



Booleans JS

El tipo de dato primitivo **boolean** se utiliza para asignar un valor de tipo de **dato lógico** a una variable. Este dato solo puede tener como valor true o false.

```
const coding = true;  
const sleeping = false;
```

Métodos

Los tipos de dato **boolean** no tienen muchos métodos, pero uno muy útil que nos permite devolver como una cadena el valor de una constante, por ejemplo, de tipo boolean es **toString**.

```
const winner = true;  
const winnerText = winner.toString();  
console.log(winner);  
// Retorna "true";
```

Tratando los booleans

Los booleanos son un tipo de dato en JavaScript que sólo pueden tener dos valores: **true** (verdadero) o **false** (falso). Los booleanos se utilizan a menudo para representar el resultado de una comparación o de una evaluación de una expresión.

La mayoría de las expresiones en JavaScript devuelven un valor booleano, ya sea explícitamente o implícitamente. Por ejemplo, la expresión **3 > 2** devuelve **true**,

mientras que la expresión `3 < 2` devuelve `false`.

Los booleanos también se pueden utilizar en las estructuras de control de flujo, como los `if` y los `while`. Por ejemplo:

```
let number = 5;

if (number > 0) {
  console.log('El número es positivo');
}
```

En este caso, la expresión `number > 0` devuelve `true`, por lo que se ejecuta el código dentro del bloque `if`.

Los booleanos también se pueden utilizar en operaciones lógicas, como `and`, `or` y `not`. Por ejemplo:

```
let a = true;
let b = false;

console.log(a && b); // false
console.log(a || b); // true
console.log(!a); // false
```

En este caso, la expresión `a && b` devuelve `false`, ya que ambos operandos son booleanos y uno de ellos es `false`. La expresión `a || b` devuelve `true`, ya que al menos uno de los operandos es `true`. Por último, la expresión `!a` devuelve `false`, ya que el operador `!` invierte el valor booleano de su operando.

Espero que esto te haya ayudado a entender cómo funcionan los booleanos en JavaScript. Si tienes alguna duda o necesitas más información, no dudes en preguntar.

