

Dynamic Web Pages

with the DOM and jQuery

Course's Agenda

- ~~Javascript Basics~~
- ~~Advanced Javascript~~
- **Introduction to the DOM**
- **Easier DOM with jQuery**
- Transitions and Animations
- Talking to the server with AJAX

Today's Agenda

- Review
- Introduction to the DOM
- jQuery basics

Review

Cheatsheet

```
var number = 5;  
var string = "Hello";  
var array = [1, 2, 3];  
var object = {a: 1, b: 2};  
var func = function (x) { return x * x; }
```

```
if (object.a == 1) {  
    console.log("A is 1");  
}
```

```
for (var i = 0; i < array.length; i += 1) {  
    var value = func(array[i]);  
    console.log(value);  
}
```

Demo

Implement filter function in Javascript.

```
var isEven = function (x) { return x % 2 == 0; };
```

```
filter([1, 2, 3, 4], isEven); // => [2, 4]
```

Introduction to the

DOM

**The DOM is an API
for your web page**

But what is the DOM? **for serious**

A collection of Javascript objects that represent the HTML document of the current page.

DOM objects contain methods for querying and manipulating web pages, and saying what should happen when the user makes certain actions.

e.g. "When the user clicks this button, make all the links red."

What can you do with the DOM?

1. Make changes to the HTML after the page has loaded
2. Respond to a users actions (events)
3. Any combination of 1 and 2

Real World Examples

- <http://mealime.com>
- <http://next.mitchcrowe.com>
- harmonikit <https://dry-inlet-2631.herokuapp.com>

It all starts with window

- window is the global object and default receiver
- "global" functions and objects are actually properties of window
- <https://developer.mozilla.org/en-US/docs/Web/API/Window>

An HTML Document

```
<html>
  <head>
    <script>...</script>
    ...
  </head>
  <body>
    <h1>Title</h1>
    <a id='link' href="http://www.google.com">Link</a>
  </body>
</html>
```

*"The Document Object
Model represents an
HTML document as a tree
of Javascript objects"*

DOM Basics

Demo

The DOM can be a pain

- Lots of junk left over from history
- Cross-browser compatibility issues
- Very verbose
- Not very friendly on the programmer.

jQuery

Why jQuery?

- DOM made easy
- Cross-browser compatibility
- Overwhelmingly popular

Which is simpler?

```
// "vanilla" DOM
document.getElementById( 'link' ).addEventListener( "click", function() {
    console.log( 'Clicked!' );
});
```

```
// jQuery
$( '#link' ).on( 'click', function() {
    console.log( 'Clicked!' );
});
```

jQuery Basics

jQuery in a nutshell

1. Select elements
2. Manipulate selected elements
3. Add event listeners to selected elements

```
// Hide all image tags.  
$('img').hide();
```

```
// Write a message when any links are clicked.  
$('a').on('click', function() {  
    console.log('that tickles!');  
});
```

Overview

<http://overapi.com/jquery/>

Our First jQuery Page

```
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/
libs/jquery/1.11.2/jquery.min.js"></script>
    <script>
      $(document).ready(function () {
        $('h1').html('Welcome to jQuery!');
      });
    </script>
  </head>
  <body>
    <h1>Welcome to HTML</h1>
  </body>
</html>
```


Review: Dissecting an HTML element

Element, Attributes, Contents, Id, Class

```
<div id="my-div" class="my-class" width="50">  
  Hello World  
</div>
```

Selectors

"Choose which elements on the page we want to work with"

**jQuery selectors are just
CSS selectors!**

Review CSS Selectors

1. *
2. tagName
3. .class
4. #id
5. ancestor descendant
6. selector1, selector2, ...

What does each of the following CSS selectors do?

1. `img`
2. `img.large`
3. `#reset-button`
4. `ul#homework li`
5. `img, a`
6. `#reset-button, #publish-button`

Use \$ to select elements in jQuery

```
// Select all elements that match "css selector"  
$("css selector")
```

- \$ is a *normal function* that takes a string as an argument.

Exercise

Go to <http://amazon.com>, open the console, and type:

```
$( 'img' )
```

Now, hover over the returned value in the console to highlight that element on the page.

Now type:

```
$( 'img' ).length
```

What does `.length` return?

What does it select?

```
<div class="shape">  
  <a href="#" id="toggle">Toggle Me</a>  
</div>  
<div class="shape"></div>
```

1. `$('.shape')`
2. `$('#toggle')`
3. `$('a, div')`
4. `$('a div')`

Introduction to the jQuery Lab

Exercises 1 / 2

1. Select all shapes
2. Select all black shapes
3. Count the number of shapes
4. Count the number of black shapes
5. Count the number of black OR blue shapes
6. Select all h1 tags
7. Select all small circles

Exercises 2/2

1. Select all red shapes inside the orange container
2. Select all medium or small shapes inside the green container
3. Select all shapes inside any container
4. Select all link tags
5. Select all the link tags inside an "li" tag
6. Count the number of small blue circles

Discussion

- jQuery's selector method returns a "wrapped set".
- "wrapped sets" have a bunch of useful methods
- Use eq to get the nth matching element.

