

CS 2033

Multimedia and Communications

Lab 08: jQuery and Bootstrap

- Website Development -

REMEMBER TO BRING YOUR MEMORY STICK TO EVERY LAB!

EXERCISE 1: JQUERY

jQuery is a free JavaScript-based library that provides shortcuts through its minimalistic syntax and built-in functions. jQuery doesn't necessarily add anything brand new that couldn't be done in "vanilla JS" (original JS) but it is used to simplify code and save time and unnecessary file size. A nice feature of jQuery is that it can be easily combined with vanilla JS and both can still work together even in the same functions!

What you will learn in this exercise:

- How to load a hosted version of jQuery into your website
- Adding event listeners with jQuery
- Changing CSS styles with jQuery
- How to easily implement a pre-built jQuery function for smooth scrolling

Setup

1. Navigate to your USB folder (F:) and into the **cs2033** subfolder (should have been created in a previous lab).
2. Create a folder called **lab08** inside **cs2033**. Within **lab08** create two subfolders, one called **img** and one called **css**.
3. Open <http://www.csd.uwo.ca/~bsarlo/cs2033b/labs/Lab08/img/> download all the images, and move them into **lab08** > **img**.
4. Open <http://www.csd.uwo.ca/~bsarlo/cs2033b/labs/Lab08/css/>, download **styles-1.css** and **styles-2.css**, and move them into **lab08** > **css**.
5. Open <http://www.csd.uwo.ca/~bsarlo/cs2033b/labs/Lab08/>, download **exer1.html**, and move it into **lab08**.
6. Open **exer1.html** in Brackets and immediately save it in the same folder as **exer1_complete.html**. All work in this exercise should be done in **exer1_complete.html** unless otherwise specified.
7. Before we can use jQuery, we need to load it using the `<script>` tag, just like how we load our own JS files to a website. There are 2 ways to do this: (1) download the jQuery files to your computer/USB and load it as if it was any other local JS file, or (2) embed it directly from a hosted (online) version of the file. We'll just do the 2nd option here because it's easier and quicker, but either way would work (assuming the versions are the same).
8. Click in the `<head>` section of **exer1_complete.html**. To embed the online version of jQuery version 3.3.1, add the following line of code:
`<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>`

Event listeners and CSS changes

9. You know the different methods of adding event listeners in JS, but now you will see how to do it with jQuery. It's not too different but it's important to know the syntax of each.
10. Scroll down to the bottom of **exer1_complete.html** and add a `<script>` tag just before the closing body tag.

We'll be adding several event listeners in this script area so hit Enter to add empty lines.

```
<script type="text/javascript">

</script>

</body>
</html>
```

11. Within this script area we'll access the element with the ID "spec-b" from the jQuery shorthand function: \$("#spec-b"). Note that you can also get elements by class name using the same shorthand except a period would be used instead of a # (just like in CSS).
12. Add a mouseover event listener to the spec-b element. The jQuery for this is shown below:

```
<script type="text/javascript">

    $("#spec-b").mouseover(function() {

    });

</script>
```

13. As you can see, there are some similarities to pure JS event listeners but the syntax is a little different. The listener creates a function directly inside it that will be triggered when the event occurs. We will add code inside this function to change CSS.

Before, we add the code to change the CSS in here, I need to explain how it is set up and what we want to achieve. For each of the 4 "specials" sections, there is a div with the class "specialcontainer". Each has a unique ID that begins with "spec-" and is suffixed with a letter relevant to the meal type (b for breakfast, l for lunch, s for dinner/supper, and d for dessert). Within each of these divs is a large image of the special and an inner div labelled with the "-cover" class and ID. The inner (cover) div is a translucent panel that overlays on the image and contains text (in the form of h1, h2, and p) that will be displayed on top of the large image when you hover on it and it not shown otherwise. We will want this overlay to slide in so it's been set up with the cover panel to the right of the image (left:100%). In the jQuery, we will set left to 0 so that it shows up directly on top of the image.

