

■ Funciones con bucles y arrays (sin condicionales dentro de los bucles)

Regla didáctica: puedes usar condicionales (if/switch/ternario) dentro de la función, pero no dentro del cuerpo del bucle. Las decisiones se toman antes o después del bucle; el bucle solo recorre.

1) Pares del 2 al 30 (sin filtrar)

```
// Genera pares con paso +2, sin necesidad de preguntar dentro del bucle
function paresHasta30() {
  for (let n = 2; n <= 30; n += 2) {
    console.log(n);
  }
}
paresHasta30();
```

2) Cuenta atrás del 10 al 1

```
function cuentaAtras() {
  for (let i = 10; i >= 1; i--) {
    console.log(i);
  }
}
cuentaAtras();
```

3) Mostrar 10 animales (solo recorrer)

```
function mostrarAnimales() {
  const animales = ["perro", "gato", "elefante", "jirafa", "león", "tigre", "conejo", "delfín", "oso", "caballo"];
  for (let i = 0; i < animales.length; i++) {
    console.log(animales[i]);
  }
}
mostrarAnimales();
```

4) Números en posiciones pares del array (0,2,4,...)

```
function posicionesPares() {
  const datos = ["a", "b", "c", "d", "e", "f", "g", "h"];
  for (let i = 0; i < datos.length; i += 2) { // paso de 2
    console.log(i, datos[i]);
  }
}
posicionesPares();
```

5) Sumar todos los elementos de un array

```
function sumarTodo() {
  const numeros = [4, 7, 2, 9, 1, 5];
  let suma = 0;
  for (let i = 0; i < numeros.length; i++) {
    suma += numeros[i];
  }
}
```

```

    }
    console.log("Suma total:", suma);
}
sumarTodo();

```

6) Promedio de notas (decisión FUERA del bucle)

```

function promedioNotas(notas) {
    // Validación/decisión antes de bucle (permitido)
    if (!Array.isArray(notas) || notas.length === 0) {
        console.log("Debes pasar un array con al menos un número.");
        return;
    }
    let suma = 0;
    for (let i = 0; i < notas.length; i++) {
        suma += notas[i];
    }
    const promedio = suma / notas.length;
    console.log("Promedio:", promedio.toFixed(2));
}
promedioNotas([5, 8, 9, 6, 7, 10]);

```

7) Tabla de multiplicar (sin condicionales)

```

function tablaMultiplicar(n) {
    for (let i = 1; i <= 10; i++) {
        console.log(n + " x " + i + " = " + (n * i));
    }
}
tablaMultiplicar(6);

```

8) Invertir un array (recorrer hacia atrás)

```

function invertir() {
    const colores = ["rojo", "azul", "verde", "amarillo", "morado"];
    for (let i = colores.length - 1; i >= 0; i--) {
        console.log(colores[i]);
    }
}
invertir();

```

9) Generar los primeros 10 múltiplos de 3

```

function multiplosDe3() {
    const cantidad = 10;
    for (let i = 1; i <= cantidad; i++) {
        console.log(3 * i);
    }
}
multiplosDe3();

```

10) Nombres numerados (formato 1.- Ana, 2.- Pedro, ...)

```
function listarNombresNumerados() {
  const nombres = ["Ana", "Pedro", "Andrés", "Lucía", "Antonio", "Marta"];
  for (let i = 0; i < nombres.length; i++) {
    console.log((i + 1) + ".- " + nombres[i]);
  }
}
listarNombresNumerados();
```

11) Construir un array con los 12 primeros cuadrados

```
function cuadrados() {
  const resultado = [];
  for (let i = 1; i <= 12; i++) {
    resultado.push(i * i);
  }
  console.log(resultado);
}
cuadrados();
```

12) Copiar array (crear un clon elemento a elemento)

```
function clonarArray() {
  const origen = ["sol", "luna", "estrella", "mar"];
  const copia = [];
  for (let i = 0; i < origen.length; i++) {
    copia.push(origen[i]);
  }
  console.log("Original:", origen);
  console.log("Copia:", copia);
}
clonarArray();
```

