


UT2.- Manejo de la sintaxis del lenguaje

2.5.- Condicionales de JavaScript con ejemplos de Ajedrez

Desarrollo Web Entorno Cliente (DWEC)





If Conditions JS switch JS Boolean JS Logical

UT2.- Manejo de la sintaxis del
lenguaje

2.5.- Condicionales de JavaScript



If Conditions

UT2.- Manejo de la sintaxis del lenguaje

2.5.- Condicionales de JavaScript



2.5.- Condicionales de JavaScript - JS if

Sentencia if

- Se usa para ejecutar código solo si una condición es verdadera.

Sintaxis básica

```
if (condición) {  
    // código si condición es true  
}
```

En ajedrez

- Si una jugada es legal.
- Si una pieza está protegida.
- Si es turno de un jugador.





2.5.- Condicionales de JavaScript - JS if

► Ejemplo if (turno):



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





2.5.- Condicionales de JavaScript - JS if ... else

- ▶ Con else ejecutamos código alternativo si la condición **no se cumple**.
- ▶ Ejemplo if..else (mensaje jaque/no jaque)



```
1 const jaque = true;  
2  
3 if (jaque) {  
4   console.log("El rey está en jaque, debes protegerlo");  
5 } else {  
6   console.log("El rey está seguro, puedes jugar libremente");  
7 }
```





2.5.- Condicionales de JavaScript - JS if ... else if ... else

- ▶ Permite comprobar múltiples condiciones.
- ▶ Ejemplo
if..else if..else (mensaje seguimiento partida)



```
1 const piezasCapturadas = 15;  
2  
3 if (piezasCapturadas === 0) {  
4   console.log("La partida acaba de empezar");  
5 } else if (piezasCapturadas < 10) {  
6   console.log("Estamos en el medio juego");  
7 } else {  
8   console.log("Entramos en el final de partida");  
9 }
```





2.5.- Condicionales de JavaScript - JS Nested if (if anidados)

- ▶ Se usan cuando hay que comprobar una condición dentro de otra condición.
- ▶ En ajedrez: comprobar si un jugador está en jaque y además si el jaque es mate.



```
1 const enJaque = true;  
2 const sinMovimientos = true;  
3  
4 if (enJaque) {  
5     if (sinMovimientos) {  
6         console.log("¡Jaque mate!");  
7     } else {  
8         console.log("El rey está en jaque, pero puede salvarse");  
9     }  
10 }
```





2.5.- Condicionales de JavaScript - Operador ternario ()? : ;

- ▶ Sintaxis corta para un if ... else.
- ▶ Ejemplo ajedrez (mostrar a quién tiene le toca):



```
1 const turno = "blancas";  
2  
3 const mensaje = (turno === "blancas")  
4   ? "Mueven las blancas"  
5   : "Mueven las negras";  
6  
7 console.log(mensaje);
```





2.5.- Condicionales de JavaScript - Operador lógicos en if: &&(AND) || OR

- ▶ Podemos combinar condiciones con && (AND), || (OR).
- ▶ Ejemplo ajedrez (información partida):



```
1 const turno = "blancas";  
2 const enJaque = true;  
3  
4 if (turno === "blancas" && enJaque) {  
5   console.log("Las blancas deben salir del jaque en este turno");  
6 }
```

- ▶ Otro ejemplo mas complejo:



```
1 const pieza = "♔"; // rey blanco  
2 const posicion = "e1";  
3 const enJaque = false;  
4  
5 if (pieza === "♔" && posicion === "e1" && !enJaque) {  
6   console.log("El rey blanco aún no se ha movido, puede enrocar");  
7 }
```



2.5.- Condicionales de JavaScript - Resumen parte 1

- ▶ if ejecuta código solo si la condición es true.
- ▶ else permite manejar el caso contrario.
- ▶ else if comprueba varias condiciones.
- ▶ Se pueden combinar con operadores lógicos.
- ▶ En ajedrez: verificar turnos, comprobar reglas, controlar estado del rey.



¿Alguna pregunta?



UT2.- Manejo de la
sintaxis del lenguaje

2.5.- Condicionales
de JavaScript



Aula Virtual: Actividad 03

UT2.- Manejo de la sintaxis del lenguaje

2.5.- Condicionales de JavaScript



JS switch

UT2.- Manejo de la sintaxis del lenguaje

2.5.- Condicionales de JavaScript



2.5.- Condicionales de JavaScript - JS switch

Sentencia switch

- Evalúa una expresión y ejecuta el bloque que coincida con un **case**.
- Evita cadenas largas de `if...else if....`

Se suele usar para:

- Menús de opciones.
- Valores fijos de una variable.
- Decidir entre diferentes piezas o jugadas en ajedrez.