14. Now that you know how it is set up, the next step is to set the left style to 0 within the mouseover function. Again there is a shorthand way to do this in jQuery. Add the following line into the mouseover listener created above: \$(' #spec-b-cover').css("left", "0");
15. Your jQuery code should look like this now:

```

<script type="text/javascript">

    $("#spec-b").mouseover(function() {
        $("#spec-b-cover").css( "left", "0" );
    });

</script>

```

16. Save the file and refresh the browser. Try hovering over the breakfast parfait image to see what happens. You should see a yellow translucent panel show up on top of the image. However, it doesn't slide in nicely and when you take your mouse off the image, the yellow panel doesn't leave. We have a few steps left.
17. Copy the 3 lines for the mouseover function for #spec-b and paste them just below it.
18. This second one will be used for a mouseout listener, so change "mouseover" to "mouseout"
19. In the CSS line, we now want the left style to be 100% again (just like the default value as set in the actual CSS) so change the 2nd parameter from "0" to "100%" (NOTE: include the % here since there are various units. It's not necessary for the 0 since 0 of any unit is still 0!). The entire jQuery should now look like this:

```

<script type="text/javascript">

    $("#spec-b").mouseover(function() {
        $("#spec-b-cover").css( "left", "0" );
    });
    $("#spec-b").mouseout(function() {
        $("#spec-b-cover").css( "left", "100%" );
    });

</script>

```

20. Save the file and refresh the browser. Hover again over the breakfast parfait image and then un-hover. Now the yellow panel should toggle between visible and invisible.
21. The other 3 "special" images do nothing when you hover on them. This is because we only added these mouseover and mouseout listeners on #spec-b which is the breakfast special panel. We need to add them on #spec-l, #spec-s, and #spec-d as well.
22. Copy all 6 lines containing both the mouseover and mouseout functions and paste them below 3 more times so you will have 8 functions in a row (4 sets of 2 listeners).
23. Modify the pasted functions with the other 3 special panel IDs so that you will have the mouseover and mouseout functions for each of the 4 meal categories. You also have to adjust the ID in the CSS lines so that the corresponding cover panel is identified. The code should look like this now (order doesn't matter but it's easier to keep track if they are ordered and grouped together):

```

<script type="text/javascript">

    $("#spec-b").mouseover(function() {
        $("#spec-b-cover").css( "left", "0" );
    });
    $("#spec-b").mouseout(function() {
        $("#spec-b-cover").css( "left", "100%" );
    });

    $("#spec-l").mouseover(function() {
        $("#spec-l-cover").css( "left", "0" );
    });
    $("#spec-l").mouseout(function() {
        $("#spec-l-cover").css( "left", "100%" );
    });

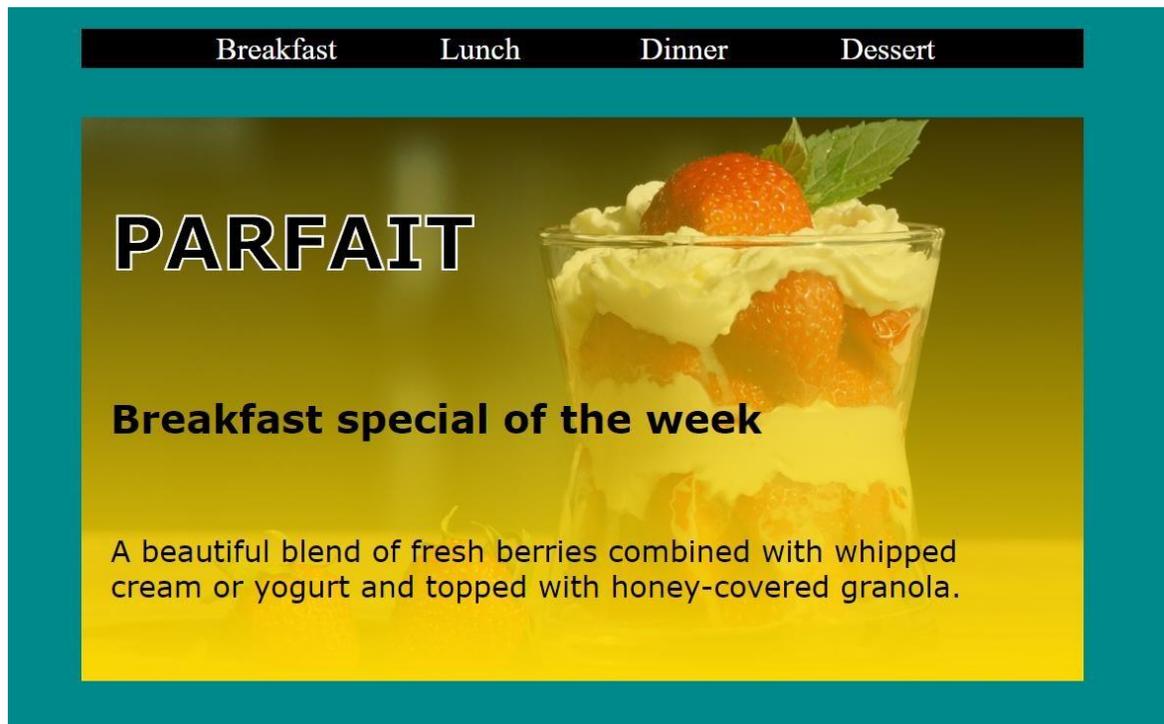
    $("#spec-s").mouseover(function() {
        $("#spec-s-cover").css( "left", "0" );
    });
    $("#spec-s").mouseout(function() {
        $("#spec-s-cover").css( "left", "100%" );
    });

    $("#spec-d").mouseover(function() {
        $("#spec-d-cover").css( "left", "0" );
    });
    $("#spec-d").mouseout(function() {
        $("#spec-d-cover").css( "left", "100%" );
    });

</script>

