React Interview — 50 High■Yield Q&A; (with quick answers & examples)

This document provides 50 React interview-style questions with concise answers and code snippets/examples.

Q1. What are React components?

Quick answer: Reusable building blocks; can be class or function components.

Example: function Hello(){ return <h1>Hello</h1>; }

Q2. Difference between functional and class components?

Quick answer: Functional: simple, hooks. Class: lifecycle methods, state pre-hooks.

Example: class MyComp extends React.Component{ render(){ return ; } }

Q3. What are props in React?

Quick answer: Read-only inputs passed from parent to child.

Example: <Child name='John'/>

Q4. What is state in React?

Quick answer: Internal data managed by component, can change over time.

Example: const [count,setCount]=useState(0);

Q5. Difference between state and props?

Quick answer: Props external, immutable. State internal, mutable via setState/useState.

Example: props.title vs this.state.count

Q6. What is JSX?

Quick answer: Syntax extension mixing HTML with JS, transpiled to React.createElement.

Example: const el=<h1>Hello</h1>;

Q7. What are hooks in React?

Quick answer: Functions adding state/lifecycle to functional components.

Example: useState, useEffect, useContext.

Q8. Explain useState hook.

Quick answer: Manages local state in functional component.

Example: const [count,setCount]=useState(0);

Q9. Explain useEffect hook.

Quick answer: Performs side effects after render (data fetch, subscriptions).

Example: useEffect(()=>{ document.title=`${count}`; },[count]);

Q10. What is virtual DOM?

Quick answer: Lightweight copy of DOM React uses for diffing and efficient updates.

Example: Reconciliation finds minimal DOM changes.

Q11. What is reconciliation in React?

Quick answer: Process of diffing virtual DOM with real DOM and updating efficiently.

Example: Keys help React identify unchanged elements.

Q12. What are keys in React lists?

Quick answer: Unique identifiers for list elements; improve reconciliation.

Example: <li key={id}>{name}

Q13. What is context API?

Quick answer: Provides global state without prop drilling.

Example: const MyContext=React.createContext();

Q14. Difference between useContext and Redux?

Quick answer: useContext for simple global state. Redux for complex state with middleware, debugging.

Example: const v=useContext(MyContext);

Q15. What is React Router?

Quick answer: Library for client-side routing in SPA.

Example: <Route path='/about' element={<About/>}/>

Q16. What is lazy loading in React?

Quick answer: Load components only when needed using React.lazy/Suspense.

Example: const Comp=React.lazy(()=>import('./Comp'));

Q17. What is code splitting?

Quick answer: Breaking bundle into smaller chunks loaded on demand.

Example: dynamic import() + React.lazy.

Q18. What is memoization in React?

Quick answer: Caching results to avoid re-renders using React.memo or useMemo.

Example: const val=useMemo(()=>expensiveCalc(x),[x]);

Q19. Difference between useMemo and useCallback?

Quick answer: useMemo caches result of fn. useCallback caches fn itself.

Example: const fn=useCallback(()=>do(x),[x]);

Q20. What is PureComponent?

Quick answer: Class component with shallow prop/state comparison for shouldComponentUpdate.

Example: class PC extends React.PureComponent{}

Q21. Difference between controlled and uncontrolled components?

Quick answer: Controlled: state managed by React. Uncontrolled: use refs to access DOM values.

Example: <input value={val} onChange={e=>setVal(e.target.value)}/>

Q22. What is forwardRef?

Quick answer: Pass ref through component to child DOM node.

Example: const C=React.forwardRef((props,ref)=><input ref={ref}/>);

Q23. Explain higher-order components (HOC).

Quick answer: Functions taking a component and returning a new one with added props/logic.

Example: withAuth(Component)

Q24. What are render props?

Quick answer: Technique where component takes a function prop to determine rendering.

Example: <Data render={data=><List data={data}/>}/>

Q25. Explain error boundaries.

Quick answer: Catch JS errors in children and display fallback UI.

