



# IES MARQUES DE COMARES

## UNIDAD 2

# Introducción a Javascript



# IES MARQUES DE COMARES

## ÍNDICE:

- 1.- Introducción**
- 2.- ¿Dónde introducir el código Javascript?**
- 3.- Constantes y Variables en Javascript**
- 4.- Tipos de datos Javascript**
- 5.- Operadores Javascript**
- 6.- Estructuras de control en Javascript**
- 7.- Comentarios**
- 8.- Funciones**



# IES MARQUES DE COMARES

## 1.- Introducción

Javascript es un lenguaje de programación del lado del cliente cuyo propósito es el de ejecutarse en un navegador, lo que implica rapidez y respuesta rápida.

Mediante Javascript no podemos crear aplicaciones que se ejecuten en un servidor ni aplicaciones locales de escritorio, por lo que solo podremos crear aplicaciones Web que se ejecuten en el navegador del cliente.

Es un lenguaje interpretado (un poco más lento porque cada vez que lo ejecute debe interpretarlo desde el principio) y no compilado

Mediante javascript podemos dar de una manera más o menos fácil interactividad a páginas web.

Se pueden utilizar librerías como JQuery que poseen multitud de funciones ya programadas.



# IES MARQUES DE COMARES

## 2.- ¿Dónde introducir el código Javascript?

Se puede ubicar en cualquier lugar del documento Web:

- Dentro de las etiquetas `<head>` y `</head>` mediante las etiquetas `<script>` `</script>`
- Dentro del body igualmente mediante las etiquetas `<script>` y `</script>`
- En un archivo externo con extensión js, normalmente entre las etiquetas `<head>` y `</head>` mediante la etiqueta `<script src="archivo js">` `</script>`



# IES MARQUES DE COMARES

## 3.- Constantes y Variables javascript

Una constante es el nombre y valor que le damos a un identificador, teniendo en cuenta que el valor de ese identificador (constante) no puede modificarse a lo largo de la estructura del programa (a excepción de los objetos), mientras que una variable es el nombre que se le da a un espacio en la memoria ram del ordenador, donde se almacenará un valor, y que podrá cambiar a lo largo de la ejecución del programa

- Los nombres de las variables y/o constantes solo pueden comenzar por una letra, el guión bajo (\_) o el símbolo de dolar (\$), seguida de cualquier combinación de los anteriores o números.
- Los nombres de variables no pueden coincidir con palabras de sintaxis javascript ni con palabras reservadas



# IES MARQUES DE COMARES

## Recomendaciones para nombrar Constantes y Variables javascript

- Usar notación **snake\_case** en el nombre de los archivos.  
Ej: `archivo_ejercicio1.js`
- Usar notación **UPPER\_CASE** en constantes.  
Ej: `const VALOR_GRAVEDAD = 9.8`
- Usar notación **UpperCamelCase** en clases  
Ej: `class MiCoche={`  
                                  `constructor (marca,color) {`  
                                  `this.marca=marca`  
                                  `this color=color`  
                                  `}`  
                                  `}`
- Usar notación **lowerCamelCase** en datos primitivos, objetos, funciones e instancias.  
Ejs:           `let miNombre="Miguel Ángel"`  
                  `let miPersona= {`  
                                  `nombre="Miguel",`  
                                  `tfno="665111111"