this is cs50

seminar / interactive websites with Bootstrap and jQuery
agenda

01 disclaimers
02 Bootstrap
03 interactivity with data-* attributes
04 jQuery
05 interactivity with Bootstrap and jQuery
disclaimers
disclaimers

> this seminar will **only cover the front-end frameworks**
  - you can integrate this with Python and Flask as in class . . .
  - . . . or with any other back-end framework of your choosing

> this is meant to be **a quick look at what's possible** with Bootstrap and jQuery — we'll use simple examples throughout and gloss over syntax

> there is **always more to learn:**
  - Claire's "Bootstrap Basics" seminar: [https://youtu.be/sDyG8z5Q-m4](https://youtu.be/sDyG8z5Q-m4)

  - Bootstrap documentation: [https://getbootstrap.com/docs](https://getbootstrap.com/docs)
  - jQuery documentation: [https://api.jquery.com/](https://api.jquery.com/)

> follow along with the code we use here: [http://bit.ly/2zWYXcl](http://bit.ly/2zWYXcl)
Bootstrap
why bother with Bootstrap

> Bootstrap is an incredibly popular CSS / HTML / JS library that can make things look very clean very easily

> the framework also helps make your content look good across platforms
> to set up Bootstrap, you just need to include a few lines in the `<head>` of your HTML file:

```html
<head>
  <!-- bootstrap css styles -->
  <link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" rel="stylesheet">

  <!-- bootstrap javascript for interactivity -->
  <script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>

  <!-- meta tag so that content scales properly on your device -->
  <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
```
understanding the basics

> Bootstrap uses **CSS classes** to style most of its elements

```
<nav class="navbar navbar-dark bg-dark"> <!-- ... content ... --> </nav>
```

> Bootstrap revolves around a simple grid layout:
- a **container** wraps around an entire chunk of content
- **rows** break lines and contain (up to) 12 **columns**

```
<div class="container-fluid">
  <div class="row">
    <div class="col-sm"> [one of two "small" columns] </div>
    <div class="col-sm"> [one of two "small" columns] </div>
  </div>
</div>
```
example: the grid in action

1 <div class="container">
2   <div class="row">
3     <div class="col-md-4">
4       <h2>Heading</h2>
5       <p>!-- [content] --></p>
6       <p>!-- [button] --></p>
7     </div>
8     <div class="col-md-4">
9       <h2>Heading</h2>
10      <p>!-- [content] --></p>
11      <p>!-- [button] --></p>
12   </div>
13   <div class="col-md-4">
14     <h2>Heading</h2>
15     <p>!-- [content] --></p>
16     <p>!-- [button] --></p>
17   </div>
18 </div>
understanding the basics

> Bootstrap has a lot of pre-built "components" including different types of text, buttons, forms, tables, etc

```html
<form>
  <div class="form-group">
    <label for="email">Email address:</label>
    <input type="email" class="form-control" id="email">
  </div>
  <div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" class="form-control" id="pwd">
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```
understanding the basics

> Bootstrap has a lot of pre-built "components" including different types of text, buttons, forms, tables, etc

```html
<form>
  <div class="form-group">
    <label for="email">Email address:</label>
    <input type="email" class="form-control" id="email">
  </div>
  <div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" class="form-control" id="pwd">
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```
understanding the basics

> Bootstrap has a lot of pre-built "components" including different types of text, buttons, forms, tables, etc

```html
<form>
  <div class="form-group">
    <label for="email">Email address:</label>
    <input type="email" class="form-control" id="email">
  </div>
  <div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" class="form-control" id="pwd">
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

> this seminar will explore two components: **alerts** and **modals**
components in Bootstrap: alerts

> alerts in Bootstrap are `<div>` that emphasize certain content

```html
<br>
```
components in Bootstrap: modals

> modals in Bootstrap are `<div>` that sit "above" the rest of the site -- modals are defined by their content — header, body, and footer

