Congratulations! You passed!

Grade received 88% **Latest Submission Grade** 88% **To pass** 80% or higher

Go to next item

1. Which of the following code snippets will set the initial state of a component to contain two people, one with brown hair and brown eyes and the other with blond hair and blue eyes? **1 / 1 point**

```
const [person, usePerson] = setState([
 { hairColor: "brown", eyeColor: "brown },
 { hairColor: "blond", eyeColor: "blue" }
]);
```

```
const [person, setPerson] = useEffect([
 { hairColor: "brown", eyeColor: "brown },
 { hairColor: "blond", eyeColor: "blue" }
]);
```

```
const [person, setPerson] = useState([
 { hairColor: "brown", eyeColor: "brown },
 { hairColor: "blond", eyeColor: "blue" }
]);
```

```
const {person, setPerson} = useState([
 { hairColor: "brown", eyeColor: "brown },
 { hairColor: "blond", eyeColor: "blue" }
]);
```

Correct

That's correct! The parameter passed in is an array containing two objects which both satisfy the requirements in the question.

2. What is the benefit of `useState` over `useReducer`? **1 / 1 point**

`useReducer` is more performant.

- ◯ There is no benefit.
- ◉ `useState` is more performant.

> **Correct**
> That's correct! The `useState` hook is more performant than `useReducer` because it does not require the overhead of creating a new dispatch function on every render.

3. Why do you use the `useEffect` hook in a react project? Select all that apply. **0.8 / 1 point**

- ☐ To update the component's state.
- ☐ To clean up effects before the component unmounts or re-renders.
- ☐ To trigger an effect *only* when the component mounts.
- ☑ To perform side effects after a component renders.

> **Correct**
> That's correct. The `useEffect` hook is often used to perform side effects after a component renders, such as making an HTTP request or updating the document title.

- ☑ To trigger an effect when a prop changes.

> **Correct**
> That's correct. The `useEffect` hook can be triggered when a prop changes, allowing it to perform side effects based on the updated prop value.

You didn't select all the correct answers

4. True or False: The `preventDefault()` method in React stops the default behavior of a new web page opening when a form is submitted. **1 / 1 point**

- ◉ True
- ◯ False

> **Correct**
> That's correct! In HTML the default behavior is to open a new web page when a form is submitted. The `preventDefault()` method in React stops this so that forms can be handled in the React way.

5. Which React hook must be utilized when calling an API using the fetch() method? **1 / 1 point**

- ◉ `useEffect`
- ◯ `useReducer`
- ◯ `useState`

Correct
That's correct! Using the `fetch()` method is considered a side effect because it is getting data from a third party, so the `useEffect` hook needs to be used.

6. Which data transfer format is most commonly used with APIs? **1 / 1 point**

- (●) JSON
- ( ) XML
- ( ) Text

Correct
That's correct! A valid endpoint is provided and the response is handled with a promise.

7. Which one of the following is a valid command for running your unit tests? **1 / 1 point**

- ( ) `test npm`
- ( ) `react test`
- (●) `npm run test`
- ( ) `run test`

Correct
That's correct! Running this command will trigger any tests written. An alternative command is npm test.

8. What is this code in JavaScript: **1 / 1 point**

```
1   /Make Your Reservation/?
```

- ( ) An array literal
- ( ) A string literal
- (●) RegExp object literal

Correct
That's correct! This code is a RegExp object literal, because it is enclosed in / delimiters.

9. Of the four major elements outlined by the WCAG (Web Content Accessibility Guidelines), which one is mostly concerned with providing features such as screen readers and synthesized speech? **0 / 1 point**

- ○ Perceivable
- ○ Robust
- ○ Operable
- ◉ Understandable

Incorrect
Not quite. Please review the reading Recap: Accessibility.

10. When importing the `useState` hook the common practice is to use object destructuring. If object destructuring was not used, and all we had was the code snippet below, how could the `useState` hook still be used?] **1 / 1 point**

```
1   import React from "react";
```

- ○ `React.get(useState)`
- ○ `useState`
- ○ It could not be used
- ◉ `React.useState`

Correct
That's correct! The `useState` hook is part of the imported `React` library, and thus can be accessed using dot notation.