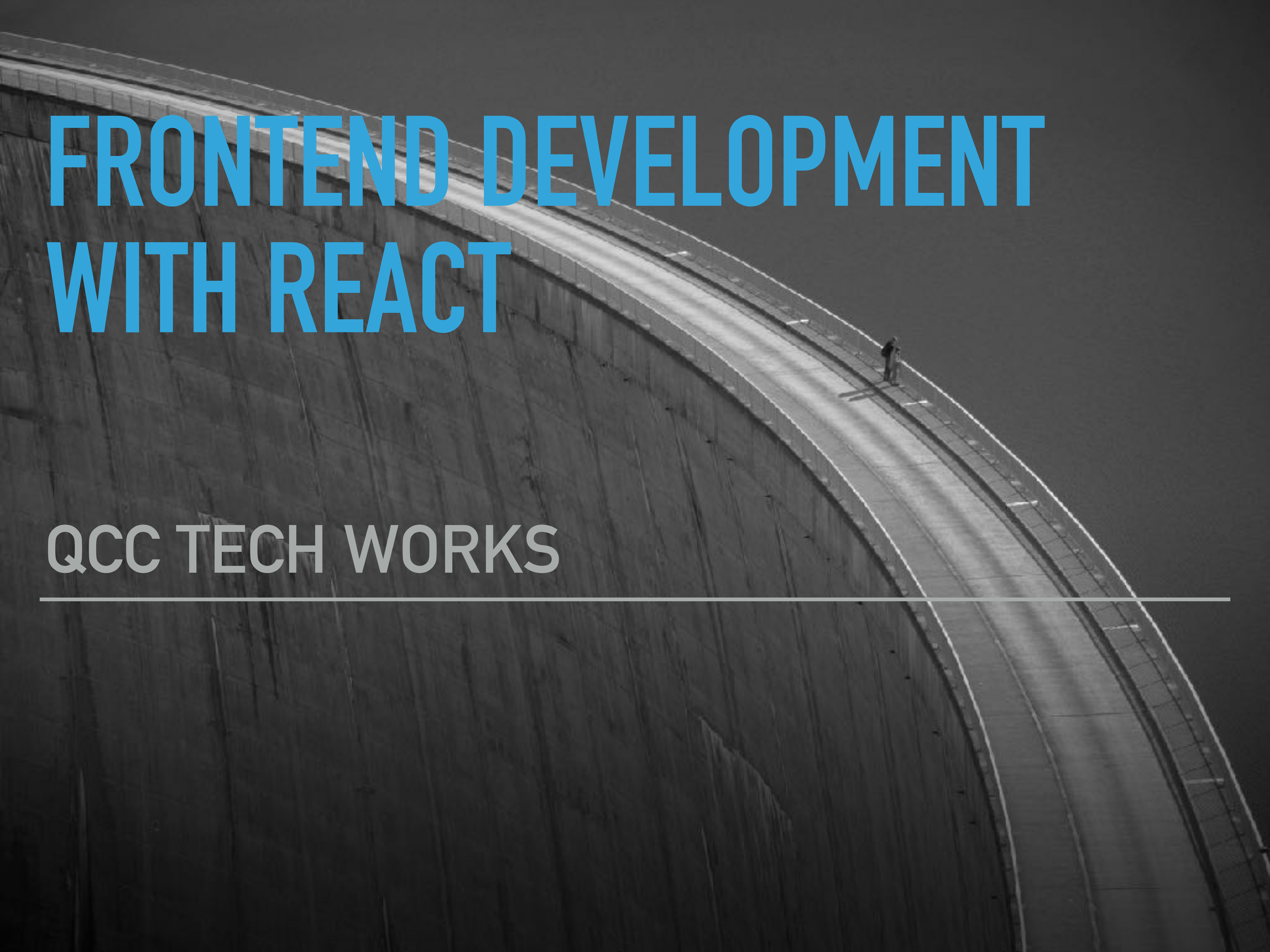
FRONTEND DEVELOPMENT WITH REACT

QCC TECH WORKS



COURSE OUTLINE

- Goals:
  - Understand what React is, and what problems it solves.
  - Gain a deeper knowledge of JSX, and Javascript (ES2015 syntax).
  - Learn to build React Components.
  - Learn to use the create-react-app module.

