Why Javascript Frameworks

INFO 253A: Frontend Web Architecture

Kay Ashaolu

We've spent a lot of time on Javascript

- We've learned the basics of Javascript
- We've learned how Javascript is used in the browser (i.e. how it manipulates the Document Object model)
- We've learned some of the newer features of JavaScript (ES6)

We've spent a lot of time on Javascript

- We did this so that you have a good foundation to write some quality JavaScript
- It is really, really important to have a good foundation for javascript

Okay, so now what?

- We could start learning how to build a complex app with what we know
- HTML, CSS, and JavaScript is all you need build an application
- You can write functions to perform actions and manage state

Okay, so now what?

- You can use the document object in JavaScript to manipulate the DOM
- We have not talked about AJAX or making HTTP Requests, but its coming
  - How you connect your web app to external systems

But that would be hard

- Remember that JavaScript feature matrix
- You run into the same issue with CSS, and even HTML
- There's even a website dedicated to this

But that would be hard

- Also writing quality modular JavaScript is hard
- Focus on writing self contained components that interact with each other
- Modularity helps with code maintainability and development

And would take a long time

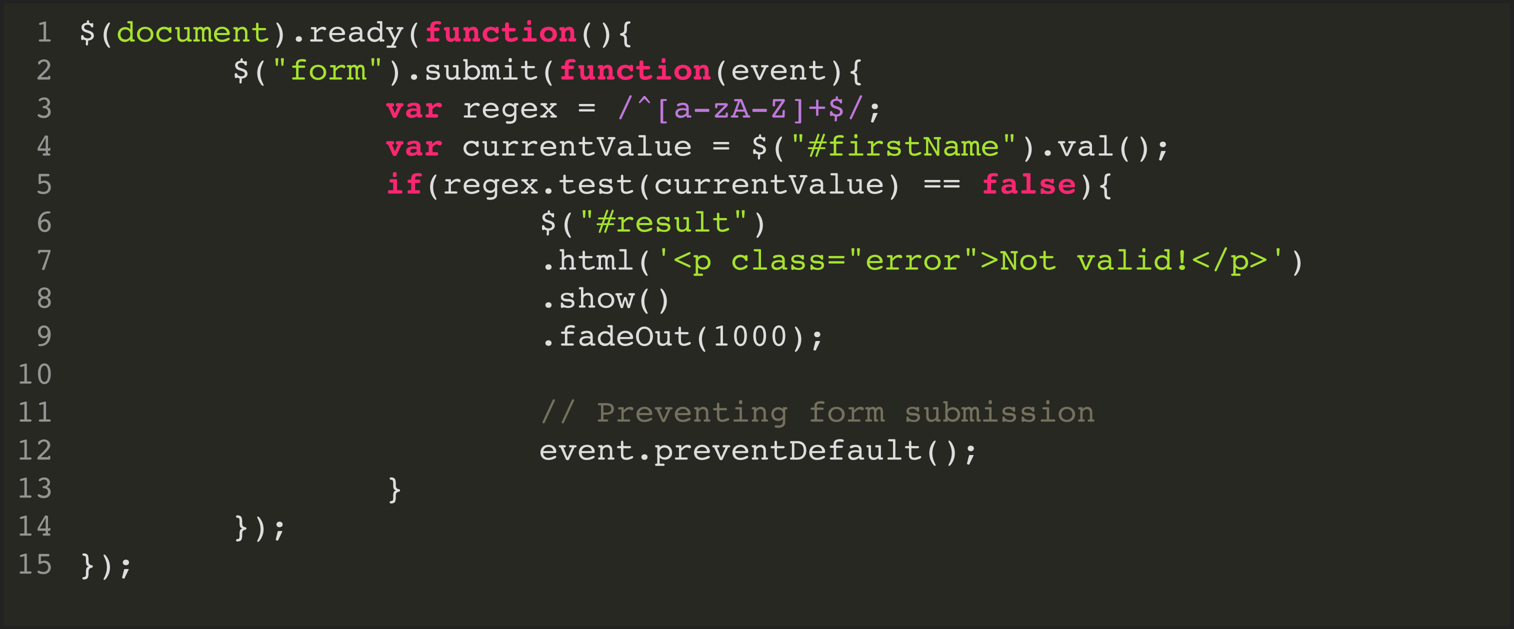
- You will find yourself writing a lot of what's called "boilerplate" code
- Code that you write over and over again
- Essentially you are creating your own "libraries" and "frameworks" to save yourself time

What is a JavaScript Library?

