**React Interview Questions & Answers**

**1. What is React?**

React is an **open-source front-end JavaScript library** that is used for building user interfaces, especially for single-page applications. It is used for handling view layer for web and mobile apps.

**2. What are the major features of React?**

The major features of React are:

- It uses VirtualDOM instead of RealDOM considering that RealDOM manipulations are expensive.

- Supports server-side rendering.

- Follows Unidirectional data flow or data binding.

- Uses reusable/composable UI components to develop the view.

**3. What is JSX?**

JSX stands for **JavaScript XML**. JSX allows us to write HTML in React. JSX makes it easier to write and add HTML in React.

**4. What is the difference between Element and Component?**

Element:

- A React element is an object representation of a DOM node.

- Elements are immutable i.e once created cannot be changed.

Component:

- A component encapsulates a DOM tree.

- The state in a component is mutable.

- Components are like functions that return HTML elements.

**5. How to create components in React?**

- A functional component is just a plain JavaScript pure function that accepts props as an argument and returns a React element(JSX).

**Syntax:**

const Car=()=> {

return <h2>Hi, I am also a Car!</h2>;

}

- A class component requires you to extend from React. Component and create a render function which returns a React element.

**Syntax:**

class Car extends React.Component {

render() {

return <h2>Hi, I am a Car!</h2>;

}

}

**6. When to use a Class Component over a Function Component?**

If the component needs state or lifecycle methods then use class component otherwise use function component. However, from React 16.8 with the addition of Hooks, you could use state, lifecycle methods and other features that were only available in class component right in your function component. \*So, it is always recommended to use Function components, unless you need a React functionality whose Function component equivalent is not present yet, like Error Boundaries \*

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

State of a component is an object that holds some information that may change over the lifetime of the component. When the state object changes, the component re-renders.

Ex.

class Car extends React.Component {

constructor(props) {

super(props);

this.state = {brand: "Ford"};

}

render() {

return (

<div>

<h1>My Car</h1>

</div>

);

}

}

To change a value in the state object, use the this.setState() method.

Ex.

class Car extends React.Component {

constructor(props) {

super(props);

this.state = {

brand: "Ford",

model: "Mustang",

color: "red",

year: 1964

};

}

changeColor = () => {

this.setState({color: "blue"});

}

render() {

return (

<div>

<h1>My {this.state.brand}</h1>

<p>

It is a {this.state.color}

{this.state.model}

from {this.state.year}.

</p>

<button

type="button"

onClick={this.changeColor}

>Change color</button>

</div>

);

}

}

**8. What is DOM and why it is used?**

The Document Object Model (DOM) is an application programming interface (API) for HTML and XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated.

**9. What are props in React?**

Props are inputs to components. They are single values or objects containing a set of values that are passed to components on creation using a naming convention similar to HTML-tag attributes. They are data passed down from a parent component to a child component.

The primary purpose of props in React is to provide following component functionality:

- Pass custom data to your component.

- Trigger state changes.

**10. React Components**

Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML.

A functional component is just a plain JavaScript pure function that accepts props as an argument and returns a React element(JSX).

A class component requires you to extend from React. Component and create a render function which returns a React element.

**11. React useState Hook**

The React useState Hook allows us to track state in a function component.

State generally refers to data or properties that need to be tracking in an application.

import { useState } from "react";

function FavoriteColor() {

const [color, setColor] = useState("");

}

**12. React useEffect Hook**

The useState hook is used for storing variables that are part of your application's state and will change as the user interacts with your website. The useEffect hook allows components to react to lifecycle events such as mounting to the DOM, re-rendering, and unmounting.

**13. Redux in react**

Redux is a pattern and library for managing and updating application state, using events called "actions". It serves as a centralized store for state that needs to be used across your entire application, with rules ensuring that the state can only be updated in a predictable fashion.

**14. Single-page Apps**

An SPA (Single-page application) is a web app implementation that loads only a single web document, and then updates the body content of that single document via JavaScript APIs.  
  
Single-page applications offer a much better user experience (UX), meaning that users can navigate easily between the different pages of an app without waiting for the pages to load.