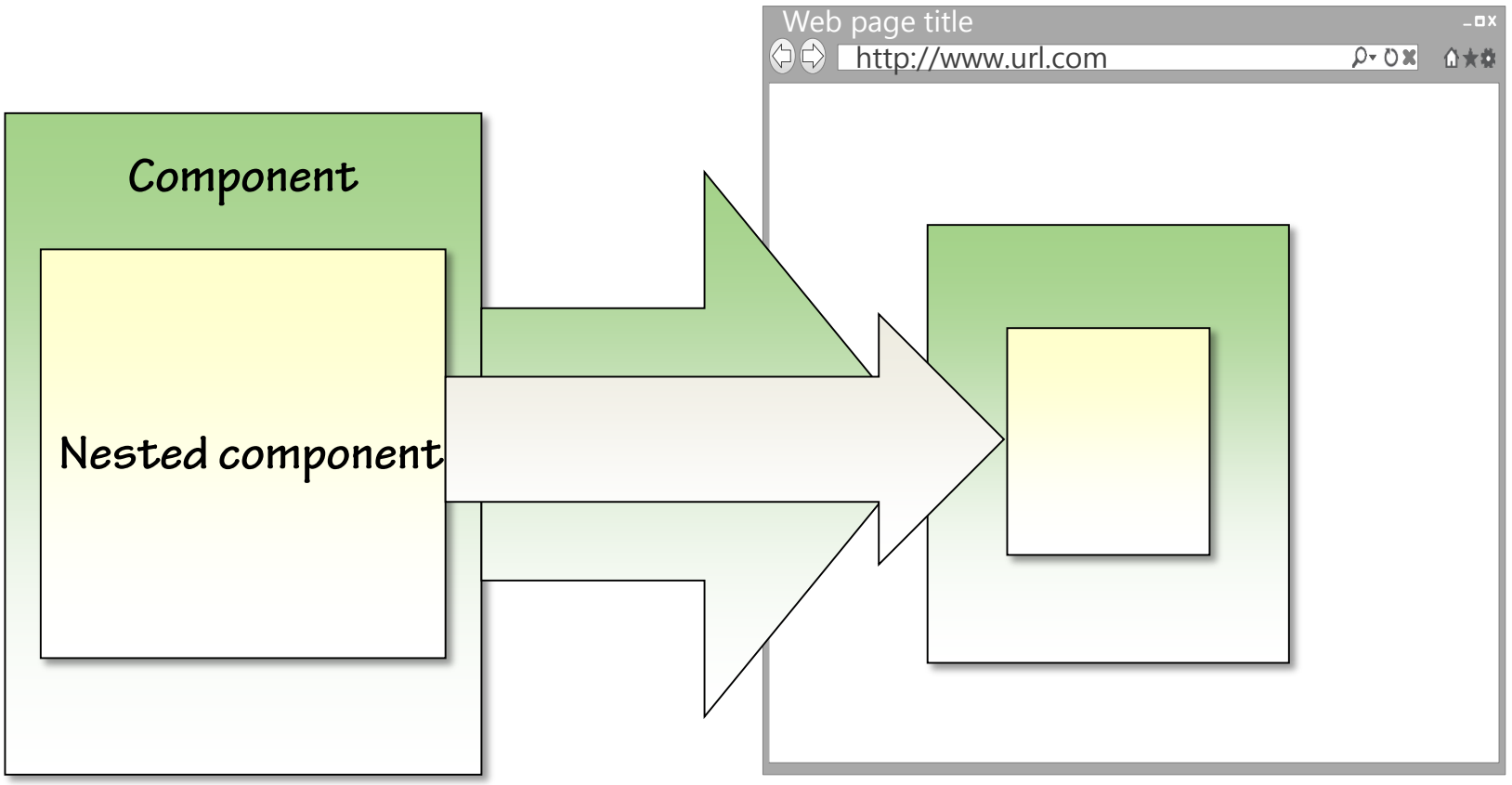
Components

Liam McLennan

pluralsight
hardcore developer training

What is a Component?



