

UT2.- Manejo de la sintaxis del lenguaje

## 2.4.- Operadores de JavaScript con ejemplos de Ajedrez

Desarrollo Web Entorno Cliente (DWEC)

**JavaScript**





# Introducción JS Arithmetic JS Assignment JS Comparisons Js Logic

UT2. - Manejo de la sintaxis del lenguaje

## Operadores de JavaScript



# Introducción

UT2.- Manejo de la sintaxis del lenguaje

## 2.4.- Operadores de JavaScript con ejemplos de Ajedrez



## 2.4.- Operadores de JavaScript - Introducción

- ▶ Los operadores permiten hacer **operaciones matemáticas y lógicas** en JavaScript: asignar valores, comparar, combinar, etc.
- ▶ Algunos tipos de operadores:
  - Aritméticos
  - Asignación
  - Comparación
  - Lógicos



es el  
operador de  
asignación



es para  
sumar o  
concatenar



para  
comparar



etc.





## 2.4.- Operadores de JavaScript - Introducción

### ► Ejemplo sumar valor piezas



```
1 <p id="info">—</p>
2 <button id="btnSumar">Sumar piezas</button>
```

JavaScript



```
1 const info = document.getElementById("info");
2 const btnSumar = document.getElementById("btnSumar");
3
4 function sumarPiezas() {
5     const blancas = 8;
6     const negras = 8;
7     const total = blancas + negras;
8     info.textContent = `Total de piezas: ${total}`;
9 }
10
11 btnSumar.addEventListener("click", sumarPiezas);
```





## 2.4.- Operadores de JavaScript - Introducción

### ► Detalles a tener en cuenta:

- El operador + también se usa para concatenar cadenas cuando uno de los operandos es un string.
- Al añadir un número y una cadena, el resultado será una cadena, por ejemplo `"5" + 5 → "55"`.





¿Alguna pregunta?



UT2.- Manejo de la  
sintaxis del lenguaje

Operadores de  
JavaScript



# JS Arithmetic

UT2. - Manejo de la sintaxis del lenguaje

## Operadores de JavaScript





## 2.4.- Operadores de JavaScript - JS Arithmetic

### Operadores aritméticos

- Realizan cálculos matemáticos con números.

### Principales operadores:

- + suma
- - resta
- \* multiplicación
- / división
- % resto
- \*\* potencia

### En ajedrez

- Se usan para contar piezas, calcular valores o movimientos posibles.





## 2.4.- Operadores de JavaScript - JS Arithmetic -

- Ejemplo calcular el número de peones en juego



```
1 <p id="resultado">—</p>
2 <button id="btnResta">Restar pieza</button>
```



```
1 const resultado = document.getElementById("resultado");
2 const btnResta = document.getElementById("btnResta");
3
4 function restarPieza() {
5     const totalInicial = 8; // peones
6     const capturados = 1;
7     const restantes = totalInicial - capturados;
8
9     resultado.textContent = `Peones restantes:
10     ${restantes}`;
11 }
12 btnResta.addEventListener("click", restarPieza);
```





## 2.4.- Operadores de JavaScript - JS Arithmetic \* /

- Ejemplo calcular el numero casillas controladas por jugador



```
1 const movs = document.getElementById("movs");
2 const btnMovs = document.getElementById("btnMovs");
3
4 function calcularCasillas() {
5     const casillasTotales = 64;
6     const jugadores = 2;
7     const porJugador = casillasTotales / jugadores;
8
9     movs.textContent = `Cada jugador empieza controlando
10     ${porJugador} casillas`;
11 }
12 btnMovs.addEventListener("click", calcularCasillas);
```





## 2.4.- Operadores de JavaScript - JS Arithmetic %

- ▶ Devuelve el resto de una división.
- ▶ Muy útil para alternar turnos.
- ▶ Ejemplo calcular turno jugador(¿Quién le toca?)



