11.2.1 JavaScript Components

Now that Dana knows where she'll encounter JavaScript in the wild—as well as some of the key differences between JavaScript and Python—it's time to get down to business. Understanding the basics of a programming language from text is one thing, but putting that understanding into practice is another. Dana is ready to dive in and start working with some of the basic components: variables and lists.

We're going to begin our practice by familiarizing ourselves with some basic JavaScript components: **variables** and **arrays**.

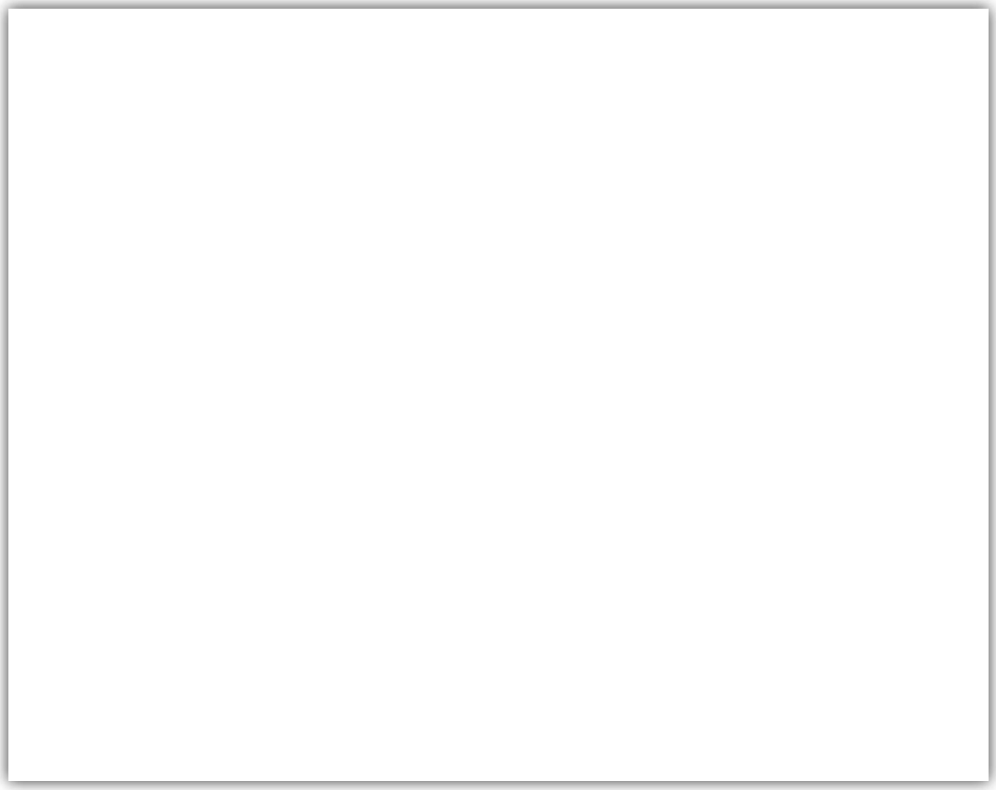
Variables

Before ES6 came along, there was a single way to declare a variable: `var`. You've already worked with variables in Python, but this concept in JavaScript is a bit different. Let's compare a Python variable to a JavaScript variable:

Python	JavaScript
y = 2	`var` y = 2;

Python's way of assigning a variable is quite simple: type the name of the variable followed by its value: `y = 2`.

JavaScript is similar, but with two additions: add `var` before the variable, and then add a semicolon after the value, like this: `var y = 2;`.



Let's test our JavaScript variable assignment using DevTools. If you still have your console open and ready to go, great! If not, go ahead and bring another up to practice with.

