Two Kinds of Data

In JavaScript there are two different kinds of data: primitives, and objects. A primitive is simply a data type that is not an object, and has no methods.
In JS, there are six primitive data types:

- Boolean
- Number
- String
- Null
- Undefined
- Symbol

- **Boolean**

A boolean represents only one of two values: true, or false. Think of a boolean as an on/off or a yes/no switch.

```
var boo1 = true;
var boo2 = false;
```

- **Number**

There is only one type of Number in JavaScript. Numbers can be written with or without a decimal point. A number can also be +Infinity, -Infinity, and NaN (not a number).

```
var num1 = 32;
var num2 = +Infinity;
```

- **String**

Strings are used for storing text. Strings must be inside of either double or single quotes. In JS, Strings are immutable (they cannot be changed).

```
var str1 = 'hello, it is me';
var str2 = "hello, it's me";
```

- **Null**

Null has one value: null. It is explicitly nothing.

```
var nothing = null;
```

- **Undefined**

A variable that has no value is undefined.

```
var testVar;
console.log(testVar); // undefined
```

- **Symbol**

Symbols are new in ES6. A Symbol is an immutable primitive value that is unique. For the sake of brevity, that is the extent that this article will cover Symbols.

```
const mySymbol = Symbol('mySymbol');
```

What about Objects?

Objects are not a primitive data Type.

An object is a collection of properties. These properties are stored in key/value pairs. Properties can reference any type of data, including objects and/or primitive values.

```
var obj = {
 key1: 'value',
 key2: 'value',
 key3: true,
 key4: 32,
 key5: {}
}
```

Loosely Typed

JavaScript is a loosely typed language. This means you don't have to declare a variable's type. JavaScript automatically determines it for you. It also means that a variables type can change. Let's look at an example:

We'll create a variable named car and set it equal to a string value:

```
var car = 'ford';
```

Later, we realize we want the value of car to be the year it was made, so we change car to a number:

```
car = 1998;
```

It works — and JavaScript could care less. Because JS is loosely typed, we are free to change variable types as we please.