What is React?

Open source library for building user interfaces

Not a framework

Focus on UI

Rich ecosystem

Why learn React?

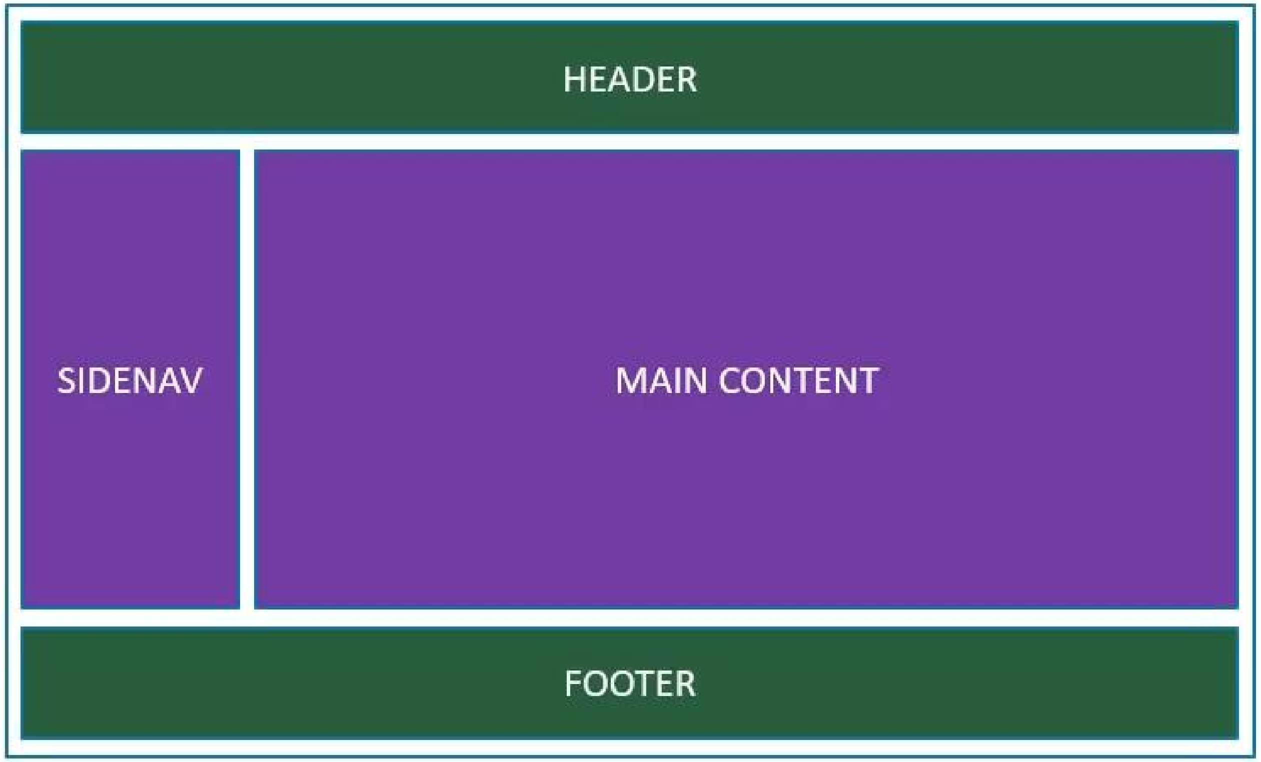
Created and maintained by Facebook

More than 100k starts on Github

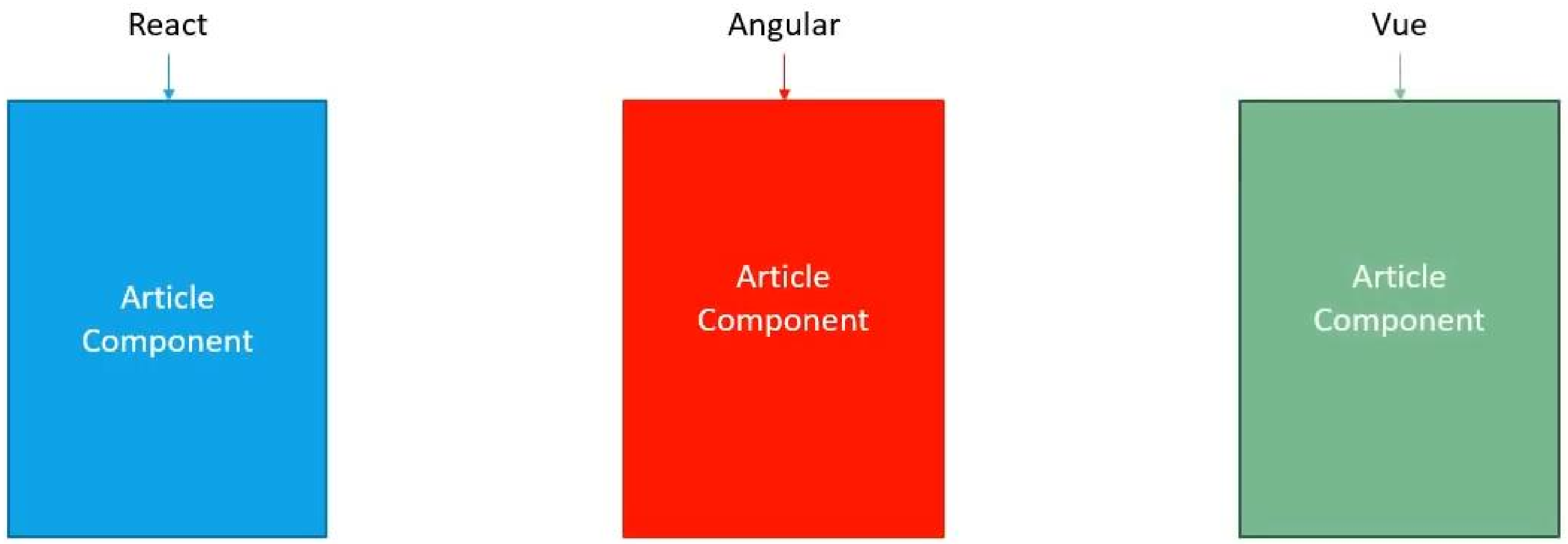
Huge community

In demand skillset

Component Based Architecture



Reusable code



React is declarative

Tell React what you want and React will build the actual UI

React will handle efficiently updating and rendering of the components

DOM updates are handles gracefully in React.

More on why React?

Seamlessly integrate react into any of your applications.

Portion of your page or a complete page or even an entire application itself.

