CS47: Cross-Platform Mobile Development

Lecture 2B: Components, Props + State, Hooks

James Landay
Abdallah AbuHashem
Tiffany Manuel
Cisco Vlahakis
Vy Mai

https://cs47.stanford.edu

cs47-fall19.slack.com

Fall 2019

Overview for today

- Live demo (throughout lecture)
  - Putting basic components to use
  - Styling layouts using Flexbox
- Class Components
  - Props
  - State
- Functional Components
  - Hooks

Overview for today

- Live demo (throughout lecture)
  - Putting basic components to use
  - Styling layouts using Flexbox
- Class Components
  - Props
  - State
- Functional Components
  - Hooks

Live Demo

Jedi ID Card

STARTER CODE

Live Demo

Jedi ID Card

STARTER CODE

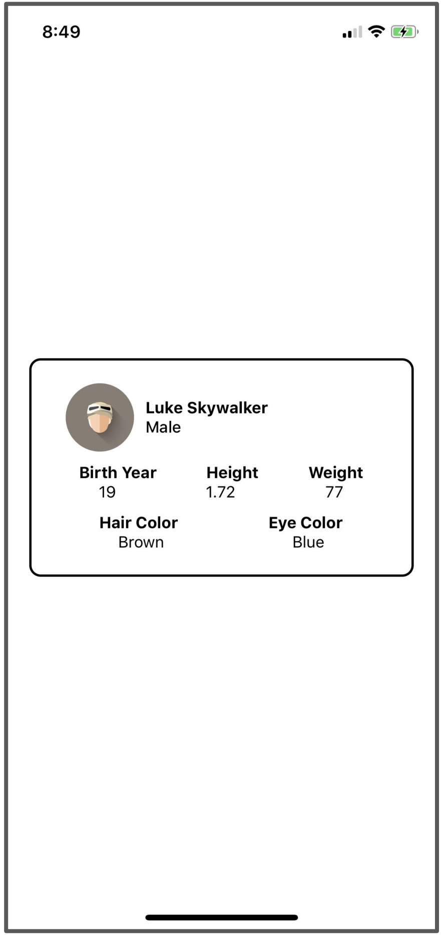
1) Run `npm install`
2) Open with Expo

Live Demo

Jedi ID Card

Create an application that shows the following details about a Jedi:

- Name
- Gender
- Birth Year
- Height
- Weight
- Hair Color
- Eye Color
- Picture?



8:49

Luke Skywalker
Male

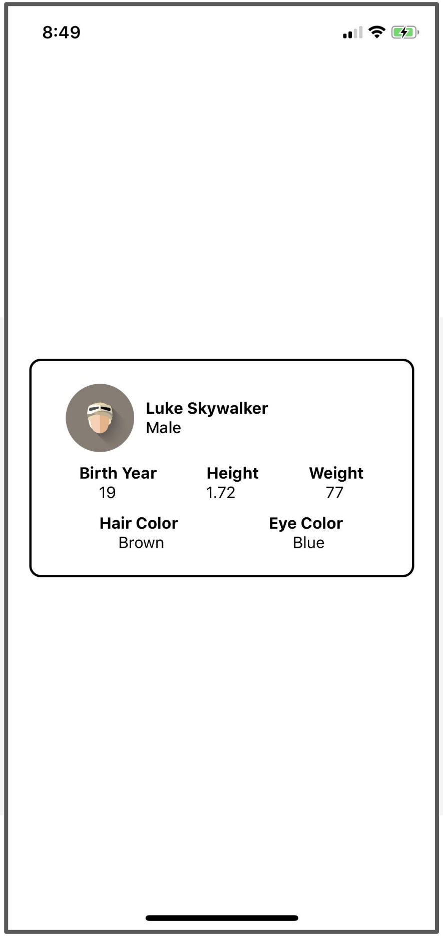
Birth Year	Height	Weight
19	1.72	77

Hair Color	Eye Color
Brown	Blue

Design to Code

BEST PRACTICES

1. Break down the design
2. Choose a component for each part
3. Code the component tree and style it



8:49

Luke Skywalker
Male

Birth Year 19

Height 1.72

Weight 77

Hair Color Brown

Eye Color Blue

USEFUL LINKS / React Native Docs / React Native Express / Cheatsheet

Overview for today

- Live demo (throughout lecture)
  - Putting basic components to use
  - Styling layouts using Flexbox
- Class Components
  - Props
  - State
- Functional Components
  - Hooks

