11.1.3 Writing JavaScript

While JavaScript is clearly capable of a variety of tasks, Dana plans to start with something a bit more manageable. Instead of building an entire dashboard right away, first she'll create a filterable table to display the data. She decides to dig into the syntax of the language. It's very different from other languages Dana has encountered before, so she wants to be sure she understands the basics before she begins to build anything.

One major component of each coding language is its **syntax**. For example, Python is a pretty clean and easy-to-read language; there aren't many semicolons, and the indentation and spacing makes sense. SQL, on the other hand, includes semicolons, but it also has guidelines and requirements when it comes to indentation and spacing.

JavaScript is no different: there are guidelines and requirements for writing it. But because JavaScript can be added to an HTML page, there are more guidelines and requirements than for languages that can only live in a `.js` file or Jupyter notebook such as Python. There are a few important things to remember about JavaScript syntax. We'll start with the following:

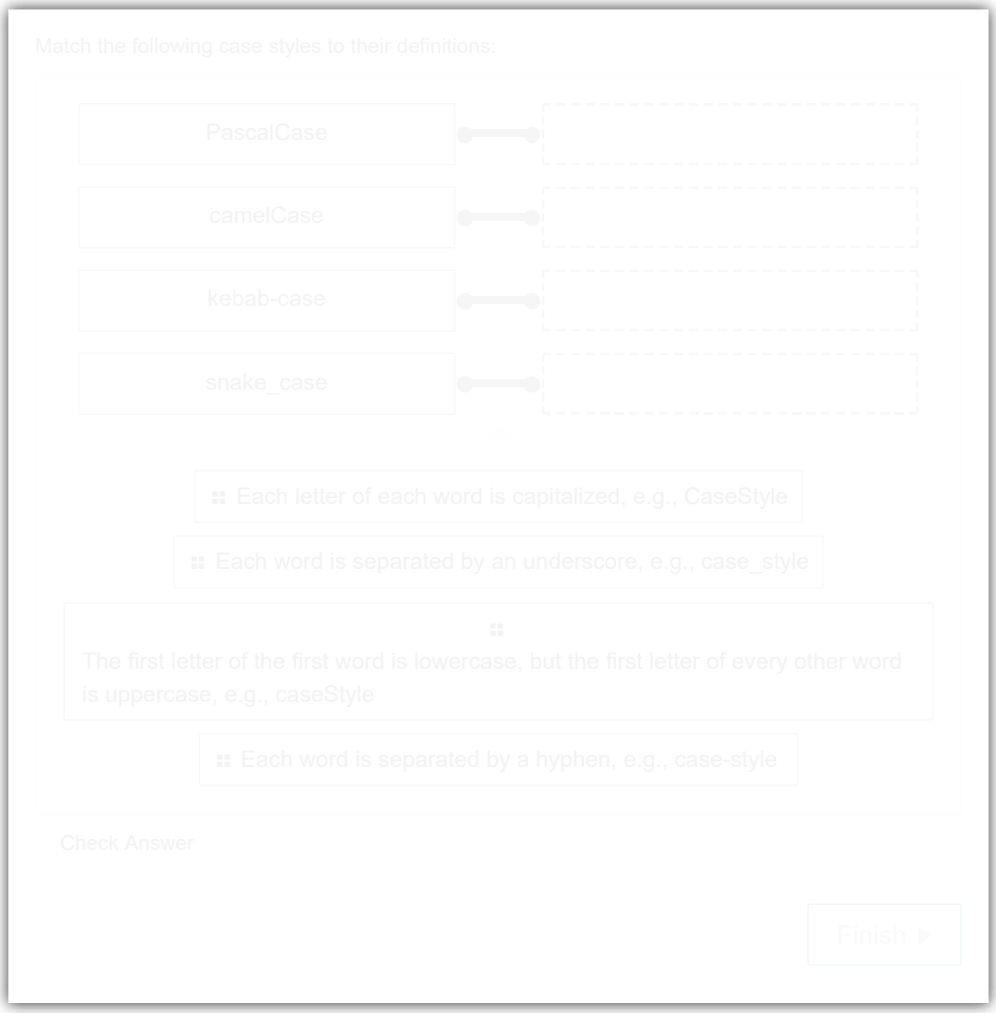
- Case sensitivity
- Semicolons
- Statements and expressions
- Code blocks

We'll be sure to get in lots of practice so that Dana can feel 100% confident in her skills.