WEEK #1: AN INTRODUCTION TO REACT

- Topics we will cover:
    - An introduction to how React is used to build UI's
    - Use JSX to build React Components
        - Stateful and Function Components
    - Learn about Props and State
    - Arrow Functions

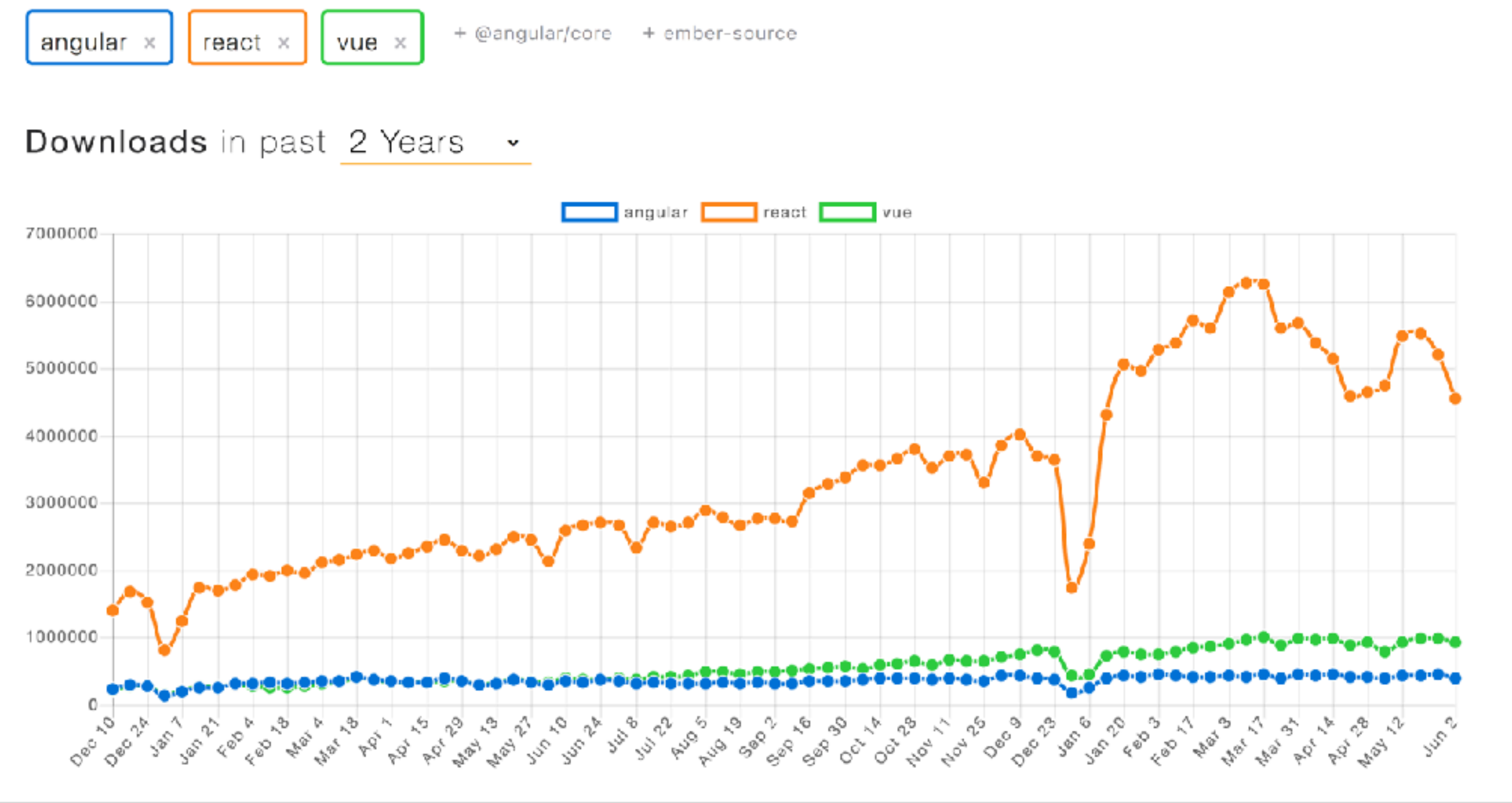
WHAT IS REACT?

- Officially - React is a JavaScript library developed by Facebook in 2011 for building user interfaces.
- Unofficially - React is a SWEEET JavaScript library used to develop responsive web pages.