```html
<div class="modal" tabindex="-1" role="dialog">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <!-- modal header content -->
    </div>
    <div class="modal-body">
      <!-- modal body content -->
    </div>
    <div class="modal-footer">
      <!-- modal footer content -->
    </div>
  </div>
</div>
```
interactivity with data-* attributes
introduction: data-* attributes

> sometimes, it's useful to store additional information about certain HTML tags -- for example, ids for each item in a list of items

> with HTML5, this becomes possible with **user-created "data" attributes**
  - the attributes can have (almost) any name and store string values

```html
<!-- restaurant ingredients inventory -->
<ul class="list-group">
  <li class="list-group-item" data-product_id="1001">potatoes</li>
  <li class="list-group-item" data-product_id="1121">carrots</li>
  <li class="list-group-item" data-product_id="0615">eggplant</li>
</ul>
```

> we can then access these data-* attributes in our CSS and JS code
data-* attributes in BS: alerts

> in Bootstrap, data-* attributes are used to add interactivity to components by linking certain buttons to styled HTML elements

> for example, we can enable dismissing an alert:

```html
<div class="alert alert-warning alert-dismissible fade show" role="alert">
  <strong>Holy guacamole!</strong> You should check in on some of those fields below.
  <button type="button" class="close" data-dismiss="alert" aria-label="Close">
    <span aria-hidden="true">×</span>&times;</button>
</div>
```

Holy guacamole! You should check in on some of those fields below.
data-* attributes in BS: alerts

in Bootstrap, data-* attributes are used to add interactivity to components by linking certain buttons to styled HTML elements.

for example, we can enable dismissing an alert:

```html
<div class="alert alert-warning alert-dismissible fade show" role="alert">
  <strong>Holy guacamole!</strong> You should check in on some of those fields below.
  <button type="button" class="close" data-dismiss="alert" aria-label="Close">
    <span aria-hidden="true">×</span>
  </button>
</div>
```

Holy guacamole! You should check in on some of those fields below.
data-* attributes in BS: alerts

> in Bootstrap, data-* attributes are used to add interactivity to components by linking certain buttons to styled HTML elements

> for example, we can enable dismissing an alert:

```html
<div class="alert alert-warning alert-dismissible fade show" role="alert">
  <strong>Holy guacamole!</strong> You should check in on some of those fields below.
  <button type="button" class="close" data-dismiss="alert" aria-label="Close">
    <span aria-hidden="true">×</span>
  </button>
</div>
```

**Holy guacamole!** You should check in on some of those fields below.
data-* attributes in BS: modals

> we can also use data-* attributes for modal components:

```html
<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal">Launch</button>

<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <!-- some header content -->
        <button type="button" class="close" data-dismiss="modal"></button>
      </div>
      <div class="modal-body"> <!-- modal body content --> </div>
      <div class="modal-footer">
        <!-- some footer content -->
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
      </div>
    </div>
  </div>
</div>
```
data-* attributes in BS: modals

> we can also use data-* attributes for modal components:

```html
<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal">Launch</button>

<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <!-- some header content -->
        <button type="button" class="close" data-dismiss="modal">Close</button>
      </div>
      <div class="modal-body"> <!-- modal body content --> </div>
      <div class="modal-footer">
        <!-- some footer content -->
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
      </div>
    </div>
  </div>
</div>
```
data-* attributes in BS: modals

> similarly, we can use the **data-dismiss** attribute for modal components
> in the example below, we can dismiss the modal with two buttons:
introduction

> jQuery is a JavaScript library that makes programming in JS a lot easier, especially when it comes to manipulating elements of the DOM.

> Bootstrap interactivity is implemented with jQuery under the hood.
  - as you saw, you can use it in HTML with data- attributes.
  - or, as we'll see now, you can use jQuery to directly.

> to get started with jQuery, just link it in your HTML `<head>`:

