Let's make a todo list app with React Native!

First workshop!

Nathaniel Budijono

UMN ADC

January 25, 2022

UMN App Developers

Logistics...

Streaming

- https://z.umn.edu/adc-zoom
- Recordings will be posted as unlisted YouTube videos as linked at https://adcumn.org/meetings

Logistics...

Streaming

- https://z.umn.edu/adc-zoom
- Recordings will be posted as unlisted YouTube videos as linked at https://adcumn.org/meetings

In-person

- Tuesdays 6-7pm
- Tate Hall 120

Officer openings!

- Workshop instructors

DM us on the discord!

https://z.umn.edu/ADCdiscord

Guide

We will be following this guide: https://github.com/ADC-UMN/todotorial. Feel free to go ahead of the workshops at your own pace.

What is React?

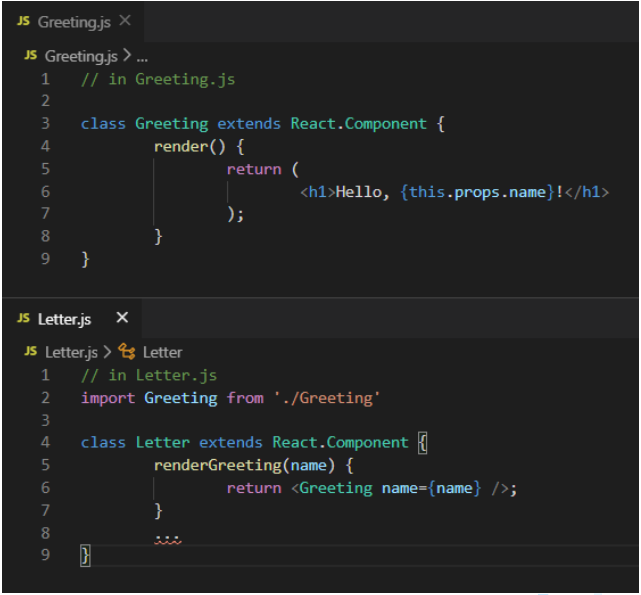
React is a *web application framework*.

What is React?

React is a *web application framework*.

It is *declarative*. The building blocks are *components*, which may be *functional* or *class components*.

React class components



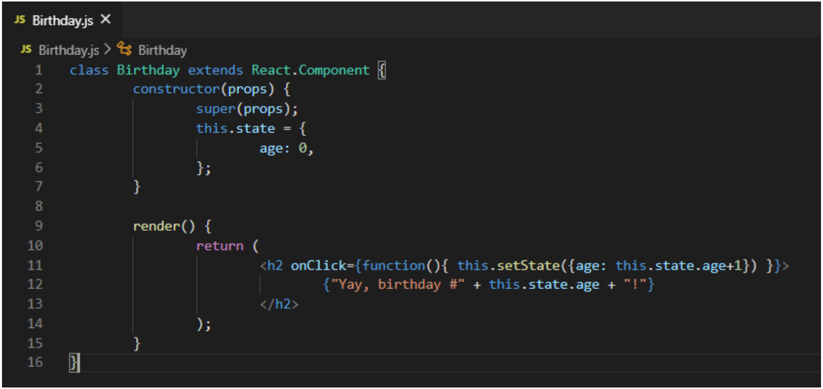
```
// in Greeting.js

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

```
// in Letter.js
import Greeting from './Greeting'

class Letter extends React.Component {
        renderGreeting(name) {
                return <Greeting name={name} />;
        }
        ...
}
```

State



```
class Birthday extends React.Component {
        constructor(props) {
                super(props);
                this.state = {
                        age: 0,
                };
        }

        render() {
                return (
                        <h2 onClick={function(){ this.setState({age: this.state.age+1}) }}>
                                {"Yay, birthday #" + this.state.age + "!"}
                        </h2>
                );
        }
}
```

What is React Native?

React Native allows you to write *cross-platform* React code instead of a *native* codebase for each platform. However, you cannot use HTML. Instead, there are native component libraries for you to use.

What is React Native?

React Native allows you to write *cross-platform* React code instead of a *native* codebase for each platform. However, you cannot use HTML. Instead, there are native component libraries for you to use.

Built-in components

React Native	Android	iOS	HTML
<View>	<ViewGroup>	<UIView>	<div>
<Text>	<TextView>	<UITextView>	<p>
<Image>	<ImageView>	<UIImageView>	<img>

Built-in components

React Native	Android	iOS	HTML
<View>	<ViewGroup>	<UIView>	<div>
<Text>	<TextView>	<UITextView>	<p>
<Image>	<ImageView>	<UIImageView>	<img>

See https://reactnative.dev for more.

Getting started

Snack allows you to edit and test your React Native apps in the browser at https://snack.expo.io.

Getting started

Snack allows you to edit and test your React Native apps in the browser at https://snack.expo.io.

You should instead use the Expo command line interface (CLI) so that you can build your app locally and upload it for distribution.

Installing Node.js

Node is a version of JavaScript that can run outside of the browser. It is required for React Native.

Installing Node.js

Node is a version of JavaScript that can run outside of the browser. It is required for React Native.

Go to https://nodejs.org/en/download/ to install.

Installing Node.js

Node is a version of JavaScript that can run outside of the browser. It is required for React Native.

Go to https://nodejs.org/en/download/ to install.

Confirm your installation and Node version with `node -v`.

Installing the Expo CLI

Now that we have Node, we use the Node Package Manager (NPM) to install Expo's CLI.

Installing the Expo CLI

Now that we have Node, we use the Node Package Manager (NPM) to install Expo's CLI.

Run `npm install -g expo-cli`.

Installing the Expo CLI

Now that we have Node, we use the Node Package Manager (NPM) to install Expo's CLI.

Run `npm install -g expo-cli`. You can now create a template for a new project with `expo init` and test it with `expo start`.

A look at what's ahead

1. Components, Styling, and Testing
2. Data Manipulation and State Handling
3. Persistent Lists using AsyncStorage
4. Deployment to Google Play Store

A look at what's ahead

1. Components, Styling, and Testing
2. Data Manipulation and State Handling
3. Persistent Lists using AsyncStorage
4. Deployment to Google Play Store

Here's the finished product: https://snack.expo.dev/@nathanielbd/todotorial.

A look at what's ahead

1. Components, Styling, and Testing
2. Data Manipulation and State Handling
3. Persistent Lists using AsyncStorage
4. Deployment to Google Play Store

Here's the finished product: https://snack.expo.dev/@nathanielbd/todotorial. What data should be kept track of using the state?