

UT2.- Manejo de la sintaxis del lenguaje

2.3.- Variables de JavaScript con ejemplos de Ajedrez

Desarrollo Web Entorno Cliente (DWEC)





Introducción JS let JS const JS Types

UT2.- Manejo de la sintaxis del lenguaje

2.3.- Variables de JavaScript con
ejemplos de Ajedrez



Introducción

UT2.- Manejo de la sintaxis del lenguaje

2.3.- Variables de JavaScript con
ejemplos de Ajedrez



2.3.- Variables de JavaScript

Introducción

- ▶ Declarar una variable: reservar un nombre para almacenar un valor.
- ▶ Palabras clave:
 - `let` --> declarar variable que puede cambiar.
 - `const` --> declarar valor constante que no debe cambiar.
 - `var` --> sintaxis antigua, con reglas de alcance problemáticas (menos recomendada).
- ▶ Se pueden declarar varias variables en una sola línea con comas.
- ▶ Una variable sin valor inicial tiene valor `undefined`.
- ▶ Buenas prácticas:
 - preferir `const` cuando el valor no deba cambiar,
 - solo usar `let` cuando sea necesario cambiar ese valor.
 - Nunca usar variables “implícitas” (sin declarar).





2.3.- Variables de JavaScript

Introducción -

Ejemplo con buenas prácticas



HTML



```
1 <p id="info">Movimiento: -</p>
2 <button id="btnMover">Mover peón</button>
```

JavaScript



```
1 const info = document.getElementById("info");
2 const btnMover = document.getElementById("btnMover");
3
4 function moverPeon() {
5     const numeroPasos = 2;           // cuántas casillas avanzar
6     let posicionInicial = 2;         // posición actual
7     let nuevaPosicion = posicionInicial + numeroPasos;
8     info.innerHTML = `El peón avanza a la casilla ${nuevaPosicion}`;
9 }
10
11 btnMover.addEventListener("click", moverPeon);
```





JS let

UT2.- Manejo de la sintaxis del lenguaje

2.3.- Variables de JavaScript con
ejemplos de Ajedrez



2.3.- Variables de JavaScript

JS let

- Introducido en ES6 (2015).
- Permite declarar **variables con alcance de bloque** ({...}).
- Se puede reasignar el valor de la variable.
- No se puede redeclarar en el mismo bloque.

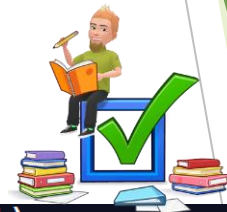




2.3.- Variables de JavaScript

JS let

Ejemplo ajedrez con buenas prácticas



```
1 const info = document.getElementById("info");
2
3 function cambiarPeon() {
4     let pieza = "♙"; // peón blanco
5     pieza = "♜";      // ahora se convierte en peón negro
6     info.innerHTML = `Nueva pieza: ${pieza}`;
7 }
```





JS const

UT2.- Manejo de la sintaxis del lenguaje

2.3.- Variables de JavaScript con
ejemplos de Ajedrez



2.3.- Variables de JavaScript

JS const

- Introducido en ES6 (2015).
- Sirve para **valores que no deben cambiar**.
- Obligatorio inicializar la variable al declararla.
- No significa que el valor sea inmutable (ejemplo: arrays y objetos pueden modificarse internamente).





2.3.- Variables de JavaScript

JS const

Ejemplo ajedrez con buenas prácticas



```
1 const info = document.getElementById("info");
2
3 // Número de piezas iniciales nunca cambia
4 const PIEZAS_INICIALES = 32;
5
6 function mostrarPiezas() {
7     info.innerHTML = `Piezas iniciales: ${PIEZAS_INICIALES}`;
8 }
```





JS Tipos de datos (Types)

UT2.- Manejo de la sintaxis del lenguaje

2.3.- Variables de JavaScript con ejemplos de Ajedrez



2.3.- Variables de JavaScript

JS Types (Tipos de Datos)

► Tipos primitivos:

