

Operadores en TS

Antonio Espín Herranz

Operadores

1. Operadores Aritméticos

Se usan para realizar operaciones matemáticas básicas:

- `+` (suma): `a + b`
- `-` (resta): `a - b`
- `*` (multiplicación): `a * b`
- `/` (división): `a / b`
- `%` (módulo o resto): `a % b`
- `**` (exponenciación): `a ** b`

Operadores

2. Operadores de Asignación

Se usan para asignar valores a variables:

- `=` (asignación): `a = b`
- `+=` (suma y asigna): `a += b` (equivale a `a = a + b`)
- `-=` (resta y asigna): `a -= b`
- `*=` (multiplica y asigna): `a *= b`
- `/=` (divide y asigna): `a /= b`
- `%=` (módulo y asigna): `a %= b`

Operadores

3. Operadores de Comparación

Se usan para comparar valores y devuelven un valor booleano:

- `==` (igualdad débil): `a == b`
- `===` (igualdad estricta): `a === b` (compara valor y tipo)
- `!=` (diferente débil): `a != b`
- `!==` (diferente estricto): `a !== b`
- `<` (menor que): `a < b`
- `>` (mayor que): `a > b`
- `<=` (menor o igual que): `a <= b`
- `>=` (mayor o igual que): `a >= b`

Operadores

4. Operadores Lógicos

Se usan para trabajar con valores booleanos:

- `&&` (AND): `a && b` (verdadero si ambos son verdaderos)
- `||` (OR): `a || b` (verdadero si uno de los dos es verdadero)
- `!` (NOT): `!a` (invierte el valor booleano)

Operadores

5. Operadores Bit a Bit

Realizan operaciones a nivel de bits:

- `&` (AND bit a bit)
- `|` (OR bit a bit)
- `^` (XOR bit a bit)
- `~` (NOT bit a bit)
- `<<` (desplazamiento a la izquierda)
- `>>` (desplazamiento a la derecha)
- `>>>` (desplazamiento a la derecha sin signo)

Operadores

6. Operadores de Incremento y Decremento

Se usan para aumentar o disminuir el valor de una variable:

- `++` (incremento): `a++` o `++a`
- `--` (decremento): `a--` o `--a`

Operadores

7. Operadores Ternarios

- `(condición ? valor_si_verdadero : valor_si_falso)`

Operadores

8. Operadores de Tipo

- `typeof` (devuelve el tipo de una variable): `typeof a`
- `instanceof` (verifica si un objeto es una instancia de una clase): `a instanceof Clase`

Operadores

9. Operadores de Nullish Coalescing (ES2020)

- `??` (nullish): Devuelve el operando de la derecha si el de la izquierda es `null` o `undefined`. Ejemplo:

Typescript

```
let valor = null ?? "Valor por defecto"; // "Valor por defecto"
```

Operadores

9. Operadores de Nullish Coalescing (ES2020)

- `??` (nullish): Devuelve el operando de la derecha si el de la izquierda es `null` o `undefined`. Ejemplo:

Typescript

```
let valor = null ?? "Valor por defecto"; // "Valor por defecto"
```

Operadores

10. Operadores de Encadenamiento Opcional

- `?.` (encadenamiento): Accede a una propiedad solo si no es `null` o `undefined`.

Ejemplo:

TypeScript

```
let usuario = { nombre: "Antonio" };  
console.log(usuario?.nombre); // "Antonio"  
console.log(usuario?.edad);   // undefined
```

Operadores

11. Operadores de Desestructuración (ES6)

Aunque no son operadores en el sentido clásico, se usan para trabajar con objetos y arrays:

- **Array:** `[a, b] = [1, 2]`
- **Objeto:** `{ x, y } = { x: 10, y: 20 }`