Bookstore:**React JS**

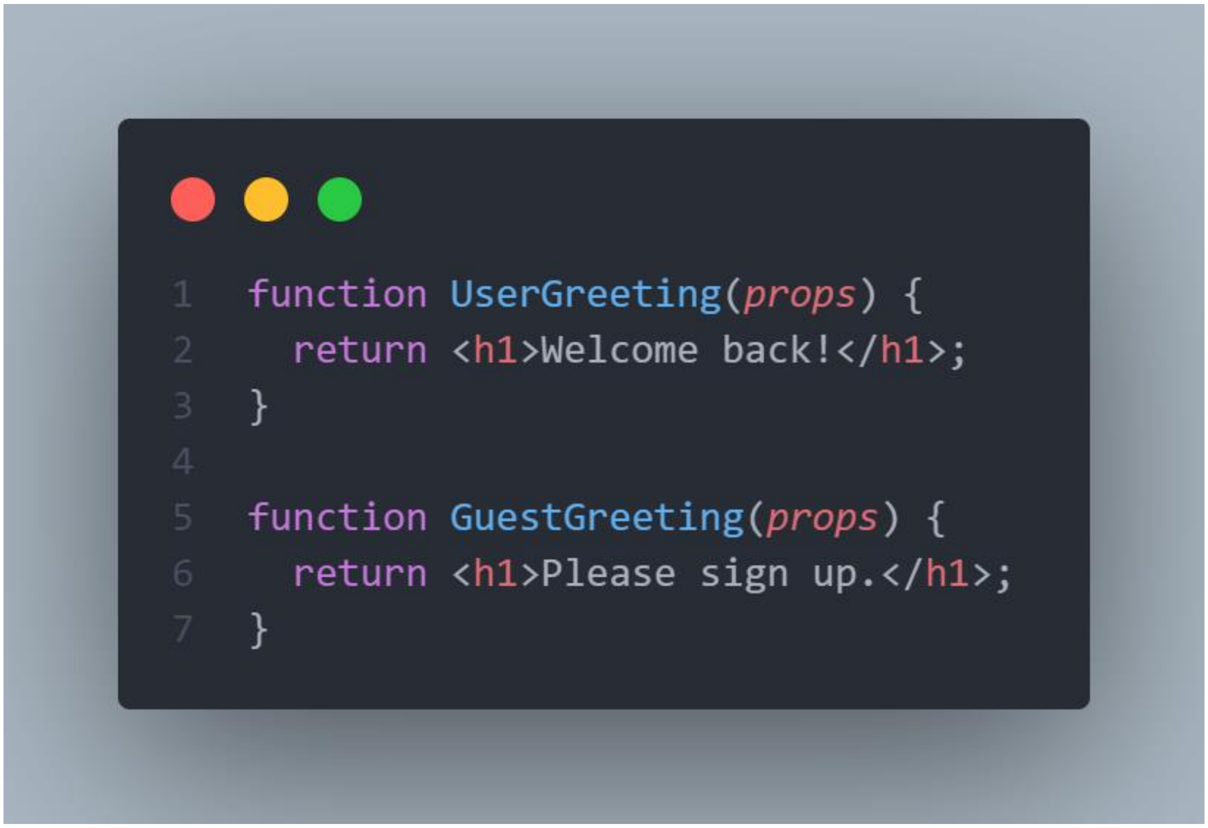
Programming language:**JavaScript**

Topic:**Conditional Rendering**

What is conditional rendering in React?

