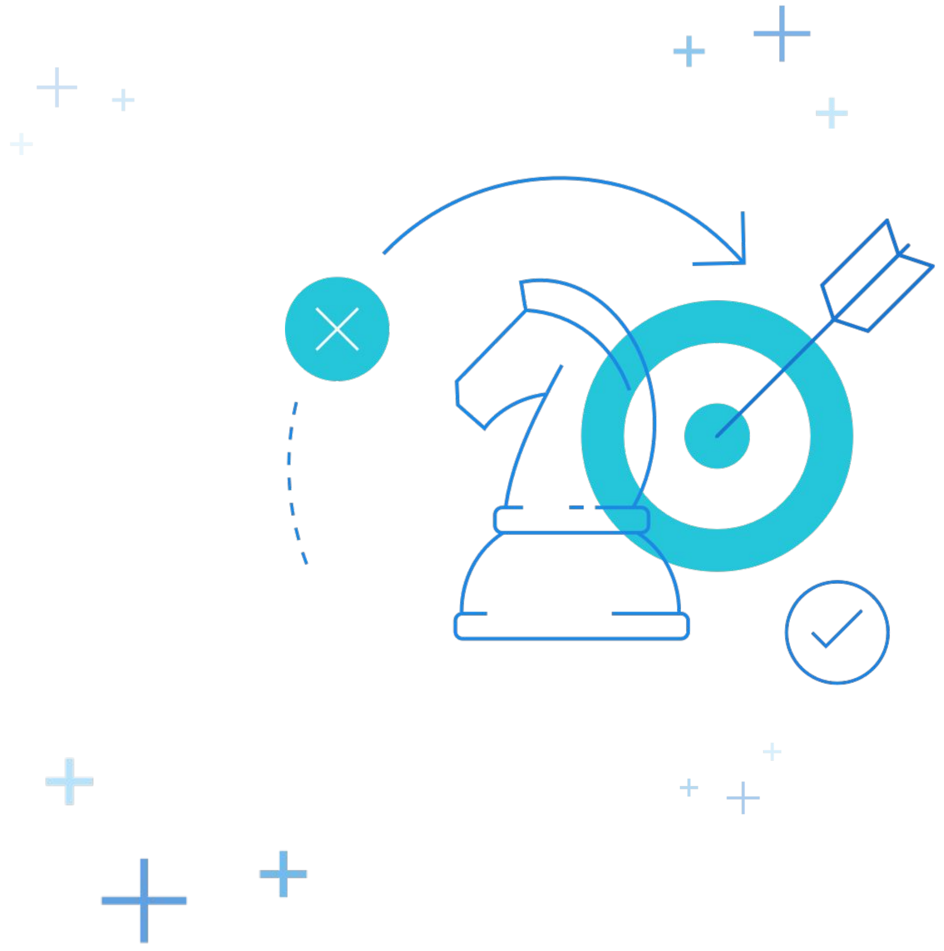
LEARNING ACTORS

React

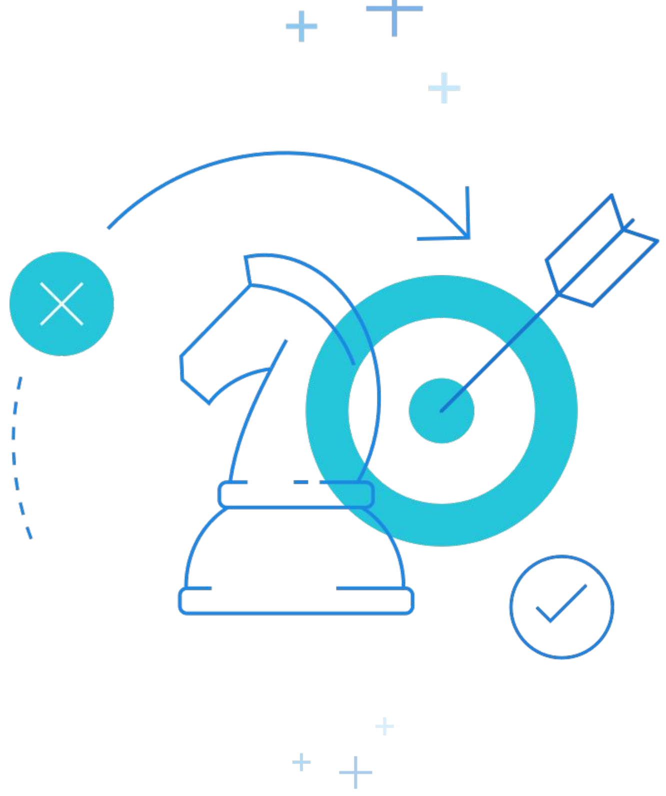
Redefining MVC

React

The V of MVC



Contents



Preparation in class

- Environment setup and test
- Setup Atom
- Set up React - https://github.com/facebookincubator/create-react-app
- Set up Redux
- Set up Redux Devtools

