JQuery

AKA: WHY DIDN'T WE LEARN THIS EARLIER??!

Aside: CSS and hyphens

http://clagnut.com/blog/2395

Security…

Interesting read: Hyphenation and CSS

https://medium.freecodecamp.org/google-publishes-a-javascript-style-guide-here-are-some-key-lessons-1810b8ad050b

A Great Video about JavaScript (really!)

This video is about JavaScript's event loop, and gives some really great insight into what's going on "under the hood"

https://youtu.be/8aGhZQkoFbQ

Learning Objectives

Learn how to use JQuery to modify your css and html content

Learn how to install and access JQuery

Learn basic JQuery syntax, including:

- selectors
- filters
- handling events

What is JQuery

Free, open source JavaScript library

Works on all modern browsers

- and abstracts away some of the differences between browsers (nice!)

Gives you a more compact way to write your code

Provides solutions to common patterns

Uses css-like syntax

Works on sets of elements instead of individual items

Using statement chaining

Highly extensible

What is JQuery?

Download it from here:

- http://jquery.com/download/

Link to it in your code

- `<script src="../jquery-3.3.1.js"></script>`

In a script element make calls to JQuery:

- `$("document").ready(...);`

Basic Syntax

jQuery is designed for selecting elements and applying some action to them:

- $(selector).action()
- $ indicates that we're making a jQuery call
- (selector) is some kind of query that gets one or more HTML elements
- action() is then performed on the selected elements

Hello World in jQuery

Download and install jQuery

- Note: we're using the uncompressed development version
  - for deployment you'll want to switch to the compressed version

Write our first jQuery "program"

Take a look at how this would look if we just used JavaScript

Selectors and Filters

These let you extract and find information in web pages

Selectors

- css-like way to access elements in a page
- returns a *matched set*, or list of items
- we can then apply things to all or some of these items
- remember they return jQuery objects, not DOM objects

Filters

- lets you refine the results you get back from selectors
- can use css-like parameters
  - classes
  - ids
- can also filter by:
  - first match
  - negation

Selectors

Selectors give you a powerful way to get a list of all elements that match some quality

Suppose we want to:

- process all elements
- add event handlers to a bunch of elements
- get all images in a page and work with them

http://api.jquery.com/category/selectors/

Selectors

some types of selectors:

- $(“tagname”)
  - all elements with that tag name
- $(“#identifier”)
  - the one element with that id
- $(“.className”)
  - all elements with that class name
- $(“tag.className”)
  - all elements with both that tag name and class name
- $(“tag#id.className”)
- $(“*”)
  - all elements (wild card)

Filters

Filters are used in conjunction with Selectors to refine the set that is returned

Lets you do things like find the first, last, with specific index etc.

Filters

Some useful filters:

- :first, :last
  - first and last elements that matched
- :even, :odd
- :gt(), :lt(), :eq() items that are greater than, less than, equal to
- :animated
  - all that are currently animated
- :focus
  - the one that currently has focus
- :not(expr)
  - not the ones that match expr

Advanced Selectors

Use CSS-style syntax to further refine our result set

child selector: “div > p”

- give me the paragraphs that are immediate children of a div

ancestor descendant selector: “div p”

- select all paragraphs that descendants of div

adjacent selector: “ul + div”

- select all divs that immediately follow a ul

next adjacent selector: “#para1 ~ p”

- selects all paragraph siblings of para1

Advanced Filters

Attribute filters

- test for existence of attribute
- test if an attribute has a particular value
- test if an attribute value contains a value
- test if an attribute value starts with a value

Content filters

- let you filter results by looking at the content of elements

Attribute Filters

use square brackets

- “p[class]” all paragraphs that have a class attribute
- “p[id=para1]” the paragraph with the id value para1
- “p[id^=para]” all paragraphs with ids that start with para
- “p[lang*=en-]” all paragraphs with lang attributes that contain en-