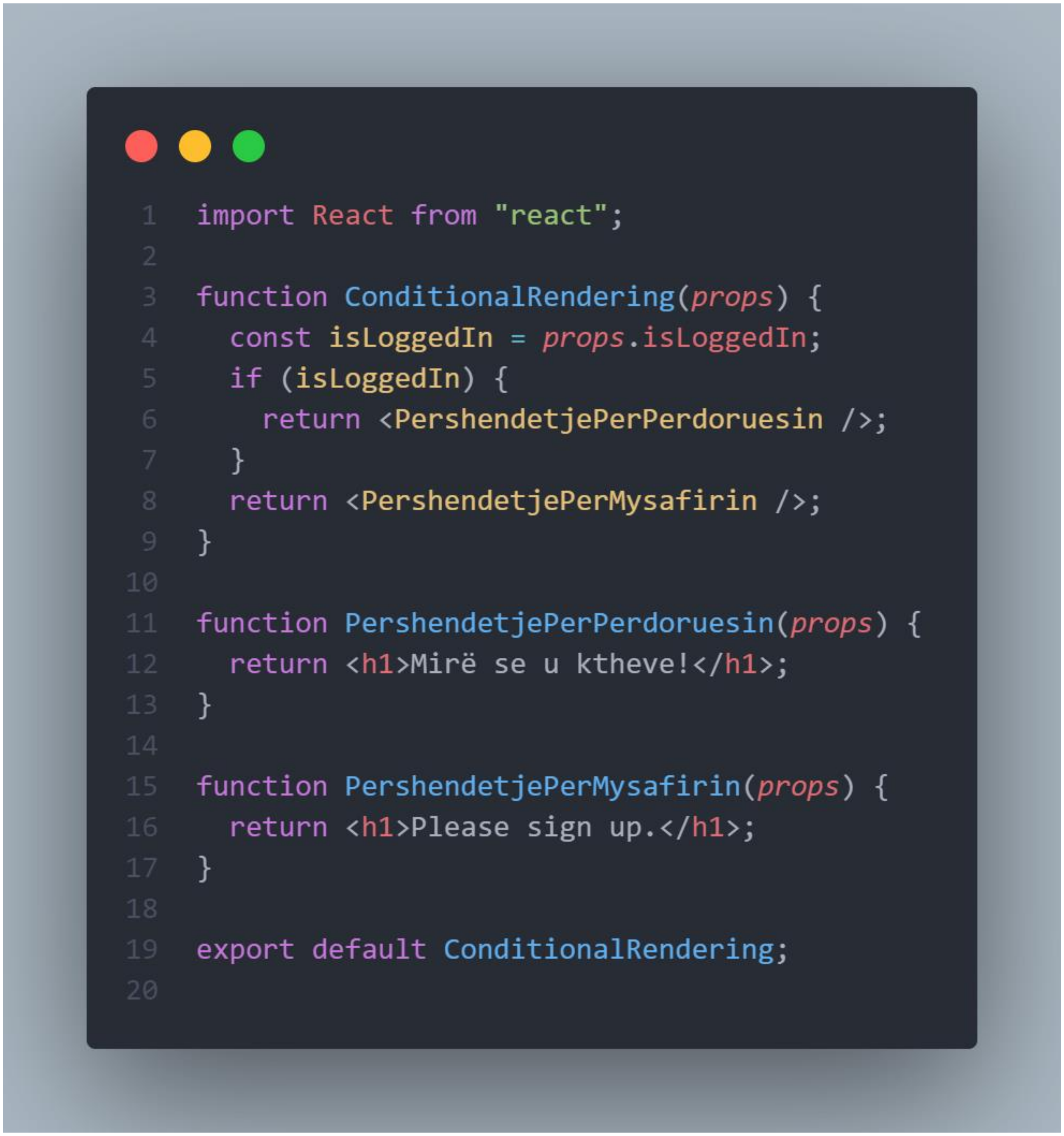
Conditional rendering in React works the same way as work conditions in JavaScript. Use JavaScript operators like if or conditional operator to create elements that represent the condition current and let React update the UI to iu adapted to them.

