

React Js Interview Questions with indepth explanations.

by Anshuman Kundu

1. What is ReactJs?

1. It is an Open-source Javascript library used for building User interfaces introduced by Facebook.
2. Combines HTML, CSS and JS to create Single Page Web applications using JSX.
3. Frequent Data changes can take place in applications without requiring a page reload.
4. React mainly focusses on its component based structure which inherits the code reusability concept in Web development.

2. Explain the key features of ReactJs.

1. Virtual DOM
2. Component based structure (code reusability)
3. JSX - Javascript extension for writing HTML, CSS and JS code all together.
4. Huge Ecosystem.
5. Hooks
6. Inbuilt state management tools.
7. Unidirectional (data flow can be only from parent to children)

3. What is JSX?

1. Javascript extension is an extension which helps coders write readable code combining HTML, CSS and JS.
2. Every tag in HTML is a DOM element which helps build the website.
3. While creating a DOM element in React without JSX we will have to **const element = React.createElement('h1', null, 'Hello, World!');** The first parameter is the type of DOM Element, the second is props and the third is the content the element will display in the browser.
4. While creating websites we need to create lots and lots of DOM elements and by the above code it seems very rigourous to code.
5. With JSX we can simply write `<h1>Hello World</h1>` inside a component and it displays what we need.

4. But the browser does not understand JSX, how will it process the code?

1. The browser understand JS syntax only and not JSX syntax.
2. There are inbuilt libraries in Javascript like **Babel** which converts the JSX code to JS syntax before processing.

5. What is DOM and what is Virtual DOM?

1. DOM - Document Object Model.
2. It is a tree like structure representing all the elements and attributes need to build a Website. It includes elements like div, h1, p, attributes like class, id, and text nodes.
3. When we create a website using HTML, CSS and JS, let suppose user triggers any action and certain things like variable values change, it directly changes the things in the DOM tree which is the actual website tree. This slows a lot of things in the website due to the re rendering of the entire DOM although the changes are too small.
4. React actually copies the real DOM and when a user triggers any action, the changes are first given to the Virtual DOM. Then the virtual dom compares the changes with the Real DOM and only the changes are updated instead of the whole DOM being re rendered which actually fastens DOM procedure and eventually the website.

6. What are components in React?

1. Components are reusable code fragments where actually the code is written.
2. It can be used any where in the React src directory.
3. So when u have written a code for a Navbar and several pages in the website needs the navbar, you donot have to write the code for it everytime in every page that needs it. Just write it once save it using `Navbar.jsx` and it can be imported wherever needed like `<Navbar/>`

7. What are functional and class components?

1. Class Components: Defined as Javascript classes -> `class AppComponent extends Component{ render(){ return(your code goes here) }}`
2. Functional Components: Defined as Javascript functions -> `function AppComponent(){ return (write code goes here.) }`

8. What are purpose of props in React?

1. In a React component **A** there is a variable named **name**.
2. There is a child component **B** being called inside **A** which also needs **name**, then you donot need to create a new variable inside **B** rather you can just pass the variable **name** from A to B like: `function A() { return { variable name='Anshuman'; <"B nameProp={name}"> } } function B(props) { return { <"h1"> {props.nameProp}</h1"> } }`

9. What is a state and how do you update state in React?

1. State is the current value of a variable.
2. React provides inbuilt State hook for state management called **useState()**.
3. Can be updated using : `const [count, setCount] = useState(1);` here is the we declared a variable count with initial value 1 and a function setCount which directly changes the count value. **setCount(count+1)** will directly change the current count value adding 1 to it.

10. What is the significance of keys in React Lists?

1. Keys are unique ids provided to list items which helps in processing the DOM changes.
2. When certain list items are changed in React Lists, the Virtual DOM identifies the unique list keys which are changed and differs the Real DOM accordingly.

11. What is the meaning of Lifting a state up?

1. When we remove the state values from a child component to its parent component such that more child components can access the state values is termed as Lifting the state up.

12. How do we handle events in React?

1. React JSX has inbuilt functions/event handlers like onClick, onChange, onSubmit, etc. which can be assigned to DOM elements in order to provide you own actions to those events.

13. What is prop drilling?

1. It refers to the passage of props from parent -> child -> grand child -> and so on.

14. Why to avoid prop drilling?

1. It makes code readability poor and complex.
2. Debugging is difficult as you have to go through numerous components.
3. Affects code performance.

15. What are the methods to avoid prop drilling?

1. Use Context API
2. Use callback functions
3. Custom hooks

16. What is this keyword?

1. **this** keyword points to the instance of the class and any parameter being stored can be accessed using this keyword inside the same class.
2. Just like other languages which supports OOPs concept, this works same in all.