React native for mobile applications

Prerequisites

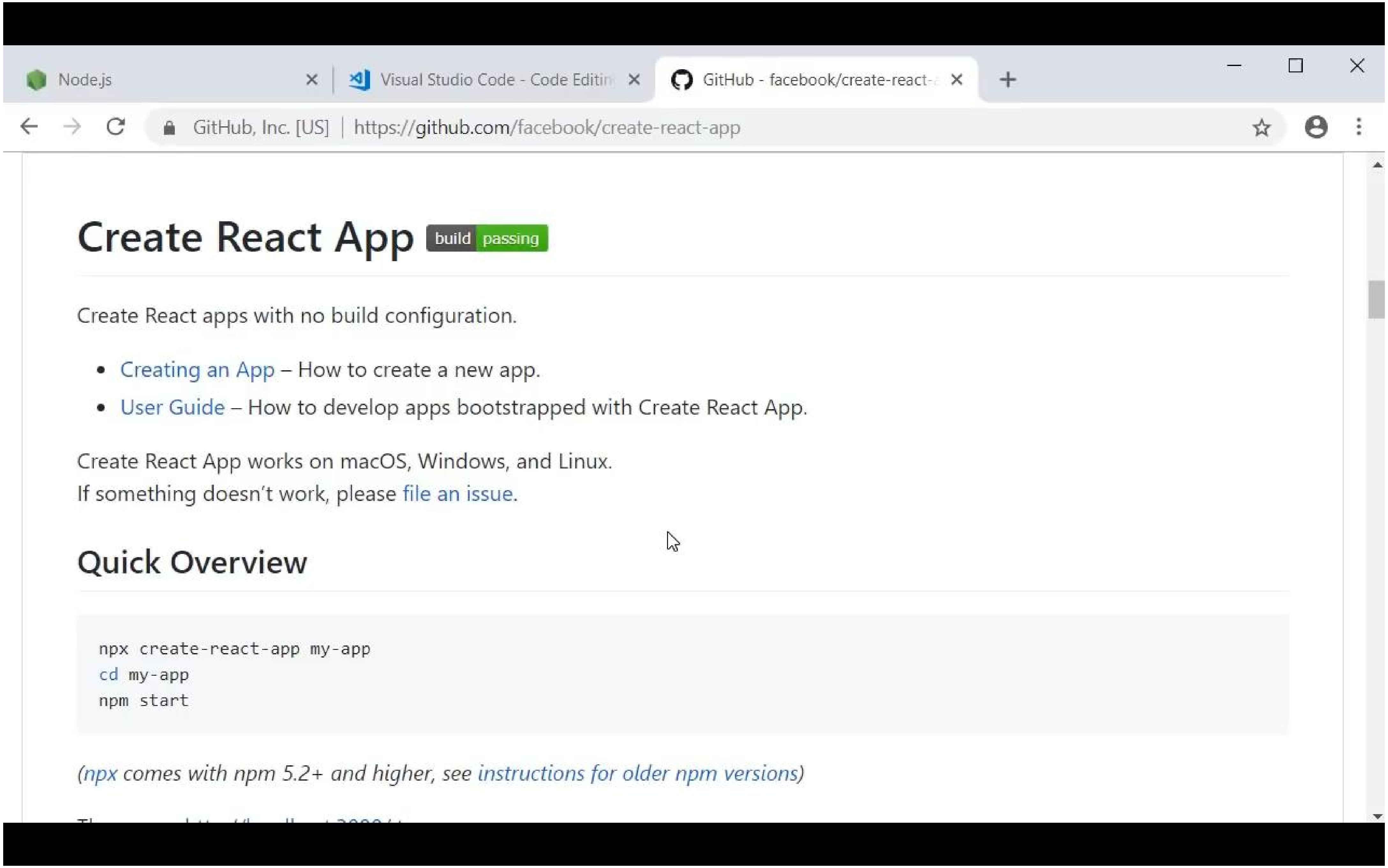
HTML, CSS and JavaScript fundamentals

ES6

JavaScript – ‘this’ keyword, filter, map and reduce

ES6 – let & const, arrow functions, template literals, default parameters, object literals, rest and spread operators and destructuring assignment.

React from scratch


Node.js
Visual Studio Code - Code Editin
GitHub - facebook/create-react-
GitHub, Inc. [US] | https://github.com/facebook/create-react-app

Create React App build passing

Create React apps with no build configuration.

- Creating an App – How to create a new app.
- User Guide – How to develop apps bootstrapped with Create React App.

Create React App works on macOS, Windows, and Linux.
If something doesn't work, please file an issue.

Quick Overview

```
npx create-react-app my-app
cd my-app
npm start
```

(npx comes with npm 5.2+ and higher, see instructions for older npm versions)

