Frontend Development with React

Kirk Sullivan

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.
- Gain a better understanding of the DOM and Virtual DOM.
- Learn to build React application using Create-react-app boilerplate.
- Build and deploy a personal resume website.

Week #1: An Introduction to React

Topics we will cover:

- Be introduced on how React is used to build UI's
- Learn about the Virtual DOM
- Use JSX to build React Components
- Learn how react uses a one-way data flow
- Use ES6 Arrow Functions
- Gain familiarity with the SetState function
- Create Stateful Components
- Create Functional Components
- Learn about Props & State

Key takeaways for this lecture

- **What is React**
- **JSX**
- **Components**
- **Props**
- **State**
- **render()**
- **Virtual DOM**

What is React?

Officially - React is a JavaScript library developed by Facebook in 2011 for building user interfaces.

Unofficially - React is a SWEEET JS library that is used to develop responsive web pages.

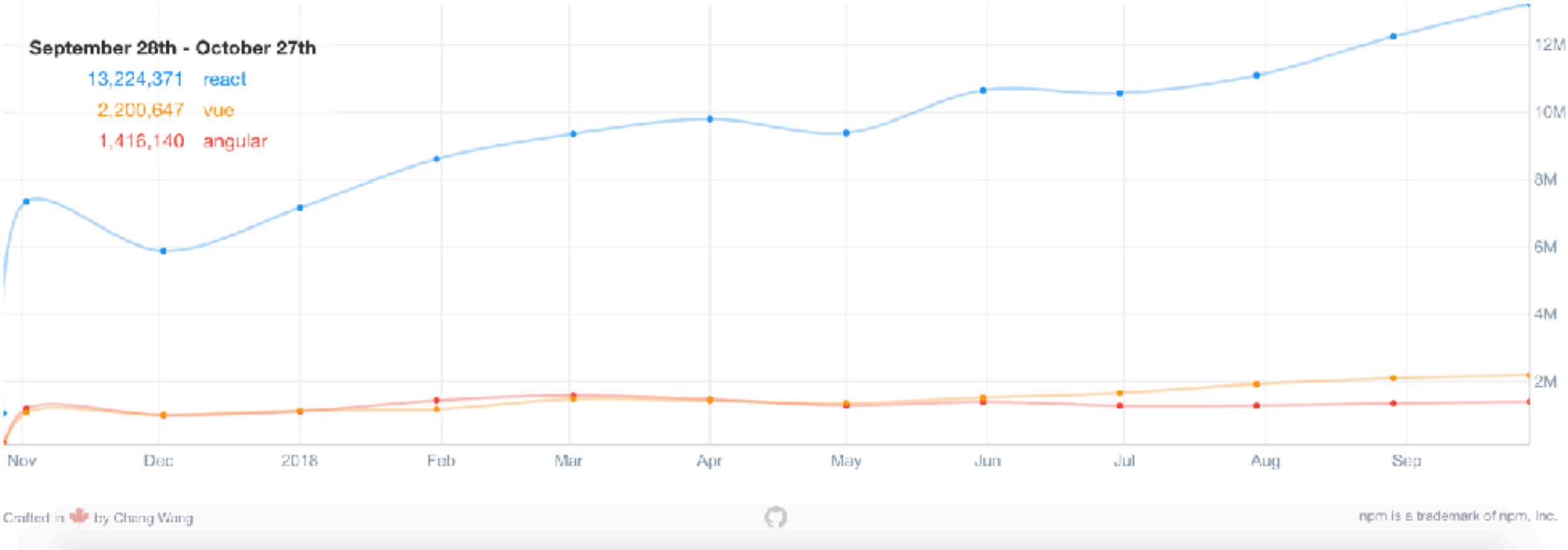
What does React do?

React renders your UI and responds to events.

AKA: The V in the MVC

Plays nicely with your stack

Is React Popular?



Who uses React?

- **BBC**
- **Codeacademy**
- **Dropbox**
- **Facebook**
- **Flipboard**
- **Github**
- **Imgur**
- **Instagram**
- **Khan Academy**
- **Netflix**
- **Paypal**
- **Reddit**
- **Salesforce**
- **Venmo**
- **WhatsApp**
- **Wired**

Sites Using React:
https://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 Advantages of Using React

JSX

- **Stands for Javascript Extension**
- **Allows us to write markup in Javascript**

```
const HelloWorld = () => {
  return <h1>Hello World!</h1>;
};
```

```
function HelloWorld() {
  return <h1>Hello World! Yay!!</h1>;
}
```

What markup in my Javascript!?!?

But Seriously

- **This allow us to focus on building components, not templates.**
- **This also allow us to reduce context switching by combining markup and Javascript.**
- **In the end JSX compiles down to basic Javascript via the babel complier.**

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

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

<u>React DOM Elements</u>

Components

Super Technical Explanation

Components are Just State Machines

React thinks of UIs as simple state machines. By thinking of a UI as being in various states and rendering those states, it's easy to keep your UI consistent.

In React, you simply update a component's state, and then render a new UI based on this new state. React takes care of updating the DOM for you in the most efficient way. -- Interactivity and Dynamic UIs

Less Technical

- **Components are the fundamental building block for React**
- **Have internal state and external props**
- **Can be nested inside each other and 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

Class Component
AKA 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.

Two Main Types of Components (continued)

Functional Component
AKA a Stateless Component

```
1   import React from 'react';
2
3   const Hello = ({name}) => {
4     return (
5       <h1>Hello {name}</h1>
6     )
7   }
8
9   export default Hello;
10
```

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

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 super duper diffing algorithm.**
- **State is a keyword.**

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

Props

Think of props as “options” passed to a component to customize its functionality.

Props are conceptually and syntactically very similar to an HTML property.

All Props are passed into a component become key-value pairs on that component’s “props” object.

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

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

render()

Think of the render method as your template

- **Called whenever the state changes**
- **Here we decide what the user should see based on the state**
- **Notice that even though we are calling render on every state change, not everything is getting reloaded on the page. The entire DOM is not re-rendering**

```
14    render() {
15      return (
16        <div>
17          <Hello name={this.state.name} />
18          <p>
19            Start editing to see some magic happen :)
20          </p>
21        </div>
22      );
23    }
24  }
```

Virtual DOM

Think of the virtual DOM as React's local and simplified copy of the HTML DOM. It allows React to do its computations within this abstract world and skip the "real" DOM operations, often slow and browser-specific.

—React KungFu

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/Voodoobrew/workshop-1