Class Components in ReactJS

Hunter Jorgensen

Class Components

As with Functional Components, React classes takes a *component* in building apps.

A React component extends a Component class and it MUST implement a render method.

The render method allows for JavaScript and HTML (technically XML) to intermingle

JSX provides a lot of benefits, but the best part is the syntax it gives us for writing HTML (instead of a bunch of strings.)

When a single Component is updated, only THAT component and its children are updated (all others are not affected). We'll explore this in a bit.

Demo

The *render* Method

In case it wasn’t clear, the render function in a React component has some unique properties:

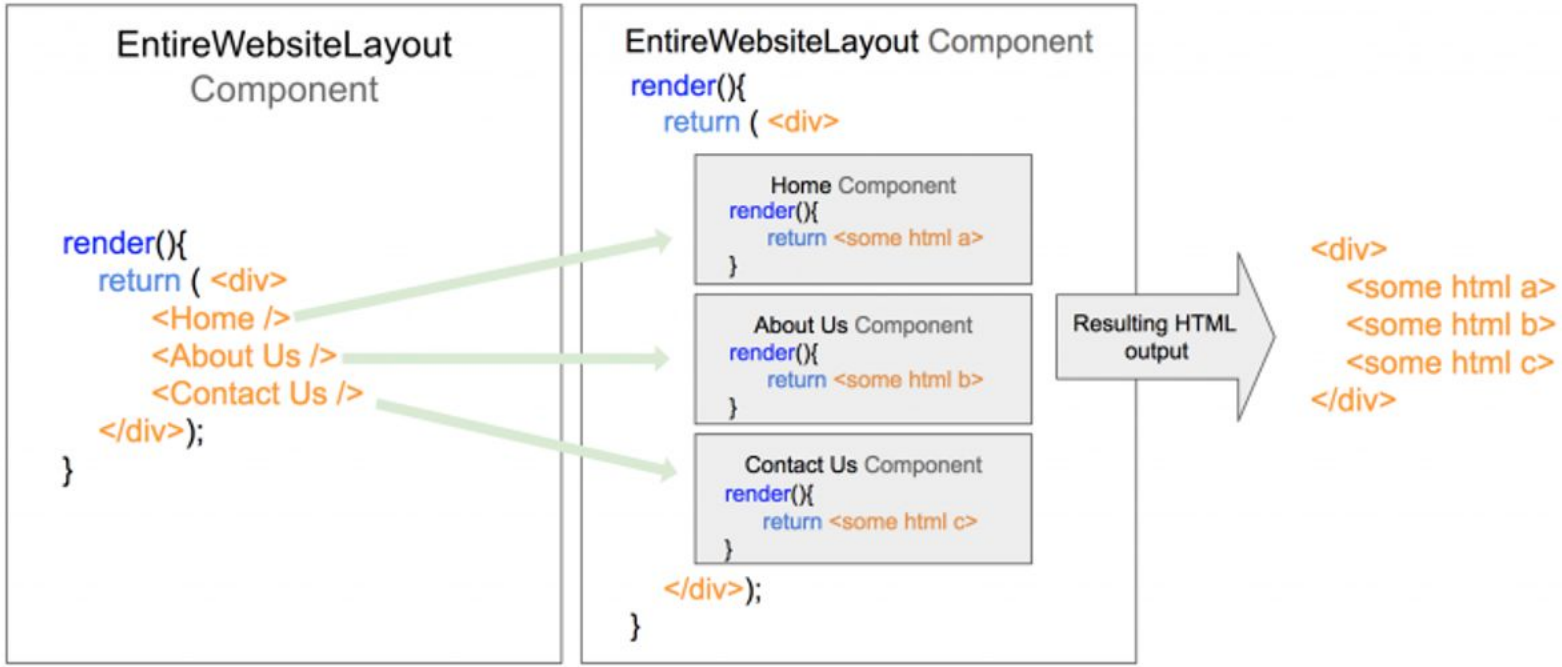
- HTML can be written between 2 parenthesis (it works best with a single parent)
- Code can be referenced IN the HTML by using curly brackets
- It is the ONLY required function on a Component

Components

Notice that Components can be called by other Components: we can see this how the first file imports the second file, but this can be done more explicitly. But this is a good practice if we want to reduce common code. Let's try to minimize the input field logic and are basically making custom HTML tags.

Demo

Components: Visualizing The Behavior



Component State

To make an app worth any value, it will need to manage and handle data.

In React, every component has an object called state that manages any data associated with it. State is immutable (except for in the constructor), but can be updated by calling setState.

Components also have an constructor (optionally required) to set the state or prepare any data before the component is initialized.

Demo

Component Properties

Like HTML, Components in React have custom attributes called properties (or properties for short.) These properties take arguments just like properties in HTML.

Demo

This is pretty confusing syntax, but the attribute is an argument and the data passed in is a value. In the child component, this is set in an object known as ‘props’. This is also accessible via through the constructor.

Functional Components

There is a second common way to write React components, and that is by writing a function.

We will explore functional components in the next lecture.

Quick Review: Components

React is composed of components that can be created by either extending a class or passing a function.

If you choose to extend a class, you MUST implement the render method.

On the class, we only care about the setState, constructor and render APIs. You are welcome to explore these more on your own!

Quick Review: State

State is a super important concept in React, as the idea of how and where data is stored is key for the rest of the semester.

Every Component has a state, which is data stored locally on the component. If you add a constructor, you must first set the state, otherwise you must ONLY modify the state through `this.setState(someObj)`. This kicks off an asynchronous process that replaces the old state with a new one, forcing the component to re-render.

When reading from the state, you may use `this.state.attribute`

Quick Review: Prop(ertie)s

Props, short for properties, is data passed from a PARENT to a CHILD component (remember the tree structure from the DOM lectures?)

Each attribute that becomes argument that can accept a value; it is then up to the child component to figure out what to do with it.

Props can be set in the state (which will cause a re-render, if important) OR passed to child components and referenced as is.

Demo

PropTypes and Default Props

You can specify what kinds of arguments you're getting in a component and if it's missing that argument, what the default should be. This is similar to the default parameters functionality of JavaScript.

Demo

Note: these only throw errors in the console and React will still try to compile your code as best as possible