Variables and Operators

Foundations Course

Introduction

In the previous sections you learned how to structure webpages with HTML and style them with CSS. The next step is to make the webpage *interactive*, which is exactly what JavaScript is for. In this section, we will focus on the fundamentals of JavaScript and how you can use it to manipulate all the various interactions between the web page and user.

Lesson overview

This section contains a general overview of topics that you will learn in this lesson.

- How do you declare a variable?
- What are three different ways to declare a variable?
- Which one should you use when?
- What are the rules for naming variables?
- What are operators, operands, and operations?
- What is concatenation and what happens when you add numbers and strings together?
- What are the different types of operators in JavaScript?
- What is the difference between `==` and `===`?
- What are operator precedence values?
- What are the increment/decrement operators?
- What is the difference between prefixing and postfixing them?
- What are assignment operators?
- What is the Unary Plus Operator?

How to run JavaScript code

All JavaScript we will be writing in the majority of the Foundations course will be run via the browser. Later lessons in Foundations and the NodeJS path will show you how to run JavaScript outside of the browser environment. Outside of these lessons, for now you should always default to running your JavaScript in the browser unless otherwise specified, otherwise you may run into unexpected errors.

The simplest way to get started is to create an HTML file with the JavaScript code inside of it. Type the basic HTML skeleton into a file on your computer somewhere:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Page Title</title>
</head>
<body>
  <script>
    // Your JavaScript goes here!
    console.log("Hello, World!")
  </script>
</body>
</html>
```

Save and open this file up in a web browser (you can use "Live Server" on Visual Studio Code to do this!) and then open up the browser's console by right-clicking on the blank webpage and selecting "Inspect" or "Inspect Element". In the 'Developer Tools' pane find and select the 'Console' tab, where you should see the output of our `console.log` statement.

> *`console.log()` is the command to print something to the developer console in your browser. You can use this to print the results from any of the following articles and exercises to the console. We encourage you to code along with all of the examples in this and future lessons."*

Another way to include JavaScript in a webpage is through an external script. This is very similar to linking external CSS docs to your website.

```
<script src="javascript.js"></script>
```

JavaScript files have the extension `.js` similar to `.css` for stylesheets. External JavaScript files are used for more complex scripts.

Variables

You can think of variables as "storage containers" for data in your code.

Until recently there was only one way to create a variable in JavaScript — the `var` statement. But in the newest JavaScript versions we have two more ways — `let` and `const`.

1. This tutorial on JavaScript variables will explain everything you need to know! Be sure to do the **Tasks** at the end. Information won't stick without practice!

The above tutorial mentioned this, but it's important enough to note again: `let` and `const` are both relatively new ways to declare variables in JavaScript. In *many* tutorials (and code) across the internet you're likely to encounter `var` statements. Don't let it bother you! There's nothing inherently wrong with `var`, and in most cases `var` and `let` behave the same way. But sometimes the behavior of `var` is *not* what you would expect. Just stick to `let` (and `const`) for now.

Numbers

Numbers are the building blocks of programming logic! In fact, it's hard to think of any useful programming task that doesn't involve at least a little basic math... so knowing how numbers work is obviously quite important. Luckily, it's also fairly straightforward.

1. This W3Schools lesson on JavaScript arithmetic followed by this on JavaScript numbers, are good introductions to what you can accomplish with numbers in JavaScript.
2. This MDN article on JavaScript math covers the same info from a slightly different point of view, while also teaching you how to apply some basic math in JavaScript. There's much more that you can do with numbers, but this is all you need at the moment.
3. Read through (and code along with!) this article on JavaScript operators. Don't forget to do the "Tasks" at the bottom of the page! It will give you a pretty good idea of what you can accomplish with numbers (among other things!) in JavaScript.

Assignment

Try the following exercises (and don't forget to use `console.log()`!):

1. Add 2 numbers together! (just type `console.log(23 + 97)` into your HTML file)
2. Add a sequence of 6 different numbers together.
3. Print the value of the following expression: `(4 + 6 + 9) / 77`
   - Answer should be approximately `0.24675`
4. Let's use variables!
   - Type this statement at the top of the script tag: `let a = 10`
   - In the console `console.log(a)` should print `10`
   - Try the following in the console: `9 * a`
   - and this: `let b = 7 * a` (returns `undefined` *) and then `console.log(b)`
5. You should be getting the hang of this by now... try this sequence:
   - Declare a constant variable `MAX` with the value `57`
   - Set another variable `actual` to `MAX - 13`
   - Set another variable `percentage` to `actual / MAX`
   - If you type `percentage` in the console and press `Enter` you should see a value like `0.7719`
6. Take a few minutes to keep playing around with various things in your script tag. Eventually, we will learn how to actually make those numbers and things show up on the webpage, but all of this logic will remain the same, so make sure you're comfortable with it before moving on.

> As you might have noticed by running JavaScript code in the console, the console prints the result of the code it executes (called a return statement). You will learn more about these in the next lessons, however for now it is good to remember that a declaration with an assignment (such as `let b = 7 * a`) returns `undefined` and so you cannot declare and assign a value to a variable and read its value in the same line.

Knowledge check

The following questions are an opportunity to reflect on key topics in this lesson. If you can't answer a question, click on it to review the material, but keep in mind you are not expected to memorize or master this knowledge.

- Name the three ways to declare a variable
- Which of the three variable declarations should you avoid and why?
- What rules should you follow when naming variables?
- What happens when you add numbers and strings together?
- How does the Modulo (%), or Remainder, operator work?
- Explain the difference between `==` and `===`.
- When would you receive a `NaN` result?
- How do you increment and decrement a number?
- Explain the difference between prefixing and postfixing increment/decrement operators.
- What is operator precedence and how is it handled in JS?
- How do you access developer tools and the console?
- How do you log information to the console?
- What does unary plus operator do to string representations of integers? eg. `+"10"`

Additional resources

This section contains helpful links to related content. It isn't required, so consider it supplemental.

- The differences between `var` and `let` are explained in this JavaScript.info article titled the old "var".
- This MDN article on what is JavaScript explains a bit more about it on a high-level.