WHAT DOES REACT DO?

- React renders your UI and responds to events.
- AKA - the V in MVC.
- Plays nicely with your stack.

IS REACT POPULAR?



Downloads in past 2 Years

WHO USES REACT?

- BBC
- Codeacademy
- Dropbox
- Facebook
- Instagram
- Github
- Reddit
- Venmo
- Salesforce
- WhatsApp
- Wired
- PayPal
- Netflix
- Me

Sites using React:

http://github.com/facebook/react/wiki/Sites-Using-React

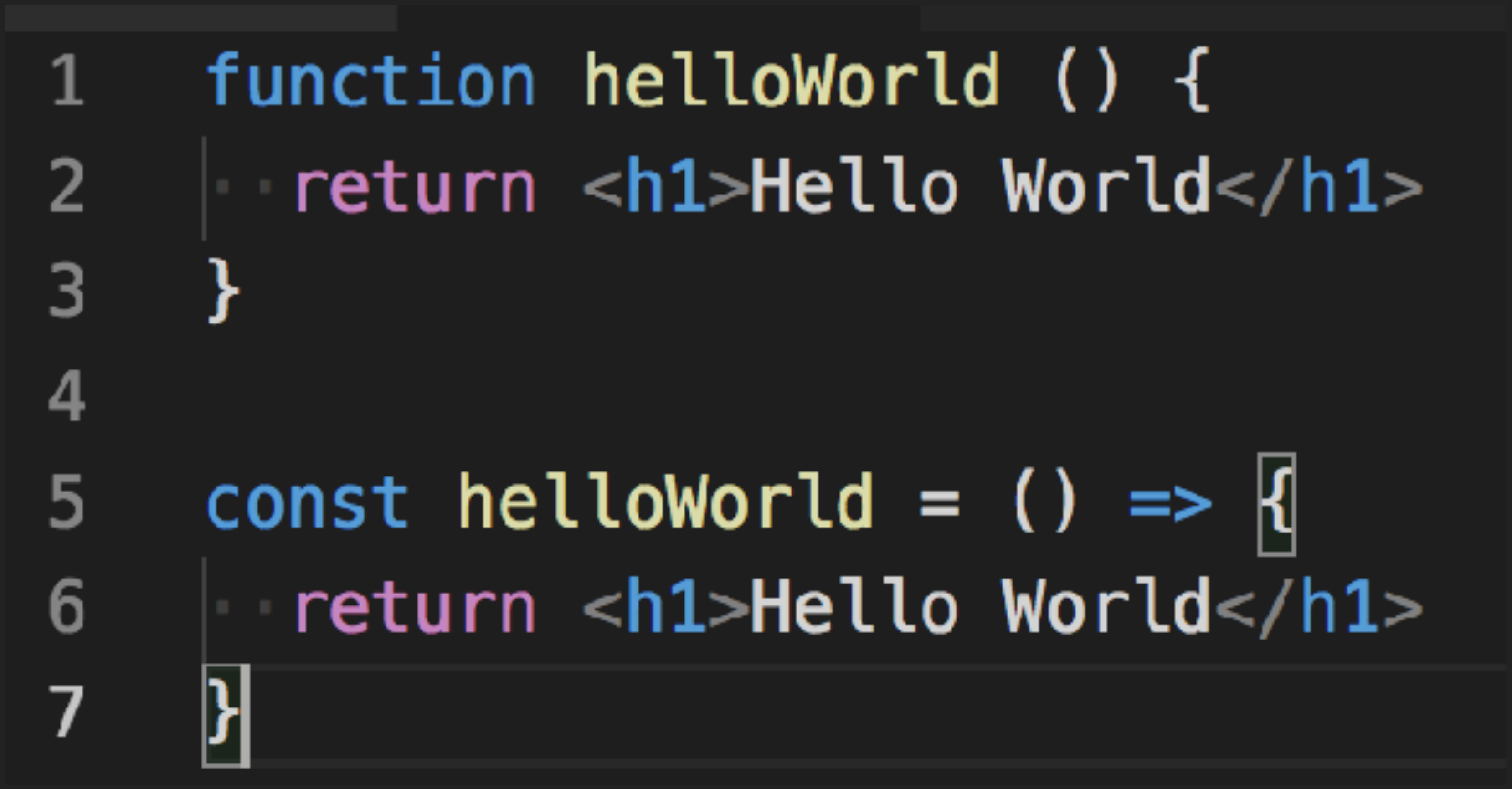
WHY USE REACT?

