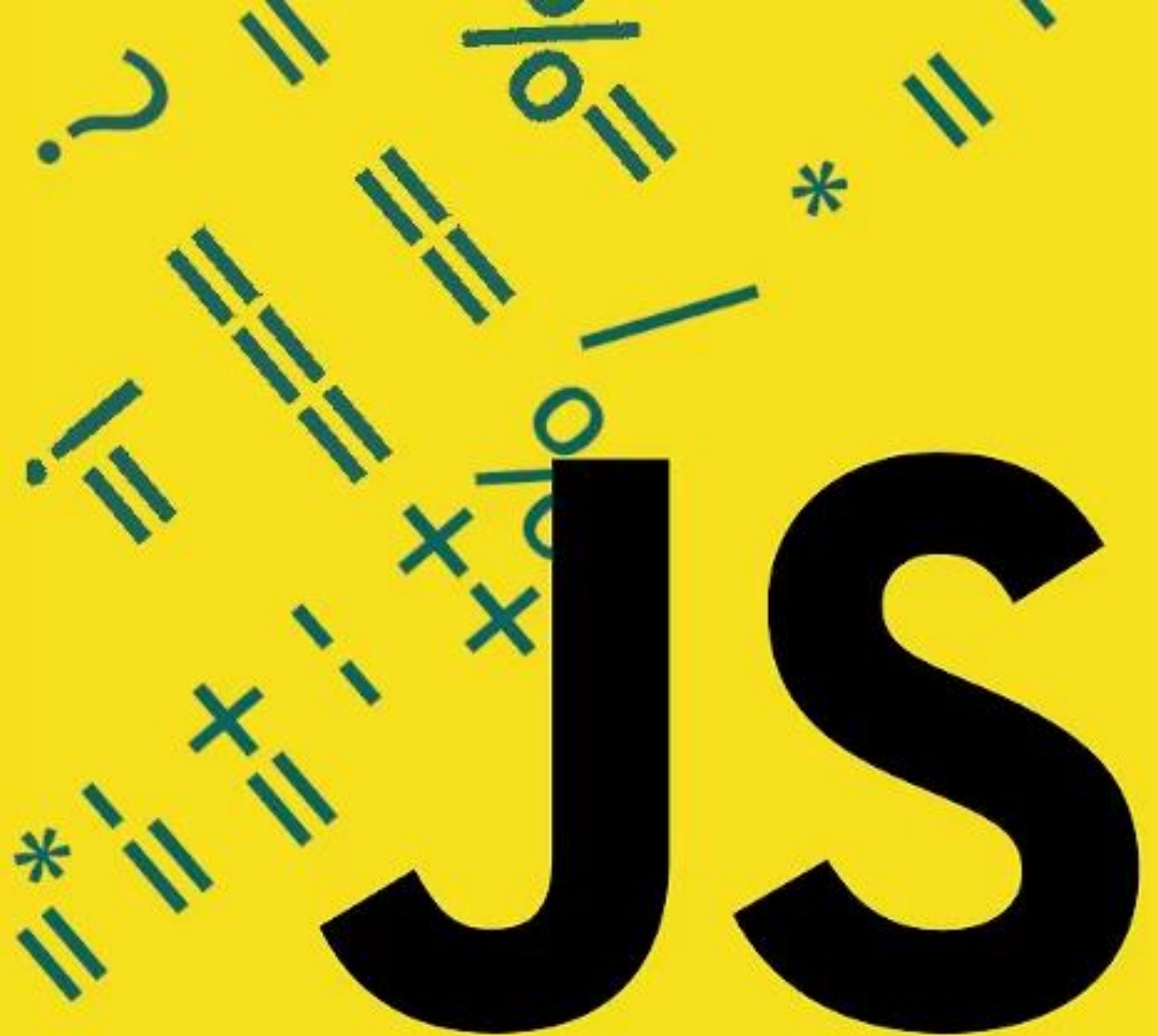


# Operadores aritméticos en JavaScript

## Integrantes

- Saenz Marcelo
- Nestares Fabricio
- Nuñez Patrick



Los operadores aritméticos en JavaScript se usan para realizar operaciones matemáticas básicas, como suma (+), resta (-), multiplicación (\*), división (/), módulo o resto (%) y exponenciación (\*\*). El operador de suma (+) también puede usarse para concatenar (unir) cadenas de texto en lugar de realizar una suma numérica.