```

24. Save the file and refresh the browser. Now you should be able to hover over all 4 category special images and see the corresponding yellow panel show up and then leave based on your mouse entering and leaving the image.
25. The last step is to add a transition so that the panel slides in and out rather than just appearing and disappearing immediately. Think back to the CSS we learned about several weeks ago. Remember the transitions we used to make rollovers and other style changes look animated? We can apply these also to styles that change from JS or jQuery.
26. Go into css > styles-1.css and look for the selector for the "special-cover" class.
27. Somewhere within that rule-set, add the following CSS style:
transition: left .25s;
28. Save this CSS file and refresh the browser. Hover over each of the 4 special images and then un-hover. The yellow panel should now slide in from the right to the left, and then leave by sliding back out to the right. Pretty nifty!



Smooth scroll

29. Before beginning this portion, look at the site in Chrome. Click each of the 4 tabs in the main navigation bar. Because they are all bookmark anchors, it should bring you to the corresponding section as you click the links. That's good but it's a little boring! Let's spice it up with some jQuery.
30. Open <http://www.csd.uwo.ca/~bsarlo/cs2033b/labs/Lab08>, download **smoothScroll.js**, and move it into **lab08**.
31. Open smoothScroll.js in Brackets and have a look at the code. You don't need to understand exactly what it's doing, but just review it briefly.
32. Go back into exer1_complete.html in Brackets and scroll to the bottom near where you added the jQuery.
33. Load the smoothScroll JS file near the bottom of this file:

```
<script type="text/javascript" src="smoothScroll.js"></script>
```

NOTE: I did not create this smoothScroll.js function. It was obtained online from <https://csstricks.com/smooth-scrolling-accessibility/>.

34. The smoothScroll function will automatically apply a transition/animation to any scrolling created from clicking a bookmark anchor on the page. Just by embedding it, this should already take effect!
35. Save these files and refresh the browser. Click each of the 4 tabs in the main navigation bar again. You should see a smooth transition to bring you to the corresponding section.

EXERCISE 2: BOOTSTRAP

What you will learn in this exercise:

- Loading the Bootstrap library
- Adding Bootstrap components to the site
- Seeing what else is doable through online examples

Pre-exercise

Before beginning this exercise, familiarize yourself with some of the common Bootstrap components. Open <https://getbootstrap.com/docs/4.3/components/alerts/> in Chrome and look through 12-15 of the different components listed along the side. You don't have to understand all the code but spend a minute or so on each component to see how it looks and the different variations. They include a lot of examples so check these out! Then spend a few minutes perusing this documentation on the Bootstrap grid layout: <https://getbootstrap.com/docs/4.0/layout/grid/>.

1. Now that you have looked over several components, you can begin this exercise in which you will create a webpage using some of these components.
2. Navigate to <http://www.csd.uwo.ca/~bsarlo/cs2033b/labs/Lab08/exer2>. You don't have to download any of these files but keep this tab open as you will need to go into each of these .txt files to copy code shortly. This page will be referred to as the "resources" page throughout this exercise.
3. Open Brackets and create a new file. Save it immediately into your **lab08** folder with the name `exer2_complete.html`.
4. Start by adding the basic shell of a webpage (refer to Lab 4 if you need a refresher).
5. Within the head section, add the `<title>` tag to set the page title to "My First Bootstrap Website".
6. In the head section, add a `<link>` tag to reference the CSS file `styles-2.css`:
`<link rel="stylesheet" href="css/styles-2.css">`

```
<!DOCTYPE html>
<html>

<head>

  <title>My First Bootstrap Website</title>
  <link rel="stylesheet" href="css/styles-2.css">

</head>

<body>

</body>
</html>
```

