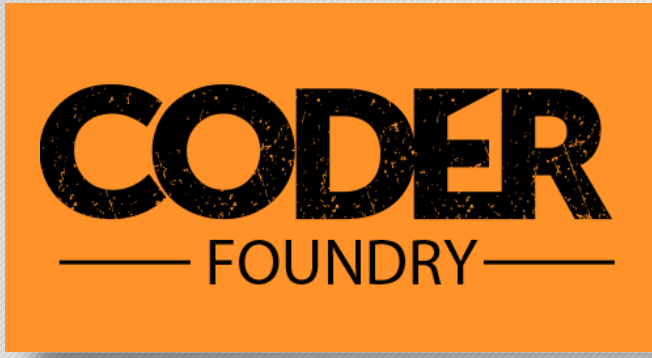
jQuery, AJAX, JSON



What is jQuery?

CODER FOUNDRY

- Lightweight, "write less, do more", JavaScript library
- Makes using JavaScript on your website much easier
- Common tasks:
  - JavaScript = many lines of code
  - jQuery = single line of code
- Simplifies complicated JavaScript actions such as AJAX calls and DOM manipulation.

jQuery

- The jQuery library contains the following features:
  - HTML/DOM manipulation
  - CSS manipulation
  - HTML event methods
  - Effects and animations
  - AJAX
  - Other Utilities

Adding jQuery to Web Pages

CODER FOUNDRY

- Download the jQuery library from jQuery.com
  - Production version - this is for your live website because it has been minified and compressed
  - Development version - this is for testing and development (uncompressed and readable code)
  - Reference in your HTML:

```
<head>
    <script src="jquery-1.11.2.min.js"></script>
</head>
```

Adding jQuery to Web Pages

CODER FOUNDRY

- Include jQuery from a CDN
  - jquery.com Reference

```
<head>
    <script src="//code.jquery.com/jquery1.11.2.min.js"></script>
</head>
```

  - Other CDNs
    - Microsoft
    - Google
    - CDNJS
    - jsDelivr

jQuery Versions

- 1.11.x
  - Supports all major browsers
- 2.1.3
  - Does not support Internet Explorer 6, 7, or 8
- Both versions available in Development and Production releases
  - jquery.com/download

jQuery Syntax

CODER FOUNDRY

- $(selector).action()
  - A $ sign to define/access jQuery
  - A (selector) to "query (or find)" HTML elements
  - A jQuery action() to be performed on the element(s)
- Examples:
  - $(this).hide() - hides the current element
  - $("p").hide() - hides all <p> elements
  - $(".test").hide() - hides all elements with class="test"
  - $("#test").hide() - hides the element with id="test"

document.ready() Event

- Prevent any jQuery code from running before the document is finished loading by placing ALL jQuery code inside the *ready* event:

```
$(document).ready(function(){

    // jQuery methods go here...

});
```

jQuery Events

- Selectors are used to assign events just as to gain access to HTML element data
  - For example:

```
$("p").click(function(){
        // action goes here
});
```

    assigns a click event to all paragraph elements on the page

jQuery Events

- And…

```
$("#p1").hover(function(){
        alert("You entered p1!");
},
function(){
        alert("Bye! You now leave p1!");
});
```

assigns a hover event action to the element with an id property of “p1”

Getting/Setting jQuery Content

CODER FOUNDRY

- Three simple methods
  - 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
  - ```
    $("#btn1").click(function(){
      alert("Text: " + $("#test").text());
    });
    ```
    ```

Lots More to jQuery

- We have provided essential concepts
- Further independent study for additional concepts
  - DOM element enumeration and traversing
  - DOM element addition and removal
  - CSS manipulation
  - BOM window dimension manipulation
  - And more

AJAX

CODER FOUNDRY

- What is it?
  - Asynchronous JavaScript and XML
  - Exchanging data with a server and updating web pages, all without reloading the whole page
  - Application examples:
    - Gmail, Google Maps, Youtube, and Facebook tabs

AJAX and jQuery

CODER FOUNDRY

- jQuery provides methods for AJAX
  - .load()
  - .get()
  - .post()
  - See http://www.w3schools.com/jquery/jquery_ref_ajax.asp for a complete reference

.load()

- Syntax:
  - $(selector).load(URL,data,callback);
  - URL parameter is required and specifies the URL you wish to load
  - data parameter is optional, specifies a set of querystring key/value pairs to send along with the request
  - callback parameter is optional, specifies the name of a function to be executed after the load() method is completed

GET and POST

CODER FOUNDRY

- jQuery get() and post() methods are used to request data from the server with an HTTP GET or POST request
  - GET - Requests data from a specified resource
  - POST - Submits data to be processed to a specified resource
- $.get() method requests data from the server with an HTTP GET request
  - $.get(URL,callback);
  - callback is optional
- $.post() method requests data from the server using an HTTP POST request
  - $.post(URL,data,callback);
  - data and callback are optional

JSON

- What is it?
  - JavaScript Object Notation
  - syntax for storing and exchanging data.
  - easier to use alternative to XML
  - follows key/value pair methodology
  - easily evaluates to JavaScript objects

```
{"employees": [
    {"firstName":"John", "lastName":"Doe"},
    {"firstName":"Anna", "lastName":"Smith"},
    {"firstName":"Peter", "lastName":"Jones"}
] }
```

JSON to JavaScript

- var text = '{"name":"John Johnson","street":"Oslo West 16", "phone":"555 1234567"}'
- var obj = JSON.parse(text);
- document.getElementById("demo").innerHTML =
  obj.name + "
" +
  obj.street + "
" +
  obj.phone;

JSON Rules

- JSON syntax is a subset of the JavaScript object notation syntax:
  - Data is in name/value pairs
    - "firstName":"John"
  - Data is separated by commas
  - Curly braces hold objects
  - Square brackets hold arrays

JSON Rules

- JSON values can be:
  - number (integer or floating point)
  - string (in double quotes)
  - boolean (true or false)
  - array (in square brackets)
  - object (in curly braces)
  - null

JSON and jQuery

- Use jQuery and AJAX to get JSON data from server:

```
$("button").click(function(){
        $.getJSON("demo_ajax_json.js",function(result){
            $.each(result, function(i, field){
                $("div").append(field + " ");
            });
        });
});
```

jQuery, AJAX, JSON

CODER FOUNDRY

- Use all three together to:
  - quickly and easily get/post from/to server
  - any data
  - from any data source
  - without reloading pages
- Dynamic web pages
  - The heart of Bootstrap
  - We'll see this whole concept "on steroids" when we discover AngularJS in week 7 of the Coder Foundry curriculum