

Concepts in Focus

- Missing Export Statement
- Missing Import Statement
- Missing Extending the React Component Class
- class vs className
- onclick vs onClick
- Event Listeners inside Class Component
- Passing Event Handler
- Modifying the State Directly
- Calling `setState()` from `render()`
- Invoking Event Handler
- `setState()` is Asynchronous
- React Component should return a single JSX element
 - Fragments
- Props are Read-only

1. Missing Export Statement

Mistake:

File: src/App.js

```
1  import Counter from './components/Counter'
2
```

JSX

```
3 ▾ const App = () => {  
4   return <Counter />  
5 }  
6  
7 export default App
```

File: src/components/Counter/index.js

```
1 import { Component } from 'react'  
2  
3 import './index.css'  
4  
5 ▾ class Counter extends Component {  
6   render() {  
7     return (  
8       <p className="counter">Counter</p>  
9     )  
10  }  
11 }
```

JSX

Collapse ^

2. Missing Import Statement

Mistake:

File: src/App.js

```
1 ▾ const App = () => {  
2   return <Counter />  
3 }  
4  
5 export default App
```

JSX

```
5 export default App
```

File: src/components/Counter/index.js

JSX

```
1 import { Component } from 'react'
2
3 import './index.css'
4
5 class Counter extends Component {
6   render() {
7     return (
8       <p className="counter">Counter</p>
9     )
10  }
11 }
12
13 export default Counter
```

Collapse ^

3. Missing Extending the React Component Class

Mistake:

File: src/App.js

JSX

```
1 import Counter from './components/Counter'
2
3 const App = () => {
4   return <Counter />
5 }
6
7 export default App
```

File: src/components/Counter/index.js

JSX

```
1 import './index.css'
2
3 class Counter {
4   render() {
5     return (
6       <p className="counter">Counter</p>
7     )
8   }
9 }
10
11 export default Counter
```

Collapse ^

4. class vs className

Mistake:

File: src/components/Counter/index.js

JSX

```

1 import { Component } from 'react'
2
3 import './index.css'
4
5 class Counter extends Component {
6   render() {
7     return (
8       <p class="counter">Counter</p>
9     )
10   }
11 }
12
13 export default Counter

```

JSX

Collapse ^

5. onclick vs onClick

Mistake:

File: src/components/Counter/index.js

```

1 import { Component } from 'react'
2
3 import './index.css'
4
5 class Counter extends Component {
6   onIncrement = () => {
7     console.log("on increment called")
8   }
9   render() {
10    return (
11      <p className="counter">Counter</p>
12      <button onclick={this.onIncrement}>Increase</button>

```

JSX

```
13     )
14   }
15 }
16
17 export default Counter
```

Collapse ^

6. Event Listeners inside Class Component

Mistake:

File: src/components/Counter/index.js

```
1 import { Component } from 'react'
2
3 import './index.css'
4
5 class Counter extends Component {
6   state = { count: 0 }
7   onIncrement() {
8     const {count} = this.state
9     console.log(count)
10  }
11  render() {
12    return (
13      <p className="counter">Counter</p>
14      <button onClick={this.onIncrement}>Increase</button>
15    )
16  }
17 }
18
19 export default Counter
```

JSX

Collapse ^

Solution:

JAVASCRIPT

```
1 onIncrement = () => {  
2   const {count} = this.state;  
3   console.log(count);  
4 };
```

In Arrow functions,

`this` refers to the context in which the code is defined.

7. Passing Event Handler

Mistake:

File: src/components/Counter/index.js

JSX

```
1 import { Component } from 'react'  
2  
3 import './index.css'  
4  
5 class Counter extends Component {  
6   state = {count: 0}  
7   onIncrement = () => {  
8     this.setState(prevState => ({  
9       count: prevState.count + 1  
10    }))  
11  }  
12  render() {  
13    return <button onClick={onIncrement}>Increase</button>  
14  }
```

```
14 }  
15 }  
16  
17 export default Counter
```

Collapse ^

Solution:

JSX

```
1 <button onClick={this.onIncrement}>Increase</button>
```

8. Modifying the State Directly

Modifying the state directly won't trigger the

`render()` method.