Introduction - reference

```
class HelloMessage extends React.Component {
  render() {
    return <div>Hello{this.props.name}</div>;
  }
}

ReactDOM.render(
  <HelloMessage name="John" />, mountNode
);
```

Understanding JSX - reference

JSX is a preprocessor step that adds XML syntax to JavaScript. You can definitely use React without JSX but JSX makes React a lot more elegant. Just like XML, JSXtags have a tag name, attributes, and children. If an attribute value is enclosed in quotes, the value is a string

JSX produces React "elements"

We will explore rendering them to the DOM

JSX Represents Objects

```
// The following
const Element = (
  <h1 className="greeting">
    Hello, world!
  </h1>
);
// Is the same as
const Element = React.createElement(
  'h1',
  {className: 'greeting'},
  'Hello, world!'
);
```

LEARNING ACTORS

JSX Represents Objects reference

```
// and the following...
const Element = (
  <h1 className="greeting">
    Hello, world!
  </h1>
);
// generates… (almost…)
const Element = {
  type: 'h1',
  props: {
    className: 'greeting',
    children: 'Hello, world'
  }
};
```

React Component

- Simple component example

React State and Lifecycle

- A problem and a problematic approach
- React State
  - ❏ this.state
  - ❏ this.setState
  - ❏ an example that does not work…
  - ❏ state and component lifecycle example

LEARNING ACTORS

React Events

- React events are named using camelCase
  - ❑ onClick instead of onclick.
- With JSX you pass a function as the event handler, rather than a string
  - ❑ `onClick={() => doSomething()}` instead of `onclick="function() {...}"`
- example

AGILE ACTORS

Conditional Rendering

- Using If else example
- Using Element Variables example
- Using short syntax example
- Using short syntax with else example
- Prevent element from rendering: when render returns null, false, undefined, true

Lists and Keys

- Rendering Collection of Components example
- List Component example
- Keys
  - ❑ Keys help React identify which items have changed, are added, or are removed
  - ❑ where keys should be used
- Keys Must Only Be Unique Among Siblings example

Forms

- A simple Form Component example
- Select example
- Handling multiple inputs example

Prop Type Checking

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

Greeting.propTypes = {
  name: React.PropTypes.string
};
```

- [reference]

Composition vs Inheritance

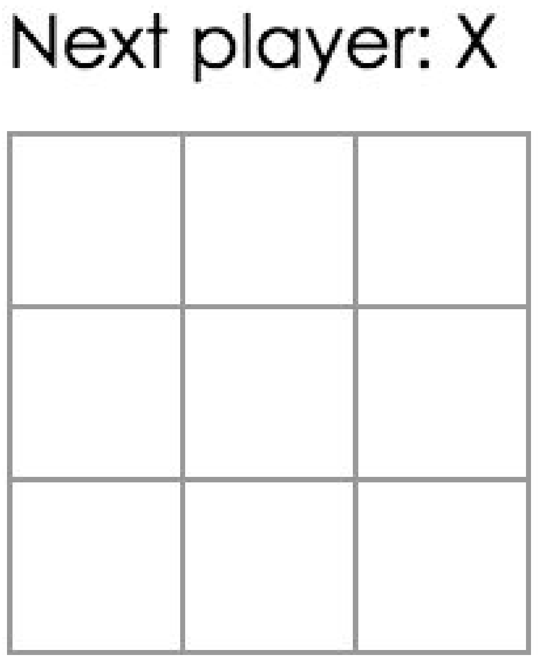
At Facebook, we use React in thousands of components, and we haven't found any use cases where we would recommend creating component inheritance hierarchies.

Props and composition give you all the flexibility you need to customize a component's look and behavior in an explicit and safe way. Remember that components may accept arbitrary props, including primitive values, React elements, or functions.

reference

Application Spec

- Hands on practice



Next player: X

LEARNING ACTORS

Thank you!

www.agileactors.com | www.linkedin.com/company/agile-actors/