1. It facilitates the overall process of writing components
2. It boosts productivity and facilitates further maintenance
3. It ensures faster rendering
4. It guarantees stable code
5. It is SEO friendly
6. It comes with a helpful developer toolset
7. There is React Native for mobile app development
8. It is focused and easy-to-learn
9. It is backed by a strong community
10. It is used by both Fortune 500 companies and innovative startups

– Top 10 Advantage s of Using React

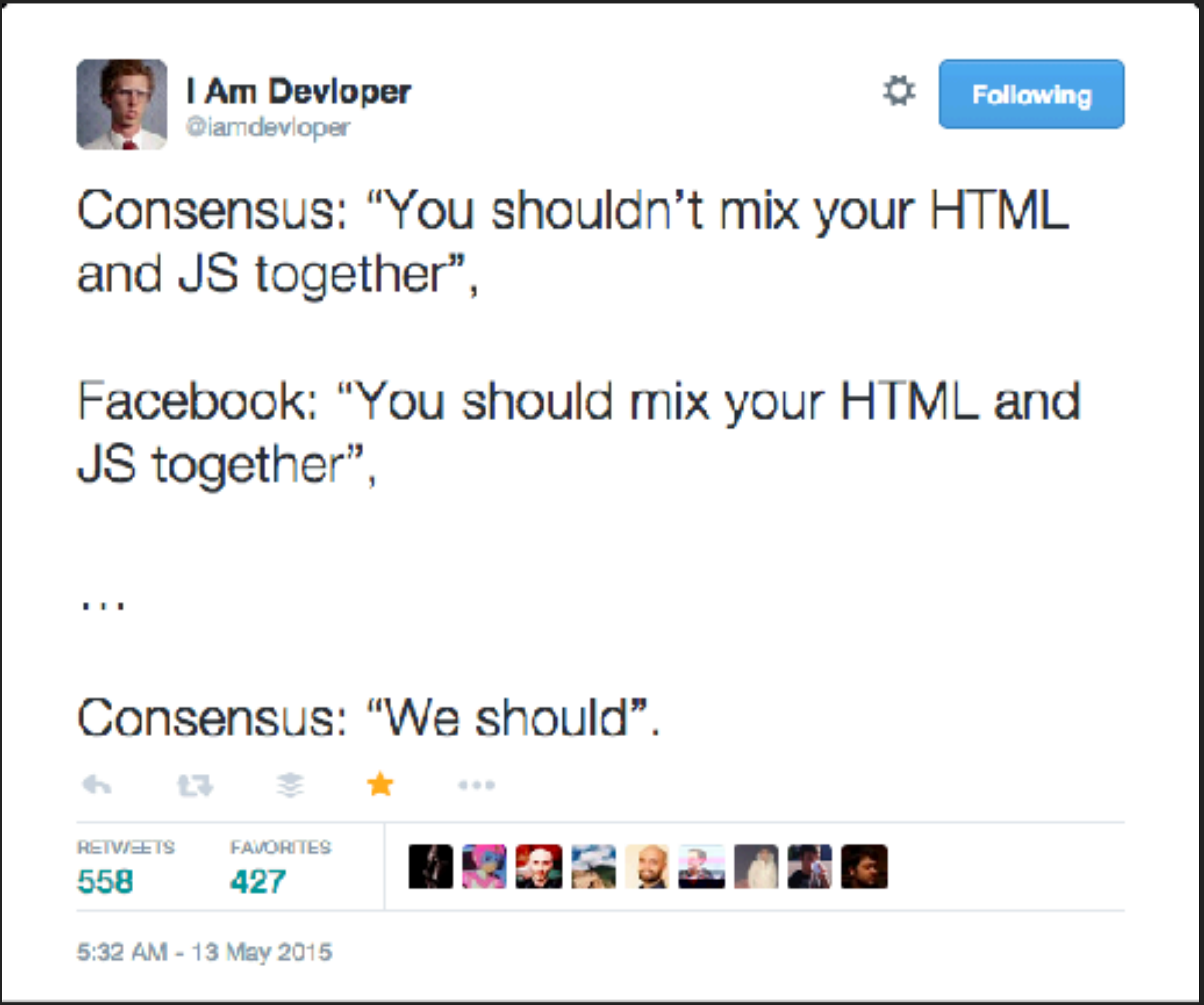
WHAT IS JSX

- Stands for JavaScript Extensions
- Allow us to write markup in JavaScript



```
function helloWorld () {
  return <h1>Hello World</h1>
}

const helloWorld = () => {
  return <h1>Hello World</h1>
}
```

WHAT!?!? MARKUP IN MY JAVASCRIPT?!?!



