
Apuntes: Bucles en Programación

Los bucles son estructuras de control que permiten ejecutar un bloque de código repetidamente. Existen varios tipos, cada uno con sus características y usos.

Tipos de Bucles:

- **for**: Ideal para iterar un número específico de veces.

- **Sintaxis:** **for** (*inicialización*; *condición*; *incremento/decremento*)
{ ... }

- **Ejemplo:**

```
1 for (let i = 0; i < 10; i++) {  
2   console.log(i); // Imprime del 0 al 9  
3 }
```

- **Explicación:** *i* se inicializa en 0. El bucle continúa mientras *i* sea menor que 10, incrementando *i* en 1 tras cada iteración.

- **while**: Ejecuta un bloque de código mientras una condición sea verdadera. ¡Cuidado con los bucles infinitos!

- **Sintaxis:** **while** (*condición*){ ... }

- **Ejemplo:**

```
1 let i = 0;  
2 while (i < 10) {  
3   console.log(i); // Imprime del 0 al 9  
4   i++;  
5 }
```

- **Explicación:** El bucle se ejecuta mientras *i* sea menor que 10. *i++* incrementa *i* en cada iteración.

- **do...while**: Similar a **while**, pero la condición se evalúa *después* de ejecutar el bloque de código, garantizando al menos una ejecución.

- **Sintaxis:** **do** { ... } **while** (*condición*);

- **Ejemplo:**

```
1 let i = 0;  
2 do {  
3   console.log(i); // Imprime del 0 al 9  
4   i++;  
5 } while (i < 10);
```

-
- **Explicación:** El bloque dentro de **do...while** se ejecuta una vez. Luego se evalúa **i < 10**. Si es verdadera, el bucle continúa; si es falsa, termina.

- **for...in:** Itera sobre las propiedades enumerables de un objeto.

- **Sintaxis:** **for** (**const** property in object){ ... }

- **Ejemplo:**

```
1 const obj = { a: 1, b: 2, c: 3 };
2 for (const property in obj) {
3   console.log(`${property}: ${obj[property]}`);
4 }
```

- **Explicación:** Itera sobre las propiedades 'a', 'b' y 'c' de **obj**.

- **for...of:** Itera sobre los valores de un iterable (arrays, strings, maps, sets).

- **Sintaxis:** **for** (**const** value of iterable){ ... }

- **Ejemplo:**

```
1 const arr = [1, 2, 3];
2 for (const value of arr) {
3   console.log(value); // Imprime 1, 2, 3
4 }
```

- **Explicación:** Itera sobre los valores de **arr**.

Control de flujo dentro de bucles:

- **break:** Termina la ejecución del bucle por completo.
- **continue:** Salta a la siguiente iteración del bucle.

Ejemplos adicionales (Completar):

- Calcular la suma de los números del 1 al 100:

```
1 let suma = 0;
2 for (let i = 1; i <= 100; i++) {
3   suma += i;
4 }
5 console.log(suma); // Imprime 5050
```

- Imprimir los números pares del 1 al 20:

```
1 for (let i = 2; i <= 20; i += 2) {
2   console.log(i);
3 }
```

-
- Recorrer un array y encontrar el elemento mayor:

```
1 const arr = [5, 2, 9, 1, 5, 6];
2 let mayor = arr[0];
3 for (let i = 1; i < arr.length; i++) {
4   if (arr[i] > mayor) {
5     mayor = arr[i];
6   }
7 }
8 console.log(mayor); // Imprime 9
```