**jQuery Notes**

1. **What is jQuery?**
   1. Compare JavaScript to legos, with an infinite amount of time you could build an entire city
   2. jQuery is like pre-made lego buildings, you can build the city in much less time
   3. Each pre-made structure is equivalent to a jQuery method
   4. Methods add dynamic behaviors to HTML elements
2. **The jQuery Function**
   1. Takes a parameter that targets an HTML element, and turns it into an object
   2. Now you can call any jQuery method on said jQuery object (the old HTML element)
      1. Developers often save jQuery objects in variables
      2. Ex: const $jQueryObject = $(‘.someClass’);
      3. Because the jQueryObject variable contains a jQuery object, we decided to start the name with $ to remind the reader it is a jQuery object
      4. Now we can call methods, ex: $jQueryObject.hide();
3. **Key Words**
   1. $ - an alias for the jQuery function, they are interchangeable
   2. ‘document’ - targets the HTML document (all of it)
4. **Targeting**
   1. By Class: $(‘.someClass’).methodToCall();
   2. By ID: $(‘#someID’).methodToCall();
5. **jQuery Event Listeners** - https://developer.mozilla.org/en-US/docs/Web/Events
   1. Examples:

****

1. **jQuery Event Handlers**
   1. Event Handlers are composed of two parts
      1. Event Listener - a method that listens for a specified event to occur, like a click event
      2. Callback Function - a function that executes when something triggers the event listener



* + 1. In the above example, on(‘click’ is the Event Listener, and ()=>{} is the Callback Function
  1. .on() - adds event handlers to jQuery objects
     1. Takes two parameters
        1. The Handler - a string declaring the event to listen for
        2. A Callback Function - to fire when the event is detected



* + 1. When a user clicks the HTML element with ID ‘login’, the loginForm appears

Note: .on() is used with jQuery objects - i.e. called with $()

Note: .addEventListener is used on ‘vanilla’ JavaScript (cannot be used on jQuery objects)

* 1. Mouse Events
     1. Mouse Events can be chained together to do cool things, example:



* + 1. Use the ‘currentTarget’ attribute to single out a specific element without creating unique id properties; example:



* + 1. ‘click’ - triggers when clicked
    2. ‘mouseenter’ - triggers when the mouse enters the elements space
    3. ‘mouseleave’ - triggers when the users mouse leaves the elements space

1. **jQuery Effects** - Methods that add dynamic behavior to websites, ex: sliding menus

**------------NOTE:** You may need to pass an argument ‘event’ into your jQuery functions and set the first line of the function to ‘event.preventDefault();’ in order to stop the HTML page from refreshing and making the effect invalidated**------------**



* 1. Documentation: http://api.jquery.com/category/effects/
  2. Example Syntax:



* 1. .fadeIn() - appear over a given period of time
     1. Has two optional parameters, duration, complete
        1. Duration: time in MS or ‘fast’, ‘slow’
        2. Complete: a function to call when animation is complete
  2. .fadeOut() - disappear over a given period of time
     1. Has two optional parameters, duration, complete
        1. Duration: time in MS or ‘fast’, ‘slow’
        2. Complete: a function to call when animation is complete
  3. .fadeToggle() - a combination of .fadeIn() and .fadeOut()
     1. Has three optional parameters, duration, easing, complete
        1. Duration: time in MS or ‘fast’, ‘slow’
        2. Easing: transition style, ‘swing’ or ‘linear’
        3. Complete: a function to call when animation is complete
  4. .hide() - hides the jQuery object as if it does not exist
     1. Has two optional parameters, duration, complete
        1. Duration: time in MS or ‘fast’, ‘slow’
        2. Complete: a function to call when animation is complete
  5. .show() - shows a jQuery object that is hidden
     1. Has two optional parameters, duration, complete
        1. Duration: time in MS or ‘fast’, ‘slow’
        2. Complete: a function to call when animation is complete
  6. .slideDown() -
     1. Has two optional parameters, duration, complete
        1. Duration: time in MS or ‘fast’, ‘slow’
        2. Complete: a function to call when animation is complete
  7. .slideUp() -
     1. Has two optional parameters, duration, complete
        1. Duration: time in MS or ‘fast’, ‘slow’
        2. Complete: a function to call when animation is complete
  8. .slideToggle() -
     1. Has two optional parameters, duration, complete
        1. Duration: time in MS or ‘fast’, ‘slow’
        2. Complete: a function to call when animation is complete
  9. .toggle() - a combination of .show() and .hide(), triggers based on current state
     1. Has two optional parameters, duration, complete
        1. Duration: time in MS or ‘fast’, ‘slow’
        2. Complete: a function to call when animation is complete

1. **Other jQuery Methods -** 
   1. .addClass(‘className’) - adds a specified class to the target element
   2. .after()
   3. .append()
   4. .animate(prop, dur, eas, compl) - animates an HTML element to custom properties
      1. Has four arguments, first is mandatory, rest are optional
         1. Properties: an object with CSS key/value pairs of the desired end state
         2. Duration: time in MS or ‘fast’, ‘slow’
         3. Easing: transition style, ‘swing’ or ‘linear’
         4. Complete: a function to call when animation is complete
   5. .before()
   6. .children() - targets all children of an element, but not grand-children
   7. .css(‘key’,’value’) - allows editing of CSS key/value pairs; example:



* + 1. Alternative syntax that allows editing multiple CSS key/value pairs:



* 1. .parent() - targets the parent of an element
  2. .prepend()
  3. .querySelector(‘input’) - replace ‘input’ with tags to grab, like .class, #ID, etc.
     1. Finds the element(s?) with the specified tags
     2. Useful to do things like: let boxToEdit = document.querySelector(‘.boxClass’)
  4. .ready() - waits until the HTML page’s DOM is ready to manipulate



* + 1. Wrap all JavaScript behavior inside the .ready() method to make sure the web page is rendered in the browser before any JavaScript code executes
  1. .removeClass(‘className’) - removes a specified class to the target element
  2. .siblings() - targets all siblings of an element

1. **jQuery Library**
   1. Basics
      1. To use jQuery in a site, the index.html file must load it with other dependencies
      2. jQuery tags should be loaded at the end of the index.html doc to prevent long load times of the HTML and CSS
   2. Content Delivery Network (CDN)
      1. A collection of servers that deliver content (in this case, the jQuery library)
   3. Loading into HTML



* + 1. ‘src’ indicates the path to a CDN
       1. Note, the link indicates version ‘3.2.1’ and ‘min’ for ‘minified’
       2. Alternatively, other versions include: uncompressed, slim, and slim minified
       3. There are dozens of CDN’s to link the jQuery library, above is one example
    2. ‘integrity’ and ‘crossorigin’ ensure the file is delivered without any third-party manipulation
    3. The jQuery library needs to be loaded before the main.js file
       1. Otherwise, the computer will not understand the jQuery used in main.js
       2. ‘main.js’ should have the ‘defer’ attribute in the HTML element

EVENT.TARGET.VALUE