



Sesión 4 – Semana 2

Bucles y Condicionales

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



Contenido

1. Estructuras cíclicas o bucles
 1. Declaración for
 2. Declaración while
2. Estructuras condicionales
 1. Condicionales simples
 2. De lo contrario: else (condicionales dobles)
 3. Condicionales múltiples:
 1. Condicionales anidadas
 2. De lo contrario, si: else if
 4. En caso de: Switch (Estructura de control)
 5. Operador ternario



Expresiones cíclicas o bucles

Sirven para ejecutar un mismo código un número determinado de veces, de forma rápida y sencilla, hasta que la condición asignada al bucle deja de cumplirse.

Hay muchos diferentes tipos de bucles, pero esencialmente, todos hacen lo mismo: repiten una acción varias veces.

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



Declaración `while`

Una declaración `while` ejecuta sus instrucciones siempre que una condición especificada se evalúe como `true`. Una instrucción `while` tiene el siguiente aspecto:

```
while (condición){  
    expresión  
}
```

Si la condición se vuelve `false`, la instrucción dentro del bucle se deja de ejecutar y el control pasa a la instrucción que sigue al bucle.

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



Declaración `while`

Ejemplo

El siguiente ciclo del `while` se repite siempre que `n` sea menor que `3`:

```
let n = 0;  
let x = 0;  
while (n < 3) {  
  n++;  
  x += n;  
}
```





Declaración `for`

Un ciclo `for` se repite hasta que una condición especificada se evalúe como false.

Una declaración `for` tiene el siguiente aspecto:

```
for ([expresiónInicial]; [expresiónCondicional];  
[expresiónDeActualización])  
    instrucción
```



Declaración for

Cuando se ejecuta un bucle for, ocurre lo siguiente:

- Se ejecuta la expresión de iniciación `expresiónInicial`, si existe. Esta expresión normalmente inicia uno o más contadores de bucle, pero la sintaxis permite una expresión de cualquier grado de complejidad. Esta expresión también puede declarar variables.
- Se evalúa la expresión `expresiónCondicional`. Si el valor de `expresiónCondicional` es verdadero, se ejecutan las instrucciones del bucle. Si el valor de condición es falso, el bucle for termina. (Si la expresión condición se omite por completo, se supone que la condición es verdadera).
- Se ejecuta la instrucción. Para ejecutar varias instrucciones, usa una declaración de bloque (`{ ... }`) para agrupar esas declaraciones.
- Si está presente, se ejecuta la expresión de actualización `expresiónDeActualización`.
- El control regresa al paso 2.

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



Declaración for

Ejemplo

Un ejemplo de utilización de este bucle lo podemos ver a continuación, donde se imprimirán los números del 0 al 10.

```
var i
for (i=0;i<=10;i++) {
    document.write(i)
    document.write("<br>")
}
```




Declaración for

Ejercicio

El factorial de un número entero n es una operación matemática que consiste en multiplicar todos los factores n :

$$n \times (n-1) \times (n-2) \times \dots \times 1.$$

Así, el factorial de 5 (escrito como $5!$) es igual a:

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

Utilizando la estructura `for`, crear un script que calcule el factorial de un número entero.

WWW.MAKAIA.ORG

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



WWW.MAKAIA.ORG



Estructuras condicionales

Las estructuras condicionales **comparan** una variable con otra u otros valores, para que con base al resultado de la comparación, se siga el curso de una acción dentro de un programa.

Asimismo, posibilitan crear ramas que permitan tomar diferentes caminos en el código dependiendo de ciertas condiciones.

Existen 3 tipos de condiciones: **Simple**, **Doble**, **Múltiples**. El resultado de la comparación en una estructura condicional tiene como resultado un valor lógico: **true** o **false**.

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



Condicionales simples

Se le conoce como “toma de decisión” y ejecuta una sentencia si una condición especificada es evaluada como verdadera.

Su sintaxis es la siguiente:

```
if (<condición>) {  
    // código que se ejecuta si se cumple la  
    condición  
}
```

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



Condicionales simples

Ejemplo

```
if (true) {  
    console.log("Hola Mundo");  
}
```

Nota: Las líneas que terminan con un corchete ({ o }) no se les agrega punto y coma (;).



