



React.js

INTERVIEW QUESTIONS

1-18



Artem
@like_a_senior

1. What is React.js?

React.js is an open-source JavaScript library developed by Facebook (Meta) for building user interfaces or UI components. It allows developers to create reusable UI components and efficiently manage the state of an application.

2. What are the key features of React?

- Component-Based
- Virtual DOM
- JSX
- One-Way Data Binding
- State and Props
- Lifecycle Methods
- React Hooks



Artem

@like_a_senior

3. What is a component in React?

A component in React is a reusable and self-contained piece of UI that can be composed together to build complex user interfaces. Components can have their own state, properties (props), and lifecycle methods.

4. What are the differences between functional components and class components in React?

- Functional Components are just JavaScript functions that return JSX. They are easier to read, write, and test. With React Hooks, functional components can also manage state and side effects.
- Class Components are JavaScript classes that extend `React.Component`. They have lifecycle methods, which can be useful for managing state and side effects. However, they have a more verbose syntax.



Artem

@like_a_senior

5. Explain Controlled Form Components.

In HTML, form elements such as `<input>`, `<textarea>`, and `<select>` typically maintain their own state and update it based on user input. In React, mutable state is typically kept in the state property of components, and only updated with `setState()`.

6. What is JSX in React?

JSX (JavaScript XML) is a syntax extension for JavaScript that allows you to write HTML-like code within your JavaScript code. React uses JSX to describe the structure of the UI components.

7. Explain the concept of conditional rendering.

Conditional rendering in React involves showing or hiding components or content based on certain conditions. This can be achieved using JavaScript conditional statements, ternary operators, or logical operators



Artem
@like_a_senior

8. Can we use **Typescript**, how?

Yes, we can use TypeScript with React. The most common way is to set up React using typescript template.

9. Explain the difference between **state, props, and context** in **React**.

- Props are used to pass data from a parent component to a child component. They are immutable and cannot be modified by the child component.
- State is used to manage a component's internal data and can be modified by the component itself. State changes trigger re-rendering of the component.
- React Context is a way to share data between components without having to pass props through each level of the component tree. It is particularly useful for sharing global or application-wide state.



Artem
@like_a_senior

10. How to start a new React project?

React recommends to use one of the React-powered frameworks:

- Next.js
- Remix
- Gatsby

We also can use Vite to start a new React project or Create React App

11. What is the purpose of the **key** prop in React?

The key prop is used to uniquely identify elements in a list of React components. It is important because it helps React efficiently update and re-render components in a list when their order or structure changes. A unique key enables React to determine which elements have been added, removed, or moved, resulting in optimized rendering.



Artem
@like_a_senior

12. What are **React Fragments**, and why are they used?

React Fragments are a way to group multiple elements without adding extra nodes to the DOM. They are often used when you need to return multiple elements from a component's render method, as JSX typically requires a single parent element.

13. Explain **Fragment Syntax**

There are two syntax of Fragment

- Long (or Standard) - `<React.Fragment>...</React.Fragment>`
- Shorthand (or Abbreviated) - `<>...</>`

There are not any differences between these two forms, but if we want to use Fragment inside a loop, we need to provide “key” and it’s possible only with Long form.



Artem

@like_a_senior

14. Explain React Data Binding.

React enforces a unidirectional data flow, where data flows in a single direction, from parent components to child components. This simplifies data management and makes it easier to understand how data changes affect the application.

15. What is Virtual DOM?

React uses a virtual representation of the actual DOM (Document Object Model) to optimize and speed up the rendering process. When changes are made to a component, React updates the virtual DOM, calculates the difference (diffing) between the previous and new virtual DOM, and then efficiently updates the real DOM to reflect the changes. This minimizes unnecessary manipulations of the actual DOM, leading to better performance.



Artem

@like_a_senior

16. What is the React lifecycle?

React components go through various lifecycle phases:
Mounting, Updating, and Unmounting

17. What is React Hooks? List the most popular ones.

React Hooks are functions that allow you to use state and other React features in functional components. They provide an alternative to class components and make it easier to manage state and side effects in functional components.

The most popular hooks:

- useState
- useRef
- useEffect
- useMemo
- useCallback
- useReducer



Artem

@like_a_senior

18. Explain React lifecycle methods. List the most popular ones.

In React, class components have lifecycle methods that allow you to execute code at specific points in a component's life. These methods can be categorized into three main phases. Class components are still supported by React, but we don't recommend using them in new code.

The most popular methods:

- `componentDidMount`
- `componentDidUpdate`
- `componentWillUnmount`
- `getSnapshotBeforeUpdate`
- `shouldComponentUpdate`



Artem

@like_a_senior