The Author Quiz

Author Quiz

Select the book written by the author shown

- The Adventures of Huckleberry Finn
- Hamlet
- The Shining
- Romeo and Juliet

Defining Components

- Top-level namespace is React

```
var Hello = React.createClass({

    render: function() {
        return    <div>
                      <h1>Hello at {this.props.now}</h1>
                  </div>;
    }
});
```

Defining Components

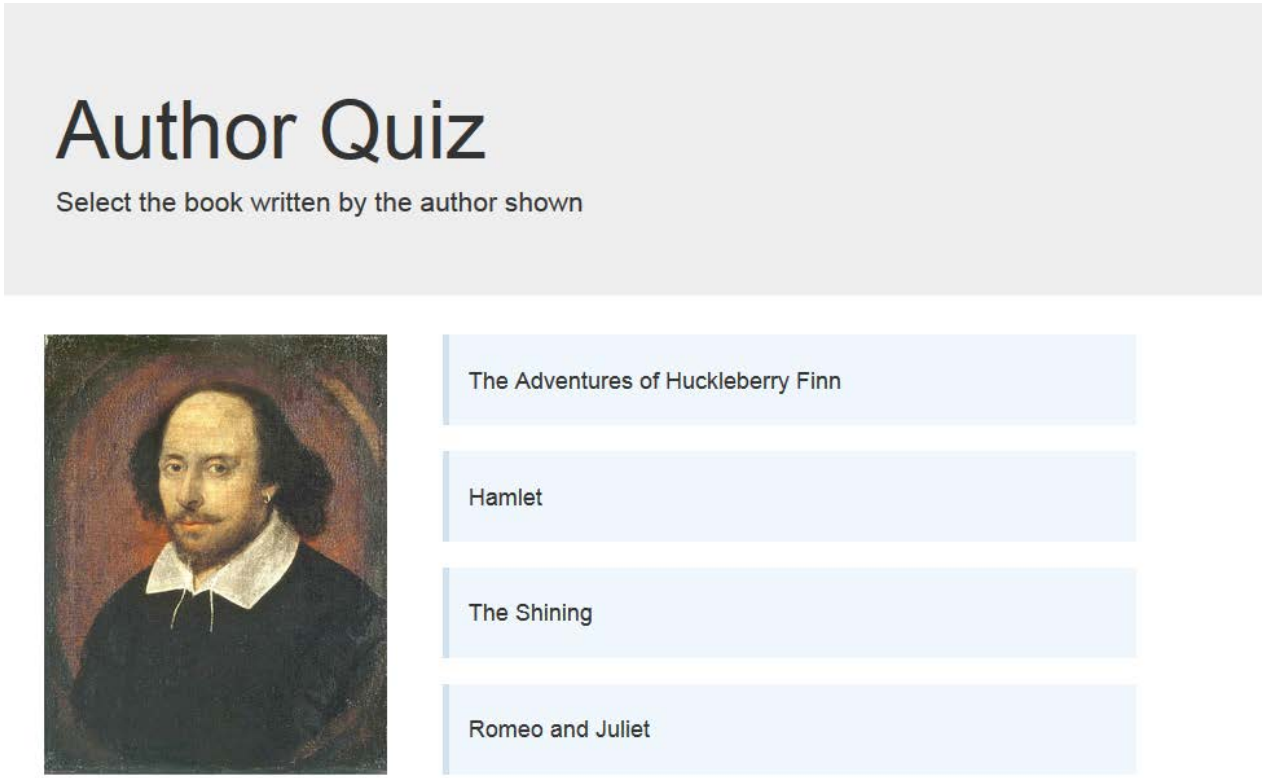
- Top-level namespace is React

```
var Hello = React.createClass({
    displayName: 'Hello',

    render: function () {
        return React.DOM.div(null,
                  React.DOM.h1(
                       null, "Hello at ", this.props.now)
                );
    }
});
```

