

Modal Windows/Pop-up

A "modal" window

- Visually shows "over" the main content
- While shown, can't interact with main content
 - That's the "mode" that "modal" refers to
- Main content often "greyed out"

Modal Considerations

- Lack of interaction isn't just mouse!
 - No keyboard focus/interactions
 - Touch can't bypass either
- How to manage scrolling?
- Good UX to allow Escape key to exit mode
 - Click off of window to exit?
- Users often look for "X" option
 - Should have a close/cancel button ALSO

Back in my day....

- Implemented modals as full page divs
 - Some opacity to "grey out"
 - Intercept, `.stopPropagation()` on click events
 - React to Escape keydown event
- These days there is a better option!

No, seriously!

- MANY Modals are implemented poorly!
 - Such as failing to trap keyboard navigation
- Follow this example!
 - Each semester students ignore and google random ways of doing this
 - I'm sure ChatGPT will happily provide a bad modal

The <dialog> element

- Technically not always modal
 - But has modal option
- Built in display conventions
 - Not perfect :(
 - But better than most alternatives
 - But workarounds exist
- Requires JS to interact with

Minimal Parts

```
<button class="open" type="button">Open</button>
<dialog class="modal">
  <p>Whatever content goes here</p>
  <button class="close" type="button">Close</button>
</dialog>
```

- Pick your own class names

```
const modalEl = document.querySelector('.modal');
const openEl = document.querySelector('.open');
const closeEl = document.querySelector('.close');

openEl.addEventListener('click', () => {
  modalEl.showModal();
});

closeEl.addEventListener('click', () => {
  modalEl.close();
});
```

Styling the Modal

- `::backdrop` on the modal element
 - `background-color` is the "grey out" part
 - Likely want some partial opacity

```
.modal::backdrop {  
  background-color: rgb(0 0 0 / 0.8 );  
}
```

<dialog> and Rendering

A `<dialog>` element has inherent state

- Is a modal open?
- Form fields also have inherent state

So don't replace these elements during a render

- At least not while in use