- A library performs specific, well-defined actions
- Typically it provides several functions for you to use
- You are in control of your code, you simply use a library to make it easier to do what you wanted to do

JavaScript Library Example

jQuery is the granddady of JavaScript Libraries



```
$(document).ready(function(){
        $("form").submit(function(event){
                var regex = /^[a-zA-Z]+$/;
                var currentValue = $("#firstName").val();
                if(regex.test(currentValue) == false){
                        $("#result")
                        .html('<p class="error">Not valid!</p>')
                        .show()
                        .fadeOut(1000);

                        // Preventing form submission
                        event.preventDefault();
                }
        });
});
```

Please note

- You are adding html code directly through your JavaScript
- Imagine you want to standardize that error throughout your web applicaton
- You would have to copy and paste that HTML code everywhere you need to add error
- You are in control

What is a JavaScript Framework?

- A framework dictates the architecture your project will follow
- You typically hook your code into a framework in order to get it to work
- The framework usually makes it easy to build a complete application quickly
- You typically have to fit the pattern that is defined by the framework

Library vs Framework

Library

- Flexible
- Developer Control
- Easy to incorporate to existing code
- More work to get started

Framework

- Quick to get started
- Quick to understand the basics
- Less Flexible

What is React?

- This is a good question
- If you look up online, people will *largely* say React is a library
- However some will make the case that React is a framework
- React is typically paired with frameworks like Flux and Redux so the lines get blurred even more
- You can even make the argument that hooks render Redux irrelevant so perhaps React is a framework

This is what I think

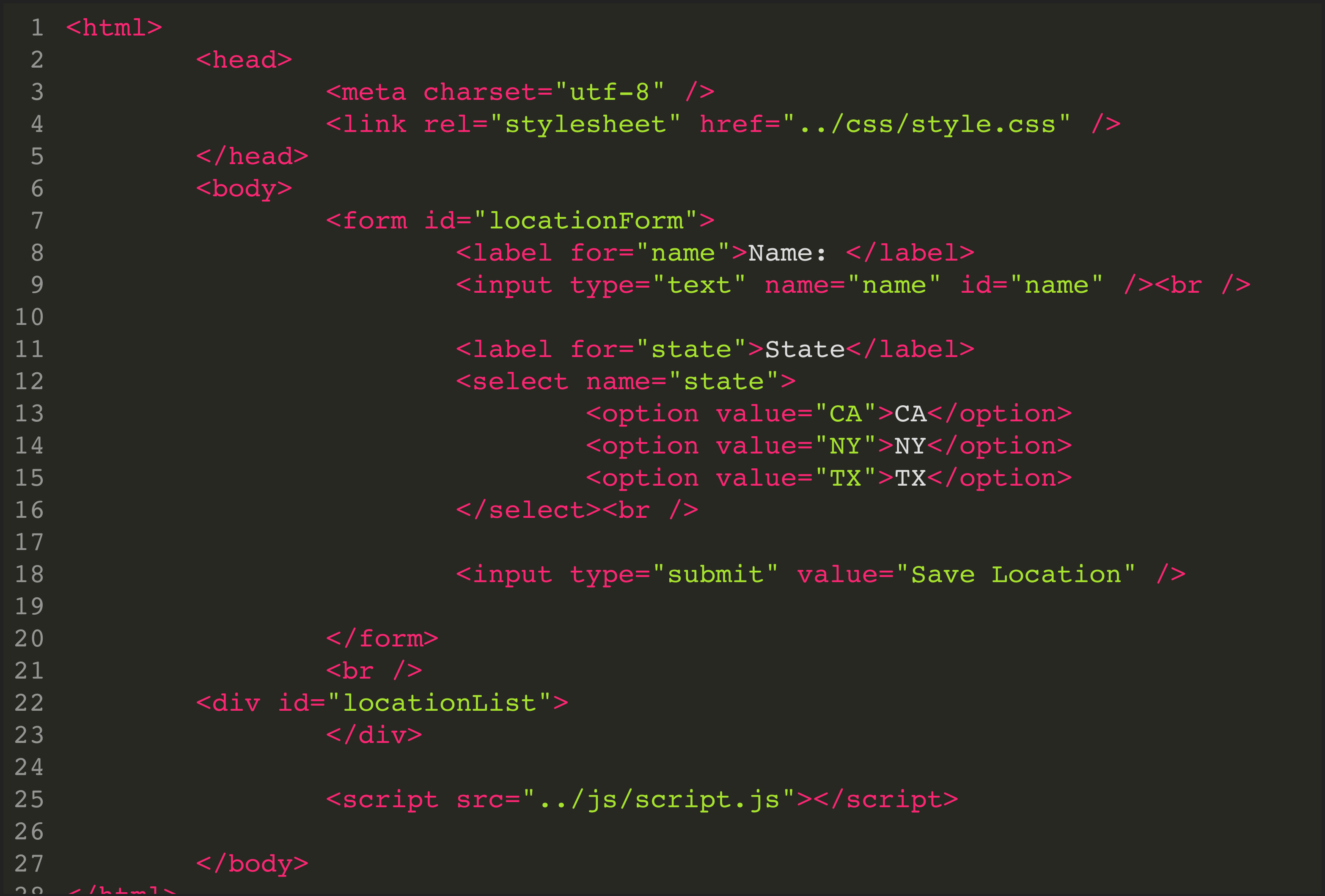
React is a **JavaScript library** that provides functionality in such a way that it **encourages the use of a new kind of framework** of developing user interfaces, centralizing the data of the application in one area and asking the developer to tell the application what it should display with the data that it has.