Create-react-app

npx

npx create-react-app <project_name>

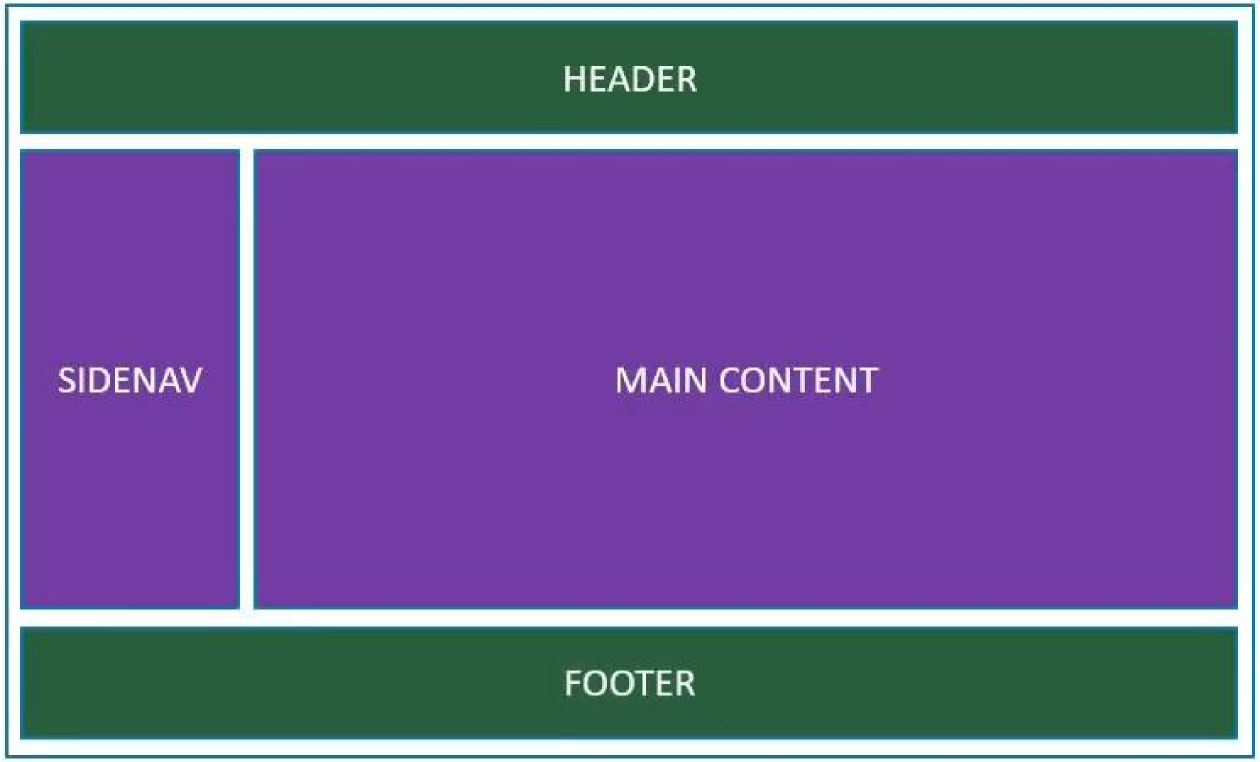
npm package runner

npm

npm install create-react-app -g

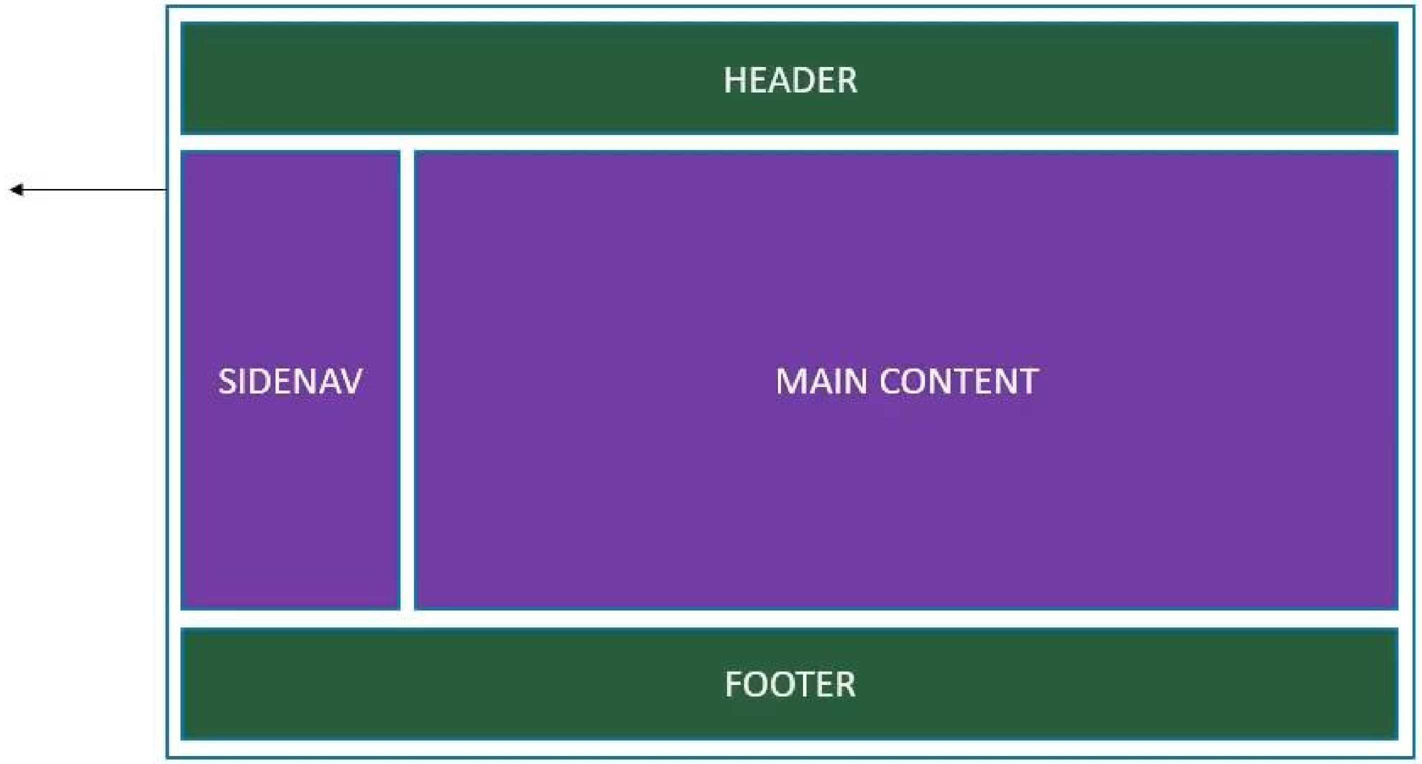
create-react-app<project_name>

Components

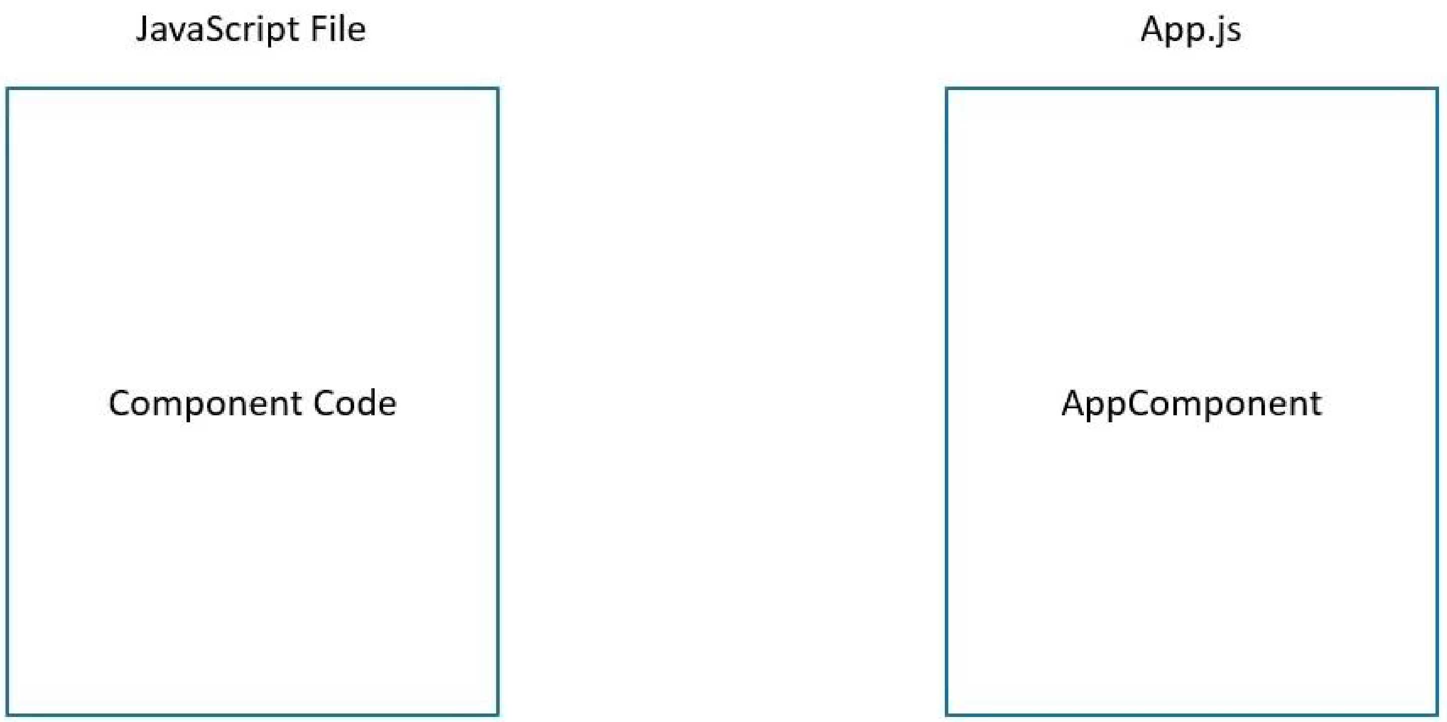


Components

Root (App) Component



Component in Code



Component Types

Stateless Functional Component

JavaScript Functions

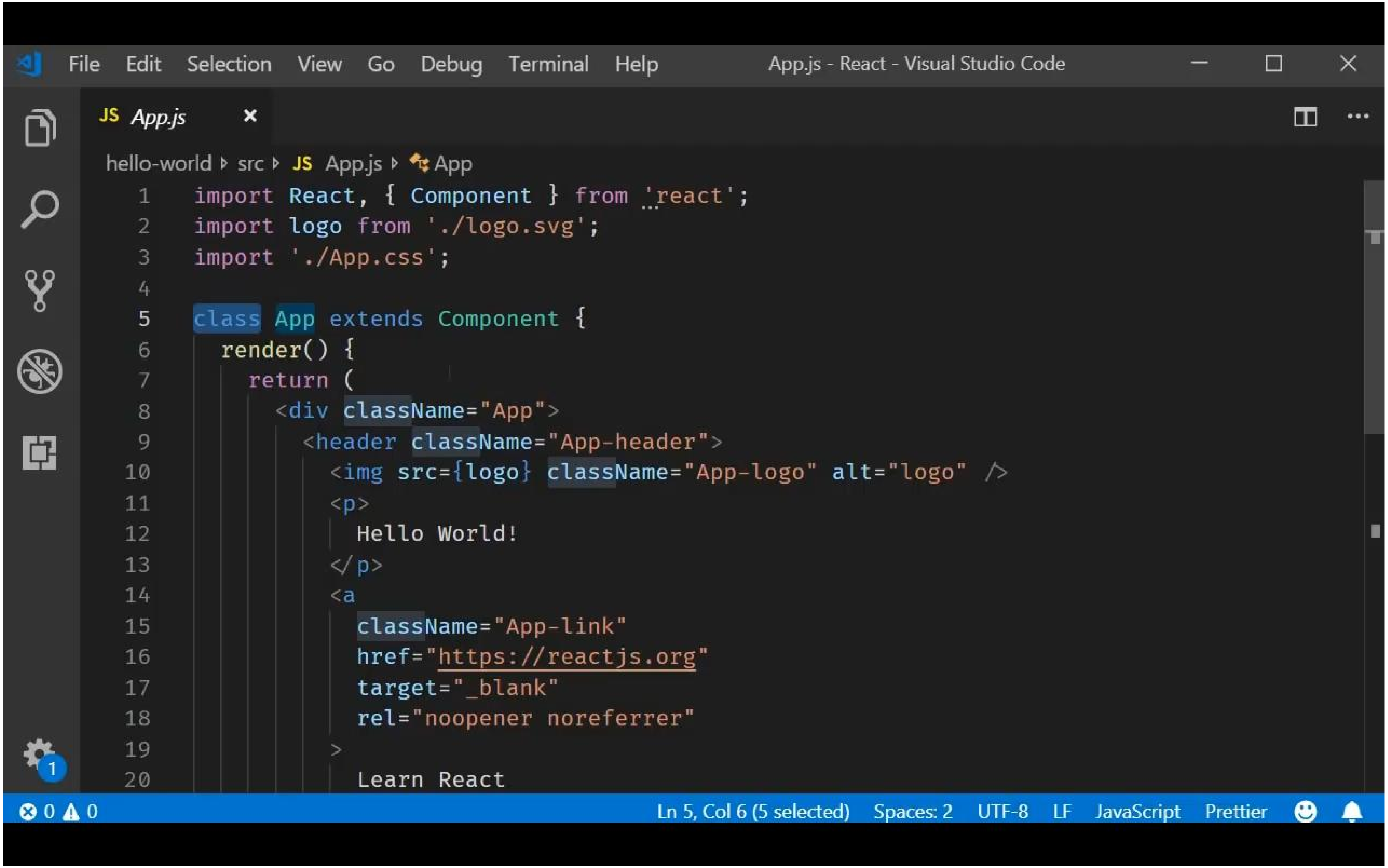
```
function Welcome(props) {
  return <h1>Hello, {props.name}</h1>;
}
```

Stateful Class Component

Class extending Component class

Render method returning HTML

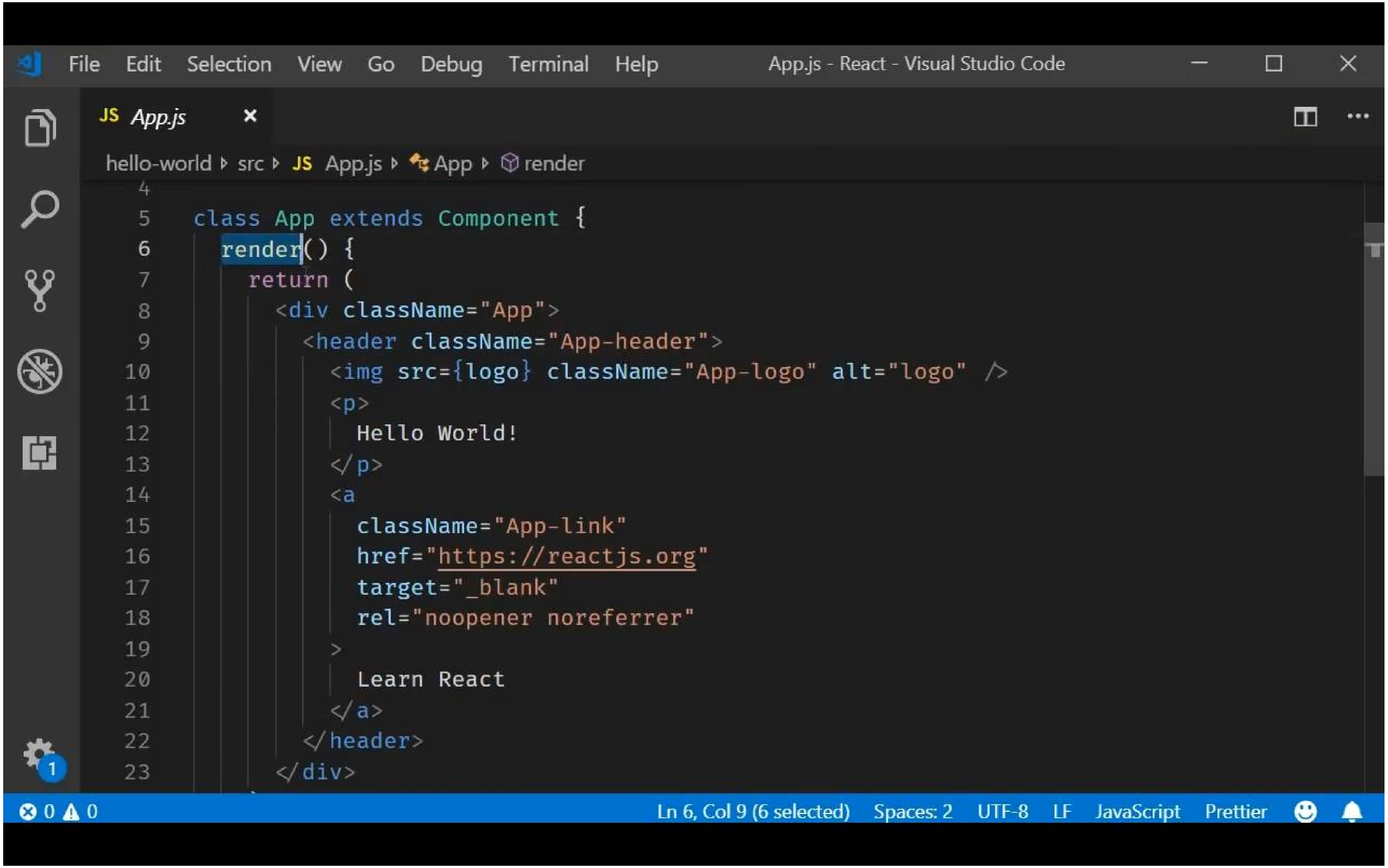
```
class Welcome extends React.Component {
  render() {
    return <h1>Hello, {this.props.name}</h1>;
  }
}
```


File Edit Selection View Go Debug Terminal Help
App.js - React - Visual Studio Code
App.js
hello-world › src › App.js › App

```
import React, { Component } from 'react';
import logo from './logo.svg';
import './App.css';

class App extends Component {
  render() {
    return (
      <div className="App">
        <header className="App-header">
          <img src={logo} className="App-logo" alt="logo" />
          <p>
            Hello World!
          </p>
          <a
            className="App-link"
            href="https://reactjs.org"
            target="_blank"
            rel="noopener noreferrer"
          >
            Learn React
```

0 0
Ln 5, Col 6 (5 selected) Spaces: 2 UTF-8 LF JavaScript Prettier


File
Edit
Selection
View
Go
Debug
Terminal
Help
App.js - React - Visual Studio Code
JS App.js
hello-world › src › JS App.js › App › render

```
class App extends Component {
  render() {
    return (
      <div className="App">
        <header className="App-header">
          <img src={logo} className="App-logo" alt="logo" />
          <p>
            Hello World!
          </p>
          <a
            className="App-link"
            href="https://reactjs.org"
            target="_blank"
            rel="noopener noreferrer"
          >
            Learn React
          </a>
        </header>
      </div>
```

