

Values and Types

This lesson explains the primary concepts of types and values used in all programming languages.

WE'LL COVER THE FOLLOWING ^

- Number
- String

A *value* is a piece of information used in a computer program. Values exist in different forms called types. The *type* of a value determines its role and operations available to it. Every computer language has its own types and values. Let's look at two of the types available in JavaScript.

Number

A *number* is a numerical value (thanks Captain Obvious). Let's go beyond that though! Like mathematics, you can use integer values (or whole numbers) such as 0, 1, 2, 3, etc, or real numbers with decimals for greater accuracy. Numbers are mainly used for counting. The main operations you'll see are summarized in the following table. All of them produce a number result.

Operator	Role
<div>+</div>	Addition
<div>-</div>	Subtraction
<div>*</div>	Multiplication
<div>/</div>	Division

String

A *string* in JavaScript is text surrounded by quotation marks, such as `"This is a string"`. You can also define strings with a pair of single quotes: `'This is another string'`. The best practice for single or double quotes is a whole political thing. Use whichever you like, but don't mix the two in the same program!

Always remember to close a string with the same type of quotation marks you started it with.

To include special characters in a string, use the `\` character (backslash) before the character. For example, type `\n` to add a new line within a string: `"This is\na multiline string"`.

```
console.log("This is \na multiline string");
```



You cannot add or subtract string values like you'd do with numbers. However, the `+` operator has a special meaning when applied to two string values. It will join the two chains together, and this operation is called a concatenation. For example, `"Hel" + "lo"` produces the result `"Hello"`.

```
console.log("Hel" + "lo");
```