I Am Devloper
@iamdevloper

Consensus: “You shouldn’t mix your HTML and JS together”,

Facebook: “You should mix your HTML and JS together”,

…

Consensus: “We should”.

RETWEETS 558

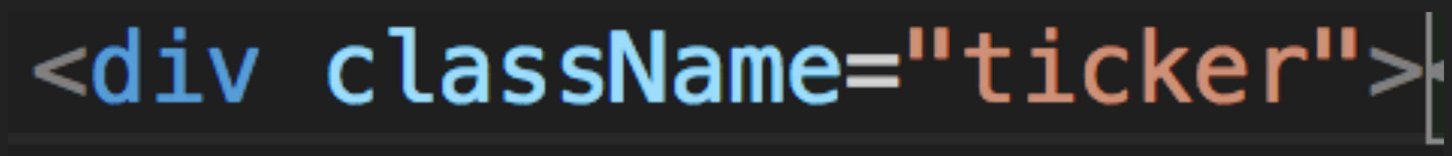
FAVORITES 427

5:32 AM - 13 May 2015

JSX

- This allows us to focus on building components, not templates.
- This also allows us to reduce context switching by combining markup and JavaScript.
- In the end JSX complies down to basic JavaScript via the bable complier.

Note: A subtile difference of JSX and HTML is you have to replace the “class ” attribute with the “className” because class is a reserved word in JavaScript.



```
<div className="ticker">
```

