

Primitive

In [JavaScript](#), a **primitive** (primitive value, primitive data type) is data that is not an [object](#) and has no [methods](#) or [properties](#). There are 7 primitive data types:

- [string](#)
- [number](#)
- [bigint](#)
- [boolean](#)
- [undefined](#)
- [symbol](#)
- [null](#)

Most of the time, a primitive value is represented directly at the lowest level of the language implementation.

All primitives are *immutable*; that is, they cannot be altered. It is important not to confuse a primitive itself with a variable assigned a primitive value. The variable may be reassigned to a new value, but the existing value can not be changed in the ways that objects, arrays, and functions can be altered. The language does not offer utilities to mutate primitive values.

Primitives have no methods but still behave as if they do. When properties are accessed on primitives, JavaScript *auto-boxes* the value into a wrapper object and accesses the property on that object instead. For example, `"foo".includes("f")` implicitly creates a [String](#) wrapper object and calls `String.prototype.includes()` on that object. This auto-boxing behavior is not observable in JavaScript code but is a good mental model of various behaviors — for example, why "mutating" primitives does not work (because `str.foo = 1` is not assigning to the property `foo` of `str` itself, but to an ephemeral wrapper object).

See also

- [JavaScript data types](#)
- [Primitive data type](#) [↗] (Wikipedia)
- [Glossary](#)
 - [JavaScript](#)
 - [string](#)
 - [number](#)
 - [bigint](#)
 - [boolean](#)
 - [null](#)
 - [undefined](#)
 - [symbol](#)

This page was last modified on Jun 8, 2023 by [MDN contributors](#).