

REACT

1. Could you give a quick overview of how React works?

- React is a JavaScript library for building user interfaces. It uses a component-based architecture and a virtual DOM to efficiently update and render UI changes.

2. Would you mind explaining what makes React components reusable?

- React components are reusable because they are modular, self-contained, and can accept customizable data through props, allowing for flexible, repeatable UI elements.

3. Important Features of React:

- **Virtual DOM:** React uses a virtual DOM to efficiently update and render components, ensuring fast performance by minimizing direct DOM manipulations.
- **Component-Based Architecture:** React builds UI using reusable, isolated components, making code more modular, maintainable, and scalable.
- **Hooks:** React hooks allow functional components to manage state and side effects, making them powerful and more flexible.

4. Explain the MVC architecture.

- The Model-View-Controller (MVC) framework is an architectural/design pattern that separates an application into three main logical components Model, View, and Controller. Each architectural component is built to handle specific development aspects of an application. It isolates the business, logic, and presentation layer from each other.

5. How to create elements in a loop in React?

- In React, you can create elements in a loop by using JavaScript's `.map()` function or a `for` loop to iterate over data and return a list of components. Here's an example of how to do it with `.map()` since it's commonly used and aligns well with React's declarative nature.

```
<div>
  { items.map(item =>
    <ItemComponent key={item.id} item={item}/>
  )}
</div>
```

6. What is the purpose of the useMemo hook?

- The useMemo hook in React is used to memoize (cache) a computed value, preventing unnecessary recalculations on re-renders. It only recalculates the value when one of its dependencies changes, improving performance for expensive calculations or large data transformations in functional components.

7. Explain props and state in React with differences.

★ PROPS:

- The Data is passed from one component to another.
- It is Immutable (cannot be modified).
- Props can be used with state and functional components.
- Props are read-only.

★ STATE

- The Data is passed within the component only.
- It is Mutable (can be modified).
- The state can be used only with the state components/class component (Before 16.0).
- The state is both read and write.

8. What is virtual DOM in React?

- The virtual DOM in React is a lightweight, in-memory representation of the real DOM. React uses it to quickly detect changes and update only the parts of the actual DOM that have changed, improving performance and efficiency.

9. What is JSX?

- JSX is a syntax extension for JavaScript used in React that allows you to write HTML-like code within JavaScript. It makes it easier to create and structure UI elements by combining JavaScript logic with HTML-like tags.

Example of JSX: The name written in curly braces { } signifies JSX

```
const name = "Learner";
```

```
const element = (  
  <h1> Hello,  
    {name}.Welcome to GeeksforGeeks.  
  </h1>  
);
```

10. How to create an event in React?

- To create an event in React, attach an event handler like `onClick`, `onChange`, etc., to a JSX element. Define the handler function to specify the action when the event is triggered, such as updating state or executing logic.
- Javascript

```
function Component() {  
  
    doSomething(e);  
  
    {  
  
        e.preventDefault();  
  
        // Some more response to the event  
  
    }  
  
    return <button onClick={doSomething}></button>;  
  
}
```

11. How to write a comment in React?

- Multi-line comment: We can write multi-line comments in React using the asterisk format `/* */`.
- Single line comment: We can write single comments in React using the double forward slash `//`.

12. Explain the use of render method in React?

- React renders HTML to the web page by using a function called `render()`. The purpose of the function is to display the specified HTML code inside the specified HTML element. In the `render()` method, we can read props and state and return our JSX code to the root component of our app.

13. What is the state in React?

- The state is an instance of React Component Class that can be defined as an object of a set of observable properties that control the behaviour of the component. In other words, the State of a component is an object that holds some information that may change over the lifetime of the component.

14. Explain the difference between functional and class component in React?

★ Functional Components :

- A functional component is just a plain JavaScript pure function that accepts props as an argument .
- No render method used.
- Also known as Stateless components .
- React lifecycle methods (for example, componentDidMount) cannot be used in functional components.
- Constructors are not used.

★ Class Components:

- A class component requires you to extend from React. Component and create a render function .
- It must have the render() method returning JSX .
- Also known as Stateful components.
- React lifecycle methods can be used inside class components (for example, componentDidMount).
- Constructor is used as it needs to store state.

15. Explain one way data binding in React?

- One-way data binding in React means that data flows in a single direction, from parent components to child components. This ensures that changes in the component's state or props are passed down to the UI consistently, making data management more predictable and easier to debug.

16. What is this.setState function in React?

- We use the setState() method to change the state object. It ensures that the component has been updated and calls for re-rendering of the component. The state object of a component may contain multiple attributes and React allows using

setState() function to update only a subset of those attributes as well as using multiple setState() methods to update each attribute value independently.

17.What is the use of ref in React?

- In React, ref is used to directly access and interact with a DOM element or React component, allowing you to manage focus, text selection, or trigger animations and other imperative actions without re-rendering the component.

Syntax:

- `const node = this.myCallRef.current;`

18.What are hooks in React?

- Hooks in React are special functions that let you use state and other React features in functional components. Introduced in React 16.8, hooks allow you to manage state (useState), handle side effects (useEffect), optimize performance (useMemo), and more, without needing class components. They simplify component logic and encourage reusable, modular code.

19.Explain the useEffect hook in react?

- The useEffect hook in React eliminates the side effect of using class based components. It is used as an alternative to componentDidMount() method. The useEffect hook accepts two arguments where second argument is optional. `useEffect(function, dependency)`

20.What is context API?

- Context API is used to pass global variables anywhere in the code. It helps when there is a need for sharing state between a lot of nested components. It is light in weight and easier to use, to create a context just need to call `React.createContext()`. It eliminates the need to install other dependencies or third-party libraries like redux for state management. It has two properties Provider and Consumer.