Condición if() en JavaScript

Una condición if() en JavaScript es una estructura de control que permite ejecutar un bloque de código solo si una condición específica es verdadera. Es una de las formas más comunes de controlar el flujo de ejecución de un programa, permitiendo que ciertas acciones se realicen solo bajo ciertas circunstancias.

# Sintaxis básica de un if

if (condición) {  
 // Bloque de código que se ejecuta si la condición es verdadera  
}

- \*\*`condición`\*\*: Es una expresión que se evalúa como `true` o `false`. Si la condición es verdadera (`true`), el bloque de código dentro del `if` se ejecuta. Si es falsa (`false`), se salta el bloque de código.

# else y else if

- \*\*`else`\*\*: Es una cláusula opcional que define un bloque de código que se ejecuta si la condición es falsa.  
- \*\*`else if`\*\*: Permite evaluar múltiples condiciones en secuencia. Si una condición `if` no se cumple, puedes verificar otra con `else if`.

Sintaxis con else y else if:

if (condición1) {  
 // Si la condición1 es verdadera  
} else if (condición2) {  
 // Si la condición1 es falsa pero la condición2 es verdadera  
} else {  
 // Si ambas condiciones son falsas  
}

# Ejemplos de uso de if()

## Ejemplo 1: Verificar si un número es positivo

let numero = 10;  
  
if (numero > 0) {  
 console.log("El número es positivo.");  
} else {  
 console.log("El número no es positivo.");  
}

\*\*Explicación\*\*: Si el número es mayor que 0, muestra "El número es positivo". Si no lo es, muestra "El número no es positivo".

## Ejemplo 2: Evaluar una calificación de examen

let calificacion = 85;  
  
if (calificacion >= 90) {  
 console.log("Excelente!");  
} else if (calificacion >= 70) {  
 console.log("Aprobaste.");  
} else {  
 console.log("Reprobaste.");  
}

\*\*Explicación\*\*: El código verifica si la calificación es mayor o igual a 90 para mostrar "Excelente", si es mayor o igual a 70 muestra "Aprobaste", y en cualquier otro caso, muestra "Reprobaste".

## Ejemplo 3: Verificar si un número es par o impar

let numero = parseInt(prompt("Introduce un número:"));  
  
if (numero % 2 === 0) {  
 console.log("El número es par.");  
} else {  
 console.log("El número es impar.");  
}

\*\*Explicación\*\*: Se solicita al usuario un número. Si el resto de la división entre 2 es 0 (usando el operador `%`), el número es par, de lo contrario, es impar.

# ¿Cómo funcionan las condiciones en JavaScript?

Las condiciones dentro de un `if()` son \*\*expresiones\*\* que se evalúan como \*\*verdaderas (true)\*\* o \*\*falsas (false)\*\*. Algunas comparaciones comunes incluyen:

1. 1. \*\*Comparaciones de igualdad y desigualdad\*\*:

- `==` (igualdad) y `===` (estrictamente igual).  
- `!=` (desigualdad) y `!==` (estrictamente desigual).

Ejemplo:

let x = 5;  
if (x == 5) {  
 console.log("x es igual a 5");  
}

1. 2. \*\*Comparaciones mayores y menores\*\*:

- `>`, `<`, `>=`, `<=` (mayor, menor, mayor o igual, menor o igual).

Ejemplo:

let y = 10;  
if (y >= 5) {  
 console.log("y es mayor o igual a 5");  
}

1. 3. \*\*Operadores lógicos\*\*:

- `&&` (AND): Evalúa si \*\*todas\*\* las condiciones son verdaderas.  
- `||` (OR): Evalúa si \*\*al menos una\*\* de las condiciones es verdadera.  
- `!` (NOT): Invierte el valor lógico de una expresión.

Ejemplo:

let edad = 20;  
if (edad >= 18 && edad <= 65) {  
 console.log("Estás en edad laboral.");  
}

# Resumen

- La condición `if()` te permite ejecutar código solo cuando una \*\*condición\*\* se evalúa como verdadera.  
- Puedes usar \*\*`else`\*\* para manejar el caso en que la condición sea falsa, y \*\*`else if`\*\* para manejar múltiples condiciones.  
- Las condiciones pueden usar operadores de comparación, operadores lógicos, y otras expresiones que produzcan un resultado booleano.