Attributes

"Set or get the value of an attribute on an element."

attr: Set and Get attributes on elements

```
// Get the value of "src" attribute.
```

```
$( 'img' ).attr( 'src' );
```

```
// Set the value of "src" attribute.
```

```
$( 'img' ).attr( 'src', 'http://google.com/logo.png' );
```

Exercises

1. Get the "href" attribute of the first link on the page
2. Set the "href" attribute of all links on the page to "http://www.codecore.ca". Try clicking one.
3. Set the "class" attribute of all links to "highlight"
4. Set the "class" attributes of all shapes to "highlight". What happened?

Discussion

- Manipulations happen to all matched elements
- Reading happens only on first matched element
- Setting "class" attribute erases existing classes.

Manipulation

**"Get or Set the
contents of the page"**

remove: Remove an element from the page

```
<a>One</a>
```

```
<a>Two</a>
```

```

```

```

```

```
// Remove all links from the page.
```

```
$( 'a' ).remove();
```

```
// Remove the first image from the page.
```

```
$( 'img' ).eq(0).remove();
```

html: Set or Get the contents of an element

```
<div class="container">  
    Original  
</div>
```

```
// Get the contents of the container.  
$( '.container' ).html();
```

```
// Update the contents of the container.  
$( '.container' ).html( 'New Content' );
```

Exercises

1. Remove all blue shapes
2. Remove all shapes in the orange container
3. Remove all small red circles
4. Get the html contents of the reset button.
5. Try to get the html contents of all links. What happened?
6. Change the reset button to read "Launch Missiles!"
7. Change all the H1 tags to read "[Your Name] is Cool!"

Discussion

- Manipulations happens to all matched elements
- Reading happens only on first matched element

Traversal

**"Make more complex selections
by building on existing ones."**

children: Select the child elements of the current selection

```
<div id="box">  
  <a>a</a>  
  <span>b</span>  
  <button>c</button>  
</div>
```

```
// Select all the children of the box  
$( '#box' ).children();
```

parent: Select the parent elements of the current selection

```
<section>  
  <a id='link-a'>a</a>  
</section>  
<section>  
  <a id="link-b">b</a>  
</section>
```

```
// Select the section that contains link-a  
$( '#link-a' ).parent();
```


Exercises

1. Select all the shapes in the purple container using "children"
2. Select the green container using "parent"
3. Select all the "li" tags that contain a link.

Effects

**"Easy ways to update or
animate the page."**

hide: Hide elements without removing them

```
<div>  
  <span>A</span>  
  <a>B</a>  
    
</div>
```

```
// Hide all links and buttons  
$( 'a, button' ).hide();
```

show: **Show elements that are hidden**

```
<div>  
  <span>A</span>  
  <a>B</a>  
    
</div>
```

```
// Show links if they are hidden  
$( 'a' ).show();
```

Exercises

1. Hide the purple container
2. Show the purple container again
3. Hide all the links.
4. Show all the links again.

Events

"When X happens, do Y "

X is an event and Y is a function

on: Add event handler

```
$(selector).on(event, handlerFunction);
```

```
<span>Click Me</span>
```

```
// Log when the span is clicked
```

```
$( 'span' ).on( 'click', function() {  
    console.log( 'clicked' );  
});
```

Most common event types:
click + mouseenter + mouseleave

Exercises 1 / 2

1. When any shape is clicked, log "shape clicked" to the console
2. When your mouse enters any blue circle, log "Blue Circle: Go away!" to the console.
3. When your mouse leaves a blue circle, log "Blue Circle: Goodbye!" to the console.
4. When your mouse enters any "tr", set its class to "highlight".
5. When your mouse leaves any "tr", set its class to "".

Exercises 2/2

1. When 'button 1' is clicked, remove all shapes.
2. When 'button 2' is clicked, disable button 2. (Set the 'disabled' attribute to true).
3. When 'button 3' is clicked, set the button message to "Button 3 was clicked"

Which specific element was clicked?

*"**this** refers to the target object inside of handler functions."*

```
$( 'tr' ).on( 'mouseenter', function () {  
    $(this).attr( 'class', 'highlight' );  
});
```

Exercises 1 / 2

1. When any shape is clicked, log the value of its "class" attribute to the console.
2. When any shape is clicked, hide it.
3. When any shape is clicked, remove its container
4. When any container is clicked, remove all the shapes inside it.

Exercises 2/2

1. When your mouse enters any link, log the value of its "href" attribute to the console. "Your mouse entered a link to: [href]"
2. When any button is clicked, set the button message to "Button [button id] was clicked"

document ready event

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

- Just a regular event
- Triggered on the document when it is loaded.
- Used to make sure the document is loaded before running jQuery code.
- No quotes around document.

Demo

Make a team picker using jQuery

Homework

Jukebox 2

Tomorrow

jQuery 2