

JavaScript Basics

Home

- [JavaScript Overview](#)
 - [Variable Declaration](#)
 - [Variable Assignment](#)
 - [Data Type](#)
 - [Conditionals](#)
 - [Loops](#)
 - [Function](#)
- [Projects](#)
- [Assignments](#)
- [Quiz](#)

JavaScript Basics

JavaScript is a cross-platform, object-oriented scripting language. JavaScript is extremely popular for a variety of reasons. It is a small and lightweight language allowing maximum flexibility for developers to take it in a bunch of different directions. JavaScript lives inside a host environment (a web browser or Node server), it can be connected to the objects of these environments to provide programmatic control over them.

- [Variable Declaration](#) JavaScript variables are containers for storing data values - imagine a cup you fill with coffee, the cup holds the coffee, a variable holds a value. All JavaScript variables must be identified with unique names. These unique names are called identifiers. `var x;`
- [Variable Assignment](#) Assignment operators assign values to JavaScript variables - our cup can now have coffee poured in it, giving our variable a value to hold. The `=` assignment operator assigns a value to a variable. `var x = 10;`
- [Data Types](#) Data types are an important concept; to be able to operate on variables you need to know the data type. There are six data types that are JavaScript primitives: Boolean - `true` or `false`; null - `null` aka nothing; Number - `42` or `3.14159`; String - `"Coding Dojo Rocks!"`; Array - `[1, 'Coding', 2, 'Dojo']`; and Object - `{first_name: 'Jane', last_name: 'Doe'}`
- [Conditionals](#) When you write code, you want to perform different actions for different decisions - hitting different code blocks based on values or conditions that have been met. You can use conditional statements in your code to accomplish this. There are the following conditional statements: `if` a specified condition is true, do this code in our code block; `else if` to specify a new condition to test, if the first condition is false; `else` we execute this block of code;
- [Loops](#) There are many different kinds of loops in every programming language, but they all essentially do the same thing: they will repeat an action some number of times. Imagine you have to run a mile, well you run around the track four times and then you stop. That's a loop!
- [Function](#) Functions are an encapsulation of a code block. When we call our function this will run that code block. Think of it as a list of instructions. As an example imagine we are putting together a desk from Ikea, we open up the instruction manual and get started, first we screw the legs to the table top; next we place the table the right way up. Done! We finished our instructions. Sadly there are a ton more tables to do so let's call our function over and over and over again.

For more useful information check out this url: [JavaScript!](#)

JavaScript Basics

Recreate the image below using only what you've learned about HTML and CSS. Use **display: inline-block/inline/block** to do all of these assignments. You can download the image here: [JavaScript Basics](#)

Before you submit your work:

Remember that you're here to learn: Don't try to find the original source code of these images.

Make sure you've validated your HTML.

Also, do NOT make this responsive. Just make this work for a fixed width (set the width to be 970px). If you have time during the bootcamp, go through our responsive design course *later*, but, for now, try to master the CSS principles for fixed width first.

The responsive design course is added at the end of the Web Fundamentals track, and you can do this later in your bootcamp.