- string: cadenas de texto --> "♔", "Rey"
- number: numéricos --> 8, 3.14
- boolean: true / false --> esTurnoBlancas
- undefined: variable declarada pero sin valor
- null: ausencia intencional de valor --> casilla vacía
- symbol: valor único e inmutable (poco usado en este nivel)
- bigint: enteros muy grandes

► Tipos compuestos:

- object: colecciones de pares clave-valor --> objeto pieza
- array: listas ordenadas





2.3.- Variables de JavaScript

JS Types (Tipos de Datos)

TIPOS DE DATOS EN JAVASCRIPT		NOMBRE	DESCRIPCIÓN aprenderaprogramar.com
	TIPOS PRIMITIVOS	String	Cadenas de texto
		Number	Valores numéricos
		Boolean	true ó false
		Null	Tipo especial, contiene null
		Undefined	Tipo especial, contiene undefined
	TIPOS OBJETO	Tipos predefinidos de JavaScript	Date (fechas) RegExp (expresiones regulares) Error (datos de error)
		Tipos definidos por el programador / usuario	Funciones simples Clases
		Arrays	Serie de elementos o formación tipo vector o matriz. Lo consideraremos un objeto especial que carece de métodos.
		Objetos especiales	Objeto global
			Objeto prototipo
			Otros





2.3.- Variables de JavaScript

JS Types (Tipos de Datos)

Ejemplo ajedrez con buenas prácticas



```
1 // Tipos primitivos
2 let rey = "♔";           // string
3 let filas = 8;           // number
4 let turnoBlancas = true; // boolean
5 let piezaCapturada;     // undefined
6 let sinMovimiento = null; // null
7
8 // Tipos compuestos
9 const piezas = ["♠", "♠", "♠"]; // array
10 const jugador = { nombre: "Ana", color: "blancas" }; // objeto
```





2.3.- Variables de JavaScript

JS Types (Tipos de Datos)

Ejemplo práctico Tipos en acción

► Versión simple



```
1 let jugador = "Carlos";  
2 let piezas = 16;  
3 document.getElementById("info").innerHTML = jugador + " tiene " + piezas + " piezas";
```

► Versión con buenas prácticas

```
1 const info = document.getElementById("info");  
2  
3 function mostrarJugador() {  
4   const jugador = { nombre: "Carlos", piezas: 16 };  
5   info.innerHTML = `${jugador.nombre} tiene ${jugador.piezas} piezas`;  
6 }
```





2.3.- Variables de JavaScript

JS Types (Tipos de Datos)

El operador typeof

► ¿Qué es typeof?

- typeof devuelve el tipo de una variable o expresión.
- Ejemplos (concepto)
 - `typeof " ♚ " // "string",`
 - `typeof 8 // "number".`





2.3.- Variables de JavaScript

JS Types (Tipos de Datos)

El operador typeof

► Versión simple



```
1 <p id="tipo">--</p>
2 <button onclick="document.getElementById('tipo').innerHTML = typeof '△'">
3   ¿Qué tipo es '△'?
4 </button>
```

► Versión con buenas prácticas



```
1 <p id="tipo">--</p>
2 <button id="btnTipo">¿Qué tipo es '△'?</button>
```



```
1 const tipoElemento = document.getElementById("tipo");
2 const btnTipo = document.getElementById("btnTipo");
3
4 =function mostrarTipoDePieza() {
5   tipoElemento.innerHTML = typeof "△"; // "string"
6 }
7
8 btnTipo.addEventListener("click", mostrarTipoDePieza);
```

JavaScript





2.3.- Variables de JavaScript

JS Types (Tipos de Datos)

String (cadenas)

► Reglas rápidas (Strings)

- Texto entre comillas simples '...' o dobles "...".
- También podemos usar **backticks** `...` --> *template literals*.
 - Permiten:
 - Escribir cadenas en **varias líneas**.
 - Incrustar variables fácilmente con `${variable}`.





