Web Front-End Development
Week 4: jQuery Intro

All web developers need a solid foundation in HTML, CSS, and JavaScript. Although some developers use "vanilla JavaScript", many use tools, libraries, and frameworks in their development. These terms are often used interchangeably, which can make figuring them out even more complicated.

JavaScript Tools
Tools aid in development and should make the development process easier, but they are not an integral part of your project. David Schaal covers many popular web dev tools in his Web Creative Development Tools course offered in the spring.

JavaScript Libraries
Libraries package together functions aimed at handling some specific functionality. There are libraries for animation, data visualization, typography, and a lot of other functionality.
https://www.javascripting.com/
Libraries normally provide a higher level of abstraction which smooths over implementation details and inconsistencies.
We're going to look at the jQuery library this week so we can understand what it means to use a library and how to approach development with a library.

JavaScript Frameworks
Frameworks are more involved than just bundling functions together in a library. There are many different types of frameworks but in general they provide more of an overall design pattern for an application. Many are also more "opinionated" meaning that you can't just pick and choose the functionality you need, you have to adopt the entire framework and adhere to their design pattern and methodologies.
We'll be looking at the React framework later in the semester.

jQuery
jQuery is a JavaScript library with functions to achieve many common JavaScript tasks, making JavaScript easier to write.
jQuery simplifies many common JavaScript tasks:

- Select elements
- Content manipulation
- Handle events
- Animations and effects
- Asynchronous server calls

Solves cross-browser compatibility issues
It's free
Very popular, large developer community
Plug-ins

So the whole purpose of using jQuery is that it provides a whole bunch of convenience functions and high-level operations without having to use JavaScript to manipulate the DOM.

Adding jQuery to your web site
http://jquery.com/ Download
jQuery is an external JavaScript file you link to

- Download and add it to your site
- Link to a version hosted at a CDN (content delivery network) https://code.jquery.com/
    - Provides the latest version
    - Might already be cached in the user's browser
    - Must be connected to the Internet

<script src="https://code.jquery.com/jquery-3.4.1.min.js" integrity="sha256-CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSFlBw8HfCJo=" crossorigin="anonymous">
</script>

- The `integrity` and `crossorigin` attributes are used for Subresource Integrity (SRI) checking. This allows browsers to ensure that resources hosted on third-party servers have not been tampered with. Use of SRI is recommended as a best-practice whenever libraries are loaded from a third-party source.

How jQuery works https://learn.jquery.com/about-jquery/how-jquery-works/

- JavaScript can't select a HTML tag to manipulate until the whole page is downloaded by the browser.
- Because jQuery manipulates a web page's content, it needs the whole web page to be loaded before jQuery functions can run.
- The **$(document).ready()** function is a built-in jQuery function that runs as soon as the document is ready to be manipulated.
    - The $ is short for jQuery() and it's used for all jQuery instructions
    - The "document" keyword indicates that we're about to perform an operation on the page itself.
    - The ready function sets up an event listener for when the DOM structure of the page is fully parsed by the browser, and is ready to be operated on.
    - The ready event allows us to operate on the page much earlier than the load event does(the load events waits for all images, including ads, to be loaded). A callback function is passed to the ready function, which is the event handler that will be fired when the ready event is triggered.
    - You only need the $(document).ready() function ONCE in your script
- Use jQuery to:
    - Select an element on the page with the jQuery() shorthand $
    - Filter to get the element(s) you want
    - Change/manipulate the HTML or CSS
    - Add new content
    - Add or remove an element or attribute
    - Extract information from an element

https://repl.it/@aileenjp/jQuery-intro

Example:

```
$("document").ready(function () {
   //put instructions here
 });
```

Selectors

Basic http://api.jquery.com/category/selectors/basic-css-selectors/

Selectors are used to select parts of the Web page using a common CSS-style syntax.

In JavaScript we used **document.getElementById()** to select one id and a few other methods to select elements on a page.

jQuery can easily select one or more elements with the jQuery object **$('selector')**

- You can use tags as the selector
  - $("p")
- For an ID you use a # (like in CSS)
  - $("#banner")
- For a class you use a .
  - $(".submenu")

The jQuery selector returns a jQuery object which have a large number of functions and properties you can use on them.

Filters

Basic Filter http://api.jquery.com/category/selectors/basic-filter-selectors/

Filters are used to further refine the results returned from selectors.

Selectors and filters work together to retrieve document content. Selectors are used to select content, as their name implies, and then filters are used to further refine the selected content.

You can think of selectors and filters as the query part of jQuery.

- $("p:first") will select the first paragraph in the returned set from $("p")

Content filters http://api.jquery.com/category/selectors/content-filter-selector/

Content filters let you filter the results of jQuery selectors by examining the content of the selectors themselves.

- $("p:contains('3')") this selector and filter combination selects p tags that contain 3
- $("div:has(p)") this selector and filter combination selects div tags that have at least one paragraph inside them

Child filters http://api.jquery.com/category/selectors/child-filter-selectors/

Child filters let you filter selectors based upon where an element is positioned within its parent.

Attribute http://api.jquery.com/category/selectors/attribute-selectors/

You can test to see if an attribute is present using [ ]

- $("p[class]") looks for p tags with a class attribute. Doesn't look at the value, just that the attribute is there.

You can test for a value by adding an equality test.

