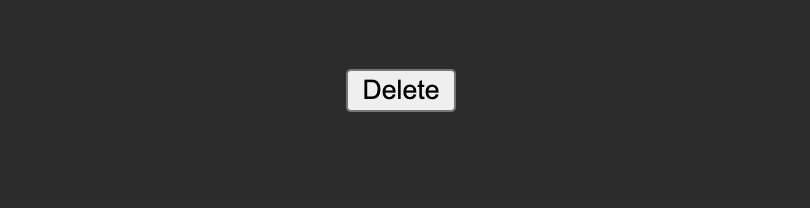
**React Task**

**Conditional Content**

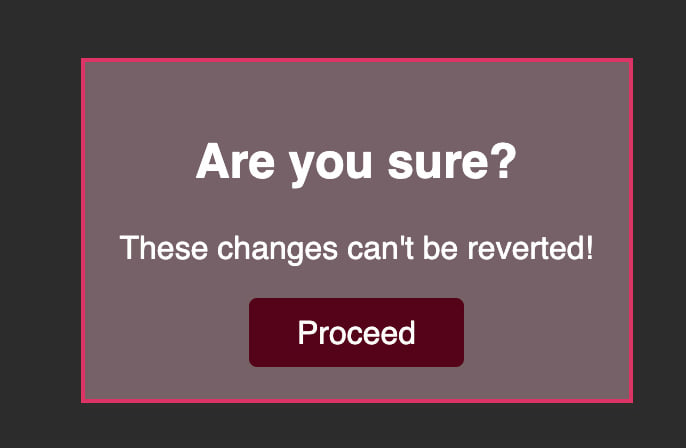
You're working on a part of a web app that's responsible for **showing a warning** when a user is about to perform a dangerous action.

Therefore, your task is to **conditionally** show a warning box once a user has **clicked** a specific button. Inside that warning dialog, another button allows users to **dismiss the warning** (i.e., remove the warning box from the screen).

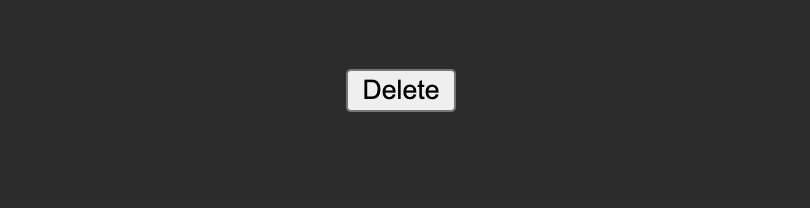
The finished app should display this UI, if the <button> has not been clicked yet:



And this UI, once the button was clicked:



Once the "Proceed" button was clicked, the warning box should be removed again:



For this task, you must react to clicks on both <button> elements that are part of the starting code. The second button, outside of the <div> with the id="alert", should show the <div id="alert"> (and all its content). The button inside that <div> should then hide it again (i.e., remove it from the DOM).

It's up to you whether you want to use a ternary expression or store the conditionally shown JSX code in a variable.

Important: In this Udemy code editor you may get an error if you use useState() - use React.useState() instead!

**Code :-**

import React from 'react';

export default function App() {

const [isDeleting, setIsDeleting] = React.useState(false);

function deleteHandler() {

setIsDeleting(true);

}

function proceedHandler() {

setIsDeleting(false);

}

let warning;

if (isDeleting) {

warning = (

<div data-testid="alert" id="alert">

<h2>Are you sure?</h2>

<p>These changes can't be reverted!</p>

<button onClick={proceedHandler}>Proceed</button>

</div>

);

}

return (

<div>

{warning}

<button onClick={deleteHandler}>Delete</button>

</div>

);

}

**Dynamic Styling**

Your task is to **dynamically** apply a CSS class (active) to the <p>Style me</p> element in the provided React app.

The class should be applied when the <button> is clicked for the first time.

Here's how the finished app should look like BEFORE the button was clicked:



Here's how it should look like AFTER the button was clicked:



Important: Use React.useState() instead of just useState() as the latter can cause problems in this Udemy code environment.

### Learning objective

**Code :-**

import React from 'react';

export default function App() {

const [highlighted, setHighlighted] = React.useState(false);

function handleClick() {

setHighlighted(isHighlighted => !isHighlighted);

}

return (

<div>

<p className={highlighted ? 'active' : undefined}>Style me!</p>

<button onClick={handleClick}>Toggle style</button>

</div>

);

}