Components Summary

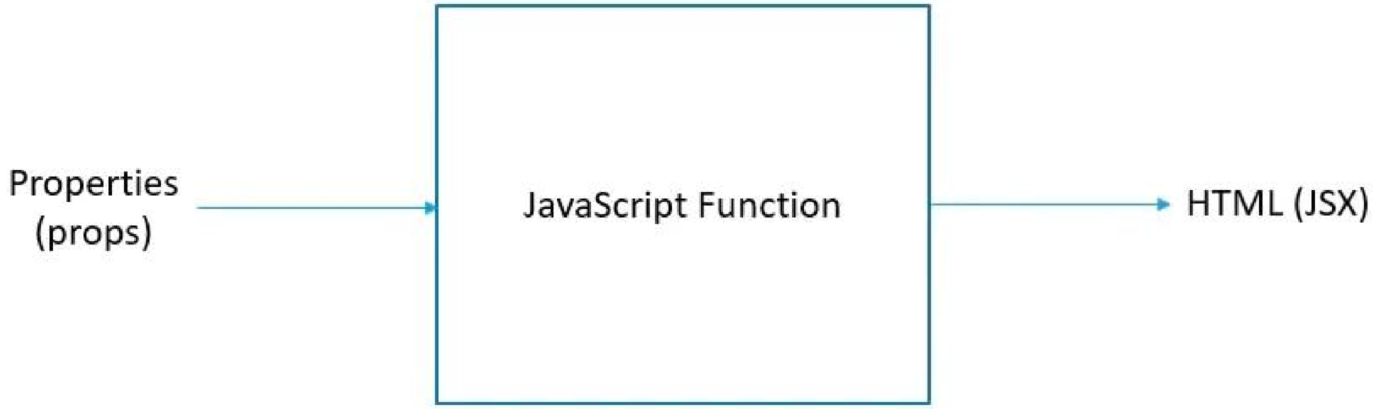
Components describe a part of the user interface

