11.5.1

Introduction to Dynamic Tables

Dana is making some headway. She's more familiar with objects and arrays, and she created a few functions, both traditional JavaScript functions and the faster arrow functions. Comparing and writing `for` loops was a bit challenging; logically, they work in the same manner as a Python `for` loop, but the code to create one in JavaScript is far more involved! Dana is realizing that becoming proficient in JavaScript requires a lot of practice and patience.

Practice is progress, though, and Dana is now ready to create her table.

Dana's code is somewhat modest right now, but it's about to get a lot more interesting. Now we're going to help her build the table to display all of the UFO sightings. We'll need to iterate through the array of objects in our data file and then append them to a table row. All of this will happen within a function, which makes the code self-contained.

Creating self-contained code makes it easier to reuse the code and keeps us organized: the code in this function will be used to fill the table with data **only**.

Let's get started by returning to our `app.js` file in the editor and, on a new line, creating a new function.

Typically, functions are named after what they do. We're building a table, so we'll name the function "buildTable." We'll also pass in "data" as the argument. Remember that we used the variable "data" earlier to import our array of UFO sightings? This is the first step in actually working with the data.

In our editor, we should have the start of a new function:

```
function buildTable(data) {

}
```

We're using a standard JavaScript function instead of an arrow function because of what we'll be inserting inside the function (hint: another function!). Let's start building out the rest of the function.

In the next line, we'll want to use code to clear existing data.
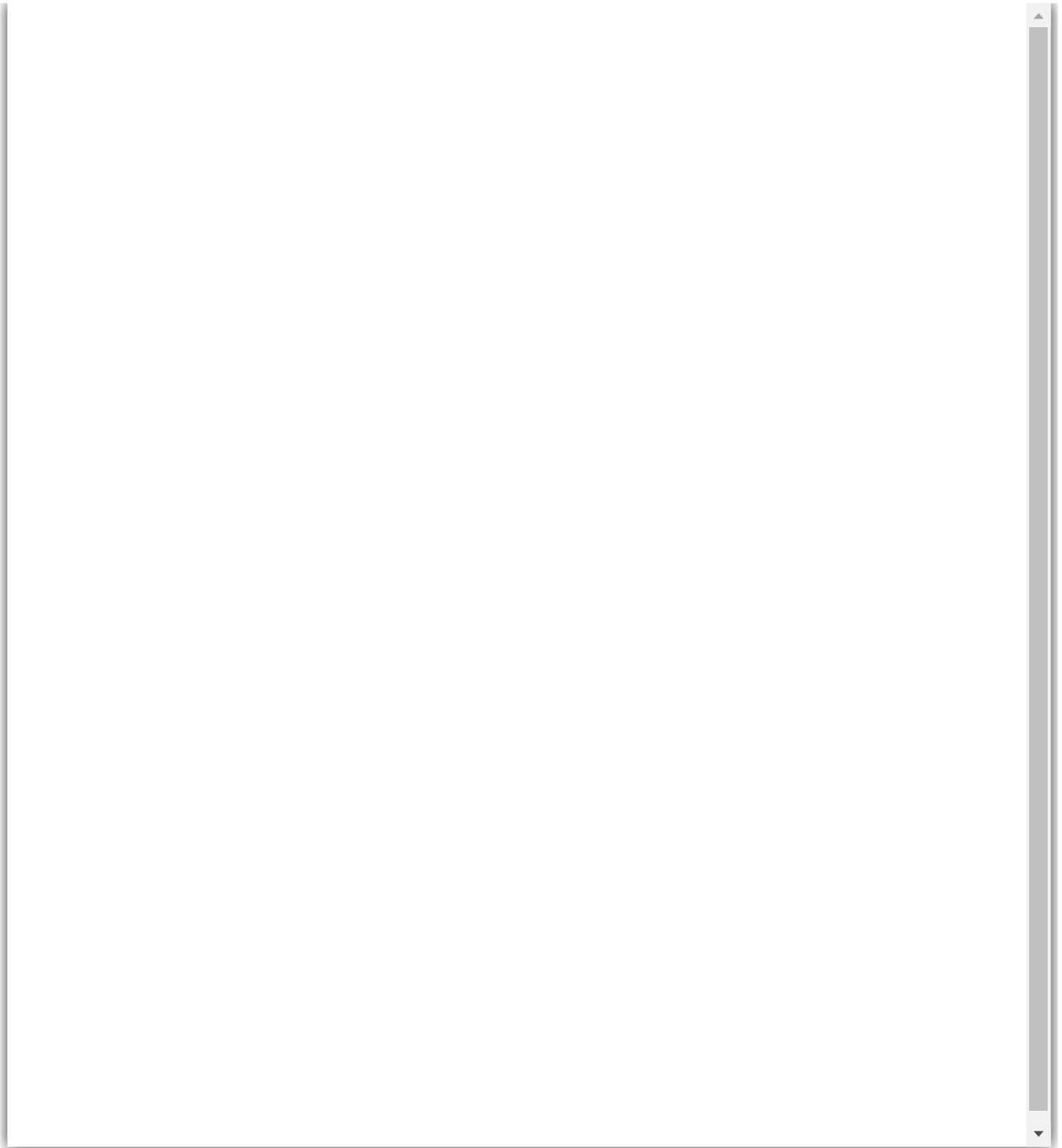
Clearing the existing data creates a fresh table in which we can insert data. If we didn't clear existing data first, then we would find ourselves reinserting data that already exists, thus creating duplicates and making a bit of a mess. It's good practice to clear the existing data first to give ourselves a clean slate to work with.

The line we'll use to clear the data is `tbody.html("");`. But how exactly is this code clearing data?

- `tbody.html` references the table, pointing JavaScript directly to the table in the HTML page we're going to build.

- The parentheses with empty quotes (`("");`) is an empty string.

Basically, this entire line—`tbody.html("");`—tells JavaScript to use an empty string when creating the table; in other words, create a blank canvas. This is a standard way to clear data.

Here is what our code looks like with the addition of this line:

```
function buildTable(data) {
  tbody.html("");
}
```

Now that we have the start of a clean table, let's apply the `forEach` function.