Basic React.js Questions

1. **What is React.js?**
    - **Answer:** React.js is a JavaScript library developed by Facebook for building user interfaces, particularly single-page applications. It allows developers to create reusable UI components and efficiently update the DOM using a virtual DOM.
2. **What are the features of React?**
    - **Answer:**
        - Virtual DOM
        - Component-based architecture
        - JSX (JavaScript XML)
        - One-way data binding
        - Unidirectional data flow
        - Lifecycle methods
        - React Hooks
3. **What is JSX?**
    - **Answer:** JSX is a syntax extension for JavaScript used with React to describe what the UI should look like. It looks like HTML but is transformed into React.createElement calls.
4. **What is the Virtual DOM?**
    - **Answer:** The Virtual DOM is an in-memory representation of the real DOM elements. When state changes, React updates the virtual DOM first, compares it with the previous version, and updates only the changed parts of the real DOM (diffing algorithm).
5. **What are components in React?**
    - **Answer:** Components are independent, reusable pieces of UI. There are two types: **Functional Components** and **Class Components**.

Component and Props

6. **What is the difference between Functional and Class components?**
    - **Answer:** Functional components are stateless (before hooks) and simpler, while class components are stateful and include lifecycle methods. With React Hooks, functional components can now manage state and side effects.
7. **What are props in React?**
    - **Answer:** Props (short for properties) are inputs to components used to pass data from parent to child.
8. **Can props be changed inside a component?**
    - **Answer:** No, props are read-only. You should use state for values that need to change.
9. **How do you pass data between components?**

- **Answer:** Data is passed from parent to child using props. For child to parent communication, callbacks (functions) can be passed as props.

10. **What is `children` prop?**

- **Answer:** The `children` prop allows you to pass elements or components between opening and closing tags of a component.

State and Lifecycle

11. **What is state in React?**

- **Answer:** State is a JavaScript object that holds dynamic data and determines the component's behavior. When state changes, the component re-renders.

12. **How to update state in React?**

- **Answer:** Using `this.setState()` in class components or `useState()` in functional components.

13. **What are lifecycle methods in React?**

- **Answer:**
  - componentDidMount()
  - componentDidUpdate()
  - componentWillUnmount()

  These methods are called at different stages of a component's life.

14. **What is the use of `componentDidMount()`?**

- **Answer:** Called once when the component is mounted. Commonly used for API calls.

15. **What is the difference between state and props?**

- **Answer:**
  - **Props**: Immutable, passed from parent.
  - **State**: Mutable, owned by the component.

Hooks

16. **What are React Hooks?**

- **Answer:** Hooks are functions that let you use state and lifecycle features in functional components.

17. **What is `useState()`?**

- **Answer:** A Hook that lets you add state to functional components.

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

18. **What is `useEffect()`?**

- **Answer:** A Hook that lets you perform side effects in functional components.

```
useEffect(() => {
  // effect
  return () => {
    // cleanup
  };
}, [dependencies]);
```

19. **What is the difference between `useEffect()` and lifecycle methods?**

- **Answer:** `useEffect()` can replicate componentDidMount, componentDidUpdate, and componentWillUnmount within one API.

20. **What are custom hooks?**

- **Answer:** Functions that use React Hooks to extract and reuse logic across components.

Event Handling and Forms

21. **How do you handle events in React?**

- **Answer:** Events are handled using camelCase syntax and passing a function:

```
<button onClick={handleClick}>Click</button>
```

22. **What is synthetic event in React?**

- **Answer:** React's wrapper around native events to ensure cross-browser compatibility.

23. **How do you handle forms in React?**

- **Answer:** Use controlled components by syncing input values with component state.

24. **What are controlled components?**

- **Answer:** Form inputs whose value is controlled by React state.

25. **What are uncontrolled components?**

- **Answer:** Form inputs that maintain their own state; access values via refs.

Advanced Concepts

26. **What is React Router?**

- **Answer:** A library used for handling routing in React applications.

27. **What are Higher-Order Components (HOC)?**

- **Answer:** Functions that take a component and return a new component with additional props or logic.

28. **What is Context API?**

- **Answer:** Provides a way to share values like themes or authentication status between components without passing props manually.

29. **What is useContext?**

- **Answer:** A Hook to consume values from a React Context.

30. **What is the difference between `useRef` and `createRef`?**

- **Answer:**
  - `useRef()` is used in functional components.
  - `createRef()` is used in class components.

Performance Optimization

31. **What is memoization in React?**

- **Answer:** Memoization is caching the result of expensive function calls using `React.memo`, `useMemo`, or `useCallback`.

32. **What is React.memo?**

- **Answer:** A higher-order component that prevents unnecessary re-renders of functional components.

33. **What is useMemo()?**

- **Answer:** A Hook that memoizes expensive calculations.

```
const memoizedValue = useMemo(() => computeExpensive(), [dependencies]);
```

34. **What is useCallback()?**

- **Answer:** Memoizes functions to avoid unnecessary re-creation.

```
const memoizedFn = useCallback(() => doSomething(), [dependencies]);
```

35. **How to optimize React performance?**

- **Answer:**
    - Use React.memo
    - Lazy loading components
    - Code splitting
    - Avoid unnecessary re-renders
    - Use useMemo/useCallback

Testing and Debugging

36. **How do you debug React apps?**

- **Answer:** Using:
    - React Developer Tools
    - console.log()
    - Browser debugger
    - Breakpoints in IDE

37. **What is Jest?**

- **Answer:** A JavaScript testing framework used with React for unit and snapshot testing.

38. **What is Enzyme?**

- **Answer:** A testing utility from Airbnb for testing React components.

39. **What is snapshot testing?**

- **Answer:** Captures a rendered component and compares it to a stored snapshot to detect changes.

40. **What is the role of `React.StrictMode`?**

- **Answer:** Helps identify potential issues in the application like deprecated lifecycle methods or unsafe code.

Miscellaneous

41. **What is reconciliation in React?**

- **Answer:** The process React uses to update the DOM by comparing the new virtual DOM with the previous one and updating the changes.

42. **What is the key prop and why is it important?**

- **Answer:** A special prop used to uniquely identify items in a list to help React track changes efficiently.

43. **What is lifting state up?**

- **Answer:** Moving state to the closest common ancestor of components that need to share it.

44. **What is prop drilling?**

- **Answer:** Passing data from a parent to deeply nested child components via intermediate components.

45. **How do you avoid prop drilling?**

- **Answer:** Using the Context API or state management libraries like Redux.

React and External Libraries

46. **What is Redux?**

- **Answer:** A state management library that allows you to manage global state using actions and reducers.

47. **What are the key principles of Redux?**

- **Answer:**
  - Single source of truth
  - State is read-only
  - Changes via pure functions (reducers)

48. **What is useSelector and useDispatch in Redux?**

- **Answer:**
  - `useSelector` is used to read values from the Redux store.
  - `useDispatch` is used to dispatch actions to the store.

49. **How does React handle forms with Formik?**

- **Answer:** Formik is a library for building forms in React. It simplifies form validation, tracking state, and submission.

50. **What is the difference between server-side and client-side rendering in React?**

- **Answer:**
  - **Client-side rendering:** React renders in the browser after JS loads.
  - **Server-side rendering (SSR):** HTML is rendered on the server and sent to the browser, improving initial load and SEO.