

# JavaScript Avanzado

## Sesión 2



Universidad  
Tecnológica  
del Perú

# ¿Tienen alguna consulta o duda sobre la clase previa?



# Logro de la sesión

Al finalizar la sesión el estudiante emplea operadores matemáticos, lógicos y de desplazamiento de bits en la solución de casos mediante el lenguaje JavaScript.

# ¿Quiénes deben participar en el diseño de las interfaces gráficas?



*¿Cuál es la  
importancia de  
los operadores  
en la  
programación?*



```
console.log( 0 == 0);           // true
console.log( "" == 0 );        // true
console.log( false == 0);      // true
console.log( undefined != 0);  // true
console.log( null == 0);       // false

console.log( '1' === 1);       // false
console.log( 2 === 2);         // true
console.log( 'abc' !== 123);   // true
```



Universidad  
Tecnológica  
del Perú

# Contenido

## Operadores en JavaScript

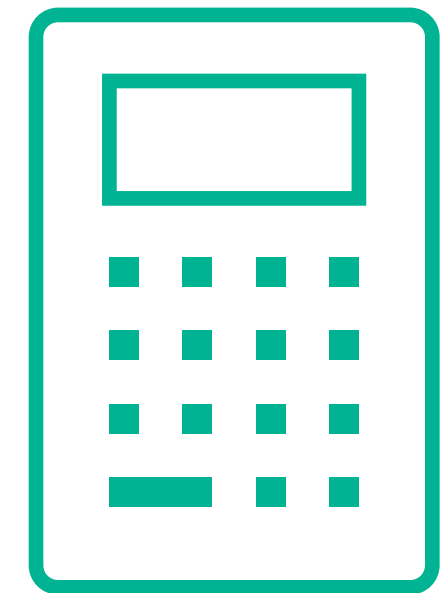
- Números en JavaScript
- Operadores matemáticos
- Operadores de desplazamiento de bits
- Operadores lógicos
- Operadores con asignación
- Sintaxis Spread
- Funciones, métodos y propiedades matemáticas.





# Operadores en JavaScript

- Son símbolos especiales que realizan operaciones específicas sobre uno o más valores (operandos).
- Son fundamentales para realizar cálculos, manipular datos y construir expresiones más complejas.



# Números en JavaScript

**IEEE 754:** JavaScript utiliza el estándar IEEE 754 para representar números de punto flotante.

- Esto significa que hay límites en la precisión y el rango de los números que se pueden representar.

**BigInt:** Un tipo de dato introducido en JavaScript para representar números enteros de precisión arbitraria.

- Es útil para trabajar con números extremadamente grandes que no caben en el rango de un número de 64 bits.



```
const biggestInt = Number.MAX_SAFE_INTEGER; // (2**53 - 1) => 9007199254740991
const smallestInt = Number.MIN_SAFE_INTEGER; // -(2**53 - 1) => -9007199254740991
```

```
Number("123"); // 123
Number("123") === 123; // true
Number("12.3"); // 12.3
Number("12.00"); // 12
Number("123e-1"); // 12.3
Number(""); // 0
Number(null); // 0
Number("0x11"); // 17
Number("0b11"); // 3
Number("0o11"); // 9
Number("foo"); // NaN
Number("100a"); // NaN
Number("-Infinity"); // -Infinity
```

```
const biggestNum = Number.MAX_VALUE;
const smallestNum = Number.MIN_VALUE;
const infiniteNum = Number.POSITIVE_INFINITY;
const negInfiniteNum = Number.NEGATIVE_INFINITY;
const notANum = Number.NaN;
```

# Operadores matemáticos

**Suma (+):** Suma dos números.

**Resta (-):** Resta un número de otro.

**Multiplicación (\*):** Multiplica dos números.

**División (/):** Divide un número por otro.

**Módulo (%):** Devuelve el resto de una división.

**Incremento (++):** Aumenta el valor de una variable en 1.

**Decremento (--):** Disminuye el valor de una variable en 1.

# Operadores de desplazamiento de bits

<<

- Desplazamiento a la izquierda.

>>

- Desplazamiento a la derecha.

>>>

- Desplazamiento a la derecha sin signo.

&

- AND bit a bit.

|

- OR bit a bit.

^

- XOR bit a bit.

~

- NOT bit a bit.

# Operadores Lógicos

&&

- Y lógico.

||

- O lógico.

!

- NO lógico.

```
console.log(!undefined); // true
console.log(!null); // true
console.log(!20); //false
console.log(!0); //true
console.log(!NaN); //true
console.log(!{}); // false
console.log(!''); //true
console.log(!'OK'); //false
console.log(!false); //true
console.log(!true); //false
```

# Operadores con Asignación

- **=:** Asignación simple.
- **\*=, +=, -=, /=, %=:** Combinan una operación aritmética con una asignación.
  - Ejemplo: **x += 5** es equivalente a **x = x + 5**.
- **<<=, >>=, >>>=, &=, |=, ^=:** Combinan una operación de desplazamiento de bits con una asignación.