SKILL DRILL

If needed, visit a webpage such as **Google (http://www.google.com)** and activate your DevTools. Click the Console tab to activate the console. Then do the following:

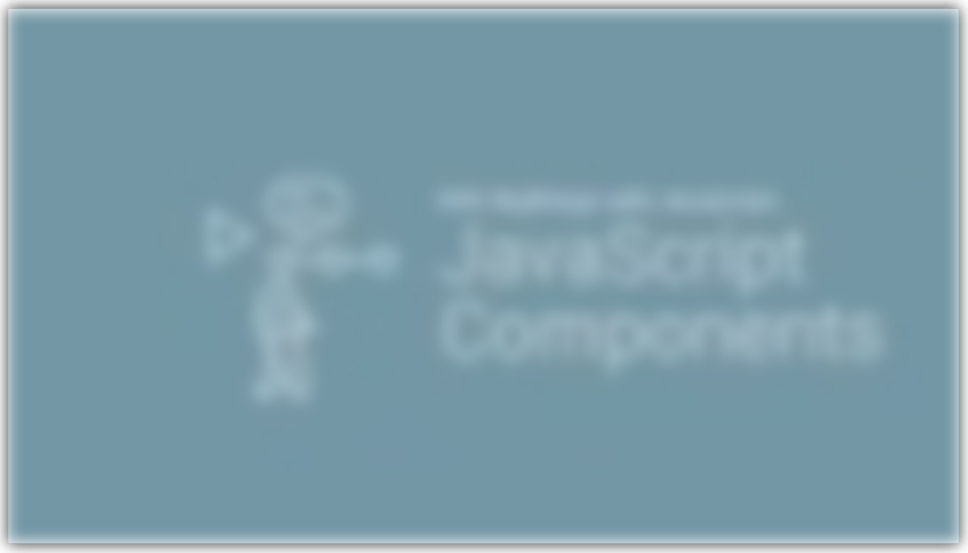
1. On the first line, type `var y = 2;` and press Enter.
2. On the next line, type `console.log(y);` and press Enter.

The value of `y` should print to your console.

Of the many additions that came along with ES6, two more ways to declare variables were also introduced: `let` and `const`.

This can be a bit trickier than it seems, because in JavaScript a variable isn't always just a variable. There are specific uses for different variables,

and using `let` and `const` instead of `var` helps developers define what the uses are. Let's check them out in more detail.



Create Variables with let

The biggest difference between `var` and `let` is that the `var` declaration is global, meaning it applies to the program instead of being contained in a block of code.

When a developer chooses to use `let`, it basically means "I might want to use this variable again later to hold different data, but in this code block I'll only use it once." In ES6+, `let` is typically used in place of `var`. We'll be using `let` in this module, but both are encountered out in the wild.

Create Variables with const

The `const` declaration is more specific than `let`. Instead of being contained within a block of code, `const` tells JavaScript that the variable won't be reassigned or redeclared, either in a block of code or within the program as a whole. The following table highlights the key differences of `var`, `let`, and `const`:

Least specific	var	Variable used in entire program
	let	Variable used in a code block
Most specific	const	Variable used once

Now that we've discovered three different ways to declare variables in JavaScript, let's take a look at arrays.

Arrays

When coding in Python, data can be grouped together in a **list**. The same is true of JavaScript. In fact, Dana was inspired to learn JavaScript because the data is already stored in a JavaScript array! Let's take a look at the data to see what we're working with. Start by downloading the JavaScript file below:

Download data.js (https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_11/data.js)

GITHUB

This `data.js` file is a large part of this project. Save it in the new repository you cloned to your computer.

When you open the file, you'll see that it contains *a lot* of data. Look at the construction of this array.

First, the name of the array, `data`, is declared with `var`:

```
var data = [
```

The structure of the array begins much like a Python list: with a square bracket. But the data inside is arranged a bit differently.

