

Equality And Type Coercion

Web Development

Equality And Type Coercion

En JavaScript existen dos formas principales de comparar valores:

Igualdad flexible (==)

Igualdad estricta (===)

La diferencia principal está en cómo manejan los tipos de datos antes de compararlos.

Loose Equality (==) — Igualdad flexible

Este operador compara valores sin importar su tipo.

Si los tipos son diferentes, JavaScript intentará convertirlos (coerción de tipo) antes de comparar.

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Equality>

Loose Equality (==) — Igualdad flexible

Reglas principales:

- Si ambos valores son null o undefined, devuelve true.
- Convierte los booleanos a números ($\text{true} \rightarrow 1$, $\text{false} \rightarrow 0$).
- Si se compara un número con un string, convierte el string a número.
- Si se compara un objeto con un string, el objeto se convierte usando `toString()` o `valueOf()`.
- Si los tipos son iguales, se comporta igual que la igualdad estricta (`===`).

Loose Equality (==) — Igualdad flexible

`5 == "5" // true` → convierte "5" a número

`0 == false // true` → convierte false a 0

`null == undefined // true`

Consejo: Aunque `==` puede parecer útil, a veces produce resultados inesperados.

Por eso se recomienda usar `===` para evitar confusiones.

Strict Equality (===) — Igualdad estricta

Este operador compara tanto el valor como el tipo.

No realiza ninguna conversión de tipo, por lo que solo devuelve true si ambos son idénticos.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Strict_equality

Strict Equality (===) — Igualdad estricta

Reglas principales:

- Si cualquiera de los valores es NaN, devuelve false.
- Si los tipos son diferentes, devuelve false.
- Si ambos son null o ambos undefined, devuelve true.
- Si ambos son objetos, devuelve true solo si son el mismo objeto en memoria.
- Si son del mismo tipo primitivo y tienen el mismo valor, devuelve true.

Strict Equality (===) — Igualdad estricta

`5 === "5" // false → tipos distintos`

`5 === 5 // true → mismo tipo y valor`

`true === 1 // false → booleano ≠ número`

`null === undefined // false`