# Sintaxis Spread (...)

Permite expandir un iterable (como un array) en una lista de argumentos individuales.

## Aplicaciones

- **Copiar arrays:** `let newArray = [...oldArray];`
- **Unir arrays:** `let combined = [...array1, ...array2];`
- **Pasar argumentos a funciones:** `function(a, b, ...rest){}`



# Funciones, métodos y propiedades matemáticas

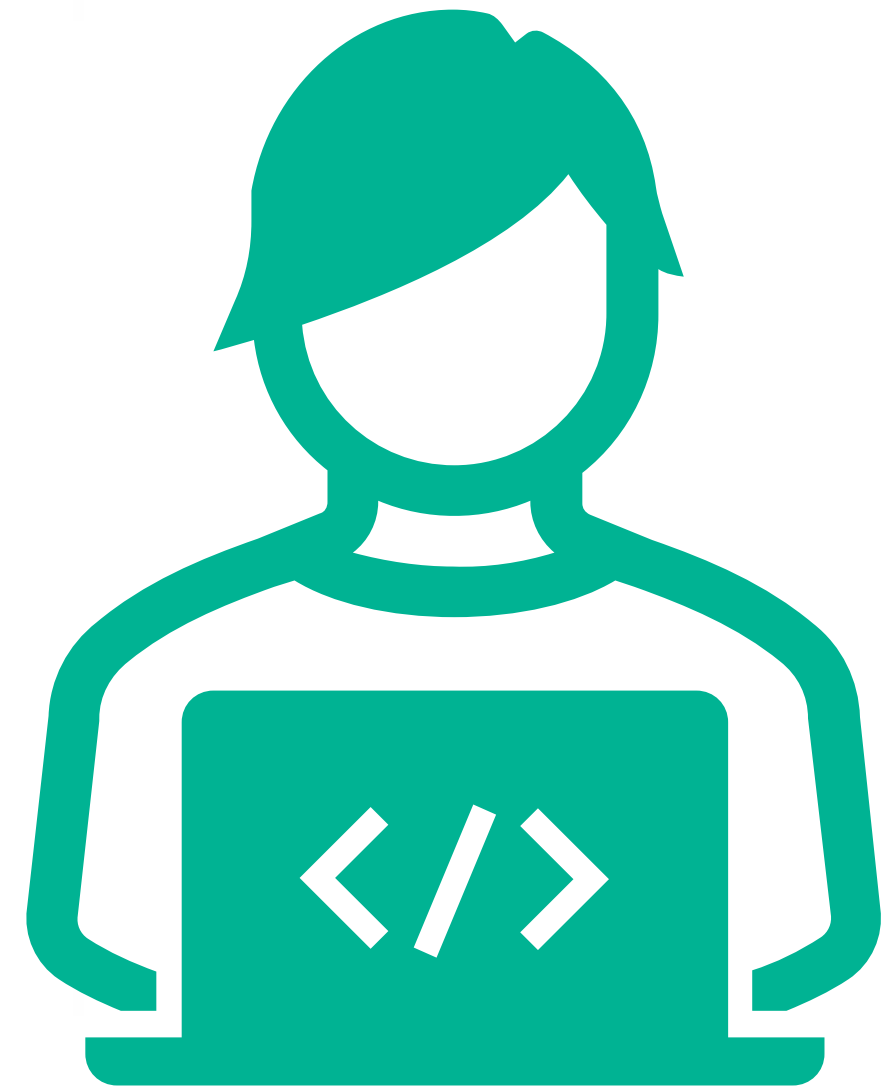
- **Math.PI:** Constante que representa el número pi.
- **Math.abs():** Devuelve el valor absoluto de un número.
- **Math.round():** Redondea un número al entero más cercano.
- **Math.ceil():** Redondea un número hacia arriba al entero más cercano.

# Funciones, métodos y propiedades matemáticas

- **Math.floor():** Redondea un número hacia abajo al entero más cercano.
- **Math.random():** Genera un número aleatorio entre 0 (inclusive) y 1 (exclusivo).
- **Math.pow():** Eleva un número a una potencia.
- **Math.sqrt():** Calcula la raíz cuadrada de un número.

# Programando

- Construir algoritmos empleando operadores de JavaScript.



# ¿Tienen alguna consulta o duda?



# Actividad



Resolver la actividad planteada en la plataforma.

# Cierre

## ¿Qué hemos aprendido hoy?



Elaboramos nuestras conclusiones sobre el tema tratado





**Universidad  
Tecnológica  
del Perú**