13) Concatenar prefijo a cada elemento (sin if en el bucle)

```
function prefijar() {
  const tags = ["js", "css", "html"];
  const resultado = [];
  for (let i = 0; i < tags.length; i++) {
    resultado.push("#" + tags[i]);
  }
  console.log(resultado);
}
prefijar();
```

14) Suma solo de posiciones pares (sin comprobar nada dentro)

```
function sumarPosicionesPares() {
  const numeros = [10, 5, 8, 11, 14, 20, 7];
  let suma = 0;
  for (let i = 0; i < numeros.length; i += 2) { // 0,2,4,6...
    suma += numeros[i];
  }
  console.log("Suma de posiciones pares:", suma);
}
```

```
sumarPosicionesPares();
```

15) Reemplazar negativos por 0 usando Math.max (sin if)

```
function reemplazarNegativos() {  
  const numeros = [4, -3, 7, -1, 5, -9];  
  const limpio = [];  
  for (let i = 0; i < numeros.length; i++) {  
    limpio.push(Math.max(numeros[i], 0)); // sin if  
  }  
  console.log("Original:", numeros);  
  console.log("Sin negativos:", limpio);  
}  
reemplazarNegativos();
```

16) Comprobar si todos son no-negativos (decisión DESPUÉS del bucle)

```
function todosNoNegativos(valores) {  
  let minimo = valores[0];  
  for (let i = 1; i < valores.length; i++) {  
    minimo = Math.min(minimo, valores[i]); // sin if  
  }  
  // Decisión fuera del bucle (permitida)  
  if (minimo >= 0) {  
    console.log("Todos son  $\geq 0$ ");  
  } else {  
    console.log("Hay valores negativos");  
  }  
}  
todosNoNegativos([4, 8, 0, 9, 1]);
```

17) Buscar un elemento (decisión fuera; el bucle solo imprime)

```
function buscarElemento(arr, buscado) {  
  const indice = arr.indexOf(buscado); // sin bucle explícito para buscar  
  // Decisión fuera de cualquier bucle  
  if (indice === -1) {  
    console.log("No está:", buscado);  
    return;  
  }  
  // Bucle solo para mostrar todos (no usa if)  
  for (let i = 0; i < arr.length; i++) {  
    console.log(arr[i]);  
  }  
  console.log("Encontrado en posición:", indice);  
}  
buscarElemento([3,8,15,22,7,14,5,30], 22);
```

18) Construir serie 1..N (o 2..N si empiezaEnDos) (if fuera, bucle limpio)

```
function serieHastaN(N, empiezaEnDos) {
```

```

let inicio = 1;
if (empiezaEnDos === true) { // decisión antes del bucle
  inicio = 2;
}
for (let i = inicio; i <= N; i++) {
  console.log(i);
}
}
serieHastaN(10, true);

```

19) Mostrar solo los 5 primeros elementos (límite decidido antes)

```

function primerosCinco(arr) {
  // Si el array tiene menos de 5, usamos su longitud
  const limite = arr.length < 5 ? arr.length : 5; // condicional FUERA del bucle
  for (let i = 0; i < limite; i++) {
    console.log(arr[i]);
  }
}
primerosCinco(["uvas", "peras", "manzanas", "kiwis", "plátanos", "mangos"]);

```

20) Serie de pares hasta N (elige dirección antes del bucle)

```

function paresHastaN(N, descendente) {
  if (descendente) { // decisión antes del bucle
    const inicio = N % 2 === 0 ? N : N - 1; // todavía fuera del bucle
    for (let i = inicio; i >= 2; i -= 2) {
      console.log(i);
    }
  } else {
    for (let i = 2; i <= N; i += 2) {
      console.log(i);
    }
  }
}
paresHastaN(20, false);

```