JavaScript



```
1 function turnoActual(movimiento) {  
2   return movimiento % 2 === 0 ? "blancas" : "negras";  
3 }  
4  
5 console.log(turnoActual(1)); // negras  
6 console.log(turnoActual(2)); // blancas
```





## 2.4.- Operadores de JavaScript - JS Arithmetic \*\*

- ▶ Eleva un número a otro.
- ▶ En ajedrez, sirve para calcular posibilidades en un número limitado de jugadas.
- ▶ Ejemplo calcular jugadas posibles:



JavaScript



```
1 const movimientosPorTurno = 20; // jugada inicial
2 const jugadas = 3;
3
4 const posibles = movimientosPorTurno ** jugadas;
5 console.log("Posibles partidas en 3 jugadas: " + posibles);
```



## 2.4.- Operadores de JavaScript - JS Arithmetic - Resumen

- ▶ `+`, `-`, `*`, `/`, `%`, `**` → operadores clave.
- ▶ Aplicación práctica en ajedrez:
  - ▶ Contar piezas capturadas o restantes.
  - ▶ Repartir casillas entre jugadores.
  - ▶ Alternar turnos con `%`.
  - ▶ Calcular combinaciones posibles con `**`.
- ▶ Siempre separar la lógica JS del HTML
  - ▶ Facilita la lectura y el mantenimiento.



¿Alguna pregunta?



UT2.- Manejo de la  
sintaxis del lenguaje

Operadores de  
JavaScript



# JS Assignment

UT2. - Manejo de la sintaxis del lenguaje

## Operadores de JavaScript





## 2.4.- Operadores de JavaScript - JS Assignment =

### Operadores asignación

- El **operador de asignación básico** en JavaScript es **=**.
- Se usa para **dar valor a una variable**.

### Operadores combinados :

- **+=   -=   \*=   /=   %=   \*\*=**

### En ajedrez

- Muy útiles para **sumar puntos, restar piezas o acumular jugadas**.





## 2.4.- Operadores de JavaScript - JS Assignment =

### ► Ejemplo asignar dama



```
1 const piezaElemento = document.getElementById("pieza");
2 const btnAsignar = document.getElementById("btnAsignar");
3
4 function asignarDama() {
5     const pieza = "♚";
6     piezaElemento.textContent = `Se ha asignado la pieza: ${pieza}`;
7 }
8
9 btnAsignar.addEventListener("click", asignarDama);
```





## 2.4.- Operadores de JavaScript - JS Assignment +=

- ▶ Sirve para sumar y reasignar al mismo tiempo.
- ▶ En ajedrez: acumular puntos por piezas capturadas.



```
1 let puntos = 0;  
2 puntos += 9; // capturamos una dama  
3 puntos += 5; // capturamos una torre  
4  
5 console.log(`Puntos acumulados: ${puntos}`); // 14
```





## 2.4.- Operadores de JavaScript - JS Assignment -=

- ▶ Resta y reasigna el valor.
- ▶ En ajedrez: piezas restantes en el tablero.



```
1 let piezasRestantes = 16; // inicial
2 piezasRestantes -= 1; // se captura un peón
3
4 console.log("Piezas restantes: " + piezasRestantes); // 15
```





## 2.4.- Operadores de JavaScript - JS Assignment $\ast=$ , $/=$ , $\%=$ y $\ast\ast=$

- ▶  $\ast=$  multiplica y reasigna.
- ▶  $/=$  divide y reasigna.
- ▶  $\%=$  aplica el módulo y reasigna.
- ▶  $\ast\ast=$  eleva a la potencia y reasigna.



```
1 let movimientos = 1;
2
3 // duplicamos movimientos posibles
4 movimientos *= 2; // 2
5
6 // dividir entre dos jugadores
7 movimientos /= 2; // 1
8
9 // alternar turnos con módulo
10 let turno = 5;
11 turno %= 2; // 1 → negras
12
13 // crecer exponencialmente
14 let posibilidades = 2;
15 posibilidades **= 3; // 8
16
17 console.log(`Posibilidades: ${posibilidades}, Turno: ${turno}`);
```



## 2.4.- Operadores de JavaScript - JS Assignment - Resumen

- ▶ = asigna valores.
- ▶ Operadores combinados simplifican el código:
  - ▶  $x += y \rightarrow x = x + y$
  - ▶  $x -= y \rightarrow x = x - y$
  - ▶  $x *= y \rightarrow x = x * y$
  - ▶  $x /= y \rightarrow x = x / y$
  - ▶  $x \% = y \rightarrow x = x \% y$
  - ▶  $x ** = y \rightarrow x = x ** y$
- ▶ En ajedrez se aplican para:
  - ▶ Acumular puntos (+=)
  - ▶ Reducir piezas (-=)
  - ▶ Simular turnos (%=)
  - ▶ Calcular combinaciones (\*\*=)



¿Alguna pregunta?