So that wasn't well worded at all

- React lets you focus on writing self contained pieces of code (modules) where you describe how you would render that module if it had a certain piece of data. What you don't have to worry about is figuring out **how** your UI is going to manipulate the dom to make that change happen

Remember this?

html/index.html



```html
<html>
        <head>
                <meta charset="utf-8" />
                <link rel="stylesheet" href="../css/style.css" />
        </head>
        <body>
                <form id="locationForm">
                        <label for="name">Name: </label>
                        <input type="text" name="name" id="name" /><br />

                        <label for="state">State</label>
                        <select name="state">
                                <option value="CA">CA</option>
                                <option value="NY">NY</option>
                                <option value="TX">TX</option>
                        </select><br />

                        <input type="submit" value="Save Location" />

                </form>
                <br />
        <div id="locationList">
                </div>

                <script src="../js/script.js"></script>

        </body>
</html>
```

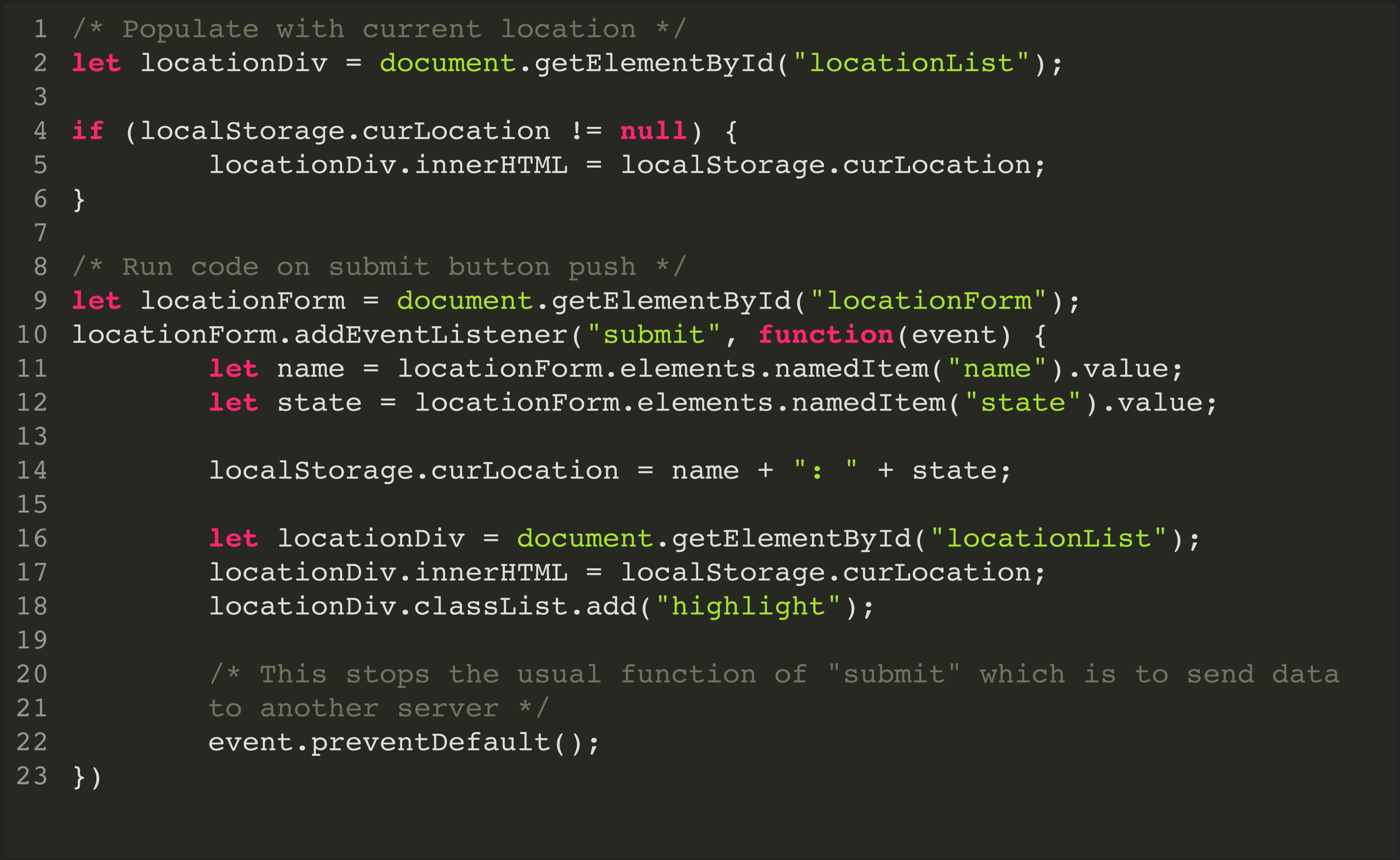
Remember this?

css/style.css

```css
.highlight {
        color: red;
        background-color: yellow;
}
```

Remember this?

js/script.js



```
/* Populate with current location */
let locationDiv = document.getElementById("locationList");

if (localStorage.curLocation != null) {
        locationDiv.innerHTML = localStorage.curLocation;
}

/* Run code on submit button push */
let locationForm = document.getElementById("locationForm");
locationForm.addEventListener("submit", function(event) {
        let name = locationForm.elements.namedItem("name").value;
        let state = locationForm.elements.namedItem("state").value;

        localStorage.curLocation = name + ": " + state;

        let locationDiv = document.getElementById("locationList");
        locationDiv.innerHTML = localStorage.curLocation;
        locationDiv.classList.add("highlight");

        /* This stops the usual function of "submit" which is to send data
        to another server */
        event.preventDefault();
})
```

Remeber this?

