1 `reduce()` Notes

Sometimes you need to analyze all of the elements in an array and return just a single value, like a number, string, or object.

With the `reduce()` method, you can use the elements of an array to compute and return any value you want.

For example, you might have an array of prices and you want to get a single value, the total of all the prices.

The parameters for `reduce()` are different than the methods we've worked with so far.

```
[].reduce( (acc, cur) => {}, initialValue);
```

The first parameter is called an *accumulator* represented here by `acc`. The second parameter, `cur`, represents the *current* array item.

The accumulator is very important. It contains the running total of the value that the `reduce()` method returns. The first time `reduce()` runs on an array, it needs an initial value. This is set as the *second* argument after the *callback function*, above this value is denoted as `initialValue`. **IF** an initial value is NOT given, `reduce()` will use the first value in the array as the *initial value*. While ommiting the initial value won't typically be a problem, it is better practice to be explicit for readability and maintainability.