

type conversion in js

Type conversion in JavaScript is the process of changing a value from one data type to another, and it can happen either **implicitly** (automatically by JavaScript, also called *type coercion*) or **explicitly** (manually by the programmer using built-in functions)^{[1] [2] [3]}.

Implicit Type Conversion (Type Coercion)

JavaScript often converts types automatically based on the operation:

- **String Concatenation:** Adding a number to a string converts the number to a string (5 + "5" becomes "55").
- **Boolean Arithmetic:** Booleans in arithmetic become 1 for true and 0 for false (true + 1 is 2).
- **Equality Comparison (==):** The loose equality operator converts operands to the same type before comparison (5 == "5" is true).
- **Arithmetic with Strings:** Using operators like -, *, / on numeric strings converts them to numbers ("4" - "1" is 3).

Explicit Type Conversion

You can manually convert values to different types with built-in global functions:

- **To Number:** Number(value), parseInt(value), parseFloat(value)

```
let n = Number("123");    // 123
let f = parseFloat("12.3"); // 12.3
```

- **To String:** String(value), value.toString(), or by adding an empty string (value + "")

```
let s = String(123);      // "123"
```

- **To Boolean:** Boolean(value)

```
Boolean("hello") // true
Boolean("")      // false
```

Falsy values: undefined, null, 0, NaN, "" all become false; everything else becomes true^{[2] [3] [4]}.

Examples

```
let a = "5" + 2;      // "52" (implicit: 2 to string)
let b = "5" - 2;      // 3   (implicit: "5" to number)
let c = Boolean(0);   // false (explicit: 0 is falsy)
let d = Number("abc"); // NaN (cannot be converted)
```

Key Points

- Implicit conversion happens automatically, often during operations or comparisons.
- Explicit conversion uses `Number()`, `String()`, `Boolean()`, etc., for manual control.
- Understanding type conversion is critical to avoid unexpected results in code [\[2\]](#) [\[3\]](#) [\[5\]](#).

JavaScript's flexibility in type conversion makes it powerful but can lead to subtle bugs if not clearly understood.

✴✴

1. <https://www.geeksforgeeks.org/javascript/javascript-type-conversion/>
2. <https://www.programiz.com/javascript/type-conversion>
3. https://www.tutorialspoint.com/javascript/javascript_type_conversions.htm
4. <https://www.scaler.com/topics/javascript/type-conversion-in-javascript/>
5. <https://webreference.com/javascript/basics/type-conversion/>