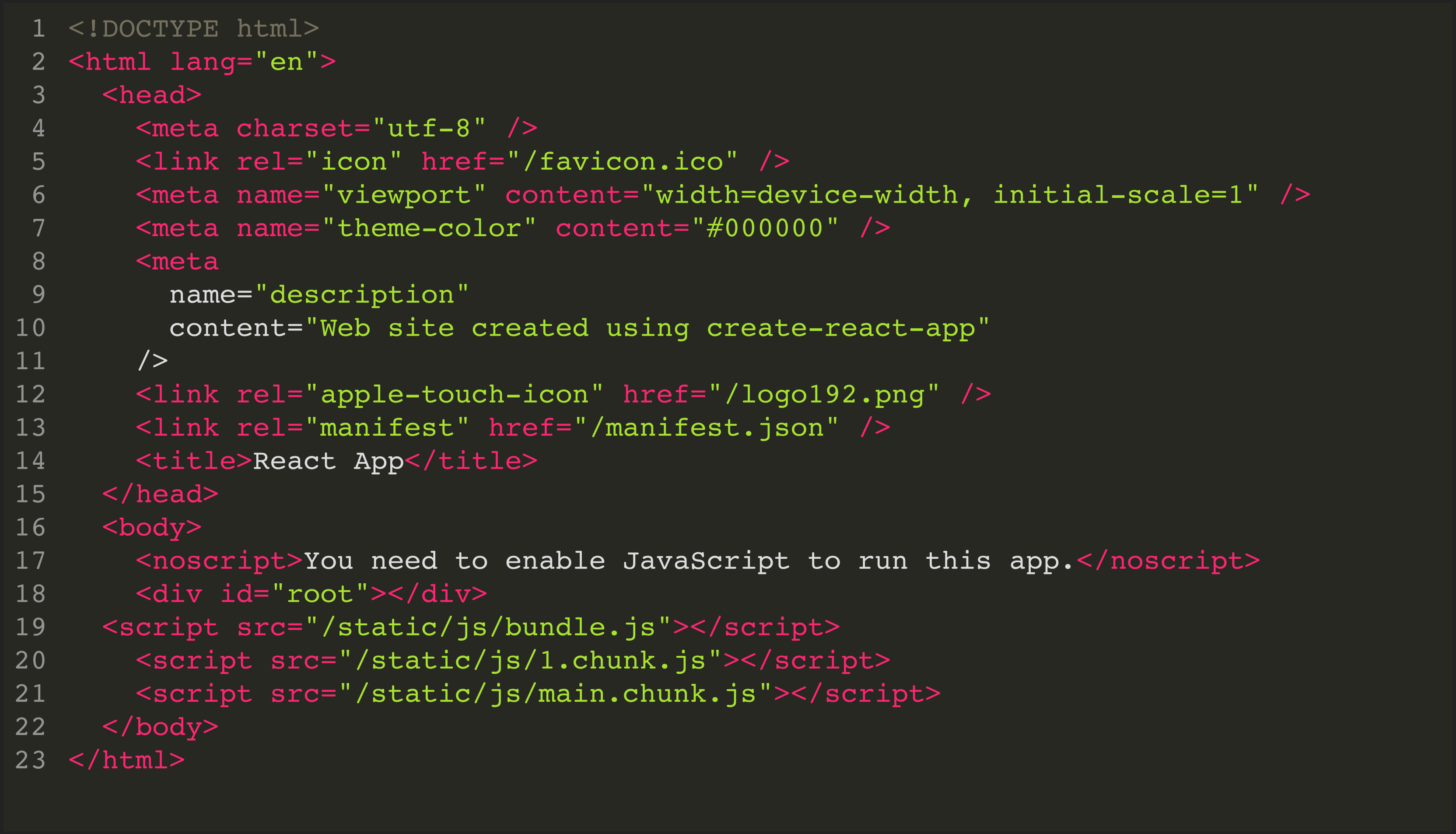
- Remember how you had to both populate #locationList on startup and on submit button event?
- You had to think about every aspect when #locationList would change, including:
    - On startup
    - On submit button

What's different with React

- You write a component that would represent #locationList
- You write code on how you would render locationList if given an array (think, like a function)
- That's it!

React Example

index.html (Autogenerated)



```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="/logo192.png" />
    <link rel="manifest" href="/manifest.json" />
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
  <script src="/static/js/bundle.js"></script>
    <script src="/static/js/1.chunk.js"></script>
    <script src="/static/js/main.chunk.js"></script>
  </body>
</html>
```

