

OPERADORES *avanzados y ternarios*

Operadores ternarios

El operador ternario es una alternativa al condicional if/else de una forma mucho más compacta y breve, que en muchos casos resulta más legible. Sin embargo, hay que tener cuidado, porque su sobreutilización puede ser contraproducente y producir un código más difícil de leer.

La sintaxis de un operador ternario es la siguiente:

```
condicion ? valor verdadero : valor falso;
```

Para entenderlo bien, vamos a escribir el ejemplo utilizando este operador ternario. Primero, repasemos un ejemplo utilizando estructuras if/else:

```
let nota = 7;
console.log("He realizado mi examen. Mi resultado es el siguiente:");

if (nota < 5) {
  // Accion A: nota es menor que 5
  calificacion = "suspendido";
} else {
  // Accion B: Cualquier otro caso diferente a A (nota es mayor o igual que 5)
  calificacion = "aprobado";
}

console.log("Estoy", calificacion);
```

Ahora, vamos a reescribirlo utilizando un operador ternario:

```
let nota = 7;
console.log("He realizado mi examen. Mi resultado es el siguiente:");

// Operador ternario: (condicion ? verdadero : falso)
let calificacion = nota < 5 ? "suspendido" : "aprobado";

console.log("Estoy", calificacion);
```

Repasemos el ejemplo:

- Observa que guardamos en calificacion el resultado del operador ternario.
- La condicion es nota < 5, se escribe al principio, previo al ?.
- Si la condicion es cierta, el ternario devuelve "suspendido".
- Si la condicion es falsa, el ternario devuelve "aprobado".

OPERADORES *avanzados y ternarios*

Otro ejemplo:

```
condition ? expressionIfTrue : expressionIfFalse;
```

En esta sintaxis, condition es una expresión que se evalúa como un valor booleano, ya sea verdadero o falso.

Si condition es verdadera, se ejecuta la primera expresión (expressionIfTrue). Si es falso, se ejecuta la segunda expresión (expressionIfFalse).

A continuación, se muestra la sintaxis del operador ternario utilizado en una expresión:

```
let variableName = condition ? expressionIfTrue : expressionIfFalse;
```

En esta sintaxis, si condition es verdadera, variableName tomará el resultado de la primera expresión (expressionIfTrue) o expresiónIfFalse.

Ejemplos de operadores ternarios de JavaScript

Tomemos algunos ejemplos del uso del operador ternario.

1) Usar el operador ternario de JavaScript para realizar múltiples declaraciones

El siguiente ejemplo usa el operador ternario para realizar múltiples operaciones, donde cada operación está separada por una coma. Por ejemplo:

```
let authenticated = true;
let nextURL = authenticated
  ? {alert('You will redirect to admin area'), '/admin'}
  : {alert('Access denied'), '/403'};

// redirect to nextURL here
console.log(nextURL); // '/admin'
```

En este ejemplo, el valor devuelto por el operador ternario es el último valor de la lista separada por comas.

2) Ejemplo de operador ternario simplificado

Veamos el siguiente ejemplo:

```
let locked = 1;
let canChange = locked !== 1 ? true : false;
```