They are re-usable and can be nested inside other components

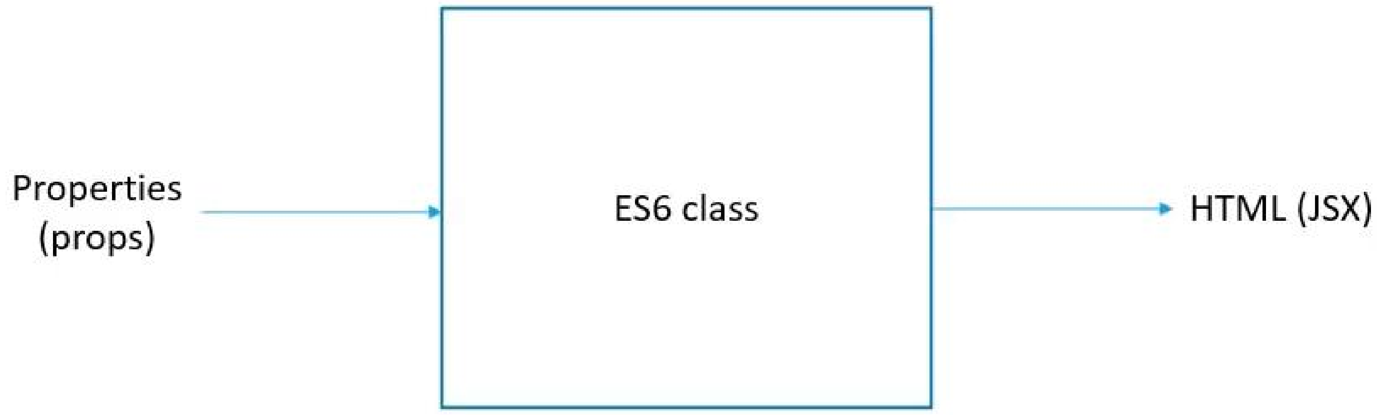
Two Types –

- Stateless Functional Components
- Stateful Class Components

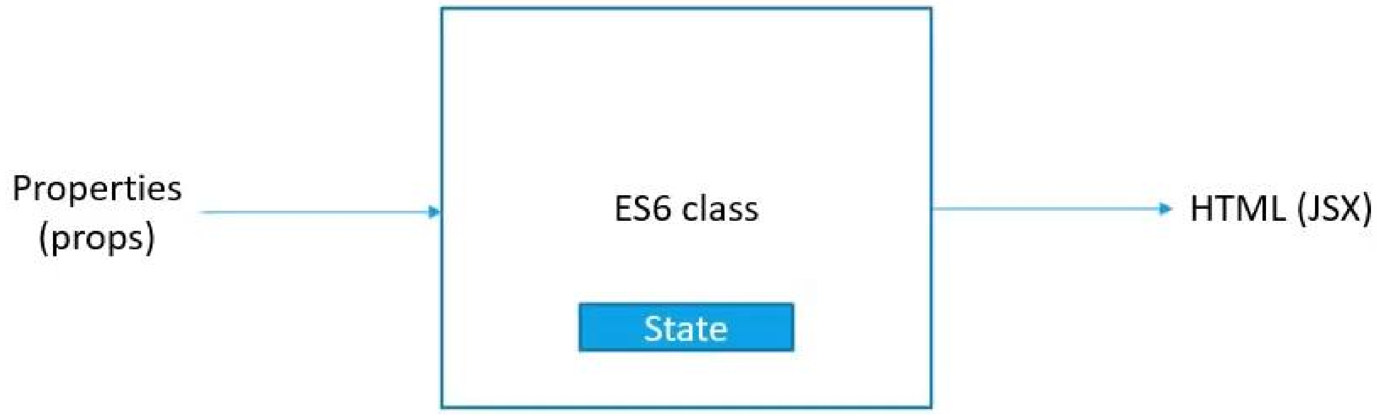
Functional Components



Class Components

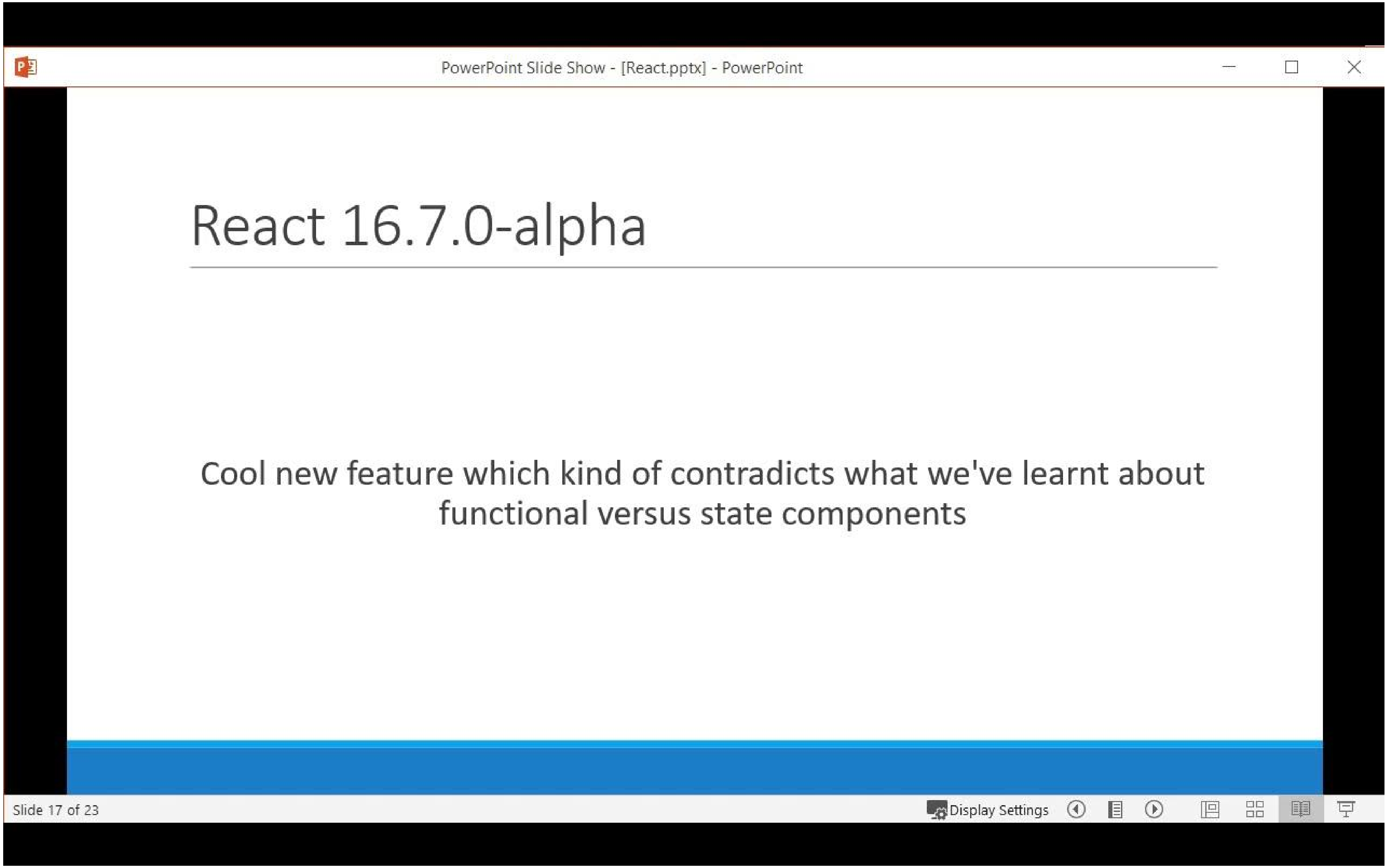


Class Components



Functional vs Class components

Functional	Class
Simple functions	More feature rich
Use Func components as much as possible	Maintain their own private data - state
Absence of 'this' keyword	Complex UI logic
Solution without using state	Provide lifecycle hooks
Mainly responsible for the UI	Stateful/ Smart/ Container
Stateless/ Dumb/ Presentational	

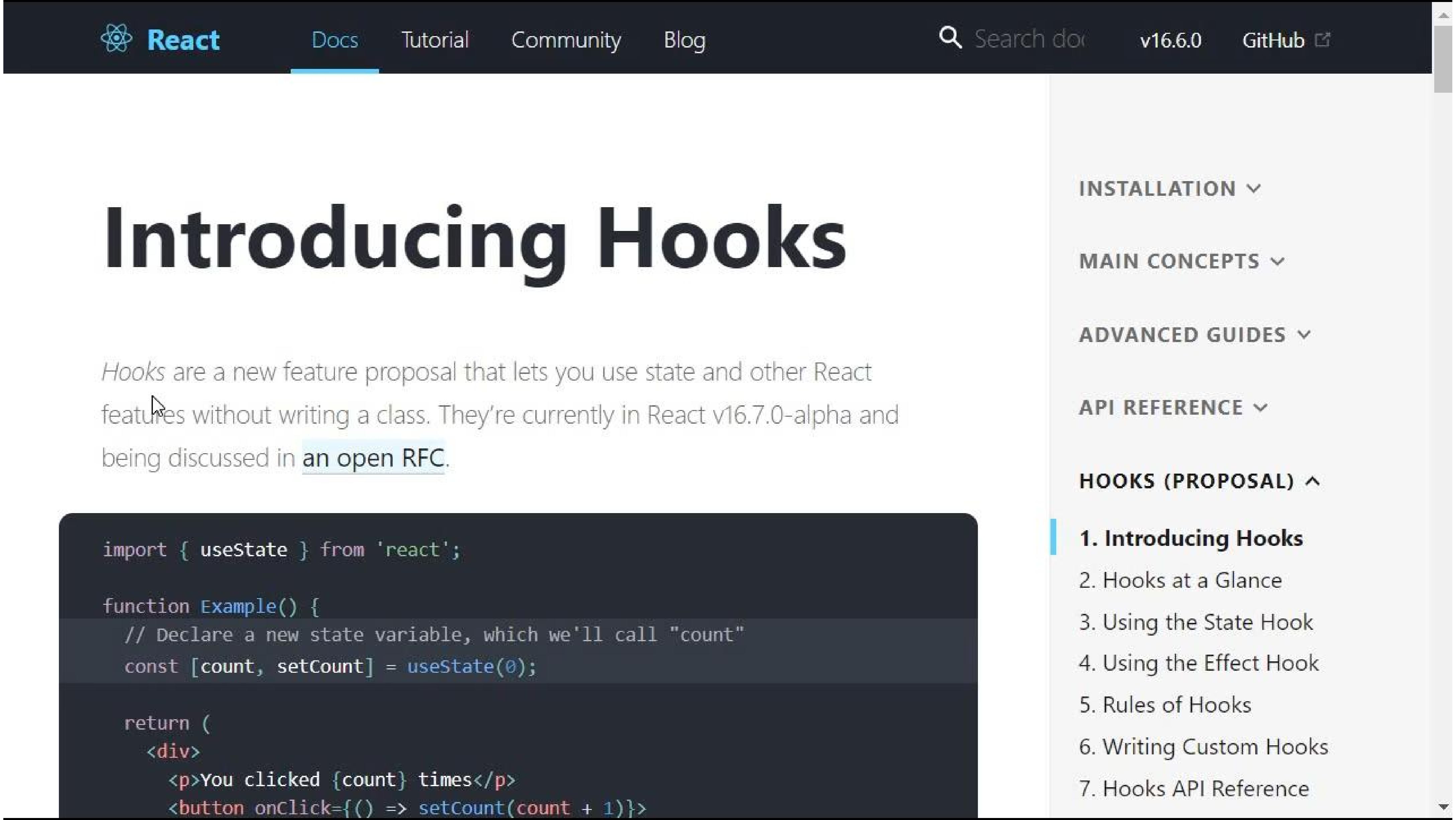


PowerPoint Slide Show - [React.pptx] - PowerPoint

React 16.7.0-alpha

Cool new feature which kind of contradicts what we've learnt about functional versus state components

Slide 17 of 23



Introducing Hooks

Hooks are a new feature proposal that lets you use state and other React features without writing a class. They're currently in React v16.7.0-alpha and being discussed in an open RFC.

```
import { useState } from 'react';

function Example() {
  // Declare a new state variable, which we'll call "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
```

JSX

JavaScript XML (JSX) – Extension to the JavaScript language syntax.

Write XML-like code for elements and components.

JSX tags have a tag name, attributes, and children.

JSX

JavaScript XML (JSX) – Extension to the JavaScript language syntax.

Write XML-like code for elements and components.

JSX tags have a tag name, attributes, and children.

JSX is not a necessity to write React applications.

JSX makes your react code simpler and elegant.

JSX ultimately transpiles to pure JavaScript which is understood by the browsers.