7. To use Bootstrap, we need to embed links to its CSS and JS files that are hosted online. Another option would be to download these files and reference them as local files, just like the options we have for jQuery. Again we'll use the hosted version for this.
8. Start by adding 3 lines into the head section to help set up the page and load the Bootstrap CSS file. Go back to the resources page you opened in step 2 in Chrome and open the file titled references.txt. Copy (Ctrl+C) the top 3 lines from in the file and then go back to exer2_complete.html and paste (Ctrl+V) them into the head section.
9. Click near the bottom of the HTML, just before the closing body tag. Then go back to Chrome to the references.txt page, copy the bottom two long lines provided in that file, and paste them into exer2_complete.html where you moved your cursor, just before </body>.
10. Now the Bootstrap library should be ready to use because the library is now loaded to your website. We'll begin adding Bootstrap components shortly. The code should look like this:

```

<!DOCTYPE html>
<html>

<head>

  <title>My First Bootstrap Website</title>
  <link rel="stylesheet" href="css/styles-2.css">

  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.
css">

</head>

<body>

  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-
q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
  <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js
" integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>

</body>
</html>

```

11. First create an anchor with the name "top" right after the opening <body> tag:
12. Next you will create the container div (similar to a wrapper) which will contain all the contents of the page. Add this immediately after the anchor. Then click between the opening and closing container div tags and hit Enter many times to create a lot of space since we will be adding code for each of the components in here. If it helps, add a comment (using <!-- and -->) after the closing div tag to help you remember what this tag is for since it may get confusing as we add more code.

```
<body>

    <a name="top"></a>
    <div class="container">

        </div> <!-- close container div -->

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
    integrity="sha384-
```

Navbar (navigation bar)

13. Go back to the resources page and you should see various other files listed there (along with the references.txt one that you used above). I have taken the code structure you need for each of the Bootstrap components we'll be using and put them into these .txt files for you because some of these components require a lot of nested divs and other code so it will be much easier to copy and paste it in.
14. Open the navbar file and select all the code. Copy it and paste it immediately below the container div code.
15. After adding the navbar, create a div with the class "space". This is not part of Bootstrap but just a class I created in styles-2.css. It simply adds vertical margin space so that the next component we add won't be cut off from the navbar. It will start beneath the navbar so that nothing is hidden. Close it immediately. This div can also be added after the other modules as you progress through this exercise. Any time two modules look too close together, add the space div to add vertical space between them. The div is made with:
`<div class="space"></div>`

Image carousel

16. Go back to the resources page in Chrome and open the carousel page. Select all the code in there and copy it and then paste it into exer2_complete.html after the space div.
17. This code involves 3 main chunks that, together, create the overall image slider (carousel) which is a very commonly used component on most modern websites. The first chunk is the ordered list (ol) tag and the list items (li) within it. This creates the lines at the bottom that indicate which image is currently displayed and they are also links to change the current

image. The next chunk is the "carousel-inner" div and this is where the images are actually added. The last portion of the code is the left and right arrow links to change to the previous or next image in the carousel.

18. Add a 4th image to the carousel by copying the 3 lines that add in the pancakes image (starting with `<div class="carousel-item">`) and paste it immediately below so that the four carousel-items are lined up in a row. For the new one, change the image to `img/burrito.jpg` and set the alt text to "Fourth slide".
19. You also need to add a new line indicator in the first chunk of this code area (in the ol part). Duplicate the line that creates the #2 indicator and change the "data-slide-to" value to 3.
20. Save the file and look at it in Chrome to ensure the carousel is working properly.

Grid blocks

21. The next section of our site will be using the Bootstrap grid blocks. This is similar to the blocks I made for you to use in your website assignment, except Bootstrap made a lot more different sized ones and they aren't as tight on space as mine were!
22. Under the carousel section, add a large comment that says GRID BLOCKS using the format that is being used in the provided Bootstrap sections. This helps us to keep the site organized because there is a lot of code that can get easily messy. The comments remind us which part is where and acts a separator between the sections.
23. Create an anchor/bookmark with the name "reviews".
24. Now create a div with the class "row". This row div can contain several divs that are shown side by side (inline-block).
25. Within the row div, create another div and give it the class "col-sm". This is a small block and you can fit about 8 of these across a row on a big computer screen, but they get a little squished as you add more.
26. Inside this inner div, add a paragraph (p) tag and give it the class "quote". Add a short review quote that someone may have given to this restaurant. Then add another paragraph tag also with the "quote" class and write someone's name or initials to indicate who wrote the review (make up a quote and name).
27. Copy and paste this inner "col-sm" div and make 3 more copies of it below (still inside the row div). Make up new quotes and names for each one, or use my samples. This section should look something like this:

```

<!--
GRID BLOCKS
-->
<a name="reviews"></a>
<div class="row">
  <div class="col-sm">
    <p class="quote">"Best burger I've ever had!"</p>
    <p class="quote">~ Megan B.</p>
  </div>
  <div class="col-sm">
    <p class="quote">"Great food and great service!"</p>
    <p class="quote">~ Tony L.</p>
  </div>
  <div class="col-sm">
    <p class="quote">"This is my favourite restaurant in town! I eat here at least twice a month"</p>
    <p class="quote">~ Nora G.</p>
  </div>
  <div class="col-sm">
    <p class="quote">"I absolutely love this restaurant."</p>
    <p class="quote">~ Jake W.</p>
  </div>
</div>