Bootstrapping the Author Quiz

1. Create the basic page layout and styles
2. Get a copy of React
3. Connect everything together



Rendering Components

- **Rendering a component means linking it to a DOM element and populating that DOM element.**

```
React.render(

    <Hello now={new Date().toString()} />,

    document.getElementById('container')

);
```

Populating Props

- Props are supplied as attributes

```
React.render(
    <Hello now={new Date().toString()} b="7" />,
    document.getElementById('container')
);
```

- Access Props via the props property

```
this.props.now
this.props.b
```

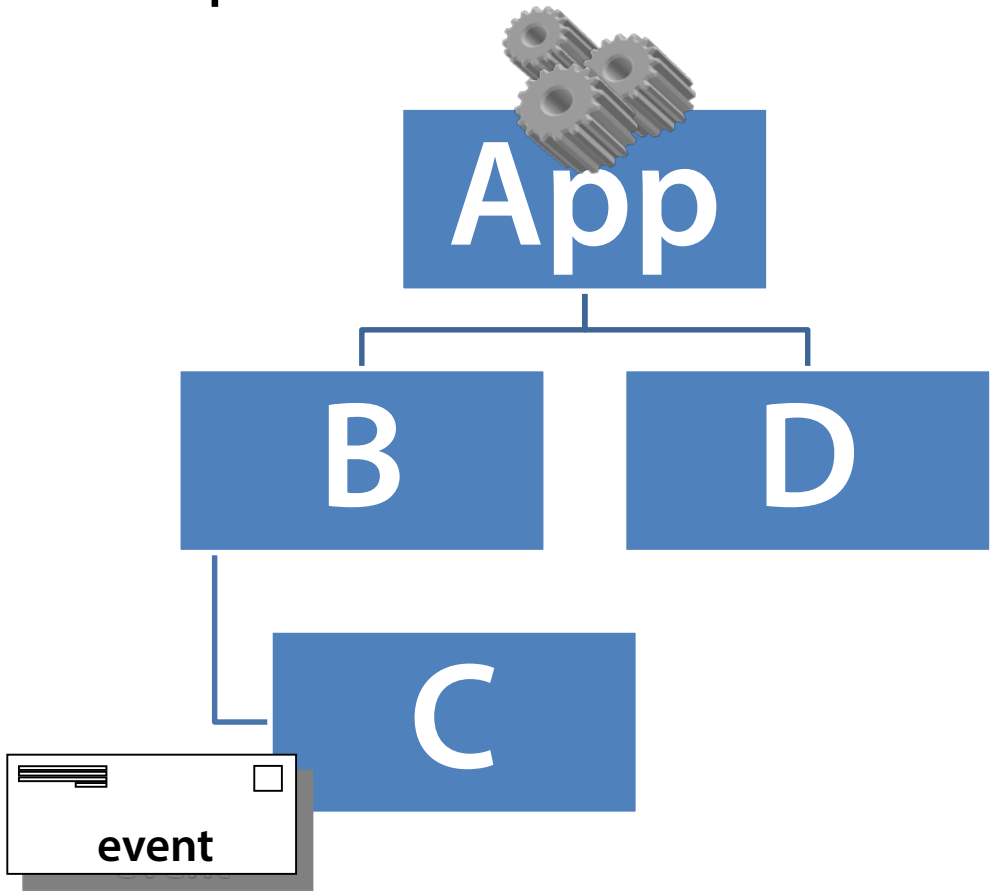
Composing Components

- Components can be nested inside other components
- Any components can be nested because components are completely self-contained

```
React.render(
    <Outer>
        <Inner />
    </Outer>),
  document.getElementById('container'));
```

