# Web Development Frameworks

Dr. Garrett Dancik

# Overview

- Web development frameworks are software frameworks for web development that simplify common tasks
- **Bootstrap** is a HTML/CSS/JavaScript framework for developing responsive, mobile first applications
  - Developed by Twitter and released to the public in 2011
- **jQuery** is a JavaScript library for JavaScript programming created in 2006
- **Angular** is a TypeScript-based web application framework
  - Developed by Google
  - Angular (beginning with Angular 2+) is a rewrite of AngularJS which was a JavaScript framework and initially released in 2010
- **React** is a JavaScript library for developing user interfaces, developed and maintained by Facebook (now Meta) and initially released in 2013

# Bootstrap

- Can be included by loading the JavaScript and CSS libraries in the header of a page
- Basic design concepts
  - Responsive to changes in screen size (e.g., desktop vs. mobile)
    - Uses a grid system consisting of 12 columns
    - Includes collapsible navbars and other elements
- Use Bootstrap by specifying *classes*, e.g.
  - *col-sm-4* will create an element is 4 columns wide
    - Example: https://www.w3schools.com/bootstrap5/bootstrap_grid_basic.php
  - *btn* and *btn-primary* for buttons
    - Example: https://www.w3schools.com/bootstrap5/bootstrap_buttons.php
- Full tutorial: https://www.w3schools.com/bootstrap5/index.php

# Selected Bootstrap examples

- Bootstrap's grid system allows up to 12 (responsive) columns across the page:
  - https://www.w3schools.com/bootstrap5/bootstrap_grid_basic.php
- Bootstrap provides a variety of button styles:
  - https://www.w3schools.com/bootstrap5/bootstrap_buttons.php
- Bootstrap provides a variety of navigation bar styles:
  - https://www.w3schools.com/bootstrap5/bootstrap_navbar.php

# jQuery

- As of February 6, 2022, jQuery is used by 78% of the top 10 million websites: (https://w3techs.com/technologies/overview/javascript_library)
- The basic jQuery framework involves applying an action to a set of elements as follows:
  - $(*selector*).*action*()
    - $ - specifies we are using jQuery (you can also use *jQuery*)
    - selector – a CSS style selector to apply the action to (e.g., *p*, *div.class*)
    - action() – a function to apply to each element with the given selector
- Examples
  - $("p").hide() - hides all <p> elements.
  - $(".test").hide() - hides all elements with class="test".
  - $("#test").hide() - hides the element with id="test".

- Tutorial: https://www.w3schools.com/jquery/default.asp

# Common jQuery actions

| Action | Description | Example |
|---|---|---|
| hide(), show(), or toggle() | Hides, shows, or toggles the element(s) | $("p").hide() |
| html() or text() | Gets the innerHTML or innerText of an element | $("p#id").html() |
| html("value") or text("value") | Sets the innerHTML or innerText of an element to the specified value | $("p#id").text("hello") |
| addClass("class"), removeClass("class"), toggleClass("class") | Adds, removes, or toggles the class of an element | $("p").addClass("fancy") |
| css("*propertyname*","*value*") | Sets the CSS property of an element | $("p").css("background-color", "yellow") |

# jQuery document.ready()

- It is good practice to call jQuery functions only after the page has been loaded. This is accomplished by including your jQuery code inside of *document.ready* event, which is the event fired after the document object model (DOM) is ready.
- Without doing this, jQuery may not be able to access all of the intended DOM elements
- In JavaScript, a *function* can be passed as an argument into another function (see JS_function example)

# jQuery document.ready()

- The following statement uses jQuery to call a *function* after a page is loaded
  - $(document).ready(*function*)
- However anonymous functions are often used:

```
$(document).ready(function(){
  // jQuery methods go here...
});
```

- The following is shortcut for the above notation:

```
$(function(){
  // jQuery methods go here...
});
```

# Handling events using jQuery

- Common events include *click()*, *doubleclick()*, *mouseenter()*, *mouseleave()*, and *hover()*
- For examples see
  - https://www.w3schools.com/jquery/jquery_events.asp
- When we handle events, we specify a function that should be called when the event is triggered, e.g.
  - $("p").click(*functionToCall*)
- However, the function to call is usually passed as an anonymous function

```
// set onclick event of all paragraphs
$("p").click(function(){
   // action goes here!!
   $(this).hide();   // hides the current
element
});
```

$(this) accesses the current element

# React

- A JavaScript library for building user interfaces
- Uses a virtual DOM to represent the real DOM
  - When an element in the virtual DOM changes, React updates that element (and only that element) in the real DOM
  - This feature makes React very efficient at rendering dynamic web pages
- React is used to build *reusable UI* components
- To use React, you it is recommended to use *Node.js* (a back-end JavaScript runtime environment), but we will run simple examples in the browser
- Tutorials:
  - W3schools: https://www.w3schools.com/REACT/DEFAULT.ASP
  - ReactJS.org: https://reactjs.org/docs/getting-started.html

# JSX

- JSX, or JavaScript XML, is an extension of JavaScript that allows you to include HTML in React code
- JSX makes React code easier to understand (though technically is not necessary)
- Creating React elements:
  - With JSX:
    - const element = <h3>Hello, world!</h3>;
  - Without JSX:
    - const element = React.createElement('h3', null, 'Hello world!');

- Rendering a React element
  - ReactDOM.render(element, document.getElementById('root'));

# Let's look at some examples

- We will focus on
  - Creating and rendering elements
  - Creating and rending re-usable components
  - Event handling