Week 09 | jQuery

jQuery is a lightweight, "write less, do more" library of JavaScript functions. It can be added to a Web page with only a single line of markup.

Adding jQuery to Web Pages

jQuery library is stored as a single JavaScript file that contains all jQuery methods. There are several ways to start using jQuery on your website:

- Download the jQuery library from jQuery.com (http://jquery.com/).

```
<head>
	<script src="jquery-1.11.3.min.js"></script>
</head>
```

- Include jQuery from a Content Delivery Network (CDN).

```
<head>
	<script src="//code.jquery.com/jquery-1.11.3.min.js"></script>
</head>
```

  Many users already have downloaded jQuery from Google or Microsoft when visiting another site. As a result, it will be loaded from cache when they visit your site, which leads to faster loading time. Also, most CDN's will make sure that once a user requests a file from it, it will be served from the server closest to them, which also leads to faster loading time.

jQuery Syntax

Basic Syntax: `$(selector).method( )` where:

- `$` - alias for jQuery
- `selector` - queries or finds HTML elements
- `method( )` - method to be performed on the elements

jQuery methods are inside a document ready event to prevent any jQuery code from running before the document is finished loading (is ready).

```
$(document).ready(function(){
	// jQuery methods go here...
});

// even shorter method

$(function(){
	// jQuery methods go here...
});
```

jQuery Concepts

1. **Selectors**

   jQuery selectors allow you to select and manipulate HTML element(s). All selectors in jQuery start with the dollar sign and parentheses: `$()`.

```
$( "body" );
$( "#myId" ); // Note IDs must be unique per page.
$( ".myClass" );
$( "input[name='first_name']" ); // Beware, this can be very slow in older browsers
$( "#contents ul.people li" );
```

2. **Event Handlers**

   The `$(document).ready()` method allows us to execute a function when the document is fully loaded.

```
// Event setup using a convenience method
$( "p" ).click(function() {
       alert( "This is a paragraph!" );
       $( this ).hide();
});

// Equivalent event setup using the `.on()` method
$( "p" ).on( "click", function() {
       alert( "This is a paragraph!" );
});

// Binding multiple events with different handlers
$( "p" ).on({
   "click": function() { console.log( "Clicked!" ); },
   "mouseover": function() { console.log( "Hovered!" ); }
});

// Bind handlers for multiple events
$( "input" ).on("click change", function() {
       console.log( "Clicked or changed?" );
});
```

To remove an event listener, the `.off()` method is used.

```
// Syntax: $(selector).off("event");
// Tearing down all click handlers on a selection
$("button").click(function(){
       $("p").off("mouseover");
});
```

jQuery provides the `.one()` method to specify that a particular handler will only once.

3. **Effects**

Syntax: `$(selector).effect(speed,`*`callback`*`);`

```
<button>Toggle</button>
<h1>Hello World!</h1>

<script>
       $("button").click(function(){
              $("p").toggle();   // fadeToggle, slideToggle
       });
</script>
```

4. **HTML Manipulation**

`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

Change Manipulation Content

```
// Syntax: $(selector).html("String");
$("p").html("This is my page.");
alert($("p").html);
```

Add HTML Content

```
// Syntax: $(selector).[append|prepend|after|before]("String");
$("p").append("See you soon."); // adds content at the end of HTML element
$("p").prepend("How are you?"); // adds content at the beginning of HTML element
$("p").after("Bye!"); // inserts HTML content after the HTML element
$("p").before("Hi there!"); // inserts HTML content before the HTML element
```

5. **CSS Manipulation**

   `addClass()` - Adds one or more classes to the selected elements
   `attr()` - Sets or returns attribute values
   `removeClass()` - Removes one or more classes from the selected elements
   `css()` - Sets or returns the style attribute

```
// Syntax: $(selector).css("property");
$("p").css("background"); //get CSS property
$("p").css("background", "yellow"); // set CSS property and value
$("p").css({"background": "yellow", "font-family": "Arial"}); // set multiple
$("p").addClass("p1 para1");
$("p").removeClass("p1");
$("input").attr("placeholder","Name");
```

Handling HTTP Requests in jQuery

jQuery provides several AJAX methods to handle HTTP Request. AJAX stands for Asynchronous JavaScript and XML. This means that AJAX is about loading data in the background and displaying it on the webpage without reloading the whole page.

With jQuery AJAX methods, it is possible to request text, HTML, XML or JSON from a remote server using both HTTP Get and HTTP Post.

- `load( )`
  The `load( )` method loads data from a server and puts the returned data into the selected element. It is similar to `$.get()` but allows you to define where in the document the returned data is to be inserted.

```
// Syntax: $(selector).load(URL, data, callback);
// Using .load() to populate an element
$( "p" ).load( "temp.html" );
```

```
// Using .load() to populate an element based on a selector
$( "#p1" ).load( "anotherpage.html", function( html ) {
      alert( "Content updated!" );
});
```

- `get( )`
  The `get( )` method performs a GET request to the provided URL.

```
// Syntax: $.get(URL, callback);
// Get plain text or HTML
$.get( "anotherpage.html", function( resp ) {
      $("#p1").append(resp);
});
```

- `post( )`
  The `post( )` method performs a POST request to the provided URL.

```
// Syntax: $.post(URL, data, callback);
$.post( "http://localhost:5000/degree-programs", {
       code: 'BSCS',
       name: 'Bachelor of Science in Computer Science'
       }, function( resp ) {
               console.log( resp ); // server response
       });
```

- `getScript( )`

  The `getScript( )` adds and loads a script to the page.

```
// Syntax: $(selector).getScript(URL, data, callback);
// Add a script to the page, then run a function defined in it
$.getScript( "/static/js/myScript.js", function() {
        functionFromMyScript();
});
```

- `getJSON( )`

  The `getJSON( )` performs a GET request, and expect JSON to be returned.

```
// Syntax: $(selector).getJSON(URL, data, callback);
// Using .load() to populate an element
// Get JSON-formatted data from the server
$.getJSON( "http://localhost:5000/students", function( resp ) {
        // Log each key in the response data
        $.each( resp, function( key, value ) {
                console.log( key + " : " + value );
        });
});
```

jQuery Template

jQuery Templates are client-side based templates. The benefits of using jQuery Templates are:

1. Easily convert JSON object to HTML without need for parsing
2. Reusability
3. Rendered and cached on client-side
4. Templates are written in pure HTML with Template Tags and simple jQuery code for magic to happen
5. Maximize the separation of UI and DATA

- `$.template()` - create named templates
- `$.tmpl()` - render the template

```
<!DOCTYPE html>
<html>
        <head>
                <title>CMSC 100 jQuery</title>
                <meta charset="utf-8" />
                <script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
                <script
src="http://ajax.microsoft.com/ajax/jquery.templates/beta1/jquery.tmpl.min.js"></script>
        </head>
        <body>
                <div id="greeting"></div>
                <script>
                        var obj = {fname: "Betel", lname: "de Robles"};
                        $.get("templates/greeting.html", function(temp){
                                $.template("markup", temp);
                                $.tmpl("markup",obj).appendTo("#greeting");
                        });
                </script>
```

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

```
<div class="names">
	<h1>Hello ${fname} ${lname}!</h1>
</div>
```

EXERCISE

REFERENCES

jQuery API Documentation. (n.d.) Retrieved from http://api.jquery.com/
jQuery Learning Center. (n.d.) Retreived from http://learn.jquery.com/
jQuery Tutorial (n.d.). Retrieved from http://w3schools.com/jquery/default.asp.
jQuery: The Write Less, Do More, JavaScript Library. (n.d.). Retrieved from http://jquery.com/