React + JavaScript Quiz

Quiz Section (Student)

1. 1. What does the 'const' keyword in JavaScript do when declaring a React component?

- A. It allows the component to be reassigned later.
- B. It prevents the component from being reassigned.
- C. It automatically binds the component to the DOM.
- D. It makes the component stateful by default.

2. 2. Which of the following best describes props in React?

- A. Global variables shared between components.
- B. Built-in constants only available in JSX.
- C. Key-value pairs passed into components.
- D. Static values that never change.

3. 3. What does the 'useState' hook return?

- A. Only the current state value.
- B. A state object.
- C. An object with methods to control state.
- D. An array with the current state and a setter function.

4. 4. What triggers a re-render in React?

- A. Any variable assignment.
- B. DOM manipulation.
- C. A call to a setter function returned by useState.
- D. Declaring a new component.

5. 5. What does the spread operator {...obj} do in JSX?

- A. Deletes all keys in obj.
- B. Loops through obj and logs each value.
- C. Copies all properties from obj into a new object.
- D. Converts the object to an array.

6. 6. What is the main reason PopoverTrigger uses 'asChild' in ShadCN/Radix UI?

- A. To apply extra CSS styling.
- B. To allow the Popover to render multiple children.
- C. To avoid adding an unnecessary wrapper DOM element.
- D. To simplify the JSX syntax.

7. 7. Which best describes how React handles state updates?

- A. It immediately mutates the DOM.
- B. It queues a re-render of the affected component.
- C. It replaces the entire HTML file.
- D. It restarts the JavaScript application.

8. 8. In JavaScript, what does object destructuring do?

- A. Loops through the object.
- B. Duplicates the object's reference.
- C. Extracts values from the object into new variables.
- D. Deletes the object's keys.

9. 9. How many arguments does a React component technically receive?

- A. Multiple individual arguments.
- B. One object containing all props.
- C. It depends on how many props are passed.
- D. None.

10. 10. What is the purpose of React's virtual DOM?

- A. To reduce the size of HTML files.
- B. To store components in memory.
- C. To compare and efficiently update the actual DOM.
- D. To manage asynchronous data.

11. 11. In React, the function component takes ______ as its argument.

____

12. 12. The ______ operator is used to copy all properties of one object into another.

____

13. 13. The useState hook returns an array with the current state and a ______ function.

____

14. 14. React components re-render when their ______ changes.

____

15. 15. The Context API allows you to avoid ______ props through many levels.

____

16. 16. Destructuring allows you to extract values from ______ or arrays.

____

17. 17. React’s rendering is triggered by changes in ______ or props.

____

18. 18. React’s internal scheduling system is called the ______.

____

19. 19. A function component in React is essentially a ______.

____

20. 20. The prop onOpenChange is usually linked to a ______ function.

____

21. 21. Explain how React determines what parts of the DOM need to be updated when state changes.

____

22. 22. Describe the difference between props and state in React.

____

23. 23. Why does React use key-value pairs instead of positional arguments for component props?

____

24. 24. What happens internally when you call setState or setIsOpen in a component?

____

25. 25. Explain the benefits of using destructuring in function components.

____

26. 26. Describe the relationship between React components and the browser’s event loop.

____

27. 27. How does React’s Context API help avoid prop drilling?

____

28. 28. Explain the lifecycle of a click event triggering a React state update and resulting DOM change.

---

29. 29. Why does React favor declarative UI code over imperative DOM manipulation?

---

30. 30. Give an example of how you could simulate useState in plain JavaScript and explain the limitations.

Answer Key

31. 1. What does the 'const' keyword in JavaScript do when declaring a React component?

Answer: B

32. 2. Which of the following best describes props in React?

Answer: C

33. 3. What does the 'useState' hook return?

Answer: D

34. 4. What triggers a re-render in React?

Answer: C

35. 5. What does the spread operator {...obj} do in JSX?

Answer: C

36. 6. What is the main reason PopoverTrigger uses 'asChild' in ShadCN/Radix UI?

Answer: C

37. 7. Which best describes how React handles state updates?

Answer: B

38. 8. In JavaScript, what does object destructuring do?

Answer: C

39. 9. How many arguments does a React component technically receive?

Answer: B

40. 10. What is the purpose of React’s virtual DOM?

Answer: C

41. 11. In React, the function component takes ______ as its argument.

Answer: props

42. 12. The ______ operator is used to copy all properties of one object into another.

Answer: spread

43. 13. The useState hook returns an array with the current state and a ______ function.

Answer: setter

44. 14. React components re-render when their ______ changes.

Answer: state

45. 15. The Context API allows you to avoid ______ props through many levels.

Answer: drilling

46. 16. Destructuring allows you to extract values from ______ or arrays.

Answer: objects

47. 17. React’s rendering is triggered by changes in ______ or props.

Answer: state

48. 18. React’s internal scheduling system is called the ______.

Answer: scheduler

49. 19. A function component in React is essentially a ______.

Answer: JavaScript function

50. 20. The prop onOpenChange is usually linked to a ______ function.

Answer: state setter

51. 21. Explain how React determines what parts of the DOM need to be updated when state changes.

Answer: React uses a virtual DOM to compare the current UI with the previous UI. It then performs a diff and calculates the minimal set of changes required. These changes are then efficiently applied to the real DOM.

52. 22. Describe the difference between props and state in React.

Answer: Props are passed from parent to child and are read-only. State is local to a component and can be updated with useState or similar hooks.

53. 23. Why does React use key-value pairs instead of positional arguments for component props?

Answer: Using key-value pairs makes code more readable, allows for optional and out-of-order arguments, and enables prop spreading with {...obj}.

54. 24. What happens internally when you call setState or setIsOpen in a component?

Answer: React schedules a re-render of the component, recalculates the JSX output, diffs it against the previous version, and updates only the changed parts of the DOM.

55. 25. Explain the benefits of using destructuring in function components.

Answer: Destructuring makes code more concise and readable by allowing direct access to props without repeatedly referencing props.propName.

56. 26. Describe the relationship between React components and the browser's event loop.

Answer: React components are rendered and updated in response to events. React relies on the browser's event loop to listen for interactions, and then it re-renders components as needed.

57. 27. How does React's Context API help avoid prop drilling?

Answer: The Context API allows data to be shared across the component tree without passing props manually at every level.

58. 28. Explain the lifecycle of a click event triggering a React state update and resulting DOM change.

Answer: Click event triggers a handler → handler calls setState → React re-renders component → virtual DOM is updated → real DOM is efficiently updated.

59. 29. Why does React favor declarative UI code over imperative DOM manipulation?

Answer: Declarative code is easier to read, reason about, and maintain. React abstracts DOM updates so you describe what the UI should look like at any point in time.

60. 30. Give an example of how you could simulate useState in plain JavaScript and explain the limitations.

Answer: By manually updating a variable and re-rendering DOM manually using innerHTML, but lacks reactivity, encapsulation, and performance optimization of React.