Consider these two components:



```
function UserGreeting(props) {
  return <h1>Welcome back!</h1>;
}

function GuestGreeting(props) {
  return <h1>Please sign up.</h1>;
}
```

We'll create a greeting component that displays one of these components depending on whether a user is logged in:

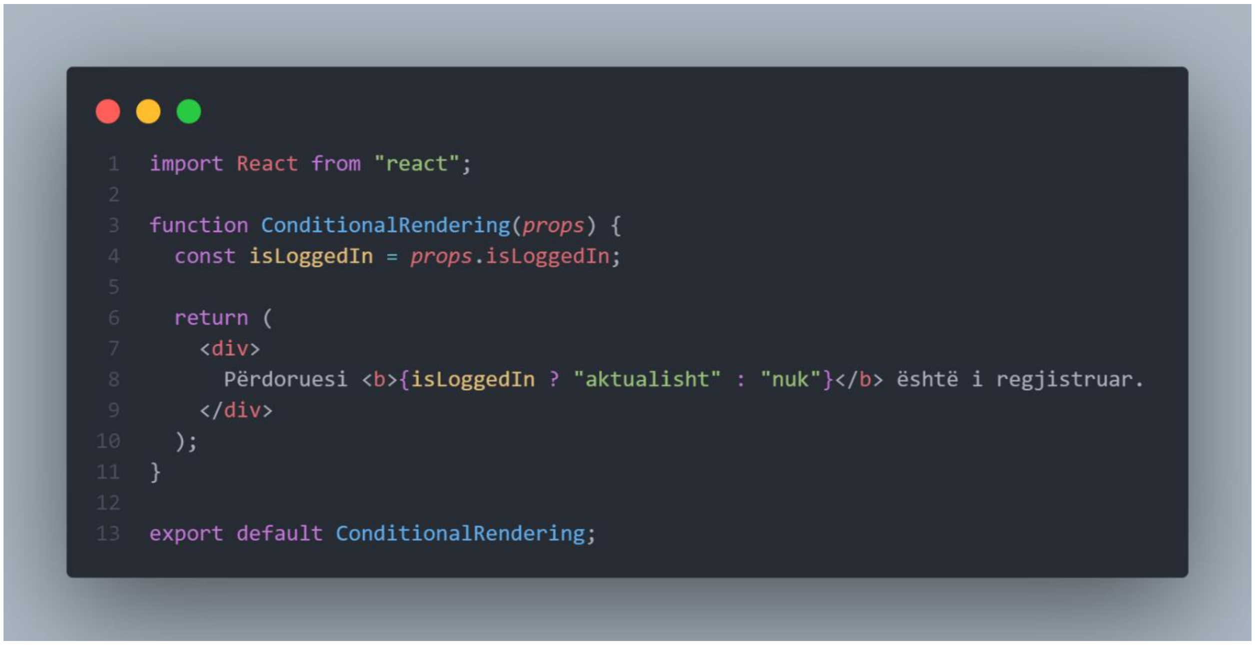


```
import React from "react";

function ConditionalRendering(props) {
  const isLoggedIn = props.isLoggedIn;
  if (isLoggedIn) {
    return <PershendetjePerPerdoruesin />;
  }
  return <PershendetjePerMysafirin />;
}

function PershendetjePerPerdoruesin(props) {
  return <h1>Mirë se u ktheve!</h1>;
}

function PershendetjePerMysafirin(props) {
  return <h1>Please sign up.</h1>;
}

export default ConditionalRendering;

```

Inline If-Else with the conditional operator

Another method for conditionally rendering elements is to use the JavaScript conditional operator condition ? true: false.