Sintáxis básica





2.5.- Condicionales de JavaScript - JS switch

Sintaxis básica:

```
switch (expresión) {  
  case valor1:  
    // código si expresión === valor1  
    break;  
  case valor2:  
    // código si expresión === valor2  
    break;  
  default:  
    // código si no coincide ningún  
case  
}
```





2.5.- Condicionales de JavaScript - JS switch

► Ejemplo mensaje según pieza:

```
1 const pieza = "♘"; // caballo
2
3 switch (pieza) {
4   case "♙":
5     console.log("La dama se mueve en todas las
6       direcciones");
7     break;
8   case "♖":
9     console.log("La torre se mueve en líneas rectas");
10    break;
11   case "♘":
12     console.log("El caballo se mueve en forma de L");
13    break;
14   case "♜":
15     console.log("El peón se mueve hacia adelante");
16    break;
17   default:
18     console.log("Pieza desconocida");
19 }
```





2.5.- Condicionales de JavaScript - JS switch

- ▶ El bloque default se ejecuta cuando ningún case coincide.
- ▶ Es importante para manejar casos no previstos.



```
1 const pieza = "♘"; // rey
2
3 switch (pieza) {
4   case "♘":
5     console.log("Caballo detectado");
6     break;
7   default:
8     console.log("Pieza no reconocida en este ejemplo");
9 }
```





2.5.- Condicionales de JavaScript - JS switch

- Se pueden agrupar varios case que ejecuten el mismo código.



```
1  const pieza = "♔"; // rey blanco
2
3  switch (pieza) {
4      case "♔":
5      case "♚":
6          console.log("Es un rey (blanco o negro)");
7          break;
8      default:
9          console.log("Otra pieza");
10 }
```



2.5.- Condicionales de JavaScript - Resumen parte 2 - Buenas Prácticas

- ▶ Buenas prácticas con switch
 - ▶ Usar siempre break (salvo que quieras fall-through intencional).
 - ▶ Mantener los case organizados y legibles.
 - ▶ default debería cubrir situaciones inesperadas.
 - ▶ Ideal cuando hay que comparar muchos valores fijos.



¿Alguna pregunta?



UT2.- Manejo de la
sintaxis del lenguaje

2.5.- Condicionales
de JavaScript



Aula Virtual: Actividad 05

UT2.- Manejo de la sintaxis del lenguaje

2.5.- Condicionales de JavaScript



JS boolean

UT2.- Manejo de la sintaxis del lenguaje

2.5.- Condicionales de JavaScript



2.5.- Condicionales de JavaScript - JS boolean

¿Qué es un Boolean?

- Es un tipo primitivo que solo puede tener **dos valores**: true o false

Se suele usar para:

- Controlar condiciones (decisiones) en el programa

Importante

- true y false se escriben sin comillas, en minúscula.
- Todos los operadores de comparación devuelven un booleano (true o false): ==, !=, <, >, etc.





2.5.- Condicionales de JavaScript - JS boolean

Valores “truthy” y “falsey”

- En JS, no solo true y false pueden comportarse como booleanos: ciertos valores se consideran “falsy” (equivalentes a false si se evalúan en un contexto booleano).
 - Algunos ejemplos de valores falsy: 0, "" (cadena vacía), null, undefined, NaN.
- Valores con contenido (números distintos de cero, cadenas con texto, objetos) se consideran “truthy”.





2.5.- Condicionales de JavaScript - JS boolean

► Ejemplos



```
1 const valorDama = 9;
2 const valorTorre = 5;
3
4 const esDamaMayor = (valorDama > valorTorre); // true
5 console.log(`¿La dama vale más que la torre? ${esDamaMayor}`);
```



```
1 function hayPieza(posicion) {
2     // posición puede ser un string con símbolo o "" si vacío
3     return Boolean(posicion);
4 }
5
6 console.log(hayPieza("♠")); // true
7 console.log(hayPieza(""));  // false
```





2.5.- Condicionales de JavaScript - JS boolean

Función Boolean() y conversión explícita

- Podemos usar la función global Boolean() para convertir valores a su equivalente booleano.
- También .valueOf() retorna el valor primitivo de un objeto booleano (aunque no es común usar objetos Boolean).
- **Advertencia:** no deberías usar new Boolean(...), pues crea un objeto, no un valor primitivo. Comparaciones entre booleanos objeto y booleanos literales pueden dar resultados inesperados.

```
1 function tienePieza(objCasilla) {  
2   // objCasilla.pieza podría ser símbolo o null/undefined  
3   return Boolean(objCasilla.pieza);  
4 }
```





2.5.- Condicionales de JavaScript - JS boolean

Ejemplos de conversiones "truthy / falsy"

