

Build Modal Dialog

Building a Modal Dialog

A **modal dialog** is a popup window that appears on top of the main content, typically used for alerts, confirmations, forms, or additional information. In this assignment, we will create a modal dialog that can be opened and closed using JavaScript.

Features

- A **button** to open the modal.
- A **close button** inside the modal.
- Clicking outside the modal (on the background overlay) will also close it.
- Smooth **CSS transitions** for better user experience.

Steps to Build the Modal

We'll break it down into three parts:

1. **HTML** – The structure of the modal.
2. **CSS** – Styling for the modal and animations.
3. **JavaScript** – Handling user interactions.

HTML (Structure of the Modal)

```
html

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Modal Dialog Example</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

  <!-- Button to Open Modal -->
  <button id="openModalBtn">Open Modal</button>

  <!-- Modal Structure -->
  <div id="modalOverlay" class="modal-overlay">
    <div class="modal">
      <span class="close-btn">&times;</span>
      <h2>Modal Title</h2>
      <p>This is a simple modal dialog box.</p>
    </div>
  </div>

  <script src="script.js"></script>
```

```
</body>
</html>
```

Explanation

- `#openModalBtn`` → The button that will trigger the modal.
- `#modalOverlay`` → The full-screen background that contains the modal.
- `.modal`` → The actual modal box.
- `.close-btn`` → The "x" (close) button to hide the modal.

2 CSS (Styling and Animations)

```
css

/* Reset styles */
body {
  font-family: Arial, sans-serif;
  text-align: center;
  margin: 0;
  padding: 0;
}

/* Button styling */
#openModalBtn {
  padding: 10px 20px;
  font-size: 18px;
  cursor: pointer;
  background-color: #007bff;
  color: white;
  border: none;
  border-radius: 5px;
  margin-top: 20px;
}

/* Modal overlay (Background) */
.modal-overlay {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background: rgba(0, 0, 0, 0.5);
  display: flex;
  align-items: center;
  justify-content: center;
  opacity: 0;
  visibility: hidden;
  transition: opacity 0.3s ease, visibility 0.3s ease;
}

/* Modal box */
.modal {
  background: white;
  padding: 20px;
  width: 300px;
  border-radius: 8px;
  text-align: center;
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.3);
  position: relative;
  transform: translateY(-50px);
  transition: transform 0.3s ease;
}

/* Close button */
```

```
.close-btn {
  position: absolute;
  top: 10px;
  right: 15px;
  font-size: 24px;
  cursor: pointer;
}

/* Show modal when active */
.modal-overlay.show {
  opacity: 1;
  visibility: visible;
}

.modal-overlay.show .modal {
  transform: translateY(0);
}
```

Explanation

- The **modal is hidden by default** (`opacity: 0; visibility: hidden;`).
- When the `show` class is added via JavaScript:
 - The overlay appears (`opacity: 1; visibility: visible;`).
 - The modal **animates in smoothly** from above (`transform: translateY(0);`).
- The close button (`x`) is positioned **at the top right** inside the modal.

JavaScript (Adding Functionality)

```
js

// Select elements
const openModalBtn = document.getElementById('openModalBtn');
const modalOverlay = document.getElementById('modalOverlay');
const closeModalBtn = document.querySelector('.close-btn');

// Function to open the modal
openModalBtn.addEventListener('click', () => {
  modalOverlay.classList.add('show');
});

// Function to close the modal
closeModalBtn.addEventListener('click', () => {
  modalOverlay.classList.remove('show');
});

// Close modal when clicking outside the modal box
modalOverlay.addEventListener('click', (event) => {
  if (event.target === modalOverlay) {
    modalOverlay.classList.remove('show');
  }
});
```

Explanation

1. Selecting Elements

- We grab the **button**, **modal overlay**, and **close button** using `document.getElementById` and `document.querySelector`.

2. Opening the Modal

- When the button is clicked, the `show` class is added to `modalOverlay`, making it visible.

3. Closing the Modal

- Clicking on the close button (`×`) removes the `show` class, hiding the modal.

4. Clicking Outside Closes the Modal

- If the user clicks anywhere outside the `.modal` box (on the overlay), it closes the modal.

Summary

Component	Functionality
HTML	Defines the structure of the modal and the button
CSS	Styles the modal, overlay, and animations
JavaScript	Handles user interaction for opening and closing the modal

Key Learnings

- ✓ **DOM Manipulation** – Using `getElementById` and `querySelector` to interact with elements.
- ✓ **Event Handling** – `addEventListener()` to detect button clicks and perform actions.
- ✓ **CSS Transitions** – Smooth animations using `transform` and `opacity`.
- ✓ **Conditional Logic** – Closing the modal only when clicking outside of it.

Full Code (For Easy Copy-Paste)

HTML (index.html)

```
html

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Modal Dialog Example</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

  <button id="openModalBtn">Open Modal</button>

  <div id="modalOverlay" class="modal-overlay">
    <div class="modal">
      <span class="close-btn">&times;</span>
      <h2>Modal Title</h2>
      <p>This is a simple modal dialog box.</p>
    </div>
  </div>

  <script src="script.js"></script>
</body>
</html>
```

CSS (styles.css)

CSS

```
body { font-family: Arial, sans-serif; text-align: center; margin: 0; padding: 0; }

#openModalBtn { padding: 10px 20px; font-size: 18px; cursor: pointer; background-color: #007bff; color: white; }

.modal-overlay { position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0, 0, 0, 0.5); }

.modal { background: white; padding: 20px; width: 300px; border-radius: 8px; text-align: center; box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1); }

.close-btn { position: absolute; top: 10px; right: 15px; font-size: 24px; cursor: pointer; }

.modal-overlay.show { opacity: 1; visibility: visible; }

.modal-overlay.show .modal { transform: translateY(0); }
```

JavaScript (script.js)

js

```
const openModalBtn = document.getElementById('openModalBtn');
const modalOverlay = document.getElementById('modalOverlay');
const closeModalBtn = document.querySelector('.close-btn');

openModalBtn.addEventListener('click', () => modalOverlay.classList.add('show'));
closeModalBtn.addEventListener('click', () => modalOverlay.classList.remove('show'));
modalOverlay.addEventListener('click', (event) => { if (event.target === modalOverlay) modalOverlay.classList.remove('show'); });
```

Let me know if you need improvements or explanations! 