Condicionales simples

Ejemplo

```
let num = 8;

if (num < 10) {
  console.log("El número es menor a 10");
}

if (num >= 10) {
  console.log("El número es igual o mayor a 10");
}
```



Condicionales dobles

if...else

Ejecuta una sentencia si una condición especificada es evaluada como verdadera. Si la condición es evaluada como falsa, otra sentencia puede ser ejecutada.

Su sintaxis es la siguiente:

```
if (condición) {  
    // código a ejecutar si la condición es verdadera  
} else {  
    // ejecuta este otro código si la condición es falsa  
}
```





Condicionales múltiples

Condiciones anidadas

Decimos que una estructura condicional es anidada cuando por la rama del verdadero o el falso de una estructura condicional hay otra estructura condicional.

Su sintaxis es la siguiente:

```
let num = 8;

if (num < 10) {
  console.log("El número es menor a 10");
} else {
  if (num > 10) {
    console.log("El número es mayor a 10");
  } else {
    console.log("El número es igual a 10");
  }
}
```



Condicionales múltiples

else if

Es preferible no tener que anidar condicionales porque son difíciles de leer y entender. Otro atajo que nos ofrece JavaScript para los condicionales es el **else if**, que significa "De lo contrario, si ..." en Inglés.

Su sintaxis es la siguiente:

```
if (<primera condición>) {  
    // código que se ejecuta si <primera condición> se cumple  
} else if (<segunda condición>) {  
    // código si <primera condición> NO se cumple, pero <segunda condición> se cumple  
} else if (<tercera condición>) {  
    // código si <primera condición> y <segunda condición> NO se cumplen, pero  
    <tercera condición> sí se cumple  
} else {  
    // código si ninguna de las condiciones se cumple  
}
```




Estructura de control Switch Case

Es muy útil cuando la condición que evaluamos puede tomar muchos valores. Las sentencias `switch` pueden tener una sintaxis más limpia que las sentencias `if else` complicadas:

```
const mascota = "perro";

if (mascota === "lagarto") {
  console.log("Tengo un lagarto");
} else if (pet === "dog") {
  console.log("Tengo un perro");
} else if (pet === "gato") {
  console.log("Tengo un gato");
} else if (pet === "serpiente") {
  console.log("Tengo una serpiente");
} else if (pet === "Loro") {
  console.log("Tengo un loro");
} else {
  console.log("No tengo mascota");
}
```

```
const mascota = "perro";

switch (mascota) {
  case "lagarto":
    console.log("Tengo un lagarto");
    break;
  case "perro":
    console.log("Tengo un perro");
    break;
  case "gato":
    console.log("Tengo un gato");
    break;
  case "serpiente":
    console.log("Tengo una serpiente");
    break;
  case "loro":
    console.log("Tengo un loro");
    break;
  default:
    console.log("No tengo mascota");
    break;
}
```





Operador ternario

Es una pequeña sintaxis que prueba una condición y devuelve un valor/expresión, si es **true**, y otro si es **false** — Esto puede ser útil en algunas situaciones, y puede ocupar mucho menos código que un bloque **if...else** si simplemente tienes dos opciones que se eligen a través de una condición **true/false**.

Su sintaxis es la siguiente:

`(condición) ? ejecuta este código : ejecuta este código en su lugar`

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



Operadores de comparación

Operador	Descripción	Ejemplos que devuelven true
Igual (==)	Devuelve true si los operandos son iguales.	3 == var1; "3" == var1; 3 == '3'
No es igual (!=)	Devuelve true si los operandos no son iguales.	var1 != 4; var2 != "3";
Estrictamente igual (===)	Devuelve true si los operandos son iguales y del mismo tipo.	3 === var1;
Desigualdad estricta (!==)	Devuelve true si los operandos son del mismo tipo pero no iguales, o son de diferente tipo.	var1 !== "3" 3 !== '3'
Mayor que (>)	Devuelve true si el operando izquierdo es mayor que el operando derecho.	var2 > var1 "12" > 2
Mayor o igual que (>=)	Devuelve true si el operando izquierdo es mayor o igual que el operando derecho.	var2 >= var1 var1 >= 3
Menor que (<)	Devuelve true si el operando izquierdo es menor que el operando derecho.	var1 < var2 "2" < 12
Menor o igual (<=)	Devuelve true si el operando izquierdo es menor o igual que el operando derecho.	var1 <= var2 var2 <= 5

```
let var1 = 3;  
let var2 = 4;
```



