Coding & Cocktails Session 9: Intro to JavaScript using jQuery

Overview

Tonight we discussed how to make our websites respond to user interaction using jQuery. For our project we're going to return to our Drink Order App that we used in March for GitHub learning and practice. This time we'll work on some basic interaction and progress to adding in a content slider, a common element on webpages.
If there is a term you don't know check out our glossary here: bit.ly/CnCgloss

Prep Work

1. Utilize our Tools worksheet (bit.ly/cnctools) to install our recommended tools prior to the session.

Project

Our first part is going to involve some common page manipulation including changing colors, hiding & showing elements and more.
Note: If you need a refresher on Git version control refer to the March worksheet here: bit.ly/CnCvers. A refresher on the Command line can be found here: bit.ly/cmdln

Part 1: Starting with common interactions

1. Fork the DrinkOrderApp repository (bit.ly/CnCDrinkApp) to your own account.

2. Clone the repository to your computer using Git Bash (Windows) or iTerm2 (Macs) and **checkout the `jquery-master` branch** to have a clean starting point for tonight's project.

3. Open the index.html file in Google Chrome and try clicking the buttons. Not very exciting yet, is it?

4. Open the DrinkOrderApp folder in Sublime Text. (Check out our #codingandcocktails Slack channel at kcwit.slack.com and search for `"Coding & Cocktails tip of the week"` (quotes included) for a reminder on how to open the whole project folder for easy access to all of your project files as you work.

5. First we need to connect our HTML and our JavaScript. This is similar to connecting our CSS files with `<link>` tags as we discussed in the HTML & CSS session as well as in the CSS Compiler session but with a different HTML tag.
    a. Add a `<script>` tag for the jQuery library (*jquery-1.12.3.min.js* found in the *assets/lib* folder) in the `<head>` section of the *index.html* page.
    `<script src="assets/lib/jquery-1.12.3.min.js"></script>`
    b. Add a `<script>` tag with a src attribute for your script file (*my -scripts.js* found in the *assets/scripts* folder) in the `<head>` section of the *index.html* page - Remember, order matters!
    Hint: bit.ly/scriptElement or bit.ly/StartjQuery

6. Open *my-scripts.js* in Sublime Text from the *assets/scripts* folder. This is where you'll be adding your jQuery code.

7. First we need to make sure we wait until our HTML document is ready before we run anything on it. Remember we do this using `$(document).ready();` **Hint:** bit.ly/docReady Don't forget that we need a function in there so we can actually run our code!

8. We'll want to be able to see our menu so we need to make the "Show Menu" button function properly. When we click the button we want to display the `#menu` div.
    a. First we have to capture the event inside your `$(document).ready(function () { ... });` code in *my-scripts.js*
        i. Select your button (check out *index.html* to find the button's id to select)
        ii. Add the click event handler to the button. **Hint:** bit.ly/CnCClick
        `Hint: $(“#my-id”).click();`
    b. Next we have to add our action:
        i. Inside of the click event handler we need to add the function to run the action we're going to do. This is similar to how we created the function to run when the document is ready.
        `Hint: $(“#my-id”).click(function () { … });`
        ii. Inside of that function, select the element you want to act on, in this case the `#menu` div.
        iii. Add a jQuery action to show that HTML div. **Hint:** bit.ly/CnCShow
        `Hint: $(“#my-id”).show();`

9. Refresh the open index.html in Google Chrome and try clicking the Show button. It should display a menu when you click on it.

10. Now that we've displayed the menu we want to be able to hide it when we don't want to see it. Hide the `#menu` div element when the hide menu button is clicked in *my-scripts.js*. This will be very similar to what we just did with the show menu button but a different action on the div.
    a. First we have to capture the event:
        i. Select your button (check out index.html to find the button's id to select)
        ii. Add the click event handler to the button. **Hint:** bit.ly/CnCClick
    b. Next we have to add our action:
        i. Inside of the click event handler we need to add the function to run the action we're going to do. This is similar to how we created the function to run when the document is ready.
        ii. Inside of that function, select the element you want to act on, in this case the #menu div. **Hint:** bit.ly/CnCSelect
        iii. Add a jQuery action to show that HTML div. **Hint:** bit.ly/CnCHide

You've added your first interactivity to your website! Celebrate with a toast with your neighbor!

