

jQuery

Making Life and Code easier for all

What's jQuery

jQuery is a JavaScript Library....okay great we're done...no

What's jQuery

jQuery is a JavaScript Library....okay great we're done...no

What's a JavaScript Library?

A bunch of pre-written javascript functions that do useful tasks. It helps us not re-invent the wheel.

So what are these useful tasks

There are a lot! jQuery makes it easier to do simple tasks such as events, styling, animations.

Let's check out some examples:

Examples:http://www.w3schools.com/jquery/jquery_examples.asp

What's that \$ sign about?

It's our jQuery entry point!

\$ is a function

Well, really it's just a variable with a weird name. The variable stores a function.

You can also use `jQuery`, but \$ is shorter and everyone uses it to work with jQuery.

What's that \$ sign about?

If the parameter passed to the \$ is another function, that function will get called when jQuery is "ready"

- "ready" means that everything is loaded on the page
- You place all of your JQuery code in that function

Example:

```
$(function () {  
  // put all your code here  
  alert("Hello, JQuery");  
});
```

Selecting Elements with \$

jQuery uses CSS selectors to retrieve references to elements in our HTML page similar to CSS.

TIME FOR AWESOME EXAMPLE!!! ARTHUR GO

Example: <http://codepen.io/Adamor/pen/azwgzQ>

Event Revisited with \$

Let's take a look at event with Plain JavaScript:

<http://codepen.io/Adamor/pen/myMyYG>

Now same event with jQuery:

<http://codepen.io/Adamor/pen/XJaJLN>

Okay, that makes sense but can I do more than click....yes!...next slide....

Event Revisited with \$

Events/Functions

.on

The .on() method attaches event handlers to the currently selected set of elements

.addClass/.removeClass

These methods add or remove class(es) to each element in the set of matched elements

.slideUp/.slideDown

.slideUp()/slideDown() methods animate the height of the matched elements.

.hide/.show

hide/show the matched elements.

.attr

This method gets the attribute value for only the first element in the matched set.

.children

This method allows us to search through the children of selected elements in the DOM tree

Code Along Lab!

Changing the Color Switcher ~ 15min

- Goto <http://codepen.io/Adamor/pen/PwmWeQ>
- We're going to add in jQuery using a CDN link
- Add classes to the body that we'll switch to change colors
- Change the event listeners in the color switcher to use jQuery instead.

Partner Lab

DOM Selector Practice Time: ~ min

- Goto <http://codepen.io/Adamor/pen/LVOKZw>
- Follow the instructions commented in the JS code.

ADVANCED JAVASCRIPT CONCEPTS

My poor brain!! There's more.???!!!!

Scope

Scope in JavaScript is best thought of as the space and time in which your variables live. They are "alive" as long as the function is running.

Each function has/create its own scope so they can have variables with the same name without conflicting.

Closures

Functions have access to the scope in which they were created.
This is called a closure! *jazz hands*
You'll use closure often as you develop more projects.

Quick Quiz

```
var A = "A";  
function fruits() {  
  var B = "B";  
  function banana() {  
    var C = "C";  
  }  
  function mango() {  
    var D = "D";  
  }  
}
```

This

First, know that all functions in JavaScript have properties, just as objects have properties. And when a function executes, it gets the `this` property—a variable with the value of the object that invokes the function where `this` is used.

- `this` is used inside a function (let's say function A) and it contains the value of the object that invokes function A.
- We need (`this`) to access methods and properties of the object that invokes function A, especially since we don't always know the name of the invoking object, and sometimes there is no name to use to refer to the invoking object.
- Indeed, `this` is really just a shortcut reference for the “antecedent object”—the invoking object.

Debugging

YOUR BEST FRIEND IS `console.log()`;

- Because it helps you get through some basic debugging issues
- Print out the value of your variables to the console
- Helps you understand if your variables are getting set properly/have the right values

YOU MAIN TOOLS WILL BE CHROME'S AND FIREFOX'S DEV TOOLS

- Chrome Developer Tools > Sources > Click on JS File
- Set breakpoints -- stops the execution of your program so you can view the value of your variables

Lab

ANOTHER LAB..SURE

This time, try to debug the JS file in the Debug folder in week6.
There may be a small problem with that HTML file as well..hehehe

Refactoring

WHAT IS REFACTORING?

Refactoring is a controlled technique for improving the design of an existing code base. Its essence is applying a series of small behavior preserving transformations, each of which "too small to be worth doing".

IT'S NOT ABOUT CODE CLEAN-UP

Refactoring defines one technique for improving the health of a code-base. Restructuring is a better term for reorganizing code that may incorporate other techniques.

Lab!

Refactor...this lab of fun!

Remember that Color scheme switcher example. You're going to refactor it to use the keyword "this". Check out the starter code for the refactor labs folder in week 6.

ADDITIONAL READING:

Scope: <http://www.smashingmagazine.com/2009/08/01/what-you-need-to-know-about-javascript-scope/>

Closures: <http://javascriptissexy.com/understand-javascript-closures-with-ease/>

This: <http://javascriptissexy.com/understand-javascripts-this-with-clarity-and-master-it/>

Homework Time!

The team from Relaxr is back...again!

They want their page to be more interactive and asked you to add JavaScript to their source code. You've been given the HTML and CSS but will need to create a JavaScript file and add a few different pieces of interactivity:

Users should be able to click a link and have the remainder of the blog's content slide down and appear on the page, both on the main content column and the side bar; users should also be able to hide the content when they are finished reading. See a detailed technical explanation below.

Homework Time!

Real-World Applications

- Practice programmatic thinking to plan before writing code

- Use basic JavaScript to create an interactive page based on a user action

- Link an external JavaScript file from HTML

Homework Time!

Technical Requirements Part 1

Prevent a form submission with the `event.preventDefault()` function

Use the `$.ready()` handler to delay your code from executing until all DOM assets have been loaded

Select the appropriate DOM elements with CSS selectors upon a user's click using the `$.on()` or `.click()` handler

Homework Time!

Technical Requirements Part 2

If a user clicks "Read More" on the primary column:

Have the text in the `<p>` tag slide down along with a "Read Less" link in the blog post using `$.slideDown()` and `$.show()`

Hide the "Read More" link using `$.hide()`

If a user clicks "Read Less" on the primary column:

Have the `<p>` slide up and hide the "Read Less" link using `$.slideUp()` and `$.hide()`

Show the "Read More" link using `$.show()`

Using the same functions as above, if a user clicks the "Learn More" link in the sidebar, have the `` inside that `<p>` slide down and hide the "Learn More" link using `$.slideDown()` and `$.hide()`