**[Number](https://javascript.info/types" \l "number)**

The number type represents both integer and floating point numbers.

There are many operations for numbers, e.g. multiplication \*, division /, addition +, subtraction -, and so on.

let n = 123;

n = 12.345;

## [String](https://javascript.info/types" \l "string)

A string in JavaScript must be surrounded by quotes.

let str = "Hello";

let str2 = 'Single quotes are ok too';

Double and single quotes are “simple” quotes. There’s practically no difference between them in JavaScript.

## [Boolean (logical type)](https://javascript.info/types" \l "boolean-logical-type)

The boolean type has only two values: true and false.

This type is commonly used to store yes/no values: true means “yes, correct”, and false means “no, incorrect”.

For instance:

let nameFieldChecked = true; // yes, name field is checked

let ageFieldChecked = false;

## [The “null” value](https://javascript.info/types" \l "the-null-value)

The special null value does not belong to any of the types described above.

It forms a separate type of its own which contains only the null value:

let age = null;

In JavaScript, null is not a “reference to a non-existing object” or a “null pointer” like in some other languages.

It’s just a special value which represents “nothing”, “empty” or “value unknown”.

## [The “undefined” value](https://javascript.info/types" \l "the-undefined-value)

The special value undefined also stands apart. It makes a type of its own, just like null.

The meaning of undefined is “value is not assigned”.

If a variable is declared, but not assigned, then its value is undefined:

let age;

alert(age); // shows "undefined"