Overview for today

- Live demo (throughout lecture)
  - Putting basic components to use.
  - Styling layouts using Flexbox.
- Class Components
  - Props
  - State
- Functional Components
  - Hooks

Overview for today

- Live demo (throughout lecture)
  - Putting basic components to use.
  - Styling layouts using Flexbox.
- Class Components
  - Props
  - State
- Functional Components
  - Hooks

Class Component

A `Component` that is declared using ES6 class syntax.

Class Component

A `Component` that is declared using ES6 class syntax.

```
import React from 'react';
import { StyleSheet, Text, View } from 'react-native';

export default class App extends React.Component {
    render() {
        return (
            <View style={styles.container}>
                <Text>Hello World!</Text>
            </View>
        );
    }
}
```

How do I make my app respond to changes?

(e.g. a button click, network request fetch, etc.)

Overview for today

- Live demo (throughout lecture)
  - Putting basic components to use
  - Styling layouts using Flexbox
- Class Components
  - Props
  - State
- Functional Components
  - Hooks

Overview for today

- Live demo (throughout lecture)
    - Putting basic components to use
    - Styling layouts using Flexbox
- Class Components
    - Props
    - State
- Functional Components
    - Hooks

Props

The parameters passed to the constructor of a `Component`.

```
<Image
  source={{uri: "https://facebook.github.io/react-native/docs/assets/favicon.png"}}
/>
```

Class Components

Props

```
<Image
  source={{uri: "https://facebook.github.io/react-native/docs/assets/favicon.png"}}
/>
```

Class Components

Props

```
<Image
  source={{uri: "https://facebook.github.io/react-native/docs/assets/favicon.png"}}
/>
```

```
<Button
  onPress={onPressLearnMore}
  title="Learn More"
  color="#841584"
  accessibilityLabel="Learn more about this purple button"
/>
```

Class Components

Props

```
<Button
  onPress={onPressLearnMore}
  title="Learn More"
  color="#841584"
  accessibilityLabel="Learn more about this purple button"
/>
```

Class Components

Props

Props

The parameters passed to the constructor of a `Component`.

Props are
only passed from
PARENT TO CHILD

Parent

```
export default class Parent extends React.Component {
     render() {
           return (
                <Image
                  source={{uri: "https://facebook.github.io/react-native/docs/assets/favicon.png"}}
                />

                <Button
                  onPress={onPressLearnMore}
                  title="Learn More"
                  color="#841584"
                  accessibilityLabel="Learn more about this purple button"
                />

           );
     }
}
```

Props

The parameters passed to the constructor of a `Component`.

Props are
only passed from
PARENT TO CHILD

Child

```
export default class Child extends React.Component {
    constructor(props) {
        super(props);

        // nothing has been rendered yet
        // you can change what is rendered based on the component’s props
        console.log(JSON.stringify(props));
    }
}
```

Child -- Image

```
export default class Image extends React.Component {
    constructor(props) {
        super(props);

        // Prints "https://facebook.github.io/react-native/docs/assets/favicon.png"
        console.log(JSON.stringify(props.source.uri));
    }
}
```

Overview for today

- Live demo (throughout lecture)
  - Putting basic components to use
  - Styling layouts using Flexbox
- Class Components
  - Props
  - State
- Functional Components
  - Hooks

