



Clase 14

JavaScript Parte 2

JS





Operadores



Los operadores nos van a servir para modificar y comprobar el valor de las variables, vamos a ver diferentes tipos de operadores:

- ✓ **Aritméticos**
- ✓ **Lógicos**
- ✓ **Relacionales**



Operadores



Aritméticos

Los operadores aritméticos nos van a permitir realizar operaciones matemáticas sobre las variables.

+	SUMA
-	RESTA
*	MULTIPLICACIÓN
/	DIVISIÓN
%	MÓDULO (RESTO)
++	INCREMENTO
--	DECREMENTO



Operadores



Aritméticos

Para permitirnos realizar operaciones matemáticas, necesitamos “asignar” los valores resultantes, esto se logra mediante los operadores de asignación:

Operador	Forma reducida	Equivalente a
=	$x = y$	$x = y$
+=	$x += y$	$x = x + y$
-=	$x -= y$	$x = x - y$
*=	$x *= y$	$x = x * y$
/=	$x /= y$	$x = x / y$
%=	$x \% = y$	$x = x \% y$



Operadores



Aritméticos

Estos operadores funcionan como en la mayoría de los lenguajes de programación.



Suma : 10 + 20

Resta: 20 - 30

Divisiones: 100 / 25

Multiplicaciones 5 * 5



Operadores



Aritméticos

En JavaScript también nos encontramos con los operadores de **incremento** y **decremento**, que añaden uno o restan uno a la variable numérica en la que sean aplicados.



```
var x = 5; //VALOR INICIAL DE X  
var y = --x //AQUI SE REALIZA 5 - 1 LO CUAL SE REALIZA CON --X
```



```
var x = 3; //VALOR INICIAL DE X  
var y = ++x //AQUI SE REALIZA 3 + 1 LO CUAL SE REALIZA CON ++X
```



Operadores



Lógicos

Los operadores lógicos son AND, OR y NOT.

AND	&&
OR	
NOT	!



Operadores



Lógicos: tablas de verdad

x	!x
true	false
false	true

x	y	x && y
true	true	true
true	false	false
false	true	false
false	false	false

x	y	x y
true	true	true
true	false	true
false	true	true
false	false	false



Operadores

Relacionales

Los operadores relaciones, permiten realizar comparaciones.

Mayor que: >

Menor que: <

Mayor o igual: >=

Menor o igual: <=

Igual que: ==

Distinto de: !=



Operadores



typeof

Este operador es especial dado que nos permite a nosotros conocer el tipo de dato que tiene la variable sobre la cual vamos a realizar una acción.

```
typeof 5; //NUMBER
typeof false; //BOOLEAN
typeof "Mariano"; //String
typeof undefined; //undefined
```



Condicionales



Las sentencias condicionales se utilizan para realizar diferentes acciones basadas en condiciones.

En JavaScript tenemos las siguientes declaraciones condicionales:

- ✓ **if** para especificar un bloque de código que se ejecutará, si una condición especificada es verdadera.
- ✓ **else** para especificar un bloque de código que se ejecutará, si la misma condición es falsa.
- ✓ **switch** para especificar muchos bloques alternativos de código que se ejecutarán.



Condicionales



Sentencia if

Se utiliza para especificar un bloque de código que debe ser ejecutado si una condición es verdadera (true).

```
if (condition){
```

código a ser ejecutado si la condición es verdadera

```
}
```



Condicionales



Sentencia else

Se utiliza para especificar un bloque de código que debe ser ejecutado si una condición es falsa (false).

```
if (condition){  
    código a ser ejecutado si la condición es verdadera  
} else {  
    código a ser ejecutado si la condición es falsa  
}
```



Condicionales



Sentencia switch

La instrucción switch se utiliza para realizar diferentes acciones basadas en diferentes condiciones.

```
switch(expression) {  
  case n:  
    code block  
    break;  
  case n:  
    code block  
    break;  
  default:  
    code block  
}
```

- ✓ La expresión del switch se evalúa una vez.
- ✓ El valor de la expresión se compara con los valores de cada caso.
- ✓ Si hay una coincidencia, se ejecuta el bloque de código asociado.
- ✓ Cuando se alcanza la palabra clave break, se rompe el bloque switch.
- ✓ Un break mejora el tiempo de ejecución porque "ignora" la ejecución de todo el resto del código en el bloque switch.
- ✓ La palabra clave default especifica el código que se ejecutará si no se produce ninguna coincidencia.



Estructuras Repetitivas



Una estructura repetitiva es un bucle (loop) que permite ejecutar un bloque de código.

Hay dos tipos de estructuras repetitivas:

- ✓ **Condicionales:** permiten que el código sea ejecutado siempre y cuando una condición especificada sea verdadera.
- ✓ **Incondicionales:** no se basan en condiciones, sino en un número finito de veces.



Estructuras Repetitivas



While

```
while (condition) {  
    bloque de código a ser ejecutado  
}
```

The number is 0
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9

```
<!DOCTYPE html>  
<html>  
<body>  
  
<h2>JavaScript while</h2>  
  
<p id="demo"></p>  
  
<script>  
var text = "";  
var i = 0;  
while (i < 10) {  
    text += "<br>The number is " + i;  
    i++;  
}  
document.getElementById("demo").innerHTML = text;  
</script>  
  
</body>  
</html>
```




Estructuras Repetitivas



For

- ✓ La sentencia 1 se ejecuta antes de que se inicie el bucle (el bloque de código).
- ✓ La sentencia 2 define la condición para ejecutar el bucle.
- ✓ La sentencia 3 se ejecuta cada vez que se ha ejecutado el bucle.

```
for (statement 1; statement 2; statement 3) {  
    bloque de código a ser ejecutado  
}
```