Conditional rendering in React refers to the process of rendering different components or content based on certain conditions. This can be achieved using various techniques within React.

Here's a basic example of conditional rendering in React using the `if` statement:

```javascript
import React from 'react';

function MyComponent(props) {
  if (props.condition) {
    return <div>This content is rendered conditionally</div>;
  } else {
    return <div>Other content when condition is not met</div>;
  }
}

export default MyComponent;
```

In this example, `MyComponent` will render different content based on the value of `props.condition`.

Another common technique for conditional rendering in React is to use the ternary operator:

```javascript
import React from 'react';

function MyComponent(props) {
  return (
    <div>
      {props.condition ? (
        <div>This content is rendered conditionally</div>
      ) : (
        <div>Other content when condition is not met</div>
      )}
    </div>
  );
}

export default MyComponent;
```

In this example, the ternary operator `props.condition ? trueCase : falseCase` determines which content to render based on the value of `props.condition`.

Additionally, you can use logical && operator for simple conditional rendering:

```javascript
import React from 'react';

function MyComponent(props) {
  return (
    <div>
      {props.condition && <div>This content is rendered conditionally</div>}
    </div>
  );
}

export default MyComponent;
```

In this example, if `props.condition` is true, then the content `<div>This content is rendered conditionally</div>` will be rendered. Otherwise, nothing will be rendered.

These are some of the basic techniques for conditional rendering in React. Depending on your specific use case, you might choose one over the other.

Example 1: Rendering a Login Button

Suppose you have a simple login page component, and you want to render a "Login" button if the user is not logged in, and a "Logout" button if the user is logged in.

```javascript
import React from 'react';

function LoginButton({ isLoggedIn, onLogin, onLogout }) {
  return (
    <div>
      {isLoggedIn ? (
        <button onClick={onLogout}>Logout</button>
      ) : (
        <button onClick={onLogin}>Login</button>
      )}
    </div>
  );
}

export default LoginButton;
```

In this example, the `LoginButton` component renders either a "Login" button or a "Logout" button based on the value of the `isLoggedIn` prop. When the user clicks the button, it triggers the corresponding `onLogin` or `onLogout` function.

Example 2: Rendering User Profile Information

Let's say you have a user profile component that displays the user's name and email. However, if the user hasn't provided an email, you want to display a message instead.

```javascript
import React from 'react';

function UserProfile({ name, email }) {
  return (
    <div>
      <h2>User Profile</h2>
      <p>Name: {name}</p>
      {email ? (
        <p>Email: {email}</p>
      ) : (
        <p>No email provided</p>
      )}
    </div>
  );
}

export default UserProfile;
```

In this example, the `UserProfile` component renders the user's name and email if the email is provided. If not, it displays the message "No email provided" instead.

These are just two simple examples of how conditional rendering can be used in React to dynamically render different content based on certain conditions.