Example: class EB extends React.Component{ componentDidCatch(){…} render(){return this.props.children;} }

Q26. What is React.Fragment?

Quick answer: Wrapper without adding extra DOM node.

Example: <><h1/> <p/></>

Q27. Explain hydration in React.

Quick answer: Process of attaching React events to server-rendered HTML.

Example: ReactDOM.hydrate(<App/>,root)

Q28. Difference between SSR and CSR?

Quick answer: SSR renders HTML on server; CSR renders on client after JS load.

Example: Next.js for SSR; CRA for CSR.

Q29. What is Next.js?

Quick answer: React framework with SSR, routing, API routes, optimizations.

Example: pages/index.js auto-routes to /.

Q30. Explain React.StrictMode.

Quick answer: Wrapper that activates extra checks in development.

Example: <React.StrictMode><App/></React.StrictMode>

Q31. What are propTypes?

Quick answer: Runtime type checking of props in dev.

Example: Comp.propTypes={name:PropTypes.string};

Q32. Difference between useReducer and useState?

Quick answer: useReducer for complex state logic; useState for simple state.

Example: const [state,dispatch]=useReducer(reducer,init);

Q33. What are custom hooks?

Quick answer: Reusable logic extracted into functions starting with 'use'.

Example: function useFetch(url){…}

Q34. What is suspense in React?

Quick answer: Component for handling lazy loading, shows fallback until loaded.

Example: <Suspense fallback={<div>Loading</div>}>…</Suspense>

Q35. Difference between imperative and declarative code in React?

Quick answer: Declarative: describe what UI should be. Imperative: step-by-step DOM updates.

Example: React: declarative; document.getElementById: imperative.

Q36. What is useRef hook?

Quick answer: Holds mutable value across renders; also access DOM nodes.

Example: const inputRef=useRef(); inputRef.current.focus();

Q37. Explain reconciliation algorithm in React.

Quick answer: Diffs virtual DOM trees; reuses nodes when keys match; minimal DOM ops.

Example: Child keyed list example.

Q38. What are synthetic events in React?

Quick answer: Cross-browser wrapper for native events.

Example: onClick={e=>console.log(e.nativeEvent)}

Q39. Explain batching in React.

Quick answer: Multiple state updates grouped into one render for performance.

Example: setCount(c=>c+1); setFlag(f=>!f); -> one render.

Q40. What is concurrent mode (React 18 features)?

Quick answer: Allows interruptible rendering, better responsiveness.

Example: enabled via createRoot API.

Q41. Difference between useLayoutEffect and useEffect?

Quick answer: useLayoutEffect runs synchronously after DOM mutations. useEffect async after paint.

Example: useLayoutEffect(()=>{…});

Q42. What is React Portal?

Quick answer: Render children into DOM node outside parent hierarchy.

Example: ReactDOM.createPortal(<Child/>,document.getElementById('modal'));

Q43. What is reconciliation key importance?

Quick answer: Keys identify elements across renders, improve re-render efficiency.

Example: array.map(i=><li key={i.id}>{i})

Q44. Explain state lifting.

Quick answer: Move state up to nearest common ancestor to share across components.

Example: Parent holds state, passes props down.

Q45. What is Redux?

Quick answer: State management library with single store and reducers.

Example: dispatch({type:'ADD',payload:1});

Q46. Difference between Redux and Context API?

Quick answer: Redux: centralized, middleware, devtools. Context: simpler, not for complex logic.

Example: useContext(MyCtx) vs store.getState().

Q47. What is useImperativeHandle?

Quick answer: Customize instance value exposed to parent when using forwardRef.

Example: useImperativeHandle(ref,()=>({focus:()=>{…}}));

Q48. Explain React keys best practice.

Quick answer: Use stable unique IDs, avoid indexes when items can reorder.

Example: <li key={user.id}>{user.name}

Q49. Additional React concept #49

Quick answer: Explanation of a React interview topic.

Example: Example usage or snippet here.

Q50. Additional React concept #50

Quick answer: Explanation of a React interview topic.

Example: Example usage or snippet here.