State

- Used when a component needs to change independently of its parent.
- Components with state have more complexity



getInitialState

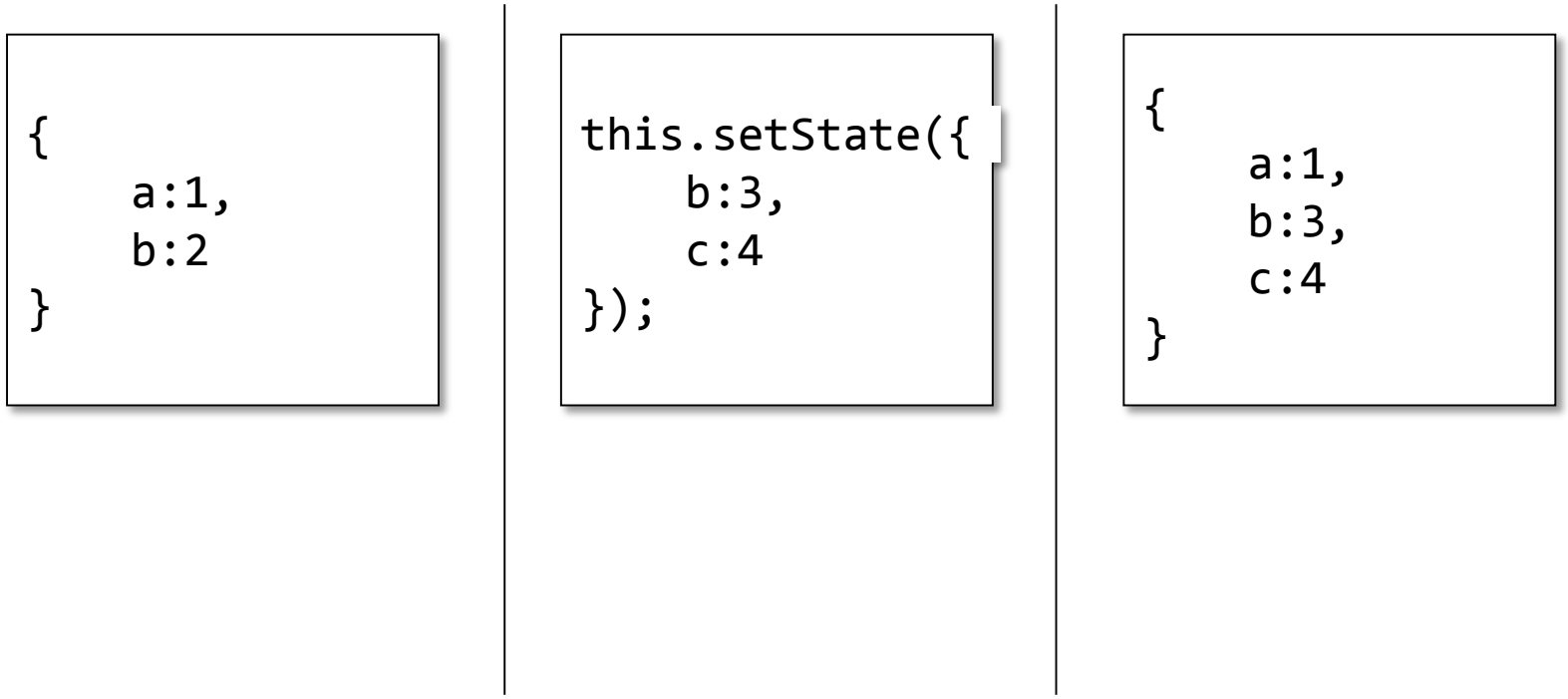
- getInitialState allows a component to populate its initial state.

```
/** @jsx React.DOM */
var ShowState = React.createClass({
    getInitialState: function () {
        return {answer: 42};
    },
    render: function() {
        return <div>my state is {this.state.answer}</div>;
    }
});
```

setState

- setState is the function used to update the state of a component.
- setState merges the new state with the old state

Previous state + setState = New state



getDefaultProps

- getDefaultProps specifies property values to use if they are not explicitly supplied.

```
var Text = React.createClass({









});
```