# Operadores Aritméticos Comunes

- ◆ **Suma (+):** Suma dos números o concatena (une) dos cadenas de texto.
  - ◆  $5 + 3$  resulta en 8.
  - ◆ `"Hola" + " Mundo"` resulta en `"Hola Mundo"`.
- ◆ **Resta (-):** Resta el segundo número del primero.
  - ◆  $10 - 4$  resulta en 6.
- ◆ **Multiplicación (\*):** Multiplica dos números.
  - ◆  $2 * 6$  resulta en 12.

A large, bold, black 'JS' logo is positioned on the right side of the slide. The 'J' and 'S' are thick and rounded, set against a solid yellow background that occupies the right half of the slide.

# JS

- ◆ División (/): Divide el primer número entre el segundo.
  - ◆  $15 / 3$  resulta en 5.
- ◆ Módulo o Resto (%): Devuelve el resto de una división.
  - ◆  $10 \% 3$  resulta en 1 (porque 10 dividido entre 3 es 3 con un resto de 1).
- ◆ Exponenciación () \*\*: Eleva el primer número a la potencia del segundo.
  - ◆  $2 ** 3$  (2 elevado a la 3) resulta en 8.



## Ejemplos de Uso

Puedes utilizar  
estos operadores  
con variables  
para realizar  
operaciones:



- ◇ `let num1 = 10;`
- ◇ `let num2 = 5;`
- ◇ `let Suma = num1 + num2;` (Suma es 15)
- ◇ `let resta = num1 - num2;` (Resta es 5)
- ◇ `let multiplicación = num1 * num2;` (50)
- ◇ `let división = num1 / num2;` (2)
- ◇ `let resto = num1 % 3;` (Resto es 1)
  - $10 \div 3 = 3$
  - Con un residuo de 1
- ◇ `let potencia = 2 ** num2;` ( $2^{**}5=32$ )



# JS

## Operadores aritméticos unarios

Estos operadores trabajan  
con un solo operando:

◆ **Incremento (++)**: Aumenta un valor numérico en uno. Puede ser prefijo (++x) o posfijo (x++).

◆ `let x = 5; x++;` (x es ahora 6)

◆ `let y = 5; ++y;` (y es ahora 6)

◆ **Decremento (--)**: Disminuye un valor numérico en uno. Puede ser prefijo (--x) o posfijo (x--).

◆ `let x = 5; x--;` (x es ahora 4)

◆ `let y = 5; --y;` (y es ahora 4)



◆ **Negación unaria (-):** Convierte un operando en su valor negativo.

◆ `let x = 5; let y = -x;` (y es -5)

◆ **Más unario (+):** Intenta convertir su operando a un número.

◆ `let texto = "123"; let numero = +texto;` (numero es 123)

# JS

## Precedencia de operadores

Al combinar múltiples operadores en una misma expresión, es importante recordar el orden de precedencia para controlar cómo se evalúan. El orden habitual es similar al de las matemáticas tradicionales:

◆ **Paréntesis ()** (para agrupar y forzar la evaluación).

◆ **Exponenciación \*\***.

◆ **Multiplicación \*, división /, y módulo %**.

◆ **Suma + y resta -**



◆ `let resultado = 5 + 3 * 2;`

(La multiplicación se hace primero. El resultado es 11)

◆ `let resultadoConParentesis = (5 + 3) * 2;`

(El paréntesis fuerza la suma primero. El resultado es 16)