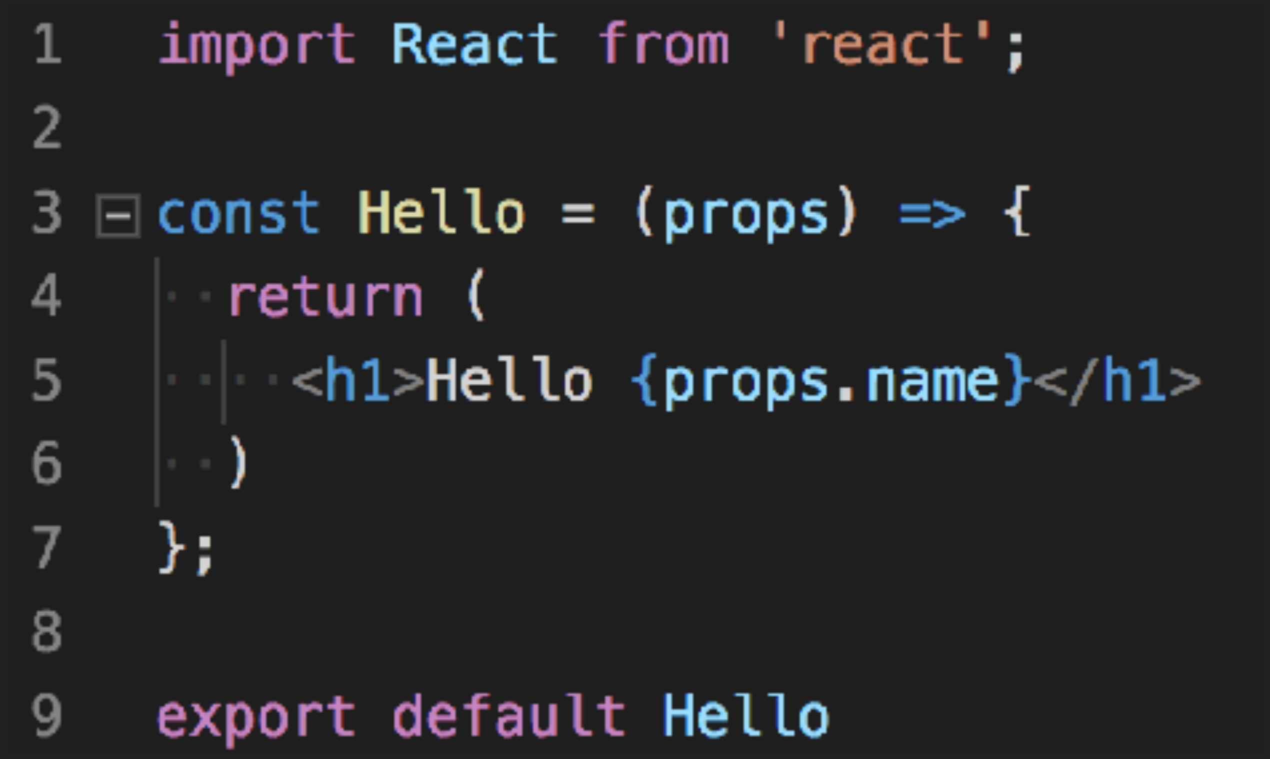
COMPONENTS

- Components are the fundamental building blocks for React.
- They can have internal state and external props
- They can also nest inside one another, as we as being used across multiple files and even projects

You can compare components with lego blocks. We use them as building blocks to build a bigger meaningful application

TWO MAIN TYPES OF COMPONENTS

- Functional Components - A.K.A. a Stateless component.

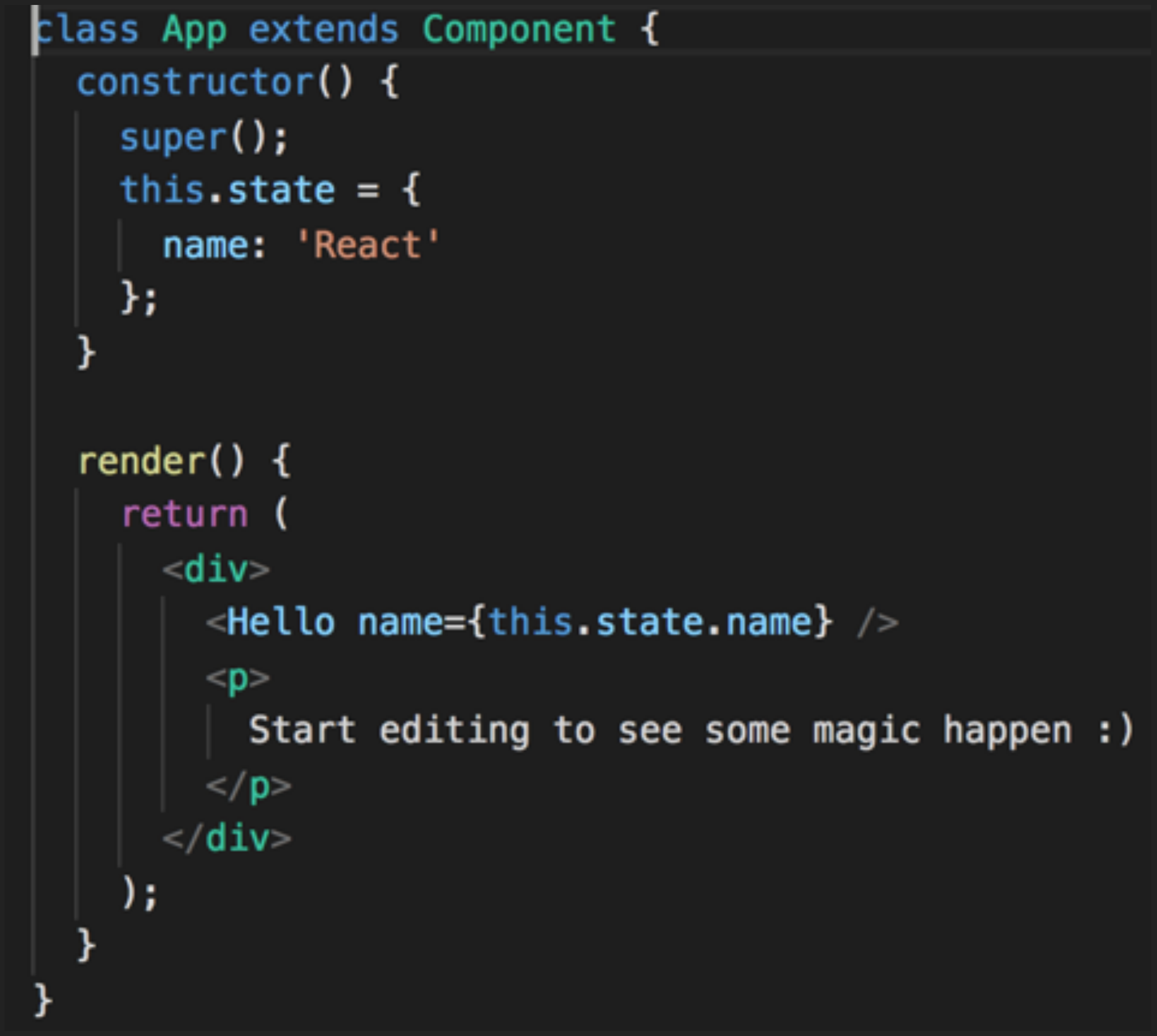


```
import React from 'react';

const Hello = (props) => {
  return (
    <h1>Hello {props.name}</h1>
  )
};

export default Hello
```

