## **Fundamentals Part 1**

Let’s dive right into JavaScript!

### **Learning Outcomes**

Look through these now and then use them to test yourself after doing the assignment:

* 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 post-fixing them?
* What are assignment operators?
* What is the “Unary +” 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 simply create an HTML file with the JavaScript code inside of it. Type the basic HTML skeleton into a file on your computer somewhere:



Save and open this file up in a web browser (you can use Live Server to do this!) and then open up the browser’s console by right-clicking on the blank web page 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.

### **Variables**

You can think of variables as simply “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 variable tutorial](http://javascript.info/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. The precise differences between var and let will be explained later.

### **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](https://www.w3schools.com/js/js_numbers.asp) followed by [this one](https://www.w3schools.com/js/js_arithmetic.asp), are good introductions to what you can accomplish with numbers in JS.
2. [This MDN article](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Math) covers the same info from a slightly different point of view. 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](http://javascript.info/operators) about operators in Javascript. 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 solution to the following equation: (4 + 6 + 9) / 77
   1. Answer should be approximately 0.24675
4. Let’s use variables!
   1. Type the following at the top of the script tag: let a = 10
   2. console.log(a) should print 10
   3. Try the following: 9 \* a
   4. 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:
   1. Declare a constant variable max with the value 57
   2. Set another variable actual to max - 13
   3. Set another variable percentage to actual / max
   4. 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 an assignment (such as b = 7 \* a) returns undefined and so you cannot assign a value to a variable and read its value in the same line.`*

### **Additional Resources**

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

[Hacker Noon on why you shouldn’t use var anymore](https://hackernoon.com/why-you-shouldnt-use-var-anymore-f109a58b9b70)

### **Knowledge Check**

This section contains questions for you to check your understanding of this lesson. If you’re having trouble answering the questions below on your own, review the material above to find the answer.

* [Name the three ways to declare a variable?](https://www.theodinproject.com/paths/foundations/courses/foundations/lessons/fundamentals-part-1#variable-declaration)
* [Which of the three variable declarations should you avoid and why?](https://www.theodinproject.com/paths/foundations/courses/foundations/lessons/fundamentals-part-1#avoid-var)
* [What rules should you follow when naming variables?](https://javascript.info/variables#variable-naming)
* [What should you look out for when using the + operator with numbers and strings?](https://javascript.info/operators#string-concatenation-with-binary)
* [How does the % operator work?](https://javascript.info/operators#remainder)
* [Explain the difference between == and ===.](https://www.w3schools.com/js/js_numbers.asp)
* [When would you receive a NaN result?](https://www.w3schools.com/js/js_numbers.asp)
* [How do you increment and decrement a number?](https://javascript.info/operators#increment-decrement)
* [Explain the difference between prefixing and post-fixing increment/decrement operators.](https://javascript.info/operators#increment-decrement)
* [What is operator precedence and how is it handled in JS?](https://javascript.info/operators#operator-precedence)
* [How do you access developer tools and the console?](https://www.theodinproject.com/paths/foundations/courses/foundations/lessons/fundamentals-part-1#access-devTools-console)
* [How do you log information to the console?](https://www.theodinproject.com/paths/foundations/courses/foundations/lessons/fundamentals-part-1#console-log)
* [What does unary plus operator do to string representations of integers?](https://javascript.info/operators#numeric-conversion-unary)