```html
<head>
  <!-- jQuery -->
  <script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
</head>
```
basic syntax

> jQuery uses a basic set-up for syntax: $(selector).action()
  - the $ is just a reference to the fact that you're using jQuery
  - the selector is a reference to an HTML element like #myElement
  - the action is a JavaScript function

```javascript
// hide the element with ID "test" from the DOM
$('#test').hide()
```

> note: selectors in jQuery are just like the CSS selectors we saw in CS50 before -- "tag", "#id", and ".class"
events in jQuery

> in addition to just running line-by-line, jQuery can also listen for "events" such as button clicks

```javascript
// change the content of paragraph elements
$('p').click(function() {
    $(this).text("contents have changed!");
});
```
interactivity with Bootstrap and jQuery
integrating jQuery and Bootstrap

> Bootstrap interactivity essentially provides "custom" jQuery functions that you can use and events you can listen for

```html
<!-- test alert -->
<div class="alert alert-primary" role="alert">This is a test alert</div>

<!-- interactive button -->
<button class="btn btn-secondary btn-block" id="close_alert">Close Alert</button>

<!-- jQuery -->
<script>
  $('"#close_alert"').on('click', function() {
    $('".alert"').alert('close')
  });
</script>
```
integrating jQuery and Bootstrap

> Bootstrap interactivity essentially provides "custom" jQuery functions that you can use and events you can listen for

```html
<!-- test alert -->
<div class="alert alert-primary" role="alert">This is a test alert</div>

<!-- interactive button -->
<button class="btn btn-secondary btn-block" id="close_alert">Close Alert</button>

<!-- jQuery -->
<script>
  $$('#close_alert').on('click', function() {
    $('.' + 'alert').alert('close');
  });
</script>
```
integrating jQuery and Bootstrap

Bootstrap interactivity essentially provides "custom" jQuery functions that you can use and events you can listen for

```html
<!-- test alert -->
<div class="alert alert-primary" role="alert">This is a test alert</div>

<!-- interactive button -->
<button class="btn btn-secondary btn-block" id="close_alert">Close Alert</button>

<!-- jQuery -->
<script>
$("#close_alert").on("click", function(){
  $(".alert").alert('close')
});
</script>
```
using jQuery to make BS dynamic

> we can use jQuery to replicate the functionality we saw earlier from 
data-toggle and data-dismiss, but we can also build on this

> let's create a modal that has different content depending on what button is pressed to open it

> buttons:

<!-- buttons -->
<button class="btn btn-primary" data-toggle="modal" data-target="#exampleModal" data-myfavoritetypeparameter="Brian">
  Feedback for Brian
</button>

<button class="btn btn-primary" data-toggle="modal" data-target="#exampleModal" data-whatever="Veronica">
  Feedback for Veronica
</button>
using jQuery to make BS dynamic

> modal:

```html
<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">New Feedback</h5>
        <button type="button" class="close" data-dismiss="modal">&times;</button>
      </div>
      <div class="modal-body">
        ...
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Submit Feedback</button>
      </div>
    </div>
  </div>
</div>
```
using jQuery to make BS dynamic

> modal:

```html
<div class="modal fade id="exampleModal" tabindex="-1" role="dialog">
  <div class="modal-dialog role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title id="exampleModalLabel">New Feedback</h5>
        <button type="button" class="close" data-dismiss="modal">Close</button>
      </div>
      <div class="modal-body">
        ...
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Submit Feedback</button>
      </div>
    </div>
  </div>
</div>
```
using jQuery to make BS dynamic

> jQuery:

```javascript
$('#exampleModal').on('show.bs.modal', function (event) {
    // extract info from button that triggered the modal
    var recipient = $(event.relatedTarget).data("myfavoriteparameter")

    // update the title of the modal
    var modal = $(this)
    modal.find('.modal-title').text("New Feedback for " + recipient)
})
```

> the jQuery code is triggered when the modal is shown
> it pulls the data-* attribute from the button that showed the modal and uses it to dynamically update the content of the modal