React

What's the big deal?

No seriously, what's the big deal

Quinn Stephens

tableXI

https://github.com/QuinnStephens/abi-intro-to-react

What is React?

- A Javascript library for creating views
- That's it

What makes React special*

- Components
- JSX
- The diffing algorithm

Components

Break up your UI into small pieces

Let them focus on one thing

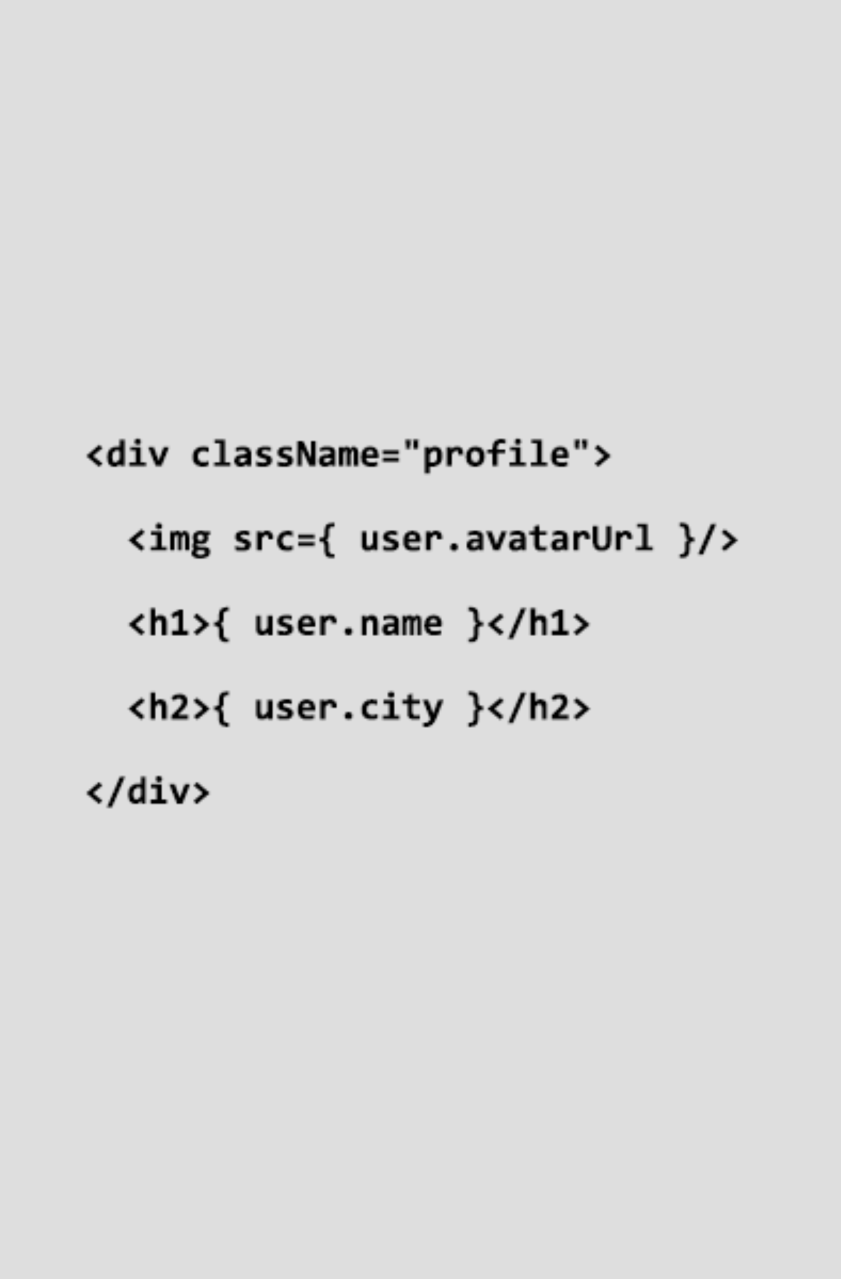
JSX

A Frankenstein's monster of HTML and JavaScript

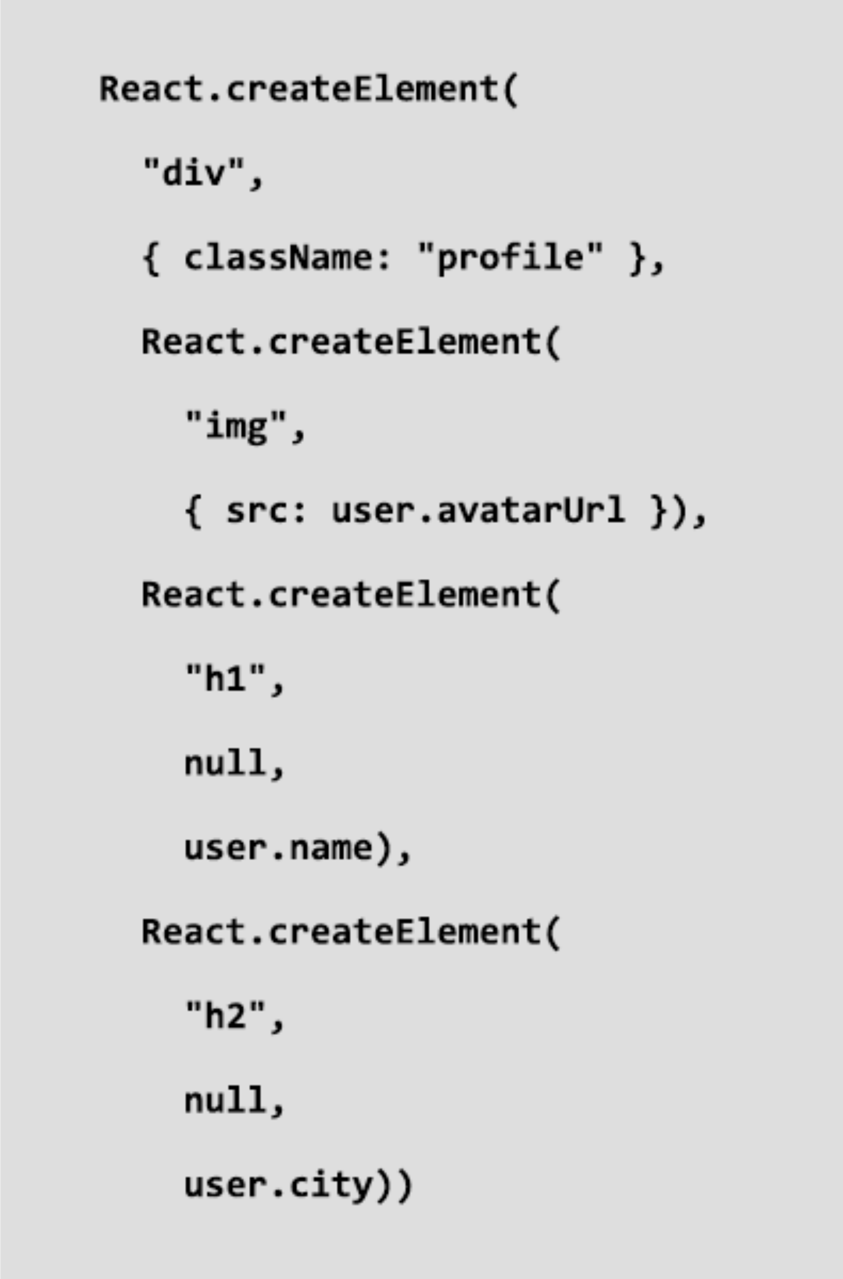
(but a good monster)

JSX is optional

But it's much nicer than the alternative



```
<div className="profile">
  <img src={ user.avatarUrl }/>
  <h1>{ user.name }</h1>
  <h2>{ user.city }</h2>
</div>
```



```
React.createElement(
  "div",
  { className: "profile" },
  React.createElement(
    "img",
    { src: user.avatarUrl }),
  React.createElement(
    "h1",
    null,
    user.name),
  React.createElement(
    "h2",
    null,
    user.city))
```

The diffing algorithm

Data changes. UI updates.

No thinking required

The rest is up to you

- React is not a framework but a set of tools
- React is just the view layer
- Structure for the app comes from other tools
  -- React frameworks (Redux)
  -- Backend frameworks (Rails, Node, et al)

Let's see an example!

- Open the source of project you created with `create-react-app`
- Run `npm  start` in the terminal from your project's root

Edit the text

Watch the diff engine at work.

Don't bother reloading

Edit the style

- Normal CSS, except…
- Use `className` instead of class

Replace text with a variable

(branch: `1-editing-and-variables`)

Move the header into its own component

- Only `render` method is required
- Pass the variable into the new component as a prop

(branch: `2-separate-components-and-props`)

Pass in a callback function as a prop

- Add a button
- Add a prop to show a simple alert in the parent component
- Make sure to bind `this` when needed!