- `Boolean(0) → false`
- `Boolean("") → false`
- `Boolean(null) → false`
- `Boolean("♘") → true`
- `Boolean(3) → true`

```
1 const casilla = { pieza: "♘" };  
2 if (Boolean(casilla.pieza)) {  
3   console.log("Hay una pieza aquí");  
4 } else {  
5   console.log("La casilla está vacía");  
6 }
```



2.5.- Condicionales de JavaScript - Resumen parte 3 - Buenas Prácticas

- ▶ Buenas prácticas con Booleanos
 - ▶ Usa booleanos (true / false) para controlar lógica de flujo (if, loops, etc.).
 - ▶ Comprendiendo valores “falsy” te evitas errores al evaluar variables no definidas o vacías.
 - ▶ Preferible usar Boolean(...) para convertir explícitamente valores a booleano cuando no esté claro el tipo.
 - ▶ Evita usar objetos booleanos (new Boolean()): pueden complicar comparaciones.



¿Alguna pregunta?



UT2.- Manejo de la
sintaxis del lenguaje

2.5.- Condicionales
de JavaScript



JS logical

UT2.- Manejo de la sintaxis del lenguaje

2.5.- Condicionales de JavaScript



2.5.- Condicionales de JavaScript - JS logical

Operadores lógicos

- permiten **combinar expresiones booleanas** (true / false).

Se suele usar:

- después de comparaciones para construir condiciones más complejas.

Operadores principales:

- && (AND) – verdadero si **ambas** expresiones son verdaderas
- || (OR) – verdadero si **al menos una** expresión es verdadera
- ! (NOT) – invierte el valor booleano (true → false, false → true)
- ?? (Nullish coalescing) – devuelve el primer operando si no es null/undefined, de lo contrario el segundo





2.5.- Condicionales de JavaScript

JS Logical && || !

► Ejemplos



JavaScript



```
1 const enJaque = true;
2 const puedeMover = true;
3
4 if (enJaque && puedeMover) {
5   console.log("El rey está amenazado pero tiene movimientos legales para escapar");
6 }
```

JavaScript



```
1 const esFinal = false;
2 const pocasPiezas = true;
3
4 if (esFinal || pocasPiezas) {
5   console.log("Se podría entrar en final de
6   partida pronto");
7 }
```

JavaScript



```
1 const enJaque = false;
2
3 if (!enJaque) {
4   console.log("El rey no está en jaque, puedes
5   continuar moviendo libremente");
6 }
```





2.5.- Condicionales de JavaScript

JS Logical

► Ejemplos combinaciones lógicas



```
1 const turno = "blancas";
2 const enJaque = true;
3 const puedeEnrocar = false;
4
5 if (turno === "blancas" && (enJaque ||
6   puedeEnrocar)) {
7   console.log("Blancas tienen que responder a
   jaque o aprovechar el enroque");
8 }
```





2.5.- Condicionales de JavaScript

JS Logical

- Ejemplos Uso de ?? (Nullish Coalescing) en ajedrez



```
1 function obtenerPropiedad(obj, propiedad) {  
2   return obj[propiedad] ?? "valor no definido";  
3 }  
4  
5 const pieza = { tipo: "Rey", color: "blanco" };  
6 const posicion = obtenerPropiedad(pieza, "posicion");  
7 console.log(`Posición: ${posicion}`); // "Valor no  
definido" si no existe propiedad
```



2.5.- Condicionales de JavaScript

Resumen parte 5

- ▶ Usa &&, ||, ! para combinar condiciones de forma legible.
- ▶ Agrupa con paréntesis para controlar precedencia.
- ▶ Usa ?? cuando quieras un valor por defecto frente a null o undefined.
- ▶ Evita lógica demasiado enrevesada (divídela en funciones si es necesario).

2.5.- Condicionales de JavaScript

Buenas Prácticas

- ▶ Usa paréntesis para hacer explícito el orden de evaluación.
- ▶ No encadenes demasiadas condiciones en una sola línea; divide en funciones si es necesario.
- ▶ Evita usar ?? en versiones antiguas de navegadores sin soporte.
- ▶ Haz que las condiciones sean lo más legibles posible comparando con valores claros.



¿Alguna pregunta?



UT2.- Manejo de la
sintaxis del lenguaje


2.5.- Condicionales
de JavaScript



Aula Virtual: Actividad 07

UT2.- Manejo de la sintaxis del lenguaje

2.5.- Condicionales de JavaScript

The background of the slide features a close-up of a hand in a dark suit sleeve moving a black chess piece on a chessboard. The chessboard has alternating light and dark squares. Several other chess pieces, both white and black, are visible on the board. The right side of the image is decorated with several overlapping, semi-transparent green geometric shapes that create a modern, abstract look.

If Conditions JS switch JS Boolean JS Logical

UT2.- Manejo de la sintaxis del
lenguaje

2.5.- Condicionales de JavaScript