Overview for today

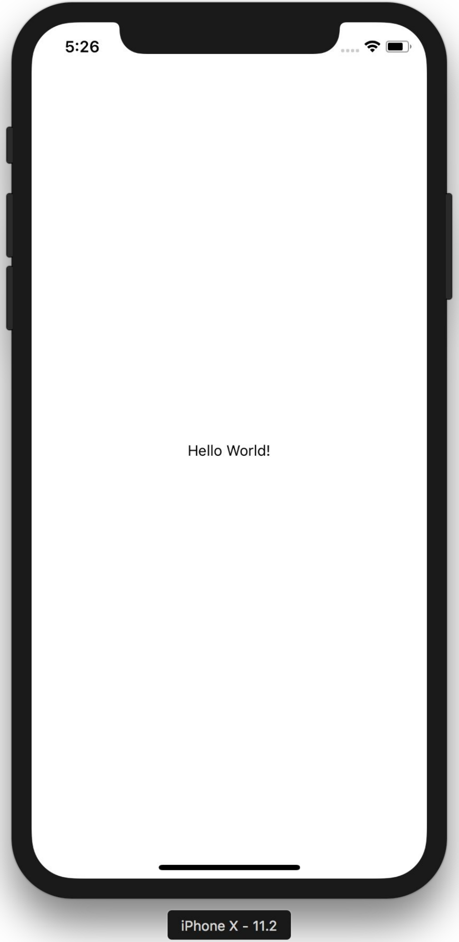
- Live demo (throughout lecture)
  - Putting basic components to use
  - Styling layouts using Flexbox
- Class Components
  - Props
  - State
- Functional Components
  - Hooks

State

An object with details about how a `Component` should render.

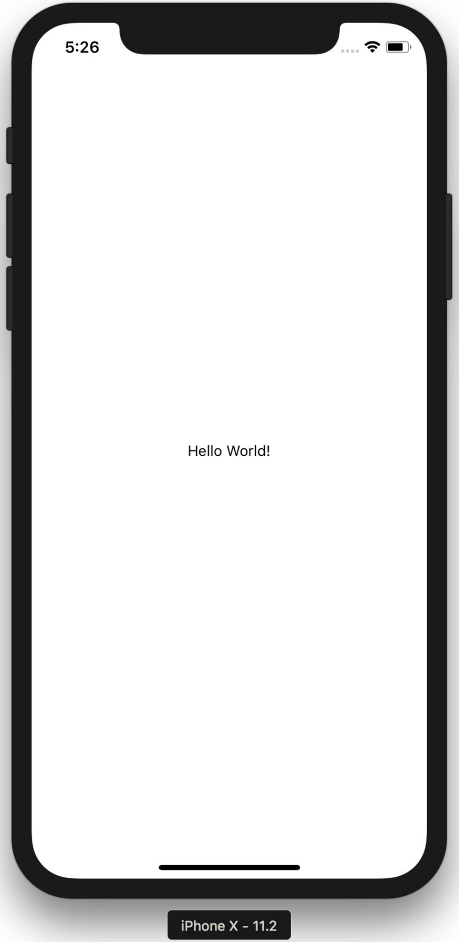
This literally is an object named `state`

```
import React from 'react';
import { StyleSheet, Text, View } from 'react-native';

export default class App extends React.Component {
  render() {
    return (
      <View style={styles.container}>
        <Text>Hello World!</Text>
      </View>
    );
  }
}
```



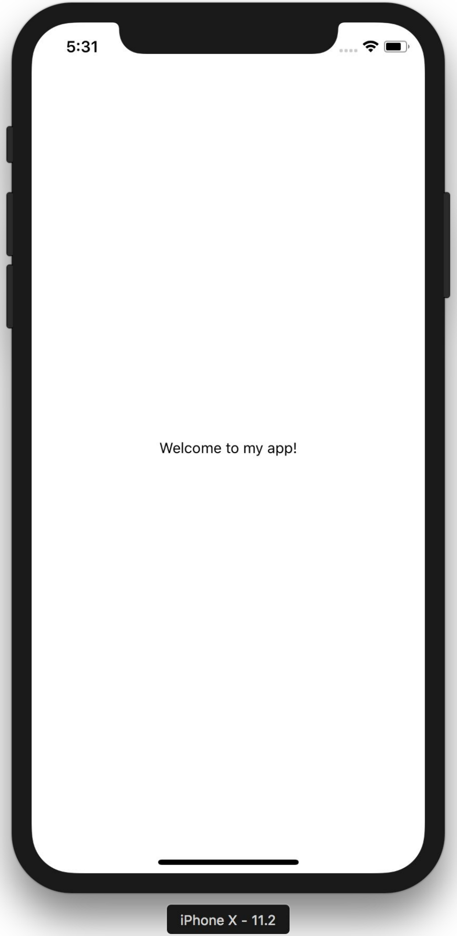
Hello World!

```
import React from 'react';
import { StyleSheet, Text, View } from 'react-native';

export default class App extends React.Component {

  state = {
    headline: 'Welcome to my app!'
  }

  render() {
    return (
      <View style={styles.container}>
        <Text>Hello World!</Text>
      </View>
    );
  }
}
```