UT2.- Manejo de la  
sintaxis del lenguaje

Operadores de  
JavaScript



# JS Comparisons

UT2. - Manejo de la sintaxis del lenguaje

## Operadores de JavaScript





## 2.4.- Operadores de JavaScript - JS Comparisons

### Operadores comparación

- Los operadores de comparación devuelven un valor booleano (true/false).
- Se usan para comprobar igualdad, desigualdad y orden.

### Principales operadores:

- == igualdad no estricta
- === igualdad estricta
- !=, !== desigualdad
- >, <, >=, <=

### En ajedrez

- Verificar si es turno de un jugador.
- Comparar valores de piezas.
- Validar posiciones en el tablero.





## 2.4.- Operadores de JavaScript - JS Comparisons ===

- Ejemplo Mostrar mensaje por consola con el turno.



```
1 const turno = "blancas";  
2  
3 if (turno === "blancas") {  
4     console.log("Es el turno de las blancas");  
5 } else {  
6     console.log("Es el turno de las negras");  
7 }
```





## 2.4.- Operadores de JavaScript - JS Comparisons !==

- Ejemplo Mostrar mensaje por consola si la pieza no es un peón.



```
1 const pieza = "♘"; // caballo
2
3 if (pieza !== "♟") {
4     console.log(`La pieza ${pieza} no es un peón`);
5 }
```





## 2.4.- Operadores de JavaScript - JS Comparisons > <

- Ejemplo Mostrar mensaje por consola comparando valor torre y caballo.



```
1 const torre = 5;  
2 const caballo = 3;  
3  
4 if (torre > caballo) {  
5   console.log("La torre vale más que el caballo");  
6 }
```





## 2.4.- Operadores de JavaScript - JS Comparisons >= <=

- Ejemplo Mostrar mensaje por consola comparando valor torre y caballo.



```
1 const movimientosHechos = 40;  
2 const limiteTablas = 50;  
3  
4 if (movimientosHechos >= limiteTablas) {  
5   console.log("Se pueden reclamar tablas por la regla de los 50 movimientos");  
6 }
```



## 2.4.- Operadores de JavaScript - JS Comparisons - Resumen

- ▶ === y !== son preferidos a == y != (más seguros).
- ▶ Usar comparaciones para:
  - ▶ Controlar turnos (¿blancas o negras?).
  - ▶ Comparar valores de piezas (mayor/menor).
  - ▶ Reglas de juego (ej. tablas a los 50 movimientos).
- ▶ Devuelven siempre true o false, ideales para condiciones en if o bucles.



¿Alguna pregunta?



UT2.- Manejo de la  
sintaxis del lenguaje

Operadores de  
JavaScript



# JS Logical

UT2.- Manejo de la sintaxis del lenguaje

## Operadores de JavaScript





## 2.4.- Operadores de JavaScript - JS Logical


### Operadores lógicos

- Permiten combinar o modificar expresiones booleanas.
- Se usan para comprobar igualdad, desigualdad y orden.

### Principales operadores:

- **&&** (AND) - devuelve true solo si ambas expresiones son verdaderas.
- **||** (OR) - devuelve true si al menos una de las expresiones es verdadera.
- **!** (NOT) - invierte el valor de una expresión booleana.
- **??** (operador nullish coalescing) -- devuelve el primer operando si no es null ni undefined, de lo contrario el segundo.





# Introducción JS Arithmetic JS Assignment JS Comparisons

UT2.- Manejo de la sintaxis del  
lenguaje

Operadores de JavaScript