```

Collapsible/expandable panel (accordion)

28. Go back to the resources page in Chrome and open the accordion page. Copy all the code from within it and then paste it into the HTML file just underneath the grid blocks section.
29. This accordion structure allows you to click each of the panels to toggle its open/closed status. Save the file and refresh the site in Chrome. Click each of the tabs to see how it opens and the others close so at most one panel can be open at once.

Large image with overlaid text

30. The next portion isn't a Bootstrap feature but just a common element in websites: a large image with text overlaid on it. You may have seen hints of this in previous labs/lectures but it's worthwhile to learn how to do it yourself.
31. Under the accordion section, create another large comment that says LARGE IMAGE WITH TEXT using the same comment format as before.
32. Then create a div with the class "large-img" which is a class I created in styles-2.css. This class uses the background-image property to embed an image into its background, similar to what you did in the parallax exercise in the previous lab.
33. Now, try adding text to it by creating an inner div (inside the large-img div) and give this one the class "large-img-caption". Write the text "NEW MENU ITEMS!" within this inner div.
34. Save the file and look at it in Chrome. Does this look acceptable to you? You'll probably notice that parts of the text are hard to read because it's white text and there are bright areas in the image, resulting in a low contrast between the two. To improve the look of this, we will add a black tint in between the image and the text.
35. Add another div that starts before the text caption but still inside the large-img div. This new div should have the class "tint" which is a translucent black background. Make sure this tint div is closed **after** the caption div closes. This code should look like this:

```

<!--
LARGE IMAGE WITH TEXT
-->
<div class="large-img">
  <div class="tint">
    <div class="large-img-caption">NEW MENU ITEMS!</div>
  </div>
</div>

```

36. Save the file and look at it again in Chrome. This should look much cleaner now with the black tint separating the caption from the background image. Note that I used absolute positioning to put the tint directly over the image and also to move the caption to center of the image. The outer div has relative positioning and the inner two have absolute positioning. This is important because without this the elements would be displayed one after the other and would look terrible! Have a look in styles-2.css at the three classes: .large-img, .tint, and .large-img-caption to see the positioning styles.

Modal window and button

37. A modal window is like a popup but it's within the website, not a secondary window. These are often used to bring the user's attention to a sale, newsletter, or some other kind of feature. We will add a modal window that pops up when a button is clicked.
38. Go into the resources page in Chrome and open the modal page. Copy all the code and paste it into the HTML file under the large image section. Notice that this code includes two chunks. The first is the HTML that creates the modal window, which is invisible by default, but still created there. The bottom of the HTML creates a button that says "Order Online". Then, we have a little section of jQuery after all the HTML and this just triggers the modal window to appear when that button is clicked.
39. Save the file and refresh Chrome. Click the blue button to see the modal window appear.

Jumbotron

40. The term "jumbotron" might make you think of a giant screen like the ones you see at a Blue Jays game or other sporting events or concerts. Don't get your hopes up – we don't have access to one of those 😞 In Bootstrap, Jumbotron is a component that displays large text to grab someone's attention or display something important.
41. Go into the resources page in Chrome and open the jumbotron file. Copy all the code and then paste it into the HTML file beneath the modal window portion.
42. This code is fairly simple, just creating some divs with the Bootstrap classes to display large text and add a button near the bottom of it, which doesn't do anything except jump to the top of the same page.
43. Save the file and have a look at it in Chrome.

Footer

44. A footer is a section at the bottom of a webpage usually containing less-important links and information than the top and main page include. Users have to scroll to the bottom to get to the footer so it's usually not going to contain anything super important.
45. Go into the resources page and open the footer file. Copy all the code from there and paste it into the HTML file below the Jumbotron section. This footer just includes a copyright message and 4 (fake) links.
46. Save the file and refresh Chrome to have a look at it.

Click the various links that jump around the page and/or open the accordion panels. If you have time, try adding other components that you found interesting in the pre-exercise component list.

This concludes this lab session. Call your TA over to check your work and receive your mark for this lab.

REMEMBER TO REMOVE YOUR MEMORY STICK FROM YOUR MACHINE AND PUT IT IN YOUR BACKPACK! (don't forget it)! 