

# JavaScript Basics

---

## Setup

You would make a JavaScript file just like any other file. You can name it however you'd like, but the filename must end in `.js`. There isn't anything you need to do in this file before writing your code, however you will want to connect your HTML to JS like we did with CSS. It's a bit different with JS though:

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

This script tag will typically be placed just before the closing `</body>` tag.

## Variables

Variables are a way to store information for the program. For example: the user's username, or keeping track of how many times the user guessed a number in a guessing game, or anything else the computer needs to remember.

Variables are 'declared' using the keywords **let** or **const**, followed by the name of the variable, and the initial value you want to set it to. Putting that together, it would look like this:

```
let variableName = value;
```

OR

```
const variableName = value;
```

The difference between let and const:

- let: The variable's value can change.
- const: The variable's value cannot change.

```
// Variables are typically named using a convention called camelCase. The first  
word is lower case, and every following word is capitalized.
```

```
let numGuesses = 0;  
const username = "Luce";
```

There is one more variable declaration keyword called **var** - you might see some variables declared like this in some older code:

```
var variableName = value;
```

The code above essentially acts the same as if you were to declare the variable with let or const. However, **var** is not specific about which variables can or cannot change value. So, in 2015, let and const were introduced as replacements to var. You can still use var, but it's not standard, and other programmers will probably look at you weird if you do.

You can name the variable whatever you want, but you're going to want to name it something that describes what data it's storing. For example, naming a variable `pancake` wouldn't make any sense if it's a variable that's used for storing the user's username, and would probably confuse you or other programmers later on. Additionally, there are certain words called **keywords** that are reserved by the language itself. Two keywords that we've already learned about are `let` and `const` - you can't use these two words for variable names, and it will give you an error if you try. There are a few other keywords that we'll talk about later.

## Reassigning Variables

If you declared a variable using `let` and you want to change the value of it later, you can just reassign it without the keyword using the equal sign.

You **cannot** reassign `const` variables - it will give you an error.

```
let numGuesses = 0;
numGuesses = 3; // Now numGuesses equals 3 instead of 0

const username = "Luce";
username = "Justin"; // Will give an error, since it was already declared with
const
```

If you want to increase or decrease the number of a variable, there are a couple ways you can do this:

```
numGuesses++; // Adds 1
numGuesses += 5; // Adds 5
numGuesses = numGuesses + 5; // This does the same thing as numGuesses += 5

numGuesses--; // Subtracts 1
numGuesses -= 5; // Subtracts 5
numGuesses = numGuesses - 5; // This does the same thing as numGuesses -= 5

// You can also do multiplication and division
numGuesses *= 2; // Multiplies by two
numGuesses /= 2; // Divides by two

numGuesses + 1; // INCORRECT, will not increase by 1. Why?
```

The `+=` means that it's adding an amount, and then assigning it to the variable at the same time. If you just did `numGuesses + 1`, it would add 1, but not assign it to the variable, so `numGuesses` would technically still be 0.

## Types

Variables each have a type. This lets the computer know whether what you're storing is a number, a word, or something else.

There are 5 different types in JavaScript.

1. Number - any number, including decimals, like 4, 4.53, -26.1239841
2. String - text like "word", 'this is a sentence'. Has single quotes or double quotes around the text (your preference).
3. Boolean - true or false
4. Null - variables that are created intentionally without any value
5. Undefined - variables that have no value or have not been declared yet

While 'null' and 'undefined' might seem similar, undefined usually happens automatically, whereas null is set on purpose.

Examples:

```
const time = 10; // Integer
let randomNumber = 34.56; // Number
const greeting = "Hello!" // String
const farewell = 'Goodbye!' // String, but with single quotes
let lightMode = true; // Boolean
let nullVar = null; // null
let noValue; // undefined

doesNotExist += 1; // Also undefined, because we didn't declare a variable called
doesNotExist. This will give an error.
```

Keep in mind that variable types in JavaScript don't really matter to the computer, and you can change the type of a variable if you need to. For instance, you could do something like this:

```
let changingType = 23;
changingType = "It's a string now!";
```

**This is not true in most other programming languages.** JavaScript is referred to as a 'weakly typed' language, meaning that variables can be reassigned to a different type. If you tried to do something like the above code in C#, C++, or Java, it would give you an error.

## Syntax

You'll notice in this file that each line is concluded with a semicolon. This lets the computer know that we're finished with a command. Unlike other languages, JavaScript automatically adds the semicolons at the end of each line, even if the programmer doesn't. However, there are edge cases where omitting them can cause unexpected behavior, so it's a good habit to include them.

Unfortunately, writing comments is different in JavaScript compared to HTML/CSS. Fortunately, it's easier - all you have to do is write two backslashes:

```
// This is a commented line

const variable = 1; // Commenting some stuff after some code
```

```
/*  
You can comment multiple lines with the backslash and asterisk  
  
const variableTwo = 2; Everything inside these asterisks will be commented  
  
End the comment with an asterisk first, then the backslash  
*/
```

Shortcut: you can hold down CTRL and press backslash to comment or uncomment the entire line you're on. You can also highlight a block of code and press CTRL backslash to comment or uncomment all of it at once.

## Logging

Lastly, in order to see and check our variables, we can use a command called `console.log()`. This will print whatever you put in between the parenthesis to the console.

```
let numGuesses = 0;  
console.log(numGuesses); // Prints 0 to the console
```

The console should automatically open when you run the JS file, or run the HTML file that's connected to the JS. In case it doesn't open, you can open up the terminal box by holding down CTRL and pressing the backtick key (the key to the left of the number 1 at the top). There should be a tab called "Debug Console" which is where it's logged.

It's also logged in the browser, if you right click on the page, select "Inspect Element" and go to the console tab on the top right of the new window that opened (you might have to refresh the page for this to work).