Content Filters

:contains(text)
- checks if elements contain specific text

:parent
- checks if an element has at least one child

:has(selector)
- lets you specify what the element has to have

:first_child
- returns the first child

:last_of_type
- returns all elements that are the last of their type

nth_child()
- returns all elements that are the nth child
- for example 2n will select the 2, 4, 6.. (note in this case we don't zero index)

Creating and Modifying page content

Create an element

- call jQuery and pass it a tag
- use function calls to append/prepend, replace html content
- need to tell jQuery where to put your new content

Modifying content

- use the .html() function
- .text function will convert text to escaped text and insert it

Manipulating Page Content

Once you have an element or set of elements, you can:

- create, delete and move content
- adjust position and sizing

Creating Content

Lets try it

- first: we'll use an alert to show what is "in" an element
- next: create some content in different ways:
  - by passing text html element to the html function
    - this replaces the content of the matched set with what we pass in
  - by creating a jQuery object and passing that to html function
  - by passing text html to the text function
    - the text function only changes the text in the element, not the html

Inserting Content

jQuery provides rich ways to insert content

- can insert into elements
- can insert outside but relative to other elements

Functions:

- append
  - appends what you pass in to content of all returned selectors
- appendTo
  - same as append but order differs
- prepend
- prependTo
- insertBefore
- insertAfter

Altering page content

jQuery provides ways to modify existing content:

- wrap content around existing content
- empty contents of elements
- remove elements
  - remove(): destroys elements, all set attributes and event handlers
  - detach(): removes the element, but attributes and event handlers not destroyed
- replace elements
  - two ways: replace() and replaceWith()
  - order of syntax differs

Manipulating attributes

we can get the value of attributes

- attr(“name”) will return all elements with that attribute set

we can set the value of attributes

- attr(“name”, “value”) will set the “name” attribute of all elements returned to value

we can remove attributes

- using the removeAttr function

we can also set the attributes of multiple elements using JavaScript object notation

Handling events

event handling is critical to web dev

jQuery makes it easy to:

- attaching and detaching event handlers
  - .on function attaches an event handler
    - pass in event and callback function
  - .off function detaches the event handler

evt will contain the information about the event

jQuery event handling

jQuery makes it easier to support older browsers

makes it easier to work on groups of objects

provides a unified way to work with events

- abstracts away differences between browsers

provides shorthand functions for common patterns

Binding and Unbinding Events

The main functions for adding/removing events are:

- on()
- off()

let’s try a demo

Event helper features

There are a bunch of short hand formats for dealing with common events

- browser events
- form events
- mouse events
- key events

hover function:

- does the same thing we did before
- we pass in which functions we want called on mouse in and mouse out
- in this case it is the same function

click function:

- pass in the function we want called

dblclick function:

- pass in the function we want called

http://api.jquery.com/category/events/browser-events/

jQuery Unified Event Object

jQuery provides a unified way to handle events

ensuring that the following are consistent across browsers

- target (where event originated from)
- relatedTarget (related element)
- pageX, pageY (where mouse was)
- which (which key/button etc. was pressed)
- metaKey
  - boolean value
  - see if meta key was pressed (command on mac, windows on windows...)
- http://api.jquery.com/category/events/event-object/

Simple Animation

jQuery make it easy to animate content in your page

call animate function on a specific element

animate takes properties and timing information

- animate({width:400}, 300) change the width and change it to 400 over 300ms
- animate({top: “+100”, {border:…

Traversing Documents

Traversing the DOM the traditional way is unwieldy

jQuery provides a streamlined way to do this

Functions:

- find
- each
- prev/next/parent
- children
- parents
- parentsUntil

http://api.jquery.com/category/traversing/

Statement Chaining

One of the interesting features of jQuery

Lets you make multiple function calls on a set of element

syntax:

- $(selector).fn1().fn2().fn3();

helps keep code less verbose

prevents you from having to get a set of elements over and over

Summary

we’ve seen the basics of using jQuery

selectors and filters

simple animation

event handling

modifying and creating content