

Exercise 7 - Modal Dialog with Event Delegation in JavaScript

🔗 Introduction

In this exercise, you'll be working on implementing a modal dialog that appears when the user clicks a button and disappears when the user clicks either the overlay or the close button. The HTML and CSS code is provided for you, along with a demonstration video of how the feature should work.

[Video demo of the result.](#)

You'll be required to write the JavaScript code that controls the behavior of the modal dialog. This exercise will involve the concept of **event delegation**, which means you'll be attaching event listeners to parent elements and using the event object to determine the specific child element that was clicked.

Requirements

1. Show the Modal Dialog:

- When the "Show Modal" button is clicked, the modal dialog (along with an overlay) must be displayed.

2. Close the Modal Dialog:

- When the overlay itself or the close button (X) inside the modal is clicked, the modal dialog must be hidden.
- This behavior must be achieved using event delegation.

Instructions

1. Understand the Structure:

- Take a moment to understand the HTML structure and the CSS styles that have been provided.
- Identify the key elements you'll be working with: the overlay, the "Show Modal" button, and the close button inside the modal.

2. Implementing the 'Show' Functionality:

- Start by writing an event listener for the "Show Modal" button.
- When the button is clicked, you'll need to change the display property of the overlay to make it visible.

3. Implementing the 'Close' Functionality with Event Delegation:

- Write an event listener for the overlay, which includes both the close button and the modal itself.
- Determine the target of the click event within the listener. You'll have to check if the target is the overlay itself or the close button. Think of how you check which element was clicked.
- If one of these elements were clicked, hide the overlay, otherwise, do nothing.

Tips and Resources

- Think about how you can use the `event.target` property within your event listener to achieve event delegation.
- You may find it helpful to read this article on [Event Delegation in JavaScript](#) to understand the concept further.

Conclusion

This exercise provides a practical application of event delegation, a powerful concept in JavaScript. Follow the instructions and use the provided resources to complete the challenge. If you have any questions or need further clarification, don't hesitate to ask your instructor.