Validating Properties

- Validate props with propTypes
- Supports validation of existence, data type or a custom condition

```
var Hello = React.createClass({
    propTypes: {
        now: React.PropTypes.string
    }
});
```

Validating Properties

- Validate props with propTypes
- Supports validation of existence, data type or a custom condition

```
var Hello = React.createClass({
    propTypes: {
        now: React.PropTypes.string.isRequired
    }
});
```

Validating Properties

- Validate props with propTypes
- Supports validation of existence, data type or a custom condition

```
var Hello = React.createClass({
    propTypes: {
        now: React.PropTypes.oneOf(['Red','Green','Blue'])
    }
});
```

Validating Properties

- Validate props with propTypes
- Supports validation of existence, data type or a custom condition

```
var Hello = React.createClass({
    propTypes: {
        count: function (props, propName, componentName) {
            if (props[propName] < 5) {
                throw new Error(propName +
                    " must be 5 or greater");
            }
        }
    }
});
```

Mixins

- Mixins allow common code to be merged into many components

```
var Highlight = {
    componentDidUpdate: function () {
        var node = $(React.findDOMNode());
        node.slideUp();
        node.slideDown();
    }
};
```

```
var Count = React.createClass({
    mixins: [Highlight]
});
```

Summary

- Components are the building blocks of React applications
- Components are composable
- Components map to equivalent DOM nodes
- createClass defines a component
- render renders a component definition into the DOM
- Props provide the immutable data for a component
- State provides the mutable data for a component
- propTypes allow basic validation of props
- Mixins allow reuse between components

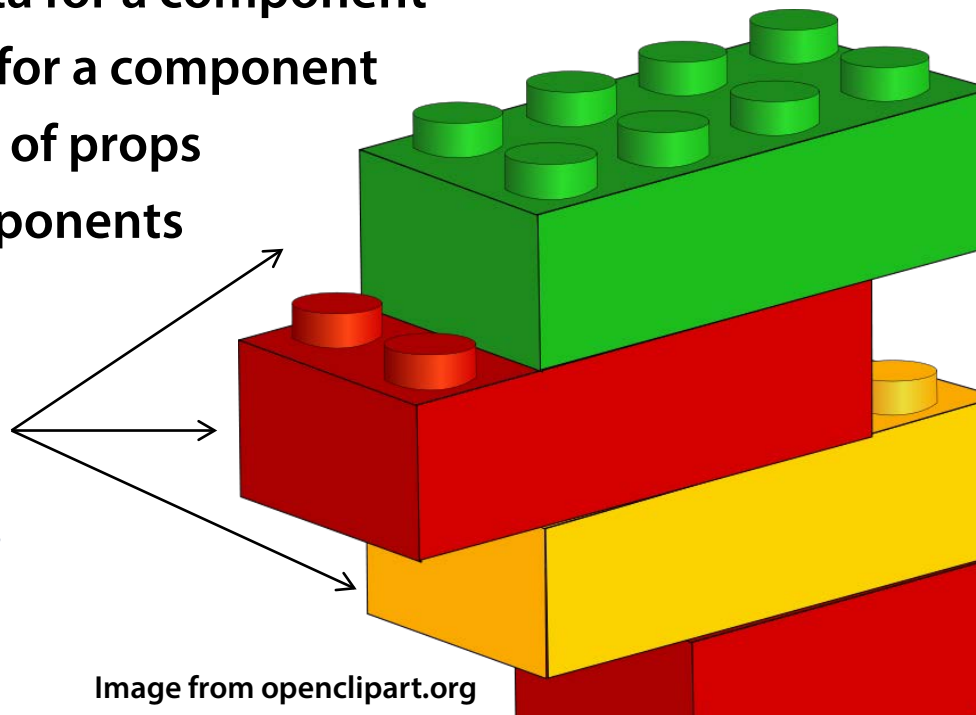


Image from openclipart.org