

## `pointer-events: none;` in CSS

The `pointer-events: none;` CSS property makes **an element ignore all mouse and pointer interactions** (such as clicks, hovers, and drag events). The element becomes "invisible" to mouse and touch, so events pass through to whatever is beneath it.

### Key Points

- The element *still appears visually*, but you cannot interact with it using the mouse, touch, or pointer devices.
- Useful for disabling buttons, overlaying elements without affecting interaction, or creating custom UI effects.

### Example Usage

```
button.disabled {  
  pointer-events: none;  
  opacity: 0.5;  
}
```

Now, clicking the button does nothing; it looks disabled, and all pointer events are ignored.

### Practical Example

```
<div class="overlay" style="pointer-events: none;">  
  <!-- This overlay won't block clicks underneath -->  
</div>  
<button>Clickable Button</button>
```

Here, the overlay is visible, but you can still click the button below it.

### Common Use Cases

- Disabling interaction for UI elements (e.g., while loading)
- Allowing clicks to "pass through" transparent overlays
- Temporarily disabling inputs or entire containers without changing their structure or removing them from the DOM

## To Re-enable Pointer Events

Set `pointer-events: auto;` to allow interactions again.

### Summary:

`pointer-events: none;` visually displays an element but blocks ALL mouse/pointer interactions, letting you control which elements react to user actions.