

React - ES6 Ternary Operator

Topics : [React JS](#)

Written on [January 02, 2024](#)

The ternary operator (`? :`) is a concise way to write conditional statements in JavaScript, including React applications. It allows you to write an inline if-else statement, making your code more compact. Here's a basic syntax of the ternary operator:

```
condition ? expressionIfTrue : expressionIfFalse;
```

In React, the ternary operator is commonly used in JSX to conditionally render different elements based on a condition. Here are some examples:

Conditional Rendering in JSX:

```
// Example 1: Simple conditional rendering
const Greeting = ({ isLoggedIn }) => (
  <div>
    {isLoggedIn ? <p>Welcome back!</p> : <p>Please log in.</p>}
  </div>
);
```

```
// Example 2: Rendering different components based on a condition
const UserType = ({ isAdmin }) => (
  <div>
    {isAdmin ? <AdminComponent /> : <RegularComponent />}
  </div>
);
```

Assigning Conditional Classes or Styles:

```
const Button = ({ isPrimary }) => (
  <button className={isPrimary ? 'primary-button' : 'secondary-button'}>
    Click me
  </button>
);
```

Using Ternary Operator in JavaScript Logic:

```
const MyComponent = ({ data }) => (
  <div>
    {data.length > 0 ? (
      <ul>
```

```
{data.map((item, index) => (  
<li key={index}> {item} </li>  
))}  
</ul>  
) : (  
<p>No data available.</p>  
)}  
</div>  
);
```

Ternary Operator with React State:

```
class ToggleComponent extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = {  
      isToggled: false,  
    };  
  }  
  
  handleClick = () => {  
    this.setState((prevState) => ({ isToggled: !prevState.isToggled }));  
  };  
  
  render() {  
    return (  
      <div>  
        <button onClick={this.handleClick}>  
          {this.state.isToggled ? 'Toggle Off' : 'Toggle On'}  
        </button>  
      </div>  
    );  
  }  
}
```