

# WHILE VS FOR

---

## ¿Cuándo usar `while`?

El `while` es útil cuando **no sabes exactamente cuántas veces** se repetirá un bloque de código. Se ejecuta **mientras una condición sea verdadera**.

### ✓ Cuándo usar `while`

- Cuando el número de repeticiones **depende de una condición externa** que puede cambiar en cualquier momento.
- Cuando quieres **detener la ejecución en cierto punto** sin recorrer todo un rango.
- Cuando la cantidad de repeticiones **no está definida** de antemano.

♦ **Ejemplo: Buscar un elemento en un arreglo (puede estar en cualquier posición)**

```
let frutas = ["Manzana", "Banana", "Cereza", "Uva"];
let buscar = "Cereza";
let i = 0;

while (i < frutas.length) {
  if (frutas[i] === buscar) {
    console.log("Encontrado en la posición:", i);
    break; // Detiene el bucle cuando encuentra la fruta
  }
  i++;
}
```

### 📌 ¿Por qué `while` aquí?

No sabemos **en qué posición** está "Cereza", así que revisamos hasta encontrarla.

---

## ¿Cuándo usar `for`?

El `for` es útil cuando **sabes exactamente cuántas veces** se repetirá el código. Se usa cuando se necesita **recorrer un rango definido de valores**.

## ✓ Cuándo usar `for`

- Cuando **se conoce de antemano el número de iteraciones**.
- Cuando necesitas recorrer **todas las posiciones de un arreglo**.
- Cuando el ciclo tiene un **inicio, una condición y un incremento claros**.

### ◆ Ejemplo: Recorrer un arreglo e imprimir cada elemento

```
let frutas = ["Manzana", "Banana", "Cereza", "Uva"];
let buscar = "Cereza";
let i = 0;

while (i < frutas.length) {
  if (frutas[i] === buscar) {
    console.log("Encontrado en la posición:", i);
    break; // Detiene el bucle cuando encuentra la fruta
  }
  i++;
}
```

### 🔴 ¿Por qué `for` aquí?

Sabemos que queremos recorrer **todo el arreglo** desde el índice `0` hasta `frutas.length - 1`, por lo que un `for` es ideal.

## Comparación rápida

Caso de uso	<code>while</code>	<code>for</code>
No sé cuántas veces se repetirá	✓ Sí	X No
Sé exactamente cuántas veces se repite	X No	✓ Sí
Recorro un arreglo completo	⚠ Se puede, pero no es lo ideal	✓ Mejor opción
Detener la ejecución cuando encuentre algo	✓ Sí	X No (requiere <code>break</code> )

### Regla general:

- Usa `for` cuando **sepas cuántas veces** debe repetirse el código.
- Usa `while` cuando **dependas de una condición externa** y no sepas cuántas iteraciones serán necesarias.