



**WESTCLIFF**  
**UNIVERSITY**  
Educate. Inspire. Empower.

# MODULE 2

Week 1: jQuery



# Agenda

- jQuery Basics
  - About jQuery
  - How to use jQuery
  - jQuery Data Types
  - jQuery Fundamentals
  - HTML DOM – A Review
- jQuery Selectors
- jQuery Attributes
- jQuery Traversing
- jQuery CSS
- In-Class Demo
- Homework

# 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>`
    - available on the **url.txt** file

# jQuery Example with Local Installation



```
<> jquery-example.html x
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 X
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 x
fullstack-bootcamp > <> jquery-function-example.html > html > head > script
1 Live Demo
2 <html>
3   <head>
4     <title>The jQuery Example</title>
5     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
6
7     <script type = "text/javascript" language = "javascript">
8     ..... $(document).ready(function() {
9     .....   $("div").click(function() {alert("Hello, world!");});
10    .....   });
11    ..... </script>
12  ..... </head>
13
14  <body>
15    <div id = "mydiv">
16      Click on this to see a dialogue box.
17    </div>
18  </body>
19 </html>
20
```

# 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();  
sample_function(1);  
sample_function("1", "2", "3");
```

```
//==> "undefined", 0  
//==> "number", 1  
//==> "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



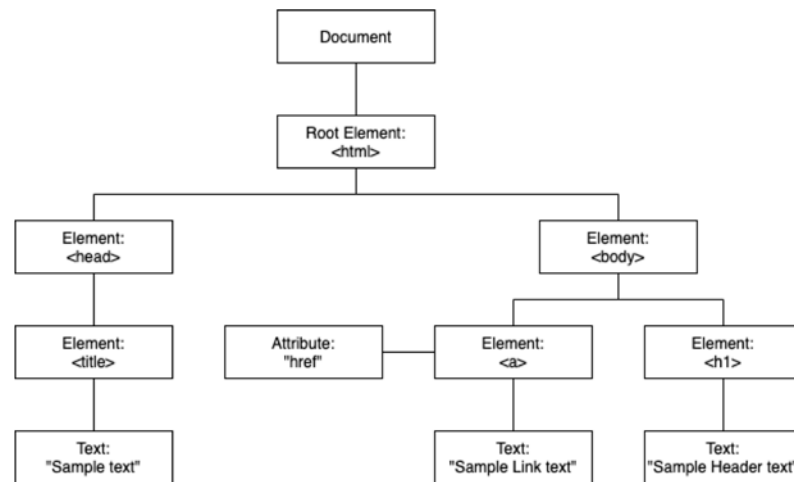
- JavaScript functions passed to some method as an argument or option.
- Some callbacks require return values, others make the return values optional.
- Ex:
  - `$("#body").click(function(event) {  
 console.log("clicked: " + event.target);  
});`
  - `$("#some-form").submit(function() {  
 return false;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

# HTML DOM – A Review

# Document Object Model



- The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document.





# HTML DOM



- A standard object model and programming interface for HTML.
- Defines:
  - The HTML elements as objects
  - The properties of all HTML elements
  - The methods to access all HTML elements
  - The events for all HTML elements

# HTML DOM Document Object

## #1



- Methods for Finding HTML Elements
  - `document.getElementById(id)`
    - Find an element by element id
  - `document.getElementsByTagName(name)`
    - Find elements by tag name
  - `document.getElementsByClassName(name)`
    - Find elements by class name

# HTML DOM Document Object

## #2



- Methods and Properties for Updating HTML Elements
  - Property
    - `element.innerHTML = new html content`
      - Change the inner HTML of an element
    - `element.attribute = new value`
      - Change the attribute value of an HTML element
    - `element.style.property = new style`
      - Change the style of an HTML element
  - Method
    - `element.setAttribute(attribute, value)`
      - Change the attribute value of an HTML element

# HTML DOM Document Object

## #3



- Methods for Add or Delete of HTML Elements
  - `document.createElement(element)`
    - Create an HTML element
  - `document.removeChild(element)`
    - Remove an HTML element
  - `document.appendChild(element)`
    - Add an HTML element
  - `document.replaceChild(new, old)`
    - Replace an HTML element
  - `document.write(text)`
    - Write into the HTML output stream

# HTML DOM Document Finding Elements



document.anchors	Returns all <a> elements that have a name attribute
document.baseURI	Returns the absolute base URI of the document
document.body	Returns the <body> element
document.cookie	Returns the document's cookie
document.doctype	Returns the document's doctype
document.documentElement	Returns the <html> element
document.documentMode	Returns the mode used by the browser
document.documentURI	Returns the URI of the document
document.domain	Returns the domain name of the document server
document.embeds	Returns all <embed> elements
document.forms	Returns all <form> elements
document.head	Returns the <head> element

document.images	Returns all <img> elements
document.implementation	Returns the DOM implementation
document.inputEncoding	Returns the document's encoding (character set)
document.lastModified	Returns the date and time the document was updated
document.links	Returns all <area> and <a> elements that have a href attribute
document.readyState	Returns the (loading) status of the document
document.referrer	Returns the URI of the referrer (the linking document)
document.scripts	Returns all <script> elements
document.strictErrorChecking	Returns if error checking is enforced
document.title	Returns the <title> element
document.URL	Returns the complete URL of the document
document.images	Returns all <img> elements

# HTML DOM Event



- [HTML DOM Events](#)
- Common HTML DOM Events

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

# 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 sample



```
<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 1<sup>st</sup> 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>
    
  </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>
<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
<code>css( name )</code>	Return a style property on the first matched element.
<code>css( name, value )</code>	Set a single style property to a value on all matched elements.
<code>css( properties )</code>	Set a key/value object as style properties to all matched elements.
<code>height( val )</code>	Set the CSS height of every matched element.
<code>height( )</code>	Get the current computed, pixel, height of the first matched element.
<code>innerHeight( )</code>	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>
<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>
```





Questions?

## Resources

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