

Introducción a la programación en JavaScript

Módulo 1 – Operadores

Operadores

Operadores: Introducción

Como en su momento habíamos establecido qué poco podíamos hacer en **JS** si no contamos con **variables**, los **operadores nos brindan herramientas** para generar que nuestra aplicación o programa cumpla su fin.

Los operadores permiten trabajar con variables, realizar operaciones matemáticas y compararlas .

Nos permiten **avanzar en nuestro programa y cumplir el fin** que nos hayamos propuesto.

Tenemos diferentes **tipos de operadores** para trabajar en **Javascript**, ¿estás preparado para trabajar?

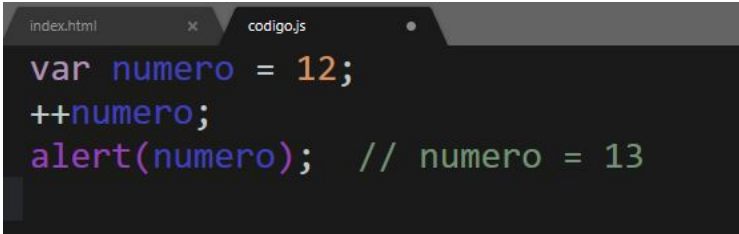


Asignación, incremento y decremento

Este tipo de operador, es simplemente aquel se utiliza para guardar un valor en una variable. Ya lo hemos utilizado previamente, pero veamos un ejemplo.

También tenemos el **operador de incremento** se indica mediante el prefijo **++**, incrementa la variable **en una unidad**, el operador de **decremento**, por el contrario, la **decrementa en una unidad**.

```
var iva = 21;
```

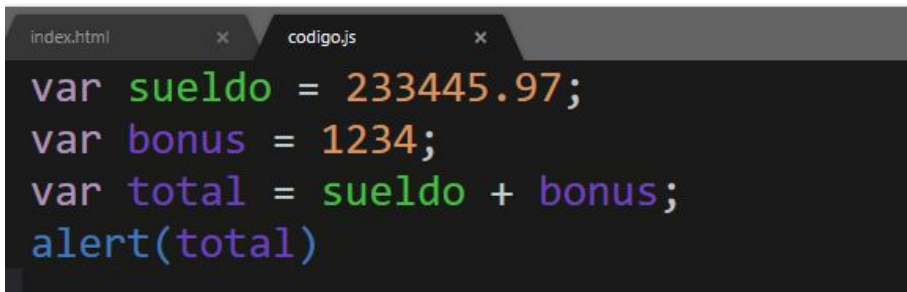


The screenshot shows a code editor with two tabs: 'index.html' and 'codigo.js'. The 'codigo.js' tab is active and contains the following JavaScript code: `var numero = 12;`, `++numero;`, and `alert(numero);` followed by a comment `// numero = 13`. The code is syntax-highlighted with colors: 'var' is blue, 'numero' is blue, '=' is orange, '12' is orange, '++' is blue, 'numero' is blue, 'alert' is purple, '(numero)' is purple, and the comment is green.

```
var numero = 12;  
++numero;  
alert(numero); // numero = 13
```

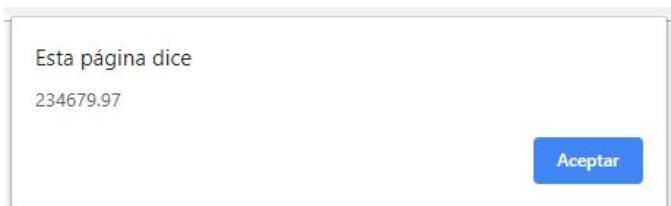
Operadores matemáticos

JavaScript permite hacer operaciones matemáticas con los operadores ya conocidos por todos como: **suma (+), resta (-), multiplicación (*) y división (/),**



```
index.html x  codigo.js x
var sueldo = 233445.97;
var bonus = 1234;
var total = sueldo + bonus;
alert(total)
```

El resultado será el siguiente.