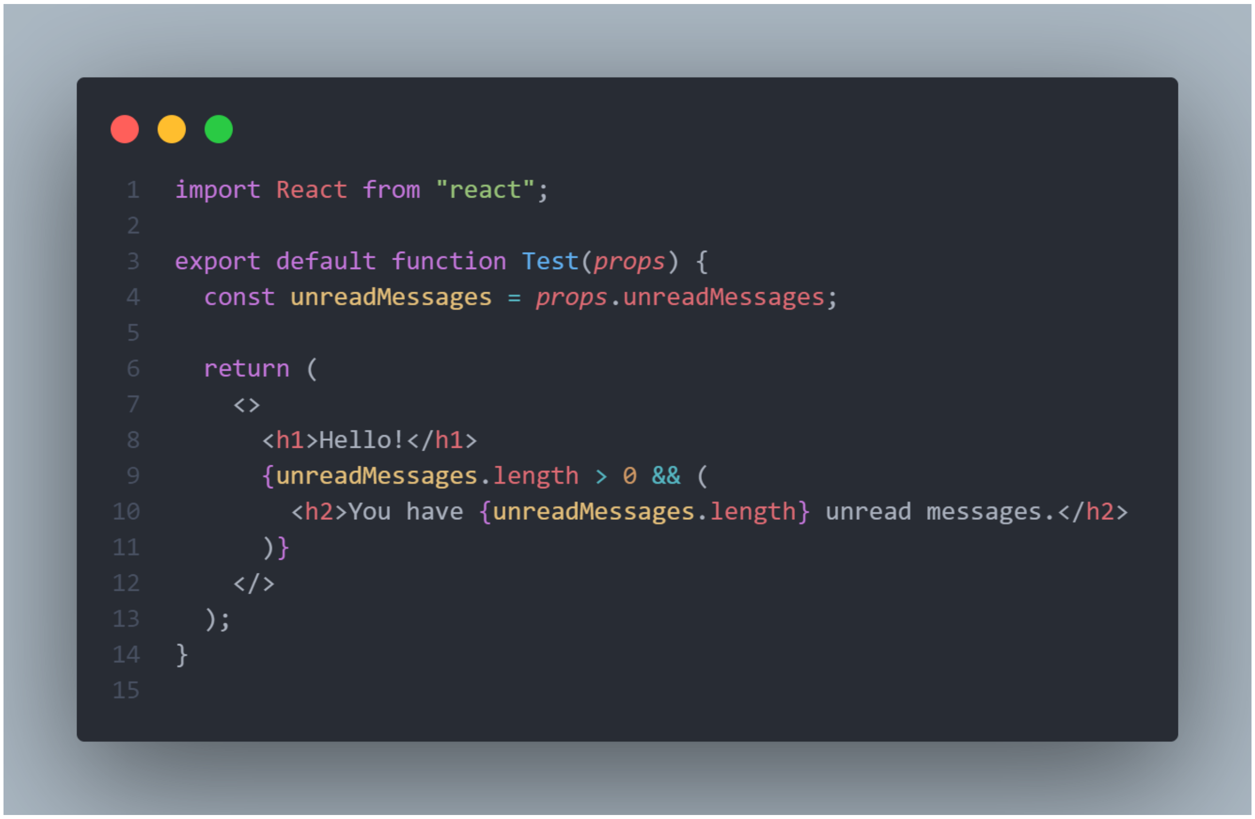
In the following example, we use it to conditionally return one small block of text.



```
import React from "react";

function ConditionalRendering(props) {
  const isLoggedIn = props.isLoggedIn;

  return (
    <div>
      Përdoruesi <b>{isLoggedIn ? "aktualisht" : "nuk"}</b> është i regjistruar.
    </div>
  );
}

export default ConditionalRendering;
```

As with JavaScript, it's up to you to choose an appropriate style based on what you and your team consider most readable. Do not forget also that whenever conditions become very complex, it can be a good time to release a component.

Inline If with logical operator &&



```
import React from "react";

export default function Test(props) {
  const unreadMessages = props.unreadMessages;

  return (
    <>
      <h1>Hello!</h1>
      {unreadMessages.length > 0 && (
        <h2>You have {unreadMessages.length} unread messages.</h2>
      )}
    </>
  );
}

```

This method works because in JavaScript, the true expression && is evaluated always in expression, and expression***false***&&is always evaluated at***false***.

Therefore, if the condition is true, the element immediately after the && will be displayed on the way out. If its***false***, React will ignore and skip it.