- $("p[id='para1']") tests for p tags with an id of para1

^= means starts with

*= means contains the text, does not need to start with it

Selectors and filters fill a pretty common task when you're working with web applications. They provide an easy way for finding and extracting information from web pages so that they can be acted upon.

jQuery functions let you manipulate the selected element(s).

CSS
http://api.jquery.com/category/css/
jQuery can also retrieve and manipulate CSS properties of elements.

- .css() lets you get or set CSS properties on the matched elements in a variety of ways
  - .css(propertyname) gets the value for propertyname
  - .css(propertyname, value) sets the value for propertyname
  - .css(propertyname, function) sets the value for propertyname to the result of the function
- .hasClass(className): determine whether a page element has a certain class
- .addClass(className): add the given CSS class to the elements in the matched set
- .removeClass(className): remove the given CSS class from the elements in the matched set
- .toggleClass(className): add or remove the given CSS class to the elements in the matched set depending on whether it is already there

Example:
How would we make the whole original trilogy yellow?
Select the id "original" and use the .css() function to change the color property.

```
$("document").ready(function () {
  $("#original").css("color", "#FFD700");
});
```

What if you want to make all the movies yellow? Remember what this looked like in JavaScript?

```
$("li").css("color", "#FFD700");
```

- $("li") returns all the li tags in jQuery called a matched set. No need to loop through them, jQuery works with sets of elements by default.

Advanced Selectors
jQuery lets you use more complicated selectors

- Descendant selectors target a tag in another tag
  - $('#banner a') all anchor tags in id banner
- Child selectors target a tag that's a child of another tag
  - $(div > p') selects all the paragraph tags that are immediate children of a div element
- Adjacent sibling selectors let you select a tag that appears directly after another
  - $('h2 + div') will select the div tags that are adjacent to a h2 tag

Example:
What if we wanted to select the movies in the original trilogy but not the div text "Original Trilogy"?

```
$("#original li").css("color", "#FFD700");
```

- selects the li elements in the div with id=original

Chaining
Chaining functions lets you user multiple jQuery functions on the same selector

- Select one or more elements
- Use dot notation to separate each function

Example:
What if we wanted to have the movies be both yellow and bold?

```
$("li").css("color", "#FFD700") .css("font-weight", "bold");
```

- calls multiple jQuery functions
- they don't need to be the same one

DOM Traversal
Traversing http://api.jquery.com/category/traversing/
Traversing the DOM can be verbose and require a lot of code.
JQuery provides a layer of API functions over the standard DOM that simplifies a lot of common operations.

- .children(): Retrieves all the child elements of the matched elements, except text nodes
  - You can deal with these as a group, no looping needed.
- .prev(): retrieves previous sibling
- .next(): retrieves next sibling
- .parent(): retrieves parent node

Example:
What if we wanted to have the movies and the year be yellow and bold but not Original Trilogy?

```
$("#original").children().css("color", "#FFD700");
```

- children(): Retrieves all the child elements of id "original" which is the span and ul nodes
- if the year wasn't in span tags it would be a text node and not selected

Content creation and manipulations
Manipulation http://api.jquery.com/category/manipulation/

Inserting content

- .html() will replace the content of the selector with new HTML content.
- .text() will replace the content of the selector with next text but doesn't take html. The html() function converts the string to HTML, the text() function doesn't.

- .append() and .appendTo() append content to the end of the selected object.
- .prepend() and .prependTo() append content to the beginning of the selected object.
- .before() and .insertBefore() insert the content before the selected elements
- .after() and .insertAfter() insert the content after the selected elements

Altering content

- .wrap() wraps the matched elements with the specified content
- .wrapAll() wraps content around the matched elements as a group
- .unWrap() removes the parents from the matched elements
- .empty() removes all the child elements from the matched elements
- .remove() removes elements from the page, including any embedded data and event handlers
- .detach() removes elements from the page, but maintains embedded data and event handlers
- .replaceAll() replaces the matched elements with the specified content

Manipulating attributes (General attributes)

- .attr() gets the value of an attribute
- .attr(name, value) sets the name attribute to value
- .removeAttr(name) removes the attribute from the element

Example:

How could you replace the Sequel trilogy years with 2015-??

```
$("#sequel span").text("2015-??");
```

- Select the span element in the section with the id sequel and then change its text

How could you add a new section after sequel for the Anthology Films?

```
$("#sequel").after("<section id='anthology'>Anthology
Films<ul></ul></section>");
```

- after the sequel id we add a new section tag with the id=”anthology”. Note there’s a ul tag in it too

How would you add a list item for Rogue One to the new Anthology section?

```
$("#anthology ul").html("<li>Rogue One</li>");
```

- $("#anthology ul") selects the ul tag in the new anthology div and replaces the html in it(there was none)
- Use .html() when the content has html and .text() when the content is text

jQuery vs. DOM manipulation

jQuery doesn’t do anything Javascript can’t do but it’s simpler and handles cross-browser support

Used on over a quarter of web sites

- Selecting elements is simpler and faster
- Event handling is simpler
- Methods affect all selected elements, no looping needed
- Methods provided for popular tasks, no need to write your own
- Cross-browser, no need for fallback code
- Need to include the jQuery library, additional size, so only use if needed

Lab: Create or modify a web page to select, filter, and manipulate the page in at least three different ways using jQuery.