

---

**ACADEMY**  
by NUMEN

**FULL STACK**

**Comenzamos en unos  
minutos**

---

ACADEMY  
by NUMEN

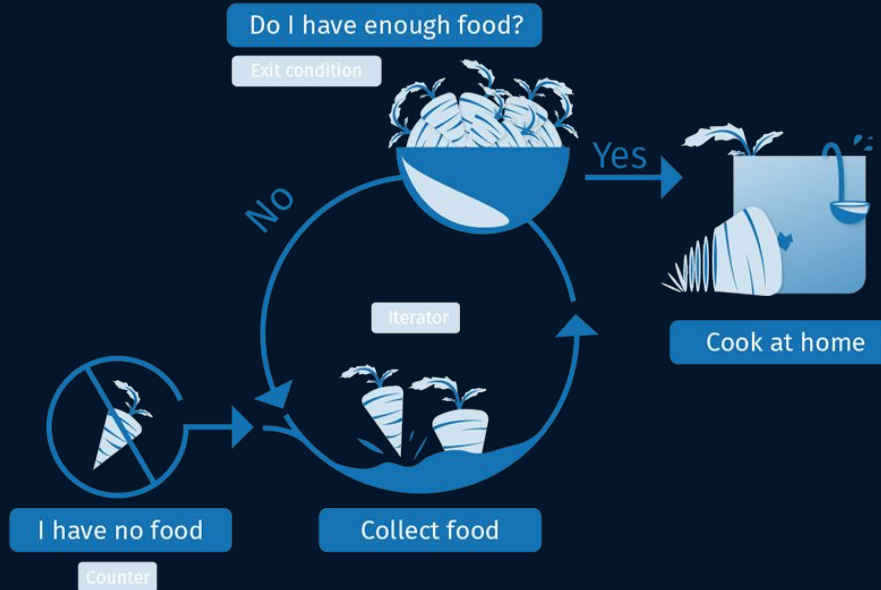
---

# Javascript: Clase 3

## Introducción a los Bucles

# ¿Qué es un bucle o ciclo?

Los bucles ofrecen una forma rápida y sencilla de **hacer algo repetidamente**. Hay muchos diferentes tipos de bucles, pero esencialmente, todos hacen lo mismo: repiten una acción varias veces. Los diversos mecanismos de bucle ofrecen diferentes formas de determinar los **puntos de inicio** y **terminación** del bucle. Hay varias situaciones que son fácilmente atendidas por un tipo de bucle que por otros.



# Bucles

## while

Crea un bucle que ejecuta una sentencia especificada mientras cierta condición se evalúe como verdadera. Dicha condición es evaluada antes de ejecutar la sentencia

```
let contador = 0;

while (contador < 10) {
  console.log(contador);
  contador++;
}
```

## do while

La sentencia (hacer mientras) crea un bucle que ejecuta una sentencia especificada, hasta que la condición de comprobación se evalúa como falsa. La condición se evalúa después de ejecutar la sentencia, dando como resultado que la sentencia especificada se ejecute al menos una vez.

```
let contador = 0;

do {
  console.log("do while" + contador);
  contador++;
} while (contador < 10);
```

# Bucles

## for

Los bucles for tienen una sintaxis única, similar a la instrucción if, pero un poco más compleja. Primero tenemos la palabra clave for, seguida de paréntesis y luego abrir y llaves. Dentro de los paréntesis necesitaremos tres cosas. Primero, debemos declarar una variable, esto es sobre lo que se repetirá el bucle. Entonces tendremos una expresión condicional, el ciclo continuará sucediendo hasta que esta declaración sea false. Tercero, incrementaremos nuestra variable. Las tres declaraciones están separadas por un punto y coma.

```
for (let i = 0 ; i < 10 ; i++ ) {  
  // | Declaramos una variable | Expresión condicional | Incrementamos la variable |  
  console.log(i);  
}
```

# Bucles

## for: iterando arrays

El Ciclo for también sirve para iterar sobre arreglos. Para esto debemos establecer una condición en la cual la variable sea menor a la longitud (length) del arreglo, de modo que luego de recorrer todo el arreglo el ciclo se corte. Lo veremos más claro en este ejemplo:

```
let arreglo = [10, 20, 30, 40, 50, 60, 70, 80, 90]

for (let i = 0; i < arreglo.length; i++) {
  console.log(arreglo[i]);
}
```

# Bucles

## forIn

La instrucción **for-in** itera sobre todas las propiedades enumerables de un objeto que está codificado por cadenas (ignorando los codificados por símbolos, incluidas las propiedades enumerables heredadas).

```
const academia = {
  nombre: "Academia",
  apellido: "Numen",
  servicio: "cursos"
}

for (const propiedad in academia) {
  console.log(propiedad);
}
```

## forOf

La sentencia **for-of** ejecuta un bloque de código para cada elemento de un objeto iterable, como lo son: string, array, objetos similares a array (por ejemplo, argumentos o listas nodo), map, set e iterables definidos por el usuario.

```
let numeros = [10, 20, 30, 40, 50];

for (const elemento of numeros) {
  console.log(elemento);
}
```



# Manejo de errores

## Try - Catch - Finally

La declaración **try...catch** señala un bloque de instrucciones a intentar (**try**), y especifica una respuesta si se produce una excepción (**catch**).

La sentencia **try** consiste en un bloque **try** que contiene una o más sentencias. Las llaves **{ }** se deben utilizar siempre, incluso para una bloques de una sola sentencia. Al menos un bloque **catch** o un bloque **finally** debe estar presente.

```
try {  
    // Código a evaluar  
} catch (error) {  
    /* Captura cualquier error  
    surgido o lanzado en el try */  
} finally {  
    /* Se ejecutará siempre al  
    final de un bloque try-catch */  
}
```

```
try {  
    let numero = "z";  
  
    if(isNaN(numero)) {  
        throw new Error(`El valor ingresado  
        no es un número`);  
    }  
  
    console.log(numero*numero);  
} catch (error) {  
    console.log(`Se produjo el  
    siguiente error: ${error}`);  
}
```

ACADEMY  
by NUMEN