

Módulo Javascript

Operadores

Objetivos: Conocer y entender los distintos operadores de Javascript

Operadores Aritméticos

Los operadores aritméticos realizan operaciones matemáticas básicas en valores numéricos. Los operadores aritméticos en JavaScript son los siguientes:

- + : Suma
- - : Resta
- * : Multiplicación
- / : División
- % : Módulo (el resto de la división)

Ejemplo de uso

```
let a = 5;
let b = 2;
console.log(a + b); // Imprime 7
console.log(a - b); // Imprime 3
console.log(a * b); // Imprime 10
console.log(a / b); // Imprime 2.5
console.log(a % b); // Imprime 1
```

Operadores de Asignación

Los operadores de asignación se utilizan para asignar un valor a una variable. El operador de asignación en JavaScript es el signo igual (=).

```
let a = 5;
a = 10;
console.log(a); // Imprime 10
```

También hay operadores de asignación compuestos que combinan una operación aritmética y una asignación. Los operadores de asignación compuestos en JavaScript son los siguientes:

- += : Suma y asigna
- -= : Resta y asigna
- *= : Multiplica y asigna
- /= : Divide y asigna
- %= : Obtiene el módulo y asigna

Ejemplo de uso:

```
let a = 5;
a += 10; // a = a + 10
console.log(a); // Imprime 15

let b = 10;
b /= 2; // b = b / 2
console.log(b); // Imprime 5
```

Operadores de Incremento y Decremento

Los operadores de incremento y decremento se utilizan para aumentar o disminuir el valor de una variable en uno. Los operadores de incremento en JavaScript son los siguientes:

- ++ : Incrementa en uno
- -- : Decrementa en uno

Los operadores de incremento y decremento pueden ser usados tanto antes como después de la variable.

Ejemplo de uso

```
let a = 5;
console.log(++a); // Imprime 6
console.log(a); // Imprime 6

let b = 10;
console.log(b--); // Imprime 10
console.log(b); // Imprime 9
```

Operadores de Comparación

Los operadores de comparación se utilizan para comparar dos valores y devuelven un valor booleano (true o false). Los operadores de comparación en JavaScript son los siguientes:

- == : Igual a
- === : Estrictamente igual a (mismo valor y tipo de dato)
- != : Diferente a
- !== : Estrictamente diferente a
- > : Mayor que
- >= : Mayor o igual que
- < : Menor que
- <= : Menor o igual que

Ejemplo de uso

```
let a = 5;
let b = 10;
console.log(a == b); // Imprime false
console.log(a != b); // Imprime true
console.log(a < b); // Imprime true
console.log(a >= b); // Imprime false
```

Operadores Lógicos

Los operadores lógicos se utilizan para combinar valores booleanos y devuelven un valor booleano. Los operadores lógicos en JavaScript son los siguientes:

- **&&** : AND (y)
- **||** : OR (o)
- **!** : NOT (no)

Ejemplo de uso

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

Operador	Descripción	Ejemplo
&&	AND (y) - devuelve true si ambos valores son true, de lo contrario devuelve false.	true && false // false
	OR (o) - devuelve true si al menos uno de los valores es true, de lo	true false // true

	contrario devuelve false.	
!	NOT (no) - devuelve el valor contrario al valor booleano proporcionado.	!true // false

Es importante recordar que los operadores lógicos solo se pueden aplicar a valores booleanos. Si se utiliza un valor no booleano (como un número o una cadena), JavaScript tratará de convertirlo a un valor booleano antes de aplicar el operador lógico.

Operador Ternario

El operador ternario es una forma corta de escribir una declaración if...else en una sola línea. El operador ternario en JavaScript es el siguiente:

- `?:` El signo de interrogación seguido de dos puntos

Ejemplo de uso

```
let edad = 18;
let mensaje = edad >= 18 ? "Eres mayor de edad" : "Eres menor de edad";
console.log(mensaje); // Imprime "Eres mayor de edad"
```

En el ejemplo anterior, si la variable "edad" es mayor o igual a 18, el valor de la variable "mensaje" será "Eres mayor de edad", de lo contrario, el valor de la variable "mensaje" será "Eres menor de edad".