Operadores lógicos

Operador	Uso	Descripción
AND Lógico (&&)	<code>expr1 && expr2</code>	Devuelve <code>expr1</code> si se puede convertir a <code>false</code> ; de lo contrario, devuelve <code>expr2</code> . Por lo tanto, cuando se usa con valores booleanos, <code>&&</code> devuelve true si ambos operandos son <code>true</code> ; de lo contrario, devuelve false.
OR lógico ()	<code>expr1 expr2</code>	Devuelve <code>expr1</code> si se puede convertir a <code>true</code> ; de lo contrario, devuelve <code>expr2</code> . Por lo tanto, cuando se usa con valores booleanos, <code> </code> devuelve true si alguno de los operandos es <code>true</code> ; si ambos son <code>falsos</code> , devuelve <code>false</code> .
NOT lógico (!)	<code>!expr</code>	Devuelve <code>false</code> si su único operando se puede convertir a <code>true</code> ; de lo contrario, devuelve <code>true</code> .



Ejercicios en clase

1. Escribe un programa que pregunte al usuario si es culpable o no. Asumiremos que:
 - En caso afirmativo el usuario responderá si
 - En caso contrario responderá no.
 - Si el usuario responde si se escribirá por el documento «irás a la cárcel».
 - Si el usuario responde no se escribirá por el documento web «irás a casa».
 - En cualquier otro caso en el documento web se escribirá «la documentación por favor».

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



Ejercicios en clase

1. Escribe un programa que responda a un usuario que quiere comprar un helado en una conocida marca de comida rápida cuanto le costará en función del topping que elija.

- El helado sin topping cuesta 1900
- El topping de oreo cuesta 1000
- El topping de KitKat cuesta 1500
- El topping de brownie cuesta 750
- El topping de lacasitos cuesta 950

En caso de no disponer del topping solicitado por el usuario el programa escribirá por pantalla «no tenemos este topping, lo sentimos. » y a continuación informar del precio del helado sin ningún topping.

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



Ejercicios de práctica

Realizar los siguientes ejercicios, con `if` y analizar su sintaxis con `Switch`:

1. Solicitar 2 números e indicar cual es el mayor o si son iguales
2. Leer 5 notas de un estudiante, calcular el promedio e indicar si está aprobado o suspendido.
3. Leer la marca de un auto, si es Ford el descuento es del 10%, si es Toyota el descuento es del 5%. Indicar el valor más el total con el descuento.
4. Leer número del mes e indicar si tiene 30 o 31 días.
5. Realizar un ejercicio con encadenamiento de `if/else` mínimo 3 encadenamientos, lógica aritmética, ciclos `for` o `while` y solicitud de datos y respuestas por consola.

- Ej: Programa que solicite cantidad de productos a comprar, solicitar la cantidad de veces indicadas el nombre y precio de cada producto. Si el producto cuesta más de 5000 pesos, lleva descuento del 5%, si cuesta más de 10000 lleva descuento del 10% y si cuesta más de 20000 lleva descuento del 20%. Mostrar la lista de productos, con el subtotal y total a pagar.

W W W . M A K A I A . O R G

Carrera 43 A # 34 - 155. Almacentro. Torre Norte. Oficina 701
Medellín (Antioquia), Colombia



W W W . M A K A I A . O R G



Fuente

1. <https://makeitrealcamp.gitbook.io/javascript-book/condicionales>
2. https://developer.mozilla.org/es/docs/Learn/JavaScript/Building_blocks/conditionals
3. <https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Statements/if...else>
4. <https://www.freecodecamp.org/espanol/news/javascript-if-else-y-if-then-sentencias-condicionales-en-js/>
5. https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Loops_and_iteration



■ WWW.MAKAIA.ORG
Info: comunicaciones@makaia.org

Corporación MAKAI A
Medellín, Colombia
Carrera 43A – 34-155. Almacentro
Torre Norte, Oficina 701
Teléfono: (+574) 448 03 74
Móvil: (+57) 320 761 01 76