17. What is the role of Fragment in JSX?

1. Fragment wraps small elements and containers within it. (`<>{code}</>`)
2. It caters the unnecessary usage of react blocks(divs) which might effect the UI of the web application.

18. What is use of spread operator in JSX?

1. `{...props}` is used to pass all the properties of a javascript object to other components in React.
2. It avoids manual listing of every key value pair being passed to any component as prop.

19. What is the difference between a transpiler and compiler?

1. Transpiler converts high level language to another high level language.
2. Converting JSX to Javascript is called transpilation.
3. Compiler converts high level language to another low level language.
4. Converting C++ to machine level code/ bytecode is called compilation.

20. What are the main files must required to start a React web application?

1. node_modules - contains all your libraries that you are going to use to build the application. Anything added using **npm install library-name** appears in these in node_modules. (It's huge in size)
2. public folder - contains **index.html**
3. src folder - contains **App.js & index.js** Q12. How does React loads the web page and displays it to the user?
4. User requests the frontend server by clicking on some website link.
5. The frontend server finds the **index.html** folder.
6. **index.html** then finds and loads the entry point of javascript which is **index.js** using React libraries.
7. **index.js** then finds the root element in the **index.html** and replaces it by **App.js**.
8. **App.js** is the root component of the web application and contains all the child components which is displayed via **index.html**.
9. package.json - It is the json file which contains the summary of the project including, name, react library versions, etc.

21. What are the differences between React and Angular?

1. React is a Javascript library -> Angular is a complete framework.
2. React is simple to learn -> Angular is complex to learn.
3. React uses Virtual DOM -> Angular uses real DOM.
4. React is fast -> Angular is slow as compared to React.
5. React uses external libraries for complex functionalities -> Angular has built in support for everything because it's a framework.

22. What are the differences between a library and a framework?

1. Library is a combination of reusable codes distributed in terms of objects, modules, classes, components, etc. -> Angular is a set of tools, libraries, etc.
2. Your application(you) controls the usage and integration of library in your code where as Framework controls the structure of the web application.
3. Library is lightweight and can be manipulated as desired -> Angular is heavy weight which has built in functionalities and its not flexible.

23. What is NPM and what is the role of node_modules in React?

1. NPM - Node Package Manager is responsible for managing and keeping all the modules, libraries and dependencies in a folder called node_modules which are necessary for the React application development.

24. What is the role of public folder in React?

1. Public folder is responsible for keeping all the static assets of a web application.
2. These assets are directly served to the web browser when website is opened first.
3. It contains favicons, static images, **index.html**, etc.

25. What is the role of index.html in React?

1. *index.html* is the entry point of our web application.
2. It contains a div with id root in the body which contains the content being served and displayed in the UI.

26. What is the role of src folder in React?

1. src folder is responsible for containing all the dynamic assets needed for our React application.
2. **App.js**, **index.js** are must in src folder, without these your web application won't run.
3. You can create components inside this folder.

27. What is the role of index.js in React?

1. index.js finds the root element in the index.html file using ReactDOM.
2. ReactDOM is a react library which is used to point reference to the DOM and deals with the DOM directly.
3. After finding the root element it replaces the content of root with App.js file.

28. What is the role of App.js in React?

1. App.js is the root component of our web application.
2. It can incorporate components, routing and usually used to structure the layout of our application.

29. Why is export default used in React?

1. Whenever a .js file is created everything is kept inside a javascript function.
2. This function is available to be imported by other components in the application only if it has been exported where it has been written.

30. What are the benefits of React Hooks?

1. When seen from behind Hooks are basically Javascript classes with specific logics.
2. Hooks are made to avoid using Javascript classes with huge code snippets and promote functional programming.
3. It keeps our code concise and makes it more readable and reusable.

31. What are two most important rules in React Hooks?

1. Hooks can only be called within React Functional components or custom hooks.
2. Hooks can only be called on the topmost level of a React Functional component or any custom hook.

32. Describe the lifecycle of a React component.

1. Mounting Phase: The phase when a component mounts.
2. Updating Phase: The phase when a component is updated.
3. Unmounting Phase: The phase when a component is unmounted(destroyed).

33. What are functions in Mounting Phase of React lifecycle?

1. constructor(): Used to initialise event handlers and states.
2. getDerivedStateFromProps(props, state): Used to update the state values when new props are received just before the component mounts.
3. render(): Renders the UI
4. componentDidMount(): Invoked just after the component mounts, can be used for API calls.

