# 4.4.10 Reflection

Great job—you made it through a complex lesson that introduced several new concepts! It's common in web development that a reference to an element, item, or data must be passed to another part of the code, such as when storing a form into a database, placing an item into a shopping cart, or clicking a thumbnail to expand the view. Early exposure to this—like you had in this lesson—will help you problem-solve later in the course and in your future career as a full-stack web developer!

In this lesson, you learned how to do the following:

- Break down a large problem into manageable steps
- Make elements draggable using a special attribute
- Listen for multiple events, including `dragstart`, `dragover`, `dragleave`, and `drop`
- Override browser behavior using the `preventDefault()` method in the event handler
- Transfer a data attribute using the `dataTransfer` object on the `DragEvent`

- Traverse the DOM using the `closest()` method to find the first ancestor of an element that matches a selector, including the element itself, if it matches.

- Look under the hood of a seemingly simple UI interaction to reveal a complex web of events

There are several JavaScript libraries that offer a drag-and-drop feature with many built-in error handling solutions. Nonetheless, it's important to understand how this type of API works.

In the next lesson, we'll finish our app by making the tasks persistent. To do this, we'll revisit our old friend `localStorage`.