

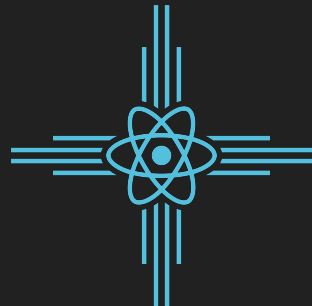
505 React

Meetup 1

Before we begin

Install Node.js for your operating system

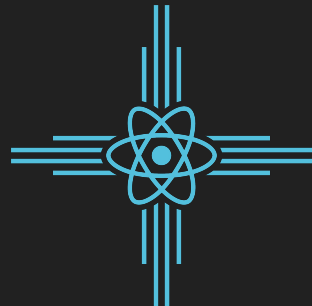
<https://nodejs.org/en/>



Welcome to 505 React

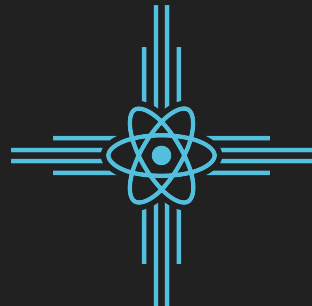
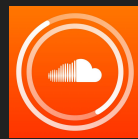
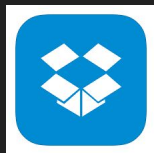
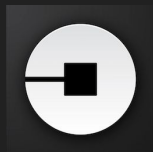
Tell us a little about yourself.

- Name
- Where you work
- What you know about React / React Native already
- What you want to know about React / React Native



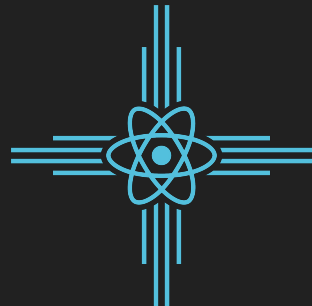
What exactly is React

- It is a JavaScript LIBRARY not a framework
- Developed by Facebook and used internally until being released in 2013
- Learn once write ANYWHERE (React and React Native work similarly and work on both platforms)



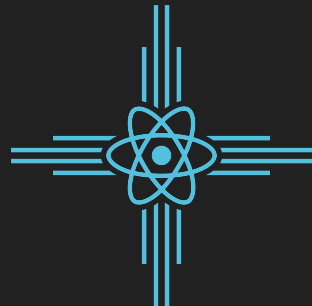
Why use React?

- It uses a **Virtual DOM** (React not React Native)
- Allows for rendering and management of data that changes **state** frequently
- **Component** based architecture makes applications highly scalable
- Utilizes **JSX** syntax (ES6, ES7, ES8)
- Allows for integrations with various other libraries and components through **npm** and **yarn**



React Pieces

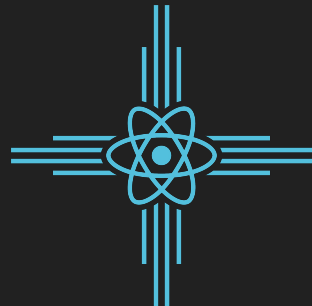
- Components
- State/Props
- Redux (sort of)



React Components

- Lifecycle
 - render()
 - componentDidMount()
- render() method is where all your UI code goes
- Example Class

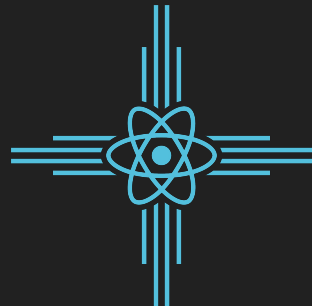
```
class 505ReactWidget extends React.Component {  
  render() {  
    return <h1>Hello, {this.props.name}</h1>;  
  }  
}
```



React Props and State

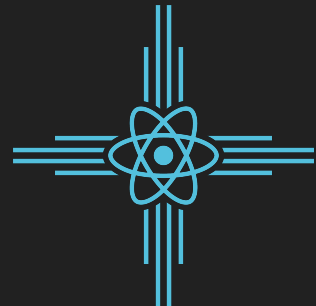
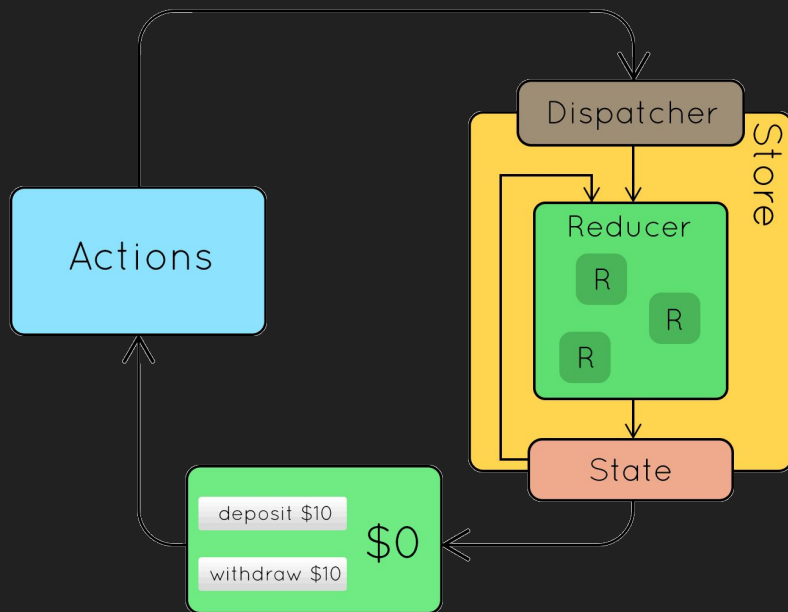
- Props/State both store data that you components need to render
- Props are passed in and State is local to your component
- Props can be used to pass both data and functions to child components

```
class Foo extends Component {  
  constructor(props) {  
    super(props);  
    this.state = {date: new Date()};  
  }  
  
  handleClick = () => {  
    this.setState({  
      date: new Date()  
    });  
    console.log('Click happened');  
  }  
  
  render() {  
    return <ChildComponent onClick={this.handleClick}>Click Me</ChildComponent>;  
  }  
}
```