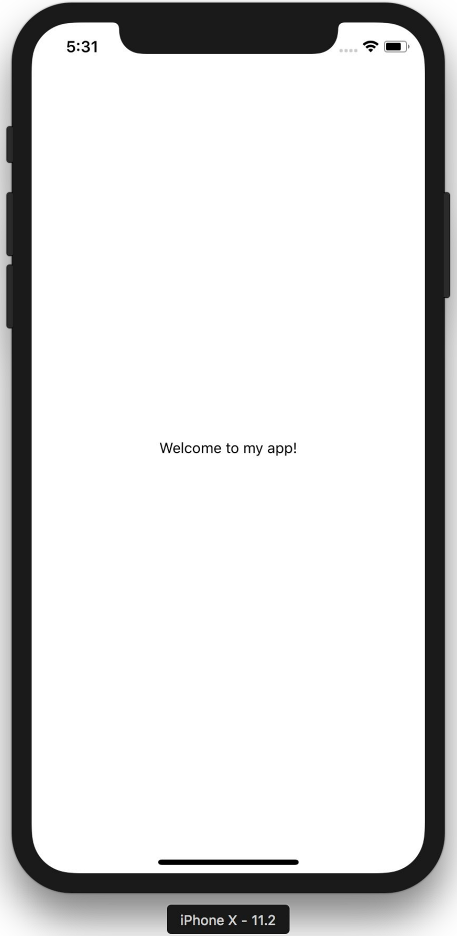
Hello World!

```
import React from 'react';
import { StyleSheet, Text, View } from 'react-native';

export default class App extends React.Component {

  state = {
    headline: 'Welcome to my app!'
  }

  render() {
    return (
      <View style={styles.container}>
        <Text>{this.state.headline}</Text>
      </View>
    );
  }
}
```

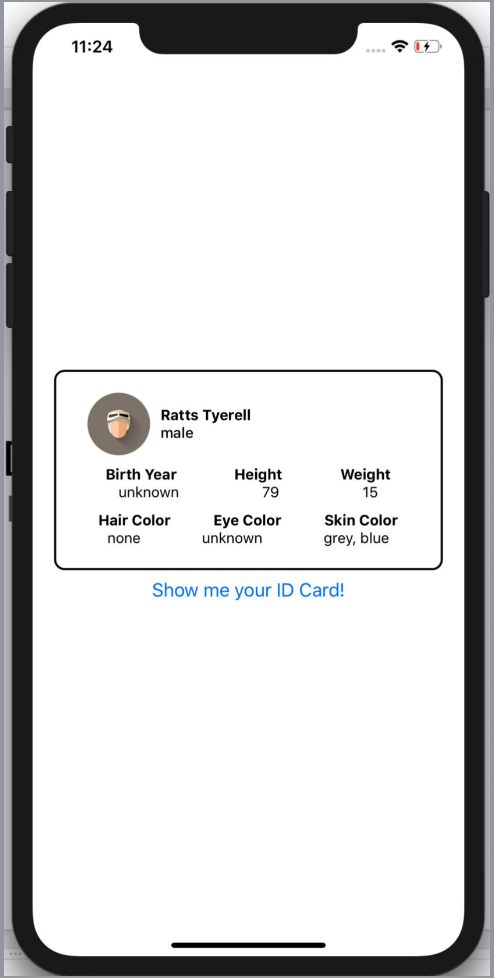


Welcome to my app!

```
import React from 'react';
import { StyleSheet, Text, View } from 'react-native';

export default class App extends React.Component {

  state = {
    headline: 'Welcome to my app!'
  }

  updateState = () => {
     this.setState({ headline: ‘Welcome back to my app!’ });
  }

  render() {
     ...
  }
}
```