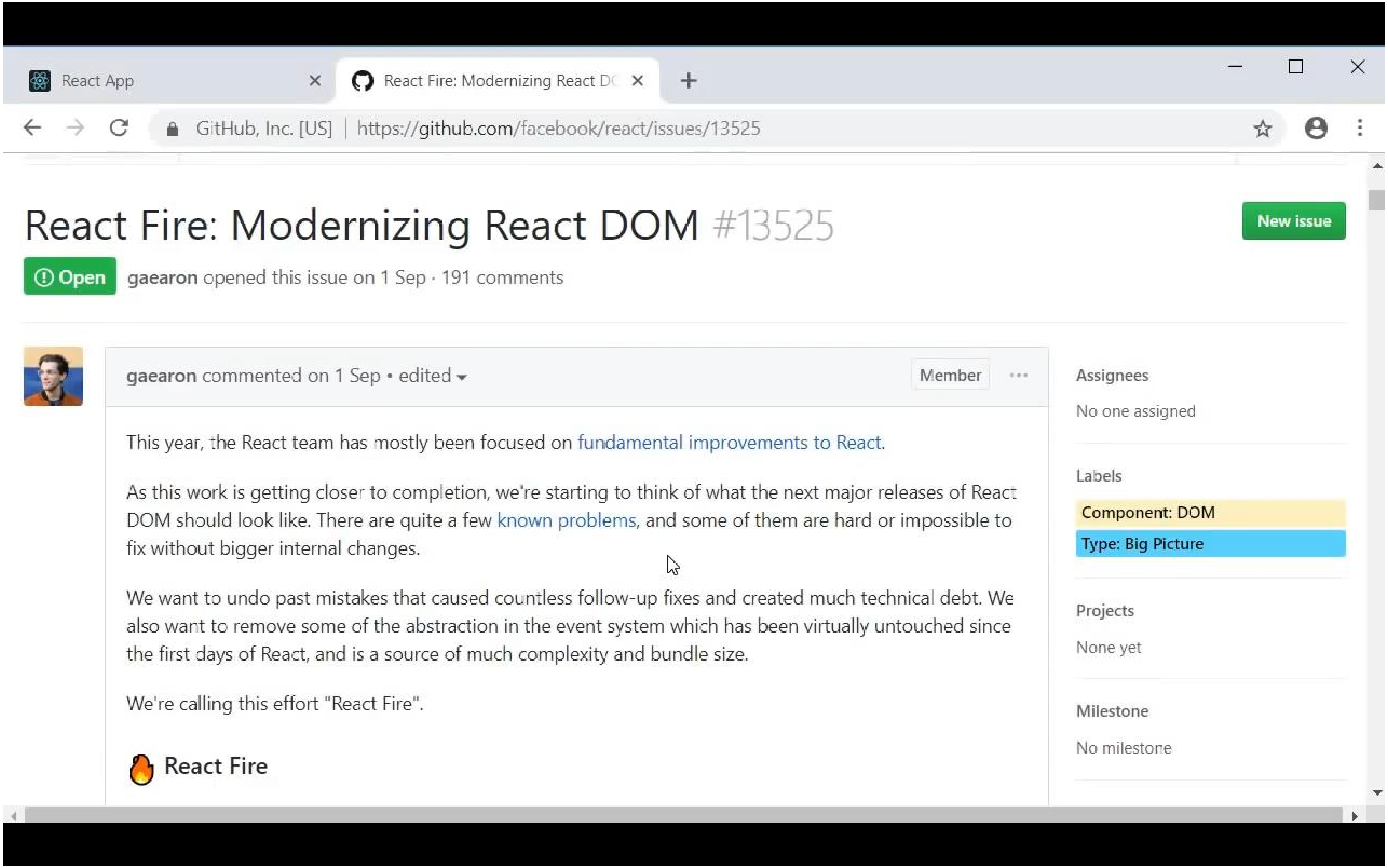
JSX differences

Class -> className

for -> htmlFor

camelCase property naming convention

- onclick -> onClick
- tabindex -> tabIndex


React App
React Fire: Modernizing React DC
GitHub, Inc. [US] | https://github.com/facebook/react/issues/13525
New issue

React Fire: Modernizing React DOM #13525

Open gaearon opened this issue on 1 Sep · 191 comments

gaearon commented on 1 Sep • edited Member

This year, the React team has mostly been focused on fundamental improvements to React.

As this work is getting closer to completion, we're starting to think of what the next major releases of React DOM should look like. There are quite a few known problems, and some of them are hard or impossible to fix without bigger internal changes.

We want to undo past mistakes that caused countless follow-up fixes and created much technical debt. We also want to remove some of the abstraction in the event system which has been virtually untouched since the first days of React, and is a source of much complexity and bundle size.

We're calling this effort "React Fire".

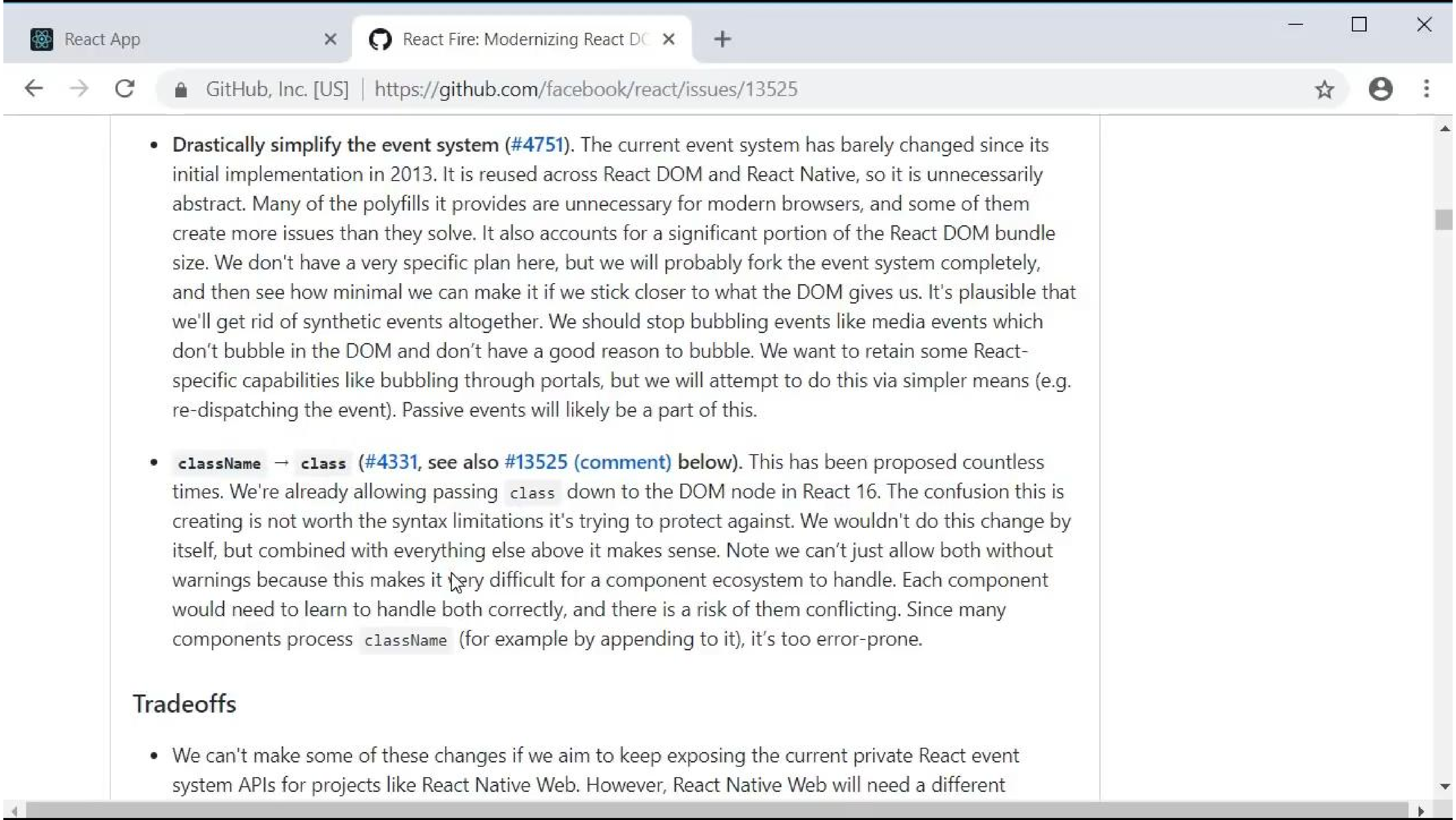
🔥 React Fire

Assignees
No one assigned

Labels
Component: DOM
Type: Big Picture

Projects
None yet

Milestone
No milestone


React App
React Fire: Modernizing React D
GitHub, Inc. [US] | https://github.com/facebook/react/issues/13525