11. Next we're going to practice another common interaction on web pages: changing element styling triggered by an event. When the checkbox next to a drink is selected, make the background of the drink label change colors.
    a. First we have to capture the event:
        i. Select your check box. Look at the HTML to see what this structure like. We'll need to utilize CSS attribute selectors this time. These look like **cssElement[“attribute=value”]** **Hint:** bit.ly/AttrSel Don't know what an attribute is? Check this out: http://bit.ly/Attribs
        ii. Add the click event handler like we did in step 8.a.ii. **Hint:** bit.ly/CnCClick

b. Next we have to add our action:

i. Inside of the click event handler we need to add the function to run the action we're going to do. This is similar to how we created the function to run when the document is ready.

ii. Inside of that function, this time we're going to need the input's parent element: the label. **Hint:** bit.ly/CnCparent

iii. We only want the color to show when the box is checked. There are a few ways to do this and often changing styling you'd use the .css() method: bit.ly/CnCCSS. This time since we want to add and remove it we're just going to toggle the CSS styling class using the jQuery method **toggleClass() Hint:** bit.ly/ToggleC We've already set up the highlight class for you in the *assets/styles/main.css* file.

12. **EXTRA CREDIT:** Make clicking the "Order" button get the value from the name radio button input and display it in an **h1** tag inside the light purple div as "{{name}}'s order is ready!"

a. First capture the event on the Order button:

i. Select the Order button like we've selected the show and hide buttons above. (check out *index.html* to find the button's id to select)

ii. Add the click event handler to the button **Hint:** bit.ly/CnCClick

b. Next we have to add our action but this time we're going to have to save the name value so we can use it again later.

i. Create your **orderName** variable. Remember when you declare a variable you must start with the **var** keyword!

ii. We need to capture the radio button input's value in this variable so let's get it selected! *Note: we only need to store the value for the input element that is "checked."* **Hint** 1: bit.ly/CnCRadio **Hint** 2: bit.ly/CnCChecked **Hint** 3: bit.ly/CnCVal **Hint 4: var myVar = $("input:input-type:checked").method();**

iii. Once you've stored the value in your variable you can check the value in Chrome's Developer Tools Console tab using **console.log(orderName);** in your script file. When you trigger that event the variable's value will then be displayed to the console. **Hint**: bit.ly/CnCOutput

iv. Now that we have captured the value, we need to display that in our div section. There are again multiple ways to do this: the **append()** method: bit.ly/CnCAppend and the **html()** method: bit.ly/CnCHTML, though one of these is a better choice than the other. Can you figure out why one of these is a better option? Talk it over with a mentor if you aren't sure! **Hint:** Try both options and push the order button multiple times for each to see what happens!

1. First select the div section you want to act on. (find the id of the element in the *index.html* file and select it)
2. Next we'll use append or html to add the h1 tag and content inside the div. See the "JavaScript String Operators" section of the JavaScript Operators page: bit.ly/CnCOps for a reminder on how to use your variable with other static text.

13. When you've worked through each piece of interaction, compare your code to our answer key here: bit.ly/CnCJQKey

Take a break and grab another drink! You've earned it!

Part 2: Content Slider

Next we're going to add in a content slider on our page. This is very useful to scroll through pictures, text, videos or any other html items on a website. We're also going to utilize a library for this called bxSlider.

1. Checkout the add-slider branch and this will be our starting point for the content slider piece. We've already included the bxSlider library in our project for you as well as the necessary HTML markup. You're going to add the interactivity!

2. First we need to connect the bxSlider script file and CSS styling to our *index.html* file so everything will work! You will find the JavaScript file and the CSS file in the *assets/lib/jquery.bxslider* folder. Don't forget one of these utilizes a script tag and the other uses a link tag! **Hint:** see step 1 of the "How to Install" section on bxslider.com

3. Look at the index.html section and see the unordered list element (ul element) with a bxslider class to see the markup for this slider.

4. Call the bxSlider on your content. **Hint**: See step 3 of the "How to Install" section on bxslider.com

5. Try making some modifications to how your slider works:
    a. Change the mode of the slider to 'vertical'. **Hint:** bit.ly/bxSExample:
    b. Add captions to your images. **Hint:** bit.ly/bxSOpts

6. Check your slider against the answer key here: bit.ly/jQFinal

Congratulations! You created an interactive website!

Homework

The more you practice, the more comfortable you'll feel! Work through the following tutorial on your own time at home to help cement the concepts in your brain. Don't forget you can always talk to us on kcwit.slack.com to get help if you feel stuck or confused.

1. Codecademy learn jQuery: bit.ly/CnCjQHmwk