**Using cloneElement() to Copy**

You can use **React.cloneElement()** to clone and modify any React element's props.

The basic syntax of **cloneElement** is as follows:

React.cloneElement(element, newProps, ...newChildren)

Here:

* **element** is the React element you want to clone.
* **newProps** is an object containing the new props you want to apply to the cloned element.
* **newChildren** are any new children you want to assign to the cloned element.

You can use **cloneElement** to modify any props of the **element**, not just the **children** prop. For example, you can use it to add, modify, or remove any other props of the element.

Here's an example that demonstrates using **cloneElement** to modify the **className** prop of a React element:

import React from 'react';

function App() {

const originalElement = <div className="original">Original Element</div>;

const modifiedElement = React.cloneElement(originalElement, {

className: "modified"

});

return (

<div>

{originalElement}

{modifiedElement}

</div>

);

}

export default App;

In this example, **cloneElement** is used to create a new React element with the **className** prop modified from "original" to "modified."

So, while **cloneElement** can be used to work with the **children** prop, it's not limited to just that; it can be used to modify any props of a React element.