Operadores matemáticos

Por otro lado, el operador matemático puede combinarse con el de asignación para lograr el siguiente resultado más rápido y conciso:

```
index.html x código.js x
var numero1 = 9;
numero1 += 4; // numero1 = numero1 + 4
numero1 -= 1; // numero1 = numero1 - 1
numero1 *= 2; // numero1 = numero1 * 2
numero1 /= 6; // numero1 = numero1 / 6
numero1 %= 5; // numero1 = numero1 % 5
```

También con el (+) podemos concatenar texto para obtener resultados accesibles y entendibles para el usuario, por ejemplo:

```
var sueldo = 233445.97;
var bonus = 1234;
var total = sueldo + bonus;
alert("El total del sueldo y el bonus es" + total)
```

Operadores matemáticos

Es decir podemos concatenar información para lograr mejores resultados para el usuario, también incluyendo etiquetas de **HTML**:

```
index.html x código.js x
var sueldo = 233445.97;
var bonus = 1234;
var total = sueldo + bonus;
document.write("<p>El total del sueldo y el <strong>bonus</strong> es " + total + "</p>")
```

La última línea fue reemplazada por **document.write()**, ya que las ventanas de alertas no soportan formato, es decir que una etiqueta en una ventana de alerta se vería de la siguiente forma:

```
index.html x código.js x
alert('<p> Esta etiqueta, se ve de verdad acá </p>')
```

Operadores matemáticos

El resultado del slide anterior será el siguiente,

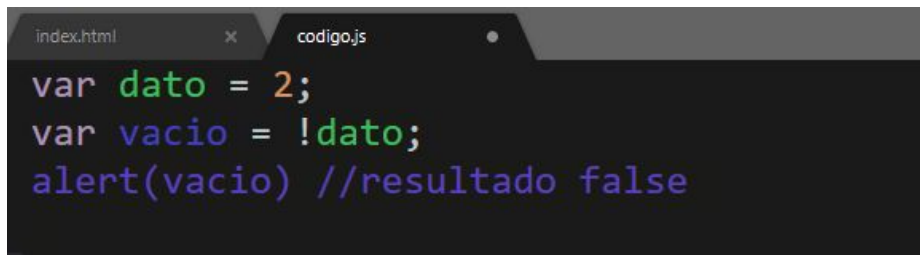


Un detalle interesante, es que seguramente llamó la atención el **operador de módulo (%)**, ya que **no es utilizado en matemática**. Este operador se utiliza para calcular **el resto de la división entre 2 valores enteros**. Tiene la misma precedencia que el producto y el cociente.

Operadores de negación

Previamente habíamos visto qué existían **variables booleanas (true/false)**, estos operadores siempre dan como resultado que algo sea **verdadero o falso**.

El de negación se utiliza para obtener el contrario o la negación de la variable.



```
index.html x  codigo.js
var dato = 2;
var vacio = !dato;
alert(vacio) //resultado false
```

Operadores and, or

La **operación and** obtiene un resultado donde ambos valores **deben ser verdaderos**:

```
index.html x código.js
var valorA = true;
var valorB = false;
resultado = valor1 && valor2; // resultado = false

valorA = true;
valorB = true;
resultado = valor1 && valor2; // resultado = true
```

En el caso de **or**, **los dos no pueden ser verdaderos**, ya que la consigna es que uno de ellos lo sea para obtener un **resultado true**.

```
index.html x código.js x
var valorA = true;
var valorB = false;
resultado = valor1 || valor2; // resultado = true

valorA = false;
valorB = false;
resultado = valor1 || valor2; // resultado = false
```

Operadores relacionales

Los operadores relacionales son iguales a los que usamos en matemáticas, nos permite realizar programas complejos, y **comparar datos**, el resultado será también al igual **que los operadores lógicos true/false**.

```
index.html x código.js
var numero1 = 1;
var numero2 = 20;
resultado = numero1 > numero2; // resultado = false, mayor que
resultado = numero1 < numero2; // resultado = true, menor que

numero1 = 10;
numero2 = 10;
resultado = numero1 >= numero2; // resultado = true, mayor o igual
resultado = numero1 <= numero2; // resultado = true, menor o igual
resultado = numero1 == numero2; // resultado = true, igual en valor
resultado = numero1 != numero2; // resultado = false, no es igual
```

Revisión

- Repase qué es un operador.
- Trabaje con variables y operadores en sus **diferentes tipos**.
- Implemente operadores y muestre los datos en una ventana.
- Muestre datos producto de operadores a través de **document.write()**
- Aplique todas las propiedades en el **proyecto integrador**.
- Realice las preguntas necesarias al/la docente antes de continuar.



¡Muchas gracias!

¡Sigamos trabajando!