Mistake:

JAVASCRIPT

```
1 onIncrement = () => {  
2   this.state.count = this.state.count + 1;  
3 };
```

Solution:

JAVASCRIPT

```
1 onIncrement = () => {  
2   this.setState(prevState => ({  
3     count: prevState.count + 1  
4   }));  
5 };
```


Updating Object

Mistake:

```
1 state = {  
2   person: {name: "Rahul", age: 30}  
3 }  
4  
5 onUpdateAge = () => {  
6   const {person} = this.state  
7   person.age = 29  
8 }
```

JSX

Solution:

```
1 state = {  
2   person: {name: "Rahul", age: 30}  
3 }  
4  
5 onUpdateAge = () => {  
6   const {person} = this.state  
7   const newPerson = {...person, age: 29}  
8   this.setState({person: newPerson})  
9 }
```

JSX

Updating List

Mistake:

```
1 state = { numbers: [1, 2, 3] }  
2  
3 onUpdateNumbers = () => {  
4   const {numbers} = this.state
```

JSX

```
4   const {numbers} = this.state
5   numbers.push(4)
6 }
```

Solution:

```
1  state = {numbers: [1, 2, 3]}
2
3  onUpdateNumbers = () => {
4    const {numbers} = this.state
5    const updatedNumbers = [...numbers, 4]
6    this.setState({numbers: updatedNumbers})
7  }
```

JSX

9. Calling `setState()` from `render()`

Mistake:

```
1  render() {
2    this.setState({ count: 0 })
3    return ({...})
4  }
```

JSX

Specifying a new state with

`setState()` function causes a new render. If we call `setState` from the `render()` function, this will cause an infinite loop.

When we call `setState` from the

`render()` function, it will cause an infinite loop.

10. Invoking Event Handler

Mistake:

```
1 import { Component } from 'react'
2
3 import './index.css'
4
5 class Counter extends Component {
6   state = {count: 0}
7   onIncrement = () => {
8     this.setState(prevState => ({
9       count: prevState.count + 1
10    }))
11  }
12  render() {
13    return (
14      <p className="counter">Counter</p>
15      <button onClick={this.onIncrement()}>Increase</button>
16    )
17  }
18 }
19
20 export default Counter
```

JSX

Collapse ^

Soution:

```
1 import { Component } from 'react'
2
3 import './index.css'
```

JSX

```

3 import './index.css'
4
5 class Counter extends Component {
6   state = {count: 0}
7   onIncrement = () => {
8     this.setState(prevState => ({
9       count: prevState.count + 1
10    }))
11  }
12  render() {
13    return (
14      <p className="counter">Counter</p>
15      <button onClick={this.onIncrement}>Increase</button>
16    )
17  }
18 }
19
20 export default Counter

```

Collapse ^

11. `setState()` is Asynchronous

`setState()` does not update state immediately.

Mistake:

```

1 import { Component } from 'react'
2
3 import './index.css'
4
5 class Counter extends Component {
6   state = {count: 0}
7   onIncrement = () => {
8     const {count} = this.state;

```

JSX

```

8     const {count} = this.state;
9     this.setState(prevState => ({
10       count: prevState.count + 1
11     })))
12     console.log(count)
13   }
14   render() {
15     return (
16       <p className="counter">Counter</p>
17       <button onClick={this.onIncrement}>Increase</button>
18     )
19   }
20 }
21
22 export default Counter

```

Collapse ^

12. React Component should return a single JSX element

Mistake:

```

1 const Welcome = () => (
2   <h1>Hello, User</h1>
3   <p>You are learning React</p>
4 )
5
6 export default Welcome

```

JSX

```

1 import Welcome from './components/Welcome'
2 const App = () => <Welcome />
3 export default App

```

JSX

12.1 Fragments

The fragment is an alternate way to return a single JSX element. It groups a list of children without adding extra nodes to the DOM.

JSX

```
1 import {Fragment} from 'react'
2
3 const Welcome = () => (
4   <Fragment>
5     <h1>Hello, User</h1>
6     <p>You are learning React</p>
7   </Fragment>
8 )
9
10 export default Welcome
```

Short Syntax:

JSX

```
1 const Welcome = () => (
2   <>
3     <h1>Hello, User</h1>
4     <p>You are learning React</p>
5   </>
6 )
7
8 export default Welcome
```

13. Props are Read-only

Mistake:

File: src/App.js

JSX

```
1 import Welcome from './components/Welcome'
2 const App = () => <Welcome name="Rahul" />
3 export default App
```

File: src/components/Welcome/index.js

JSX

```
1 const Welcome = props => {
2   props.name = 'Ramu'
3   const {name} = props
4   return (
5     <>
6       <h1>Hello, {name}</h1>
7       <p>You are learning React</p>
8     </>
9   )
10 }
11
12 export default Welcome
```

Collapse ^

✓ MARKED AS COMPLETE

[Submit Feedback](#)