- **Drastically simplify the event system (#4751).** The current event system has barely changed since its initial implementation in 2013. It is reused across React DOM and React Native, so it is unnecessarily abstract. Many of the polyfills it provides are unnecessary for modern browsers, and some of them create more issues than they solve. It also accounts for a significant portion of the React DOM bundle size. We don't have a very specific plan here, but we will probably fork the event system completely, and then see how minimal we can make it if we stick closer to what the DOM gives us. It's plausible that we'll get rid of synthetic events altogether. We should stop bubbling events like media events which don't bubble in the DOM and don't have a good reason to bubble. We want to retain some React-specific capabilities like bubbling through portals, but we will attempt to do this via simpler means (e.g. re-dispatching the event). Passive events will likely be a part of this.

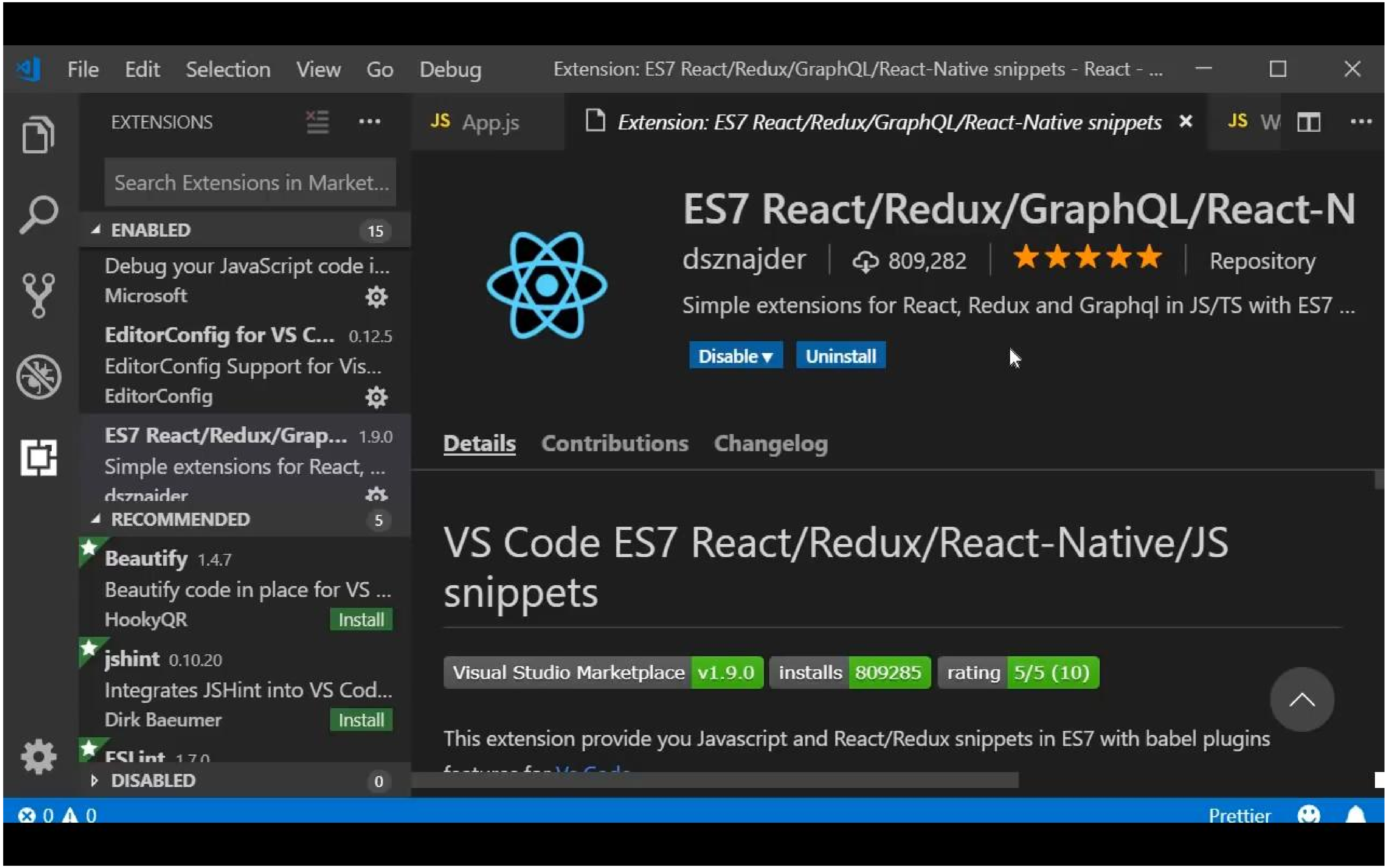
- **`className` → `class` (#4331, see also #13525 (comment) below).** This has been proposed countless times. We're already allowing passing `class` down to the DOM node in React 16. The confusion this is creating is not worth the syntax limitations it's trying to protect against. We wouldn't do this change by itself, but combined with everything else above it makes sense. Note we can't just allow both without warnings because this makes it very difficult for a component ecosystem to handle. Each component would need to learn to handle both correctly, and there is a risk of them conflicting. Since many components process `className` (for example by appending to it), it's too error-prone.

Tradeoffs

- We can't make some of these changes if we aim to keep exposing the current private React event system APIs for projects like React Native Web. However, React Native Web will need a different

props vs state

props	state
props get passed to the component	state is managed within the component
Function parameters	Variables declared in the function body
props are immutable	state can be changed
props – Functional Components this.props – Class Components	useState Hook – Functional Components this.state – Class Components


File
Edit
Selection
View
Go
Debug
Extension: ES7 React/Redux/GraphQL/React-Native snippets - React - ...
EXTENSIONS
JS App.js
Extension: ES7 React/Redux/GraphQL/React-Native snippets
Search Extensions in Market...
ENABLED 15
Debug your JavaScript code i...
Microsoft
EditorConfig for VS C... 0.12.5
EditorConfig Support for Vis...
EditorConfig
ES7 React/Redux/Grap... 1.9.0
Simple extensions for React, ...
dsznajder
RECOMMENDED 5
Beautify 1.4.7
Beautify code in place for VS ...
HookyQR
Install
jshint 0.10.20
Integrates JSHint into VS Cod...
Dirk Baeumer
Install
ESLint 1.7.0
DISABLED 0
Disable ▾
Uninstall
Details
Contributions
Changelog
Prettier

ES7 React/Redux/GraphQL/React-N

dsznajder | 809,282 | ★★★★★ | Repository

Simple extensions for React, Redux and Graphql in JS/TS with ES7 ...

VS Code ES7 React/Redux/React-Native/JS snippets

Visual Studio Marketplace v1.9.0 | installs 809285 | rating 5/5 (10)

This extension provide you Javascript and React/Redux snippets in ES7 with babel plugins

setState

Always make use of setState and never modify the state directly.

Code has to be executed after the state has been updated ? Place that code in the call back function which is the second argument to the setState method.

When you have to update state based on the previous state value, pass in a function as an argument instead of the regular object.