# Intro to JavaScript
## Primitive Data Types

**Learning objective:** By the end of this lesson, students will be able to declare and use primitive data types in JavaScript.

# Primitive data types

Primitive types are data types that are stored directly in memory. JavaScript has seven primitive data types, five of which are common and two of which are uncommon.

Here are the common primitive data types in JavaScript:

- String
- Number
- Boolean
- Undefined
- Null

## string

A string represents textual data with zero or more characters wrapped by single or double quotation marks, such as `"John"` or `'Jane'`. Either can be used; just be consistent in your code. A pair of quotes with nothing between them is still a string. Specifically, this is an empty string.

```
let myString = 'Hello World';
let myOtherString = "Hello World again";
console.log(typeof myString);
// Prints: string
```

Note that the `typeof` operator returns a string describing the data type.

## number

A number represents a numeric value.

Unlike many other programming languages, there is no distinction between integer (`15`, `3`, etc.) and floating-point/decimal types (`17.24`, `3.1416`, etc.).

Internally, JavaScript represents all numbers as floating-point values.

```
let myNumber = 15;
console.log(typeof myNumber);
// Prints: number
```

> ❓ What will the following code print to the console?
>
> ```
> console.log(typeof 12.34);
> ```

## boolean

Whereas strings and numbers can have a virtually unlimited number of different values, the boolean data type only has two possible values: `true` and `false`.

Booleans are useful for their simplicity, especially when making decisions based on data. For example, say you're building a game. When a player beats the game, you may check the user's score against the current high score of that game and see if a new record has been achieved. This would resolve in one of two ways - either the user has scored higher than the previous high score, or they didn't. This could be represented using boolean values.

## undefined

A variable that has not been assigned a value is of type `undefined`. For example:

```
let cohort;
console.log(cohort);
// Prints: undefined
```

## null

The null data type has only one value: `null`. null is the value of a variable that explicitly has no value. Thus, it represents a ***deliberate and intentional*** absence of value as opposed to `undefined`, which only represents a lack of assigned value.

`null` does have some strange behavior:

```
console.log(typeof null);
// Prints: object
```

Wait, what? This is a known bug within JavaScript, but it remains to this day because many applications that depend on this (incorrect) behavior have been written.

Because of this, the use of `null` is somewhat controversial in the JavaScript community.

> ❓ Do all variables have a data type?

# 🔨 You do

Strings, numbers, and booleans are the most common data types used to model real-world application data.

For example, in a multiplayer gaming app, we would represent a player's name using a string.

Let's brainstorm a couple of examples for each of the three common data types that would be used to represent information in that gaming app:

| Data Type | Example Data/Information |
| --- | --- |
| String | playerName |
| Number | |
| Boolean | |

# Uncommon primitive data types

There are two uncommon primitive data types in JavaScript:

- Symbol
- BigInt

You're unlikely to encounter them in most circumstances.