# Understanding the types in JavaScript

TypeScript adds a rich type system to JavaScript. Before we start to use this rich type system, we need to learn what types we have in JavaScript.

WE'LL COVER THE FOLLOWING



## Primitive types in JavaScript #

JavaScript does have some basic primitive types, but what are they?

The JavaScript code below outputs the types of four variables with various kinds of values. We'd hope for `string`, `number`, `boolean`, and `date` to be output ... but is this the case? Run the code and find out.

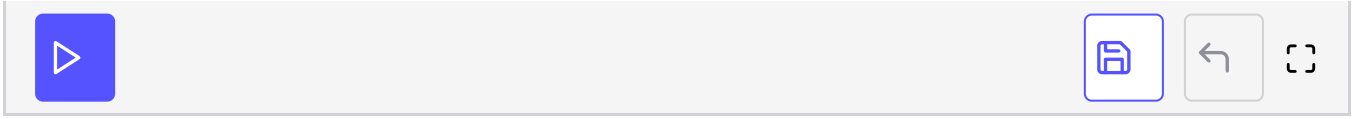
JavaScript

```
const name = "Bob";
console.log("Type of name:", typeof name);

const age = 30;
console.log("Type of age:", typeof age);

const cool = true;
console.log("Type of cool:", typeof cool);

const dateOfBirth = new Date(1989, 10, 5);
console.log("Type of dateOfBirth:", typeof dateOfBirth);
```



Nearly! JavaScript does have `string`, `number`, and `boolean` types, but there is no specific `date` type; instead, dates are of type `object`.

Let's play with the other primitive types available in JavaScript which are `bigint`, `null`, `undefined`, and `symbol`. Let's run the code below:

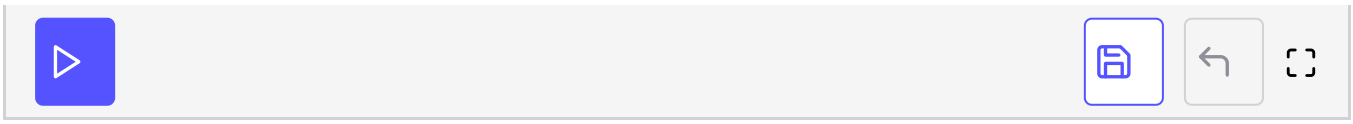
JavaScript

```
const count = BigInt(452947234234);
console.log("count", typeof count);

const address = null;
console.log("address", typeof address)

const phone = undefined;
console.log("phone", typeof phone)

const stars = Symbol('***');
console.log("stars", typeof stars)
```



You may have noticed that the type of a `null` variable is `object` and not `null` as we would expect. This is a bug in JavaScript!

For more information on the types in JavaScript, see this MDN doc.

## JavaScript is loosely typed #

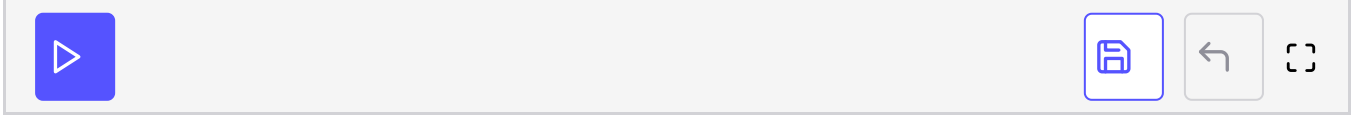
In loosely typed languages, you don't have to specify types of variables. JavaScript is a loosely typed language.

Examine the code below. The `amount` variable starts as a `number` but is then assigned to a `string.` We'd hope to get an error or a warning on line 4, but do we?

JavaScript

```
let amount = 10;
console.log("amount", typeof amount);

amount = "Eight";
console.log("amount", typeof amount);
```



We don’t because JavaScript is loosely typed. There is nothing to prevent a variable from changing its type in JavaScript.

## Wrap up #

The take away from this lesson is that we only have a few useful types in JavaScript. JavaScript does not have the capability to declare that a variable is of a particular type. This means that the type is inferred from the value it is assigned. There is no strict type checking in JavaScript either, which means that a variable can be changed to hold any value.

The lack of a rich type system with a type-checking process is what TypeScript brings to the table. We’ll learn how to use this type system in this course to make us more productive when developing React apps.

In the next lesson, we will learn how to declare variables with types in TypeScript.