Course | Discussion | Course Handouts | Progress | Additional Resources | Notes

Course > Week 3: Client-Side Frameworks for Developing Modular Web Page Components > Developing Web Applications with React > Quiz 3.3

< Previous   🎞   🎞   ✎✓   🎞   ✎✓   🎞   ✎   Next >

Quiz 3.3

🔖 Bookmark this page

Problem 1

1 point possible (graded)

I have implemented a React component that uses a callback function that contains the line: `this.state = { value : this.state.value+1 };` I see that the callback function is invoked and this line is executed, but the HTML that uses `this.state.value` never changes. What code should I instead use to fix this problem?

○ `this.state.value+1;`

○ `this.state.value = this.state.value+1;`

○ `this.setState({value : this.state.value+1});`

○ `this.setState({value + 1});`

Submit   You have used 0 of 3 attempts   ❓ Hint   💾 Save

Problem 2

1 point possible (graded)

I have a React component called "MyText" that is inserted into the HTML page using the following:

```
ReactDOM.render(
    <div><MyText value="Hi there!" /></div>,
    document.getElementById('container')
);
```

My component defines a callback function named "modifyText" and I want the function to be invoked when the user places the mouse over the text that is set as a property when this component is created. How should I implement the MyText component's "render" function?

○
```
render () {
    return (
        <span onMouseOver={modifyText}>
        {value}
        </span>
    );
}
```

○
```
render () {
    return (
        <span onMouseOver="modifyText();">
        {this.props.text}
        </span>
    );
}
```

○
```
render () {
    return (
        <span onMouseOver={modifyText}>
        {this.props.value}
        </span>
    );
}
```

○
```
render () {
    return (
        <span onMouseOver={this.modifyText.bind(this)}>
        {this.props.value}
        </span>
    );
}
```

Submit   You have used 0 of 3 attempts   ❓ Hint   💾 Save

< Previous   Next >

© All Rights Reserved