React Fundamentals
Module – Class based components

Peter Kassenaar –
info@kassenaar.com
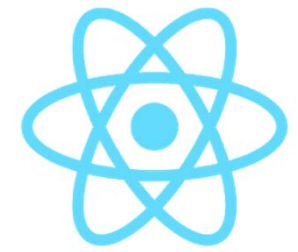
Class-based Components

Using ES6-classes for your components, methods, state and props

Types of components

Function Components

Class Components

Class-based components

- In this concept, components are just classes

- Classes have a mandatory `render()` function

- They can have an optional `constructor`, methods and a `state` object

  - `state` in the constructor

  - `state` as standalone class property

- *You don't have to learn new concepts, as all components are equal!*

- It's a matter of personal preference
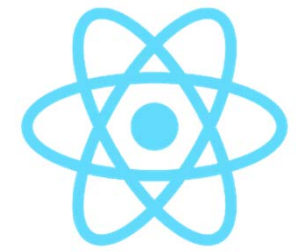
For instance: `<DisplayCounter />`

Import `{Component}` also

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

class DisplayCounter extends Component {
    // Optional constructor, to initialize th       s component
    constructor(props) {
        super(props);
        this.state = {
            // optional: some local state
        }
    }

    // mandatory render() function, to render the UI
    render() {
        return (
            <h2>
                {this.props.counter}
            </h2>
        )
    }
}

export default DisplayCounte
```
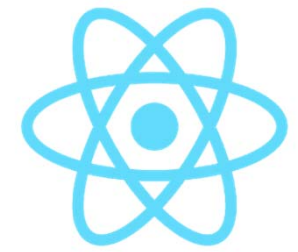
Constructor w/ `props`

Local state

`render()` **function**

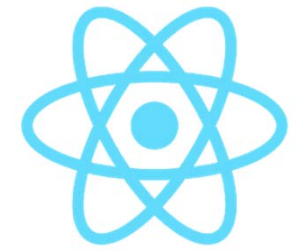**Access props and methods via** `this` **keyword**

But: no Hooks in classes!

- Use a local `state` object and the method

  `.setState({...})`

```
class App extends Component {

    state = {
        counter: 0
    };

    incrementCounter(val) {
        const newCounter = this.state.counter + val;
        this.setState({
            counter: newCounter
        });
    }
}
```
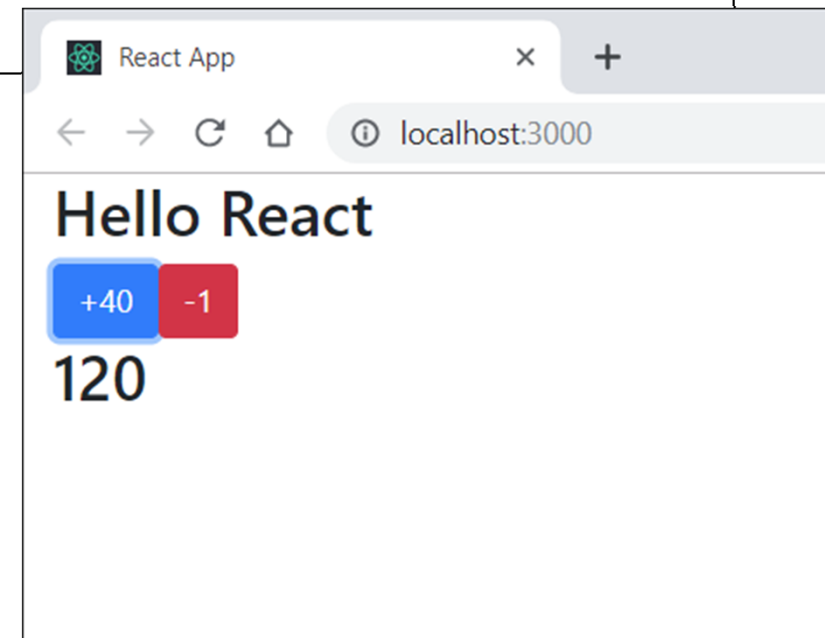
Update state via
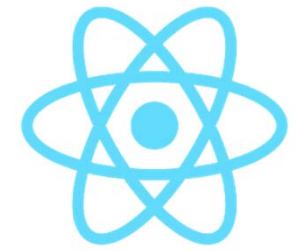**`this.setState(…)`**

Use arrow function in render

```
render() {
    return (<div className="container">
            <h2>Hello React</h2>
            <Counter increment={() => this.incrementCounter(40)} val={40}/>
            <DisplayCounter counter={this.state.counter}/>
        </div>
    )
};
```

Use the keyword `this` whenever you

access state or instance methods

More info: https://reactjs.org/docs/faq-functions.html

Passing functions to components
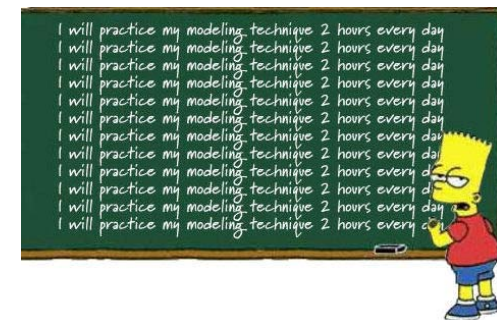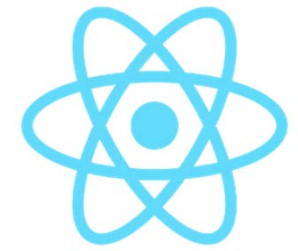
Workshop
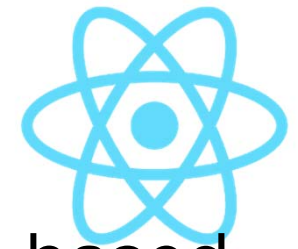
- Rewrite your previous application to use class-based components instead of function components

- OR: start with the sample application, and implement the `decrement` function

- Also pass the `className` for the button as a prop (`btn-primary, btn-success, btn-info,...`)

- Update the `<App />` component

- Example `../140-class-components`

Checkpoint

- You know about class based components

- You know how to define and access state in a class based component

- You know how to pass props and functions down to child components in a class-based component