`this.state.x = y` **VS.** `this.setState({x:y})`



11:24

Ratts Tyerell
male

Birth Year
unknown

Height
79

Weight
15

Hair Color
none

Eye Color
unknown

Skin Color
grey, blue

Live Demo

Jedi ID Card

Overview for today

- Live demo (throughout lecture)
  - Putting basic components to use
  - Styling layouts using Flexbox
- Class Components
  - Props
  - State
- Functional Components
  - Hooks

Overview for today

- Live demo (throughout lecture)
  - Putting basic components to use
  - Styling layouts using Flexbox
- Class Components
  - Props
  - State
- Functional Components
  - Hooks

Functional Component

A `Component` that is declared like a JS function.

Functional Component

A `Component` that is declared like a JS function.

```
const App = () => {
    return (
        <View style={styles.container}>
            <Text>Hello World!</Text>
        </View>
    );
}
```

Functional Component

A `Component` that is declared like a JS function.

```
const App = (greeting) => {
    return (
         <View style={styles.container}>
              <Text>{greeting}</Text>
         </View>
    );
}
```

Functional Component

A Component that is declared like a JS function.

```
const App = (greeting) => {
    return (
        <View style={styles.container}>
            <Text>{greeting}</Text>
        </View>
    );
}
```

Overview for today

- Live demo (throughout lecture)
  - Putting basic components to use
  - Styling layouts using Flexbox
- Class Components
  - Props
  - State
- Functional Components
  - Hooks

Overview for today

- Live demo (throughout lecture)
    - Putting basic components to use
    - Styling layouts using Flexbox
- Class Components
    - Props
    - State
- Functional Components
    - Hooks

Hooks

Functions that allow you to “hook into” React state and life cycle features from function components.

Hooks

Functions that allow you to “hook into” React state and life cycle features from function components.

Two Rules of Hooks

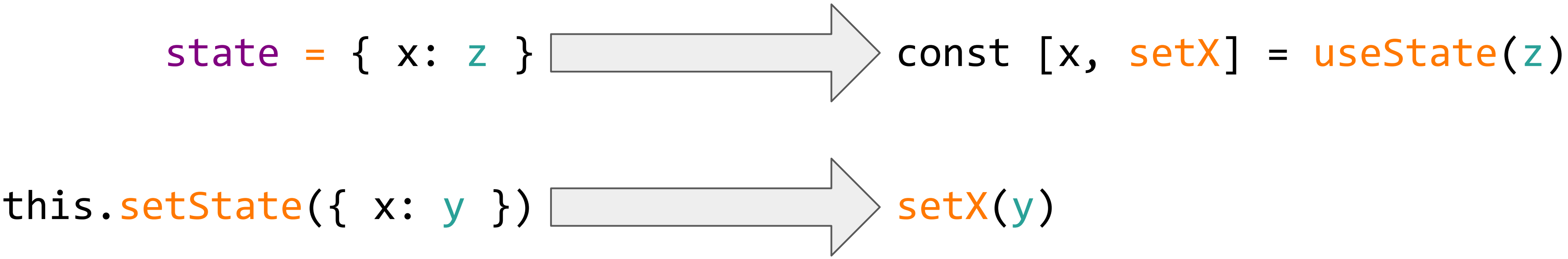
1) Only call Hooks at the **top level**

2) Only call Hooks from React **functional components**

`state` object (Class Components)

`useState` Hook (Functional Components)

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



```
state = { x: z }
```