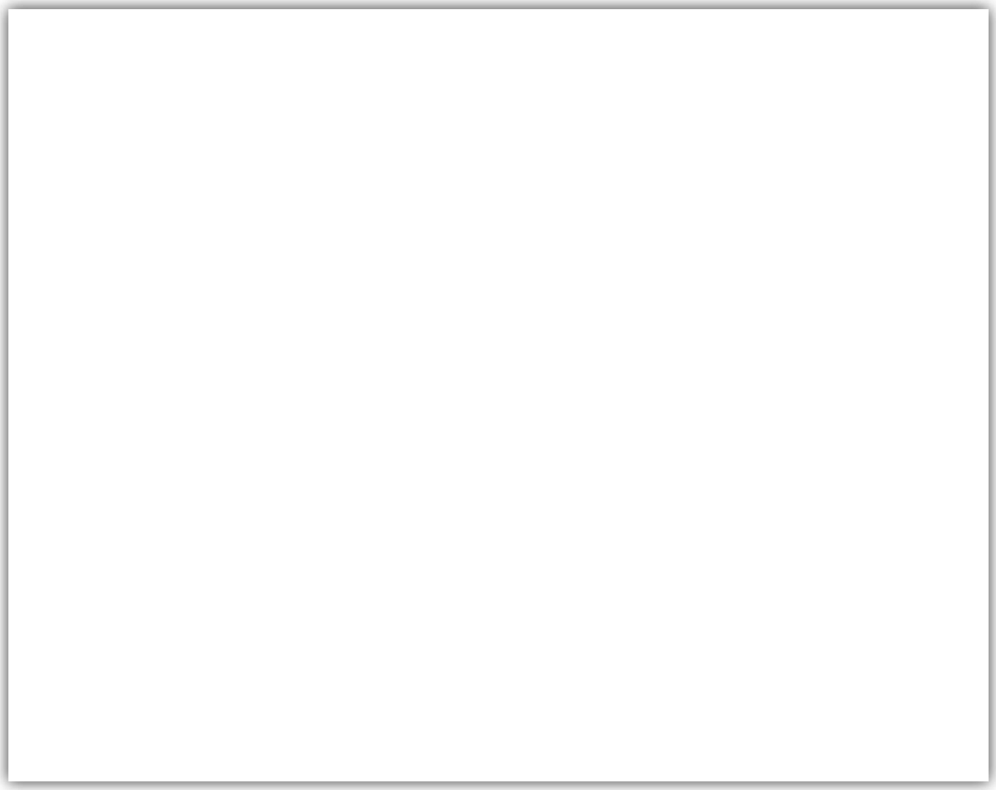
Each entry is inside the square brackets, like a Python list. That's where the similarities begin to taper off.

In this particular JavaScript array, we're not recording a single item and moving on to the next, much like a simple list (such as `[1, 2, 3]`). However, here we're recording an entire event: date, location, type, and even comments are saved inside a single array. Not only that, but multiple events are recorded. Because we have so much information, the array looks more like a Python dictionary than a simple Python list. Take a look at the example below:

```
{
  datetime: "1/1/2010",
  city: "benton",
  state: "ar",
  country: "us",
  shape: "circle",
  durationMinutes: "5 mins.",
  comments: "4 bright green circles high in the sky going in circles then
},
```

Within a set of curly brackets, we can see the key-value pairs such as the date, city, and state.

While this looks somewhat similar to a Python dictionary, there is one key difference. Scroll to the last link of the data file and see if you can spot the difference. You can use keyboard shortcuts to reach the bottom quickly, instead of scrolling through the whole file.



There's a lot going on in this file: for Dana's article to be a success, she will need to make these sightings easier for people to visually parse by converting them from their current state, a JavaScript array, into an HTML table.

Convert the Array to a Table

To convert the array to a table, we're going to take the following code and turn it into the table shown below:

```
{
  datetime: "1/1/2010",
  city: "benton",
  state: "ar",
  country: "us",
  shape: "circle",
  durationMinutes: "5 mins.",
  comments: "4 bright green circles high in the sky going in circles then one bright green light at my front door."
},
{
  datetime: "1/1/2010",
  city: "bonita",
  state: "ca",
  country: "us",
  shape: "light",
  durationMinutes: "13 minutes",
  comments: "Three bright red lights witnessed floating stationary over San Diego New Years Day 2010"
},
```

Date	City	State	Country	Shape	Duration	Comments
1/1/2010	benton	ar	us	circle	5 mins	4 bright green circles high in the sky going in circles then one bright green light at my front door.
1/1/2010	bonita	ca	us	light	13 minutes	Three bright red lights witnessed floating stationary over San Diego New Years Day 2010

The first step in transitioning the data from an array to a table is to create the appropriate variables using `var`, `let`, or `const`. Open VS Code and create a file in our repo folder named `app.js`. This is where we'll keep the code that builds the HTML table and fills it with data from `data.js`.

© 2020 - 2022 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.