name: ' + jqdata.name + '</p>');
                        $('#showinfo').append('<p>Age : ' + jqdata.age+ '</p>');
                        $('#showinfo').append('<p> Sex: ' + jqdata.sex+ '</p>');
                    });

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

    <body>
        <p>Click button to load result.json file</p>

        <div id = "showinfo" style = "background-color:■#eee;">SHOW DATA</div>

        <input type = "button" id = "btn" value = "Load Data" />
    </body>
</html>
```

jQuery Effects

jQuery Effects

jQuery provides a simple interface for doing various kind of visual effects. jQuery methods allow us to quickly apply commonly used effects with a minimum configuration. Effects can use the built-in settings, or provide a customized duration. You can also create custom animations of arbitrary CSS properties.

We will use one of the most common effects as an example: show/hide elements. The commands for showing and hiding elements are pretty much what we would expect − show() to show the elements in a wrapped set and hide() to hide them.

jQuery Effects

Here is the simple syntax for show() method:

```
[selector].show( speed, [callback] );
```

And here is the syntax for hide() method:

```
[selector].hide( speed, [callback] );
```

- **speed** − A string representing one of the three predefined speeds ("slow", "normal", or "fast") or the number of milliseconds to run the animation (e.g. 1000).
- **callback** − This optional parameter represents a function to be executed whenever the animation completes; executes once for each element animated against.

jQuery Effects

We have the following jQuery scripts:

```
$(document).ready(function() {
   $("#show").click(function () {
      $(".mydiv").show( 1000 );
   });
   $("#hide").click(function () {
      $(".mydiv").hide( 1000 );
   });
});
```

And the following HTML:

```
<div class = "mydiv">I Can Show and Hide!</div>

<input id = "hide" type = "button" value = "Hide" />
<input id = "show" type = "button" value = "Show" />
```

And CSS:

```
<style>
   .mydiv{
      margin:10px;
      padding:12px;
      font-size: 2em;
      background-color:rgb(236, 196, 65);
      width:90%;
      height:150px;
   }
</style>
```

jQuery Effects #1

Methods	Description
animate(params, [duration, easing, callback])	A function for making custom animations.
fadeIn(speed, [callback])	Fade in all matched elements by adjusting their opacity and firing an optional callback after completion.
fadeOut(speed, [callback])	Fade out all matched elements by adjusting their opacity to 0, then setting display to "none" and firing an optional callback after completion.
fadeTo(speed, opacity, callback)	Fade the opacity of all matched elements to a specified opacity and firing an optional callback after completion.
hide( )	Hides each of the set of matched elements if they are shown.

jQuery Effects #2

Methods	Description
hide(speed, [callback])	Hide all matched elements using a graceful animation and firing an optional callback after completion.
show( )	Displays each of the set of matched elements if they are hidden.
show(speed, [callback])	Show all matched elements using a graceful animation and firing an optional callback after completion.
slideDown(speed, [callback])	Reveal all matched elements by adjusting their height and firing an optional callback after completion.
slideToggle(speed, [callback])	Toggle the visibility of all matched elements by adjusting their height and firing an optional callback after completion.
slideUp(speed, [callback])	Hide all matched elements by adjusting their height and firing an optional callback after completion.

jQuery Effects #3

Methods	Description
stop([clearQueue, gotoEnd])	Stops all the currently running animations on all the specified elements.
toggle( )	Toggle displaying each of the set of matched elements.
toggle(speed, [callback])	Toggle displaying each of the set of matched elements using a graceful animation and firing an optional callback after completion.
toggle(switch)	Toggle displaying each of the set of matched elements based upon the switch (true shows all elements, false hides all elements).
jQuery.fx.off	Globally disable all animations.

UI Library Based Effects #1

Sr.No.	Methods & Description
Blind	Blinds the element away or shows it by blinding it in.
Bounce	Bounces the element vertically or horizontally n-times.
Clip	Clips the element on or off, vertically or horizontally.
Drop	Drops the element away or shows it by dropping it in.
Explode	Explodes the element into multiple pieces.
Fold	Folds the element like a piece of paper.

UI Library Based Effects #2

Sr.No.	Methods & Description
Highlight	Highlights the background with a defined color.
Puff	Scale and fade out animations create the puff effect.
Pulsate	Pulsates the opacity of the element multiple times.
Scale	Shrink or grow an element by a percentage factor.
Shake	Shakes the element vertically or horizontally n-times.
Size	Resize an element to a specified width and height.

UI Library Based Effects #3

Sr.No.	Methods & Description
Slide	Slides the element out of the viewport.
Transfer	Transfers the outline of an element to another.

jQuery Effects Example

```html
<html>
<head>
    <title>Westcliff Fullstack Bootcamp jQuery Effects Example</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript" language="javascript">
        $(document).ready(function () {
            $(".clickme").click(function (event) {
                $(".target").toggle('slow', function () {
                    $(".log").text('Transition Complete');
                });
            });
        });
    </script>
    <style>
        .clickme {
            margin: 10px;
            padding: 12px;
            border: 2px solid #666;
            width: 100px;
            height: 50px;
        }
    </style>
</head>
<body>
    <div class="content">
        <div class="clickme">Click Me</div>
        <div class="target">
            <img src="./images/jquery.png" alt="jQuery" />
        </div>
        <div class="log"></div>
    </div>
</body>
</html>
```

Questions?

Resources

https://www.w3schools.com/jquery/
https://learn.jquery.com/using-jquery-core/