Gearing Up for Development

CS130(0)

Development

Development is a coding heavy assignment!

You will need to create application using React.js (a Javascript Library).

This application will display a list of elements that can be filtered and sorted in different ways.

What types of elements will you have, and how will you filter them?

That’s up to you!

What is React?

React is a UI library developed at Facebook to facilitate the creation of interactive, stateful & reusable UI components.

As Wikipedia puts it, React allows developers to “create large web-applications that use data and can change over time without reloading the page.”

People use React.js because it’s a more efficient way of interacting with an HTML document/making live changes to a webpage.

With this in mind, React is perfect for creating a live filtering system - which is what we want you to do!

The DOM

As a bit of additional background to React, it's important to understand how the DOM works.

DOM stands for the Document Object Model - the DOM is essentially a tree containing all the HTML elements of a document.

Whenever Javascript interacts with an HTML document, it's actually interacting with the DOM, and making direct changes to this tree!

So why does this matter for React/what's the point of using React?

The DOM

React uses a cool concept called Virtual DOM.

Whenever Javascript needs to make changes to the DOM, it interacts with (and ultimately changes) the entirety of the tree!

Virtual DOM allows React to interact with (and thus change) as little of the tree as possible whenever changes to the HTML need to be made.

Thus, we use React because it optimizes interaction with the DOM.

The DOM

Understanding exactly how the DOM works isn't critical to the development of Development, but it's good to know for understanding why React is so widely used these days.

The React lab handout contains a good example of how the Virtual DOM works, as well as a link to more information regarding DOM functionality - check it out if you'd like more information!

CS1300 Javascript/React Labs!

The Development assignment has one of the largest learning curves in this course - React.js can be pretty confusing for someone using it for the first time!

The Javascript and React labs are meant to help ease this learning curve.

However, it's heavily encouraged that you look over the React lab and finish the tasks involved if you haven't already (regardless of your Javascript knowledge!).

We mean it!

The React lab is super helpful, and functions as a very good starting point for this assignment.

The lab walks you through creating a simple filter, and the framework that's set up in the lab is very easily extendable to the Development assignment.

Additionally, the lab goes over installing React on personal computers - if you plan to work remotely from the CIT, the "Getting Started" portion of the handout is pretty much mandatory!

If you haven't done the lab and/or have questions while doing it, the TAs will be happy to help you out!

React Components

Components are at the core of how React functions - they take in data, then spit out that data in the form of HTML elements that then get rendered to the screen.

React components function very similarly to the way classes do in Javascript. Classes in Javascript are nothing more than special functions (and the same is true for React components).

Your Development code must utilize three different React components. Note that if you've completed the React lab, you've already created quite a few!

React Components

We'll be going over the two main elements of a React component: the state, and the render method.

As we progress, keep in mind that all React components extend "Component." This is a file you receive upon installation of React, and one you will need to import for every component you create:

```
import React, { Component } from 'react';
```

This line imports React, alongside Component, and is necessary for each of your files.

The State

The state is initialized within a React component's constructor. This initialization looks like this:

```
constructor(props) {
        super(props);

        // State initialization
        this.state = {
                key: "value"
        };
}
```

The State

The state is where you store the variables that you want to access anywhere within your component.

As we saw in the last slide - the state stores variables as key-value pairs.

These pairs can then be accessed from the state using this.state.key and changed within your code using a call to this.setState( ).

Some things to think about when utilizing state in Development: What values are your filters currently set to? Is the user sorting?

The State (and the Constructor)

A constructor (and thus, the initialization of the state) is not a mandatory aspect of a React component.

With this in mind, you can create very useful React components that don't have constructors and don't utilize this.state at all!

However, at least one of your components will need to use the state in order to save user values (and if you've started the lab, you might know which one!).

The Render Method

Unlike the constructor, the render method is a mandatory aspect of a React component!

The render method is the way in which the content of a component shows up on a webpage:

```
render( ) {
        return (
                // HTML elements are placed in here
        );
}
```

The Render Method

You'll notice that the content being returned in the render method is pure HTML code.

It's inside this render method that you, as the developer, get to determine how your information gets displayed to the screen.

Personally, I like to think of the render method as a "draw" call! (It's the way in which content gets displayed on the screen, and the way in which content is displayed in relation to other content is determined by which render calls are called first.)

Initializing a Component in App.js

Once you've created your React app, you'll notice you have a file called App.js.

App.js will also extend Component, and will contain its own render method. Inside this render method is where your calls to your other components will be made.

The order in which you initialize each component in App.js' render method will determine the order in which your own components' render methods will be called!

What Initializing a Component Looks Like

Assuming we have nothing more than React component named HelloWorld, the render method of App.js would look like this:

```
render( ) {
        return (
                <div className="App">
                        // this is some weird syntax - it's not really HTML, but is how you'll call a component:
                        <HelloWorld />
                </div>
        );
}
```

A Few Final Tips for Development

The installation process for React can be complicated. Even if you don't code your project until the very last couple of days, try to finish your installation well before, just in case you run into any major issues!

Remember that all React components that you create will extend "Component." Additionally, every React component exports itself at the end of its file! Check and double check your imports and exports!

There's stencil code provided for the React lab - even if you don't do the lab (despite all our recommendations to), it might be a good idea to grab this code!

The stencil uses a few different build in functions (.filter and .search) - these functions are A-OK to use in your Development code, and if you want more information on how they work, feel free to ask the TAs!

Good Luck!

You've got a little over a week to complete Development!

Please come into hours to talk to the TAs if you need help with debugging or have questions about React.

Alternatively, you can post your code privately on piazza, and we can help you debug through there!

Good luck everyone!