34. What are functions in Mounting Phase of React lifecycle?

1. getDerivedStateFromProps(props, state): Handles changes in the prop values and updates the states accordingly.
2. shouldComponentUpdate(nextProps, nextState): Decides whether the component should be re-rendered or not based on the new props and state values.
3. render(): Renders the UI with the most updated changes.
4. getSnapshotBeforeUpdate(prevProps, prevState): Stores the props and state values just before the most recent render of the component.

5. `componentDidUpdate(prevProps, prevState, snapshot)`: Invoked just after the component is updated after props or state changes.

35. What are functions in Unmounting Phase of React lifecycle?

1. `ComponentWillMount()`: Invoked just before the component is unmounted or destroyed cancelling network calls, unbinding event listeners, etc.

36. Explain `useEffect` hook in React.

1. **`useEffect`** hook works as a side effect in React.
2. It is used to perform some action just after any component is mounted.
3. It works as a replacement of `componentDidMount()`, `componentDidUpdate()` and `componentWillMount()` in Functional components.

37. Explain `useMemo` hook in React.

1. It is used to memoize the result of a function.
2. It returns a memoized value.
3. Useful for optimizing performance by caching the result of an expensive computation.
4. Re-runs the function and updates the cached value only if the dependencies change.

38. Explain `useCallback` hook in React.

1. It is used to memoize a function instance.
2. Returns a memoized version of the callback function.
3. Useful for preventing unnecessary re-creation of callback functions in child components.
4. Re-creates the function instance only if the dependencies change.

39. How do you perform data fetching in React?

1. Generally two methods for fetching data in React.
2. Using `fetch` function which is a javascript native which accepts the api url as argument : **`await fetch('api-url');`**
3. Using `axios` which is a third party library and needs to be installed before usage, accepts url as argument and request method (get, post, patch, delete) should be specified: **`axios.get('api-url');`**

40. What is React router and it's key components?

1. First installation: **`npm install react-router-dom`** to use router functionality which decides which component to render on a specific frontend url path.
2. `BrowserRouter` wraps our application and provides router functionality.
3. `<"Route path="/home" component={}<"Home"/>/>` : This says when **`http:localhost:3000/home`** path will hit we should render the component named Home.
4. This meets the purpose of lazy loading which says that render a component only when it is required.

41. What is the concept of error boundaries in React?

1. It's a feature in React which prevents the UI crash whenever an error occurs.
2. We just need to create an `ErrorBoundary` class or a function which will catch the error when a component mounts and performs the action like showing some error screen or a fallback.
3. We need to wrap our application with `ErrorBoundary` component.

42. What is Server side rendering (SSR) ?

1. Server loads the full HTML page with CSS and JS and sends it to the client browser which shows the UI.
2. Initial Load time can be slow.
3. Effective for SEO because when server renders the HTML elements, the SEO crawlers can read them properly.

43. What is Client side rendering (CSR) ?

1. Client browser gets the HTML page with links of CSS and JS.
2. Client browser then executes all the code, prepares the DOM and then we see the UI.
3. Initial Load time is generally fast.
4. Problems with SEO crawlers and is ineffective for SEO.

44. What is the concept of code splitting in React?

1. Code splitting is done to reduce the whole code size in smaller chunks.
2. It helps in dynamic import (lazy loading) i.e what is required, only that should be imported.
3. Reduces the load on the application which helps in smaller bundle size and faster load times.

45. What is the use of Context APIs in React?

1. It helps in declaring global entities such as variables, themes, etc.
2. When a certain entity is needed in the whole react application, then instead of defining the entities in every variable, context apis helps in declaring them as global and using the entities more seamlessly wherever needed.
3. Can be used in User Authentication, global data, themes.

46. How to optimize React applications?

1. Code splitting
2. Lazy loading
3. Memoizing
4. Using proper keys
5. Avoiding necessary re-renders
6. Avoiding prop drilling

47. What are async functions in Javascript?

1. Async functions in JavaScript are a type of function that enables asynchronous, non-blocking behavior.
2. They allow you to write asynchronous code in a more synchronous and readable manner using the `async` keyword.

48. What is the purpose of Promise object?

1. They represent the eventual completion (or failure) of an asynchronous operation.
2. They allow you to specify what happens when the operation succeeds or encounters an error.

49. What is purpose of await keyword?

1. Async functions always return some promise.
2. Adding an `await` keyword helps in to wait for the promise to resolve before moving into next code fragment.

50. Features of ES6 in Javascript?

1. EcmaScript helps you write more understandable and concise codes in Javascript.
2. Usage of `let` and `const`.
3. Prop destructuring.
4. Template literals: `console.log(Anshuman Kundu ${some variable});`

Copyright @ Anshuman Kundu