Case Sensitivity

JavaScript is case sensitive. **Case sensitivity** means that JavaScript considers upper- and lower-case words to be different. For example, if we were to assign the words "data" and "Data" as variables, we would be able to save different information in each word. Of course, actually doing this with the word "data" could lead to confusion pretty quickly. Instead, just remember JavaScript cares about capital letters.

Similarly, JavaScript uses different naming conventions than Python that involve case sensitivity. Different languages utilize different methods to link words without using spaces, which is called a **case style**.



JavaScript's code style, according to coding guidelines and syntax, is camel case. You'll encounter this case often as you begin to practice your coding. It's especially useful when declaring variables.

NOTE

> Camel case is the preferred naming convention in JavaScript. This is especially helpful in cases where Python data is used. For example, we would know that variables named with snake case originated from the Python side of things.

Semicolons

Much like SQL, when coding in JavaScript it's good practice to end statements with a semicolon. Technically, they are optional when it comes to executing your code, but they are helpful because they tell JavaScript that a particular line or block of code is complete. It's considered a best practice to include semicolons throughout your code. You'll encounter many semicolons throughout this module.

Let's use a print statement as an example. In JavaScript, a print statement is called a **console log**. To print "Hello, world!" to the console, we would use this line:

```
// Printing a string with JavaScript
console.log("Hello, world!");
```

NOTE

> While the `print()` function does exist in JavaScript, it will actually try to print to a printer instead of our console.

This statement is almost identical to a basic Python print statement, as shown below.

```
# Printing a string with Python
print("Hello, world!")
```

Both methods will print the string (in this case, "Hello, world!"). But in addition to switching "print" with "console.log," in JavaScript, a semicolon has been added at the end of the statement.

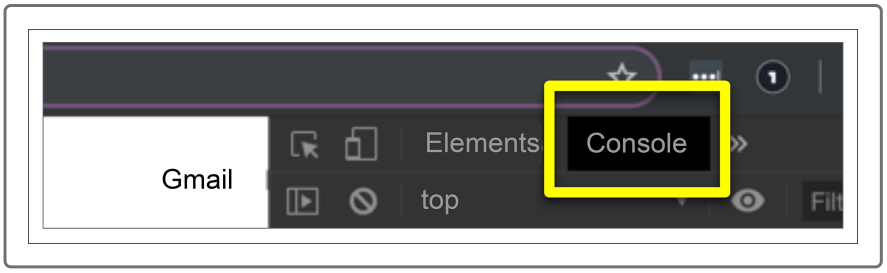
Testing Simple Statements

Simple JavaScript statements such as `console.log()` can be tested using DevTools. For example, follow these steps to test `console.log("Hello, world!")`.

1. Go to a site like **Google (http://www.google.com)** and activate your DevTools. This is where we'll access our console; the console is the command line interface tool we'll use to test JavaScript, much like our terminal is used to test Python.

   You can use any site to open your DevTools; it isn't a requirement to use the Google search page.

2. Click the "Console" tab at the top of the screen.



3. Type `console.log("Hello, world!");` on the first line and then press Enter.

The Console tab in DevTools will become a very important tool when we begin to code later on. The Console tab will allow us to see if an error has occurred and, if so, which line of code is causing the disruption.

Why will we be using DevTools to check and debug our code?

- ○ We'll use DevTools because using VS Code to run and debug JavaScript code takes longer and doesn't look as neat.
- ○ We'll use DevTools because DevTools will keep code in memory if we refresh the page.
- ○ We'll use DevTools because the console available in DevTools is our JS command line interface.

Check Answer

Finish ▶

Statements and Expressions

When describing JavaScript code, the terms "statements" and "expressions" are both used, and often. Here's how to distinguish between the two:

- Statements perform actions.
- Expressions create values.

Assigning a variable is an example of a statement. Using arithmetic to create a new value is an expression.

Code Blocks

Code blocks, which we will see more often as we start writing functions, are denoted by curly brackets. Code inside the curly brackets are typically indented two to four spaces. This isn't required to run the code, but it does make reading it easier and follows the coding guidelines.