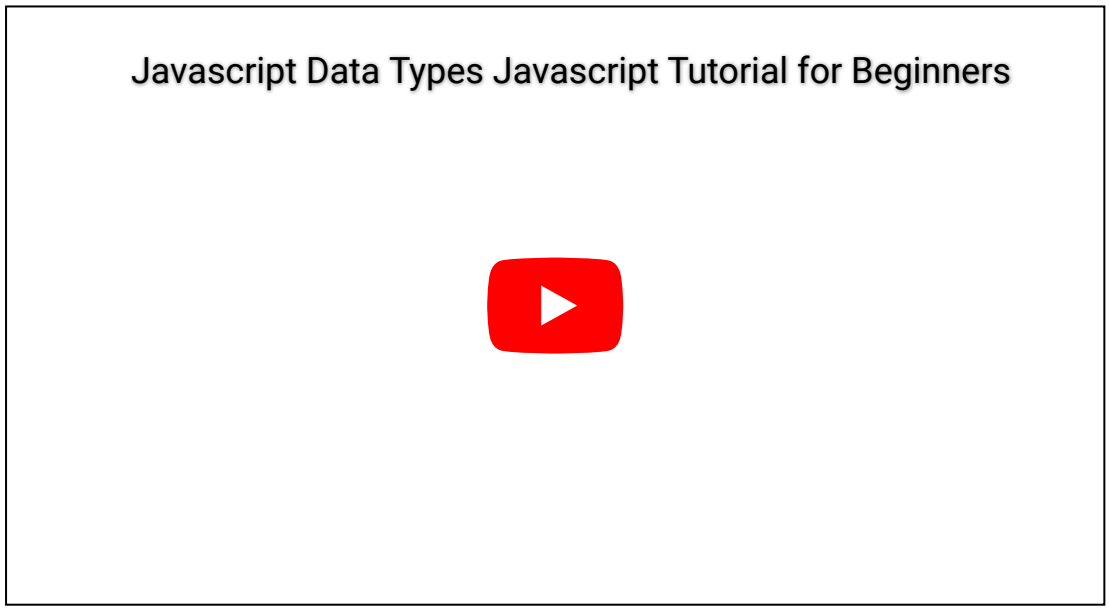
Data Types

Most programming languages help you create values that symbolize a number, a character in a text, or a longer text. You can also symbolize the concept of true and false values using booleans. You can even create values that symbolize the absence of a value.

In JavaScript, there are **six primitive types**:

1. **Boolean** (true or false)
2. **Number** (including integers like 1, -2, and floating point numbers like 3.14, 2e-3)
3. **String** (Strings are used for storing text. Strings must be inside of either double or single quotes.)
4. **Null** (Null has one value: null. It is *explicitly nothing*.)
5. **Undefined** (A variable that has no value is undefined.)
6. **Symbol** (We'll get to this later on in the course.)



Data Types

PS: Notice how `typeofnull` incorrectly prints "object"? Well that's a bug in JS. Which brings us to the second kind of data types in JavaScript:

Composite or Non Primitive Type, like the **Object** we just saw.

Objects can be: "Object literals", "Arrays", "Function", "RegEx", "Dates", ect.

For the time being, our main focus is object literals.

```
console.log(typeof(true)) // prints boolean
console.log(typeof(9000)) // prints number
console.log(typeof("Übermensch")) // prints string
console.log(typeof(anUndefinedVar)) // prints undefined
console.log(typeof(null)) // prints object
```

Data Types

A JavaScript object literal is a comma-separated list of **key-value pairs** wrapped in curly braces. These values can reference any type of data including objects and/or primitive values.

We'll be covering objects (& arrays) in more detail later on in this course. For now, let's simply keep in mind that object literals can encapsulate a multitude of data, enclose it in a tidy package, and have the structure shown in the code box.

```
var person1 = {
        name: "foulan",
        age: 9000,
        isStudent: true
}
```

Previous next