Ternary Operator in React

This lesson teaches the use of Ternary Operator in React for conditional rendering of JSX and explains it with the help of examples.

A ternary operator — also called Conditional Operator — is the only JavaScript operator which takes three operands and returns a value based on some condition. It's an alternative for if statement. This could be used for multiple purposes and comes in very handy in React too!

Displaying JavaScript strings, objects, and arrays in React is not enough. What about an if-else statement for enabling conditional rendering? You cannot use an if-else statement directly in JSX, but you can return early from the rendering function. Returning null is valid in React when displaying nothing. Just like we did in the example given below.



Did you know?

Conditional rendering in React uses JavaScript operators like if or the conditional operator to create elements representing the current state, and let React show or hide a certain UI element based on a condition.

```
import React from 'react';
require('./style.css');
import ReactDOM from 'react-dom';
import App from './app.js';
ReactDOM.render(
  <App />,
  document.getElementById('root')
);
```

However, if you want to use an if-else statement within the returned JSX, you can do it by using a JavaScripts ternary operator:

```
import React from 'react';
require('./style.css');

import ReactDOM from 'react-dom';
import App from './app.js';

ReactDOM.render(
   <App />,
    document.getElementById('root')
);
```

Another way of doing it, if you only return one side of the conditional rendering anyway, is using the && operator:

I will not go into detail why this works, but if you are curious, you can learn about it and other techniques for conditional rendering over here: All the conditional renderings in React. After all, the conditional rendering in React only shows, again, that most of the React is actually JavaScript and not anything React specific.