→

```
const [x, setX] = useState(z)
```

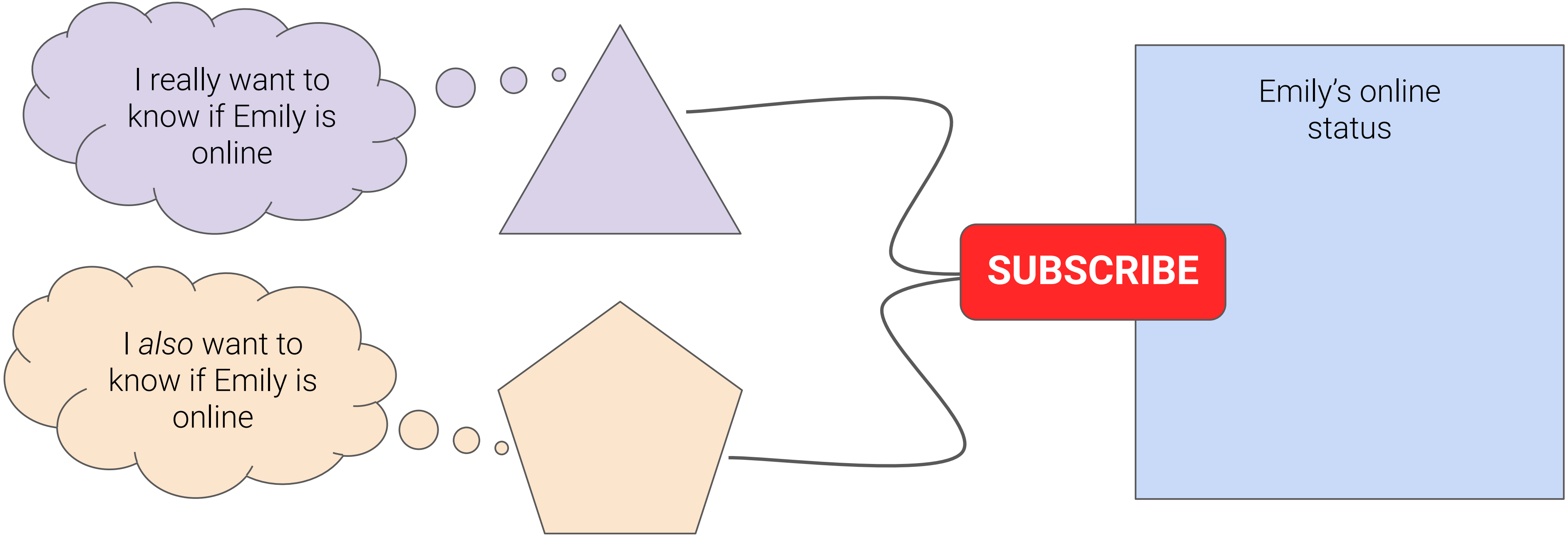
```
this.setState({ x: y })
```

→

```
setX(y)
```

Creating Your Own Hooks

JS function whose name starts with “use” (ex: useX) and may call other Hooks.
These allow functional components to “subscribe” to specific information.



I really want to know if Emily is online
I *also* want to know if Emily is online
SUBSCRIBE
Emily’s online status

How would you change the code for our Jedi ID Card to use ***functional components and the `useState` Hook*** (instead of class components and props/state)?

(We'll leave this as a ***highly recommended*** exercise, but the solution can be found here.)

USEFUL LINKS / React Hooks

Office Hours

Email us directly if you're not available at these times

Abdallah AbuHashem
Monday (12-1 PM) @ Huang Basement
By appointment

Vy Mai
Tuesday (3-4 PM) @ Old Union
By appointment

Cisco Vlahakis
Wednesday (8-9 PM) @ Huang Basement
By appointment

Tiffany Manuel
Thursday (2-3 PM) @ Huang Basement
By appointment

CS47: Cross-Platform Mobile Development

Lecture 2B: Components, Props + State, Hooks

James Landay
Abdallah AbuHashem
Tiffany Manuel
Cisco Vlahakis
Vy Mai

https://cs-47.stanford.edu

cs47-fall19.slack.com

Fall 2019