Training Day25 Report

(15th July 2024)

What are props in React?

In React, "props" (short for "properties") are a mechanism for passing data from a parent component to a child component. Props allow you to customize and configure child components by providing them with values or functions.

Key points about props in React:

1. **Passing Data:** Props are used to pass data from a parent component to a child component. They are passed as attributes to the child component when it is declared in the parent component's JSX.
   Example:

```
// Parent component

function ParentComponent() {

    const message = "Hello from parent!";

    return <ChildComponent greeting={message} />;

}



// Child component

function ChildComponent(props) {
```

```
    return <p>{props.greeting}</p>;

}
```

2. Immutable: Props are immutable, meaning that once they are set by the parent, they cannot be changed by the child component. The child component can use props for rendering, but it should not modify them.

3. Functional and Class Components: Props can be accessed in both functional and class components. In functional components, they are passed as an argument, while in class components, they are accessed using this.props.

Example:

```
// Functional component

function MyFunctionalComponent(props) {

    return <p>{props.message}</p>;

}



// Class component

class MyClassComponent extends React.Component {

    render() {

        return <p>{this.props.message}</p>;

    }

}
```

4. Default Values: You can provide default values for props using the defaultProps property for functional components or the defaultProps static property for class components.
5. Customizing Child Components: Props allow you to customize and configure child components based on the requirements of the parent component or dynamic data.
Props play a fundamental role in React's component-based architecture, enabling the creation of flexible and reusable components that can adapt to different data and scenarios. They facilitate the flow of information between components, supporting a hierarchical and modular structure in React applications.