2.3.- Variables de JavaScript

JS Types (Tipos de Datos)

String (cadenas)

► Versión simple



```
1 <p id="casilla">♖</p>
2 <button onclick="document.getElementById('casilla').innerHTML = '♖ en e4'">
3   Promover a torre
4 </button>
```

► Versión con buenas prácticas



```
1 <p id="casilla">♖</p>
2 <button id="btnPromover">Promover a torre</button>
```



```
1 const casilla = document.getElementById("casilla");
2 const btnPromover = document.getElementById("btnPromover");
3
4 function promoverAPieza() {
5   const pieza = "♖";
6   const posicion = "e4";
7   casilla.textContent = `${pieza} en ${posicion}`;
8 }
9
10 btnPromover.addEventListener("click", promoverAPieza);
```





2.3.- Variables de JavaScript

JS Types (Tipos de Datos)

Number y BigInt

► Number

- Representa tanto enteros como decimales.
- Ejemplo en ajedrez: valores asignados a las piezas (♙ =1, ♘=3, ♔=9).
- Siempre se manejan como números de precisión doble (64 bits).

► BigInt

- Sirve para enteros demasiado grandes para un Number.
- Se crean añadiendo una n al final: 123456789012345678901234567890n.
- No se usan en ajedrez normalmente, pero sirven para cálculos muy grandes (ej.: total de posibles partidas de ajedrez).





2.3.- Variables de JavaScript

JS Types (Tipos de Datos)

Number y BigInt

- Versión con buenas prácticas
- Ejemplo number: Devuelve valor pieza



```
1 <p id="valor">—</p>
2 <button id="btnValor">Comparar piezas</button>
```



```
1 const valorElemento = document.getElementById("valor");
2 const btnValor = document.getElementById("btnValor");
3
4 function compararPiezas() {
5   const dama = 9;      // valor de la dama
6   const caballo = 3;   // valor del caballo
7   const diferencia = dama - caballo;
8
9   valorElemento.textContent = `La dama vale ${diferencia} más que el caballo`;
10 }
11
12 btnValor.addEventListener("click", compararPiezas);
```





2.3.- Variables de JavaScript

JS Types (Tipos de Datos)

Number y BigInt

- Versión con buenas prácticas
- Ejemplo bigint



```
1 // Número aproximado de posibles partidas de ajedrez (muy grande)
2 const posiblesPartidas = 123456789012345678901234567890n;
3
4 // Mostrar en consola (para depuración)
5 console.log(`El número de posibles partidas es: ${posiblesPartidas}`);
```





2.3.- Variables de JavaScript

JS Types (Tipos de Datos)

Boolean, Undefined, Null, Object, Symbol

Boolean

- p. ej. `let esTurnoBlancas = true;` (usabooleans para estado).

Undefined

- variable declarada sin valor (`let pieza;` → undefined).

Null

- ausencia intencional (p. ej. `casilla.pieza = null` al capturar).

Object

- modela piezas/jugadores/tablero:

```
1 | const pieza = { tipo: "Peón", color: "blanco", posicion: "e2" };
```

Symbol

- identificadores únicos (raro)





2.3.- Variables de JavaScript

JS Types (Tipos de Datos)

Boolean, Undefined, Null, Object, Symbol

► Versión con buenas prácticas

► Ejemplo



```
1  =const pieza = {
2      id: Symbol("pieza1"),
3      tipo: "Rey",
4      color: "blanco",
5      posicion: "e1",
6      capturada: false
7  };
8
9  =function capturar(p) {
10     p.capturada = true;
11     p.posicion = null; // usamos null para indicar ausencia en tablero
12 }
```





2.3.- Variables de JavaScript

JS Types (Tipos de Datos)

Resumen y buenas prácticas aplicadas

Usa `typeof` para depurar o validar tipos cuando sea necesario.

Prefiere `const/let` frente a `var`.

Usa `textContent` si sólo pones texto; `innerHTML` si necesitas HTML.

Modela entidades complejas (piezas, tablero, jugador) con **objetos** y **arrays**;

- Reserva los tipos primitivos para valores simples.

Documenta cuándo una variable puede ser `null` o `undefined`

- Mejor explícito: `null` para “sin pieza”.





¿Alguna pregunta?



UT2.- Manejo de la sintaxis del lenguaje

2.1.- Introducción a JavaScript con
ejemplos de Ajedrez