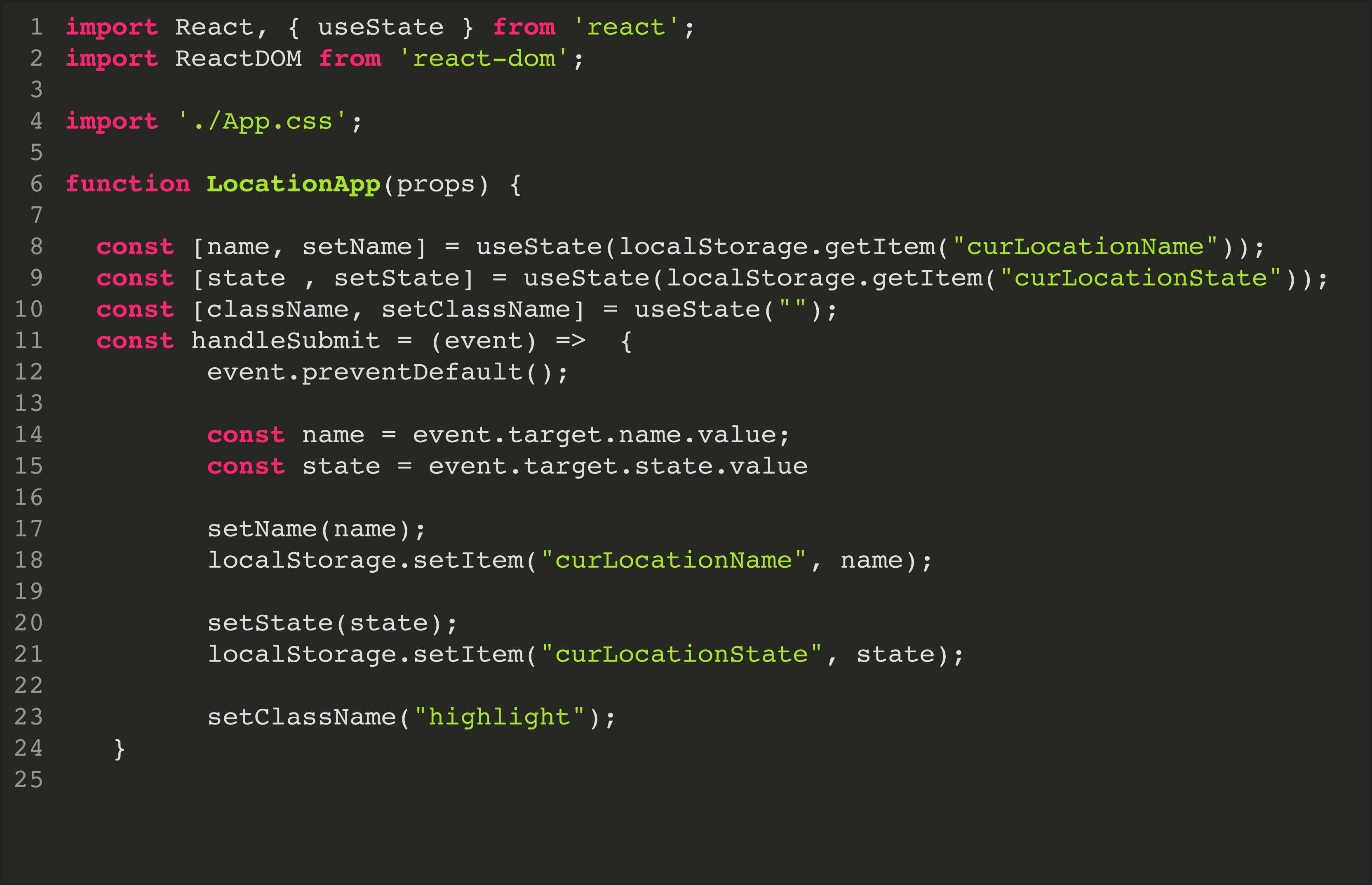
React Example

index.css

```css
.highlight {
        color: red;
        background-color: yellow;
}
```

React Example

index.js



```
import React, { useState } from 'react';
import ReactDOM from 'react-dom';

import './App.css';

function LocationApp(props) {

  const [name, setName] = useState(localStorage.getItem("curLocationName"));
  const [state , setState] = useState(localStorage.getItem("curLocationState"));
  const [className, setClassName] = useState("");
  const handleSubmit = (event) =>  {
         event.preventDefault();

         const name = event.target.name.value;
         const state = event.target.state.value

         setName(name);
         localStorage.setItem("curLocationName", name);

         setState(state);
         localStorage.setItem("curLocationState", state);

         setClassName("highlight");
   }

```

React Example

index.js

React Example explained

- There's a lot going on, but this code does the same thing as the previous code
- Note how html is in the JavaScript (called JSX)
- The real cool thing is that once LocationApp is created, you can add multiple instances anywhere you want and it will still work

Questions