

React JS Interview Questions and Answers

1. What is the difference between a functional and class component?

Functional components are JavaScript functions that return React elements. They are simpler and recommended for most use cases.

Class components are ES6 classes that extend `React.Component` and can hold state and lifecycle methods. With the introduction of hooks,

functional components can also manage state and side effects, making class components less necessary in modern React.

2. What are React hooks? Explain `useState` and `useEffect` with examples.

Hooks are functions that let you use state and lifecycle features in functional components.

- `useState`: Allows you to add state to functional components.

Example:

```
const [count, setCount] = useState(0);  
  
setCount(count + 1);
```

- `useEffect`: Handles side effects (e.g., data fetching, subscriptions).

Example:

```
useEffect(() => {  
  console.log('Component mounted or updated');  
}, [count]); // Runs when 'count' changes
```

3. How does the virtual DOM work?

The virtual DOM is a lightweight copy of the real DOM. When state changes, React updates the virtual DOM first, then compares it (diffing)

with the previous virtual DOM to find what changed. Finally, it updates only the changed parts in

the real DOM, which improves performance.

4. What are props vs. state in React?

Props are read-only inputs passed from a parent component to a child component. State is a local, mutable data store used within components.

Props make components reusable and configurable, while state handles dynamic data that changes over time.

5. What happens during the React render lifecycle?

In class components: mounting (constructor, render, componentDidMount), updating (shouldComponentUpdate, render, componentDidUpdate),

and unmounting (componentWillUnmount). In functional components, these are handled by hooks like useEffect.

6. What is the difference between let, const, and var?

- var: Function-scoped, hoisted, can be re-declared.
- let: Block-scoped, cannot be re-declared in the same scope, can be reassigned.
- const: Block-scoped, cannot be re-declared or reassigned.

7. What is the difference between == and ===?

== compares values after type coercion (loose equality), whereas === compares both value and type (strict equality).

Example: '5' == 5 -> true, but '5' === 5 -> false.

8. Explain event delegation in JavaScript.

Event delegation allows you to handle events at a higher-level element (like a parent), rather than adding event listeners to each child element.

This is efficient for dynamic elements added later in the DOM.

9. How does async/await work?

async/await is syntax to work with Promises in a more readable way.

An async function returns a Promise. await pauses the function execution until the Promise resolves.

Example:

```
async function fetchData() {  
    const res = await fetch(url);  
    const data = await res.json();  
}
```

10. What are closures?

A closure is a function that remembers variables from its outer scope even after the outer function has finished executing.

Useful for data privacy and creating function factories.

11. Explain how Redux works and its core concepts.

Redux is a state management library for JavaScript apps.

Core concepts:

- Store: The global state container.
- Actions: Descriptions of events.
- Reducers: Pure functions that specify how state changes in response to actions.
- Dispatch: Sends actions to the store to update state.

12. What is a middleware in Redux? (e.g., Redux Thunk)

Middleware in Redux intercepts actions before they reach the reducer. Useful for handling async operations, logging, etc.

Redux Thunk allows you to return a function instead of an action, to perform async logic like API calls.

13. What is the difference between local state and global state?

Local state is specific to a component and managed using `useState`. Global state is shared across multiple components and managed using tools like `Redux` or `Context API`.

14. How do you fetch data from an API in React?

Use `useEffect` to trigger data fetch and `useState` to store the data.

Example:

```
useEffect(() => {  
  fetch(url)  
    .then(res => res.json())  
    .then(data => setData(data));  
}, []);
```

15. What is the role of Webpack and Babel?

- Webpack is a module bundler that packages JavaScript files and assets.

- Babel is a JavaScript compiler that converts modern JavaScript (ES6+) into backwards-compatible version for older browsers.

16. What is NPM? What's in a package.json file?

NPM (Node Package Manager) is used to manage packages (libraries) in JavaScript projects.

`package.json` contains project metadata and lists dependencies, scripts, and configuration.