React/Redux

- Redux is not technically part of React



Short intro to JSX

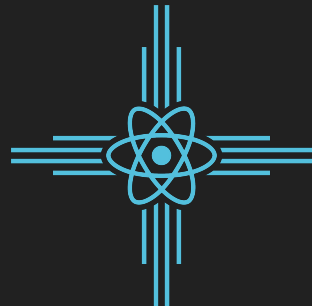
```
const element = <h1 className="greeting">Hello, world!</h1>;
```

This variable declaration is neither
html nor a string -- it's JSX!

- **JSX is a syntax extension to JavaScript (and comes with the full power of Javascript)**
- **We recommend using it with React to describe what UI should look like**
- **You can put *any* JavaScript expressions within braces inside JSX. Each React element is a JavaScript object that you can store in a variable or pass around in your program:**

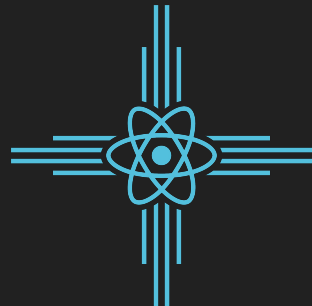
```
const name = 'Josh Perez';  
const element = <h1>Hello, {name}</h1>;
```

```
ReactDOM.render(  
  element,  
  document.getElementById('root')  
)
```



Create React App

- Is a quick boilerplate for getting a React web application up and running
- Uses Webpack
- Installed once on your system and can be used over and over

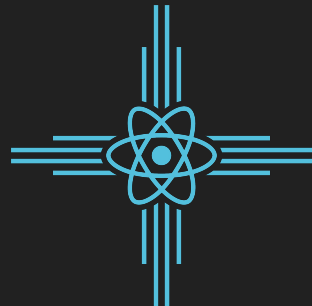


In terminal run the following command

```
sudo npm i -g create-react-app@1.4.1
```

Enter your password

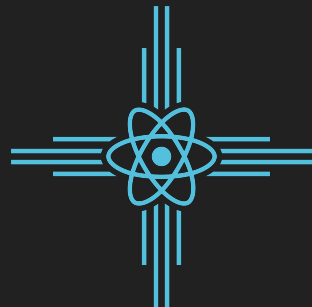
If you get an error, it may be because you installed Node via Homebrew. We recommend uninstalling node and reinstalling directly from <https://nodejs.org/en/>



In terminal run the following command

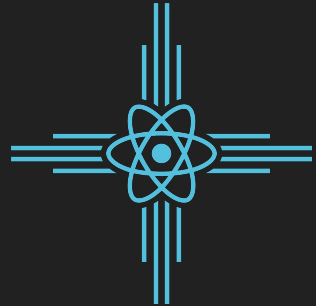
```
create-react-app  
/Users/[username]/Desktop/chat-app  
--scripts-version=1.0.14
```

This will create a new directory, named “chat-app”
on your Desktop that will hold our code



If you are using Chrome or find React Dev Tools in the Chrome plugin store, install and restart the browser.

Do the same if you are using FireFox



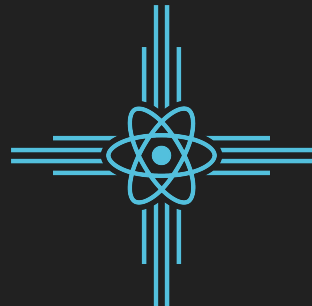
In terminal run the following commands

```
cd /Users/[username]/chat-app
```

This will get you into your new React app folder

```
npm start
```

This will launch your new app in the browser



Your First React Component

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

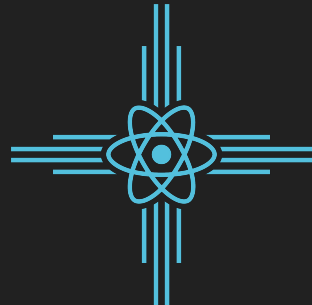
class App extends Component {
  constructor(props) {
    super(props);
    this.state = {name: "Andy"};
  }
  render() {
    return (
      <div className="App">
        <header className="App-header">
          <img src={logo} className="App-logo" alt="logo" />
          <h1 className="App-title">Welcome to React {this.state.name}</h1>
        </header>
        <p className="App-intro">
          To get started, edit <code>src/App.js</code>
        </p>
      </div>
    );
  }
}

export default App;
```

You can use an **ES6 Class** to define a component

App is a React Component class. You have your imports at the top and export at the bottom.

A component takes in parameters, called props (short for “properties”), and returns a hierarchy of views to display via the render method.



Resources we love:

- <https://www.fullstackreact.com/>
- <https://github.com/getify/You-Dont-Know-JS>