> This function is a valid React component because it accepts a single "props" (which stands for properties) object argument with data and returns a React element. We call such components "function components" because they are literally JavaScript functions.
>
> —React Documentation

COMPONENTS (CONTINUED)

- Class Component - A.K.A. Stateful component

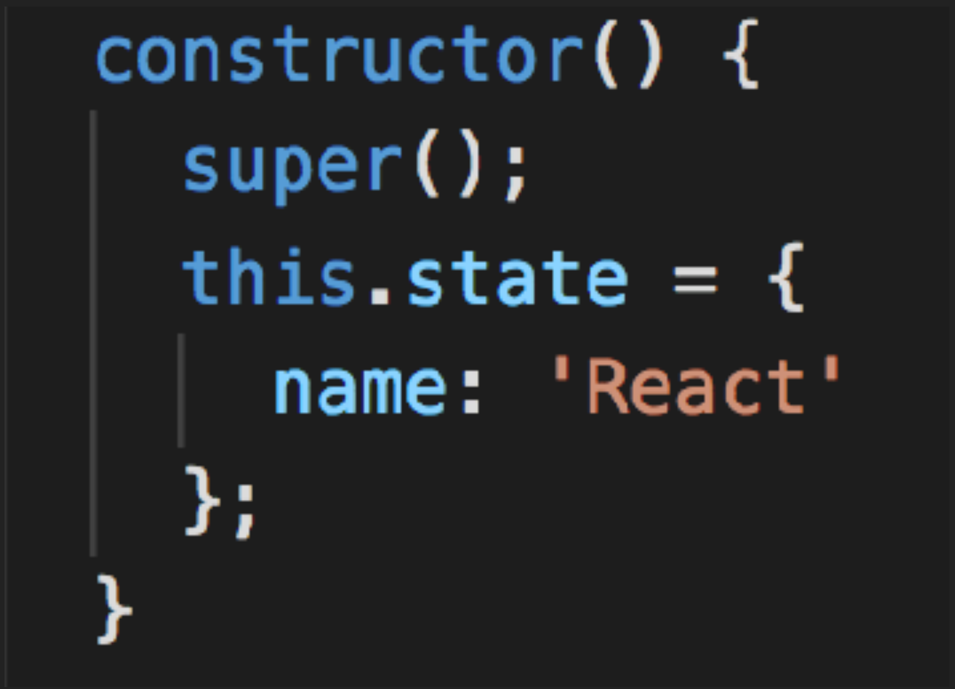


```
class App extends Component {
  constructor() {
    super();
    this.state = {
      name: 'React'
    };
  }

  render() {
    return (
      <div>
        <Hello name={this.state.name} />
        <p>
          Start editing to see some magic happen :)
        </p>
      </div>
    );
  }
}
```

These are your traditional React Components. Specifically when you need a component that needs a lifecycle method and/or State you want to use this type of component.

STATE

- Data is called “state”
  - More precisely “state” is data that changes
  - State is always an object with key value pairs
  - Is always initialized in the constructor
- When you update “state”, React re-renders the view for you
- React re-renders in a performant way. By using a riffing algorithm.
- State is a keyword.



```
constructor() {
  super();
  this.state = {
    name: 'React'
  };
}
```

RESOURCES

https://reactjs.org/docs/getting-started.html
https://reactjs.org/docs/introducing-jsx.html

GitHub Link to the workshop and files

https://github.com/CaptainKRS/lessons