(branch: `3-callback-function-as-prop`)

Move the `introText` variable to state

Update the state on button press

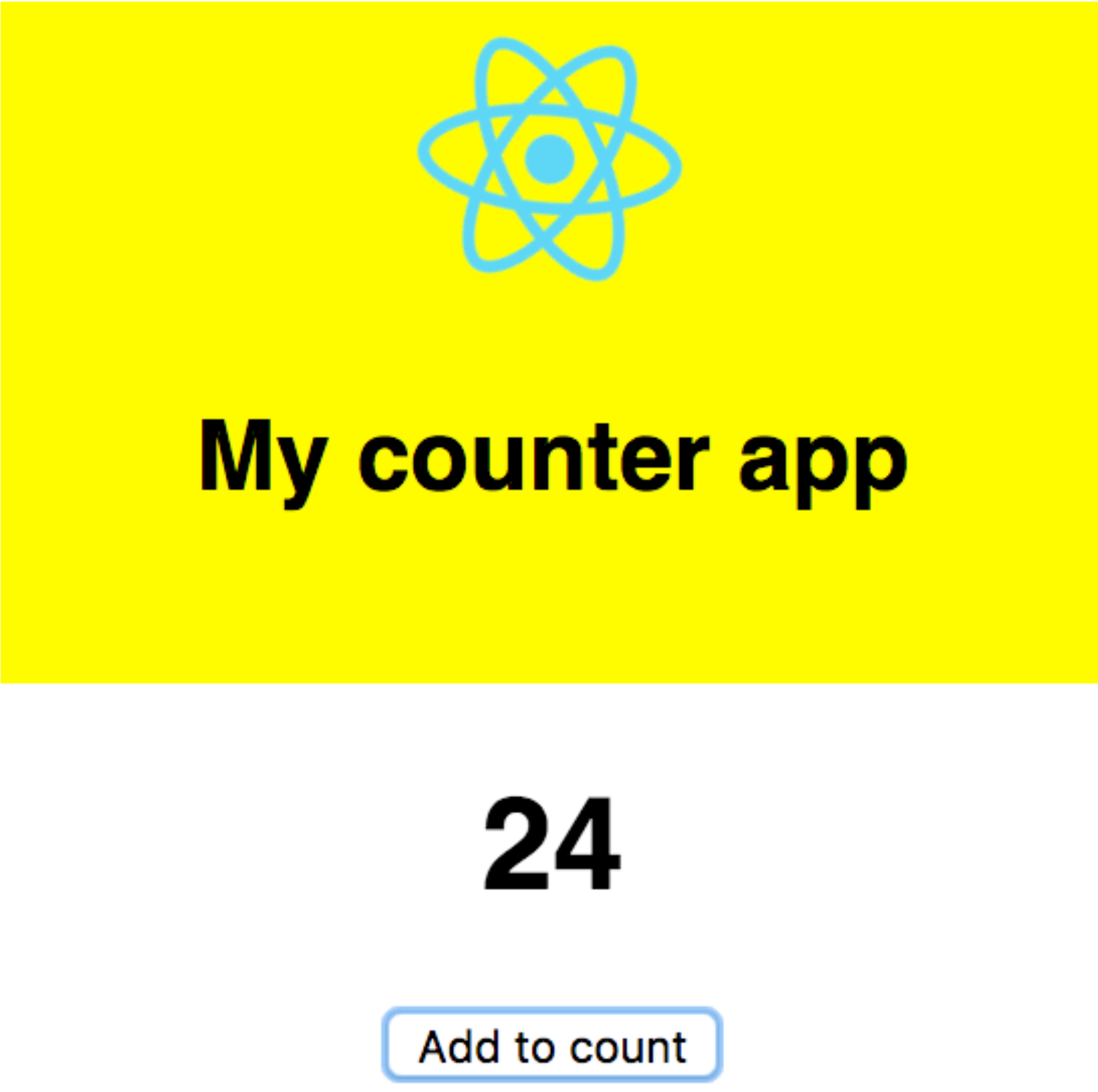
- Don't change the state directly!
- `setState` is async
- Change the state once
- Change it each time the button is pressed

(branch: `4-using-state`)

Questions?

Your turn: Create a counter component

- Keep the counter value in the App state
- Update the counter value through a button press



My counter app

24

Here's how I did it

(branch: `5-counter`)

Let's make a simple form

- Single text field
- Update state on change

Add a submit button

- Show an alert with the value

(branch: `6-form`)

Conditionally show components

- Only show a component once the user has submitted

(branch: `7-conditional-rendering`)

Your turn—do some more conditional logic

- Only show a component if the user has entered more than a certain number of characters

Here's how I did it

(branch: `8-more-conditional-logic`)

Component lifecycle methods

- `constructor`
- `componentWillMount` / `componentDidMount`
- `componentWillUnmount`
- `componentWillRecieveProps`
- `shouldComponentUpdate`

Let's make an API call

- Fetch within `componentDidMount` and update state on response
- URL: https://jsonplaceholder.typicode.com/photos/1

(branch: `9-api-call`)

Your turn: Show a loading state before the call finishes

Here's how I did it

(branch: `10-loading`)

Thank you for coming!

Quinn Stephens

https://github.com/QuinnStephens/abi-intro-to-react