

MERN INTERVIEW PREPARATION

React Interview Questions (3 Years Experience)

◆ React Basics

1. What is React and why do we use it?
 2. What are components in React? Difference between class and function components.
 3. Explain JSX. Why can't the browser read JSX directly?
 4. What are props and state? Difference between them.
 5. What are controlled vs uncontrolled components?
 6. What is the virtual DOM and how does React use it?
 7. What are keys in React? Why are they important in lists?
 8. What is reconciliation in React?
-

◆ React Hooks

9. What are React hooks? Why were they introduced?
 10. Difference between `useState` and `useReducer` . When would you use which?
 11. Explain `useEffect` . What are dependencies?
 12. How to prevent infinite loops in `useEffect` ?
 13. Difference between `useMemo` and `useCallback` .
 14. Difference between `useMemo` and `React.memo` .
 15. What is `useRef` and where is it used?
 16. Explain custom hooks. Have you written any in your projects?
-

◆ Component Rendering & Optimization

17. What causes a React component to re-render?
 18. How do you prevent unnecessary re-renders?
 19. Difference between `React.memo` , `useMemo` , and `useCallback` .
 20. What is code splitting? How do you implement it in React?
 21. What is lazy loading in React?
 22. How would you optimize a React app's performance?
-

◆ State Management

- 23. How do you manage state in React?
 - 24. Local state vs global state.
 - 25. What is Context API? When would you use it?
 - 26. Have you worked with Redux or Redux Toolkit? Explain the flow.
 - 27. Difference between Redux and Context API.
 - 28. What are actions, reducers, and store in Redux?
 - 29. What is middleware in Redux? Example of custom middleware.
-

◆ React Query / Data Fetching

- 30. How do you fetch data in React?
 - 31. Have you used React Query? What are its benefits?
 - 32. Difference between React Query and Redux.
 - 33. What is caching in React Query?
-

◆ Advanced React

- 34. What is the difference between CSR, SSR, and SSG?
 - 35. What is hydration in React (used in Next.js)?
 - 36. What are error boundaries? How do you create one?
 - 37. What is React Fiber?
 - 38. How does React handle concurrency? (React 18 concurrent features)
 - 39. What is `useTransition` ?
 - 40. What is Suspense in React?
-

◆ Forms & Events

- 41. Difference between controlled vs uncontrolled forms.
 - 42. How do you handle form validation in React?
 - 43. Difference between `onChange` and `onInput` .
-

◆ Testing

- 44. How do you test React components?

45. What are unit tests vs integration tests in React?
 46. Have you used Jest / React Testing Library?
-

◆ Real-World Scenarios

47. Suppose a list of 10,000 items is rendering slowly — how do you optimize it?
 48. How do you handle authentication in React?
 49. How do you secure tokens in React (JWT, HTTP-only cookies)?
 50. How do you implement role-based access control (RBAC) in React?
-

◆ General JavaScript (always tested with React)

51. Explain closures in JavaScript.
52. Explain event delegation.
53. Difference between `var`, `let`, `const`.
54. Explain `this` keyword in JS.
55. Difference between `==` and `===`.
56. What is `async/await`? How is it different from promises?
57. What is debouncing and throttling?
58. What are higher-order functions in JS?

JavaScript Interview Questions (3 Years Experience)

◆ Core Concepts

1. What are the different data types in JavaScript?
 2. Difference between `var`, `let`, and `const`.
 3. Explain `this` in JavaScript. How does it behave in arrow functions vs normal functions?
 4. What are closures? Can you give an example?
 5. What is hoisting in JavaScript?
 6. Explain scope: block scope vs function scope vs global scope.
 7. What are higher-order functions?
 8. Explain `call`, `apply`, and `bind` with examples.
 9. Difference between shallow copy vs deep copy in JS.
 10. What are template literals and tagged templates?
-

◆ Functions & Execution

11. What is the difference between synchronous and asynchronous JS?
 12. Explain the event loop in JavaScript.
 13. What are microtasks vs macrotasks?
 14. What is debouncing and throttling? Where do you use them?
 15. What are IIFEs (Immediately Invoked Function Expressions)?
 16. What are generators in JavaScript?
 17. Explain currying in JavaScript.
 18. What is memoization in JS?
-

◆ Objects & Arrays

19. How do you merge two objects in JS?
 20. Difference between `Object.freeze`, `Object.seal`, and `Object.preventExtensions`.
 21. How do you clone an object in JS?
 22. Difference between `map`, `forEach`, `filter`, and `reduce`.
 23. What is the difference between `in` operator, `hasOwnProperty`, and `Object.keys`.
 24. How do you remove duplicates from an array?
 25. How do you flatten a nested array?
 26. What is destructuring in JS?
-

◆ Prototypes & OOP

27. What are prototypes in JavaScript?
 28. How does prototypal inheritance work?
 29. Difference between class and function constructor in JS.
 30. Explain static methods in JS classes.
 31. What is `super` in JavaScript classes?
 32. Difference between classical inheritance and prototypal inheritance.
-

◆ Promises & Async

33. What are promises?
34. Difference between `Promise.all`, `Promise.any`, `Promise.allSettled`, and `Promise.race`.

35. Explain `async/await`.
 36. What happens if you don't use `await` inside an `async` function?
 37. How do you handle errors in `async/await`?
-

◆ DOM & Browser

38. Difference between `document.querySelector` and `getElementById`.
 39. What are event bubbling and event capturing?
 40. What is event delegation?
 41. Explain `localStorage`, `sessionStorage`, and cookies.
 42. How do you make an API call in vanilla JS?
 43. Difference between `innerHTML`, `innerText`, and `textContent`.
 44. What is the difference between synchronous and asynchronous script loading (`async` vs `defer`)?
-

◆ Advanced Concepts

45. What is `undefined` vs `null`?
 46. Difference between `==` and `===`.
 47. What is NaN in JavaScript?
 48. Explain immutability in JavaScript.
 49. What are modules in JS (ES Modules vs CommonJS)?
 50. Difference between `import` and `require`.
 51. Explain tree shaking in JavaScript bundlers.
 52. What are `WeakMap` and `WeakSet`?
 53. What is optional chaining (`?.`) and nullish coalescing (`??`)?
 54. What is the difference between `for...in`, `for...of`, and `forEach`?
 55. What are symbols in JS? Why are they used?
-

◆ Scenario / Practical Questions

56. How would you implement a polyfill for `bind`?
57. How do you check if an object is an array?
58. How do you reverse a string in JavaScript?
59. How do you implement a debounce function?
60. How do you implement a throttle function?
61. How would you deep-clone a nested object?

- 62. How would you merge two sorted arrays?
- 63. How do you detect memory leaks in JS?
- 64. How do you improve performance of a large array loop?