←

React test #3. Hooks

2023-03-27 03:00 → 2023-04-03 02:59

Description: https://github.com/rolling-scopes-school/tasks/tree/master/react/modules/module03

You must score at least 70% of points to pass. You have only 2 attempts.

Attempts: 2 attempts left.

To submit the task answer the questions.

1. Choose correct statement:

☑ const [testValue, setTestValue] = useState(42);
☑ const [testValue, setTestValue] = useState('42');
☑ const [testValue, setTestValue] = useState([{ text: 'some text' }]);
☐ const [testValue, testValue2, setTestValue, setTestValue2] = useState(42, '42');

2. Lets you subscribe to React context without introducing nesting:

○ useState
● useContext
○ useReducer
○ useMemo
○ useEffect
○ useCallback
○ useRef
○ none of above

3. It serves the same purpose as componentDidMount, componentDidUpdate, and componentWillUnmount in React classes, but unified into a single API.

○ useContext
○ useState
○ useReducer
● useEffect
○ useCallback
○ useRef
○ useMemo
○ none of above

4. When Hooks were introduced first time?

○ in 16.0 patch
● in 16.8 patch
○ in 17.0 patch
○ from the very beginning - when React were introduced

5. Choose correct statement about useLayoutEffect:

☑ it fires synchronously after all DOM mutations
☑ it used this to read layout from the DOM and synchronously re-render
☑ developers should prefer the standard useEffect when possible to avoid blocking visual updates
☑ The signature is identical to useEffect

6. Returns a mutable ref object whose .current property is initialized to the passed argument.

○ useEffect
○ useContext
● useRef
○ useReducer
○ useState
○ useCallback
○ useMemo
○ none of above

7. Choose correct statement:

□ There are plans to remove classes from React
☑ Hooks allow you to reuse stateful logic without changing your component hierarchy
□ Hooks don't let you split one component into smaller functions based on what pieces are related
☑ Hooks don’t contain any breaking changes

8. How called ESLint plugin that enforces rules of hooks?

☑ eslint-plugin-react-hooks
□ eslint-react-hooks
□ eslint-hooks-react
□ eslint-plugin-react-hooks-rules

9. Customizes the instance value that is exposed to parent components when using ref

○ useEffect
○ useMemo
○ useRef
○ useCallback
○ useState
○ useReducer
○ useContext
● none of above

10. Lets you manage local state of complex components with a reducer.

□ useEffect
□ useMemo
□ useRef
□ useCallback
☑ useState
☑ useReducer
□ useContext
□ none of above

11. What Rules of Hooks you must follow?

☑ Call Hooks from custom Hooks
□ Call Hooks from React class components.
☑ Don’t call Hooks from regular JavaScript functions.
☑ Don’t call Hooks inside loops, conditions, or nested functions.
☑ Call Hooks at the Top Level

12. Choose correct statement:

○ Custom Hooks are a React feature, rather than a convention.
○ Custom Hooks are a mechanism to reuse stateful logic and two components using the same Hook share one state
● A custom Hook is a JavaScript function whose name starts with ”use” and that may call other Hooks
○ Custom Hook gets isolated state only the first time, not each call

13. Choose correct statement:

☑ useEffect run after every render by default, but you can change it
□ useEffect called outside a component
☑ effects scheduled with useEffect don’t block the browser from updating the screen, same as like componentDidMount or componentDidUpdate
☑ using useEffect, you tell React that your component needs to do something after render

14. Returns a memoized value.

○ useContext
● useMemo
○ useRef
○ useState
○ useCallback
○ useReducer
○ useEffect
○ none of above

15. Returns a pair: the current state value and a function that lets you update it.

○ useState
○ useEffect
○ useContext
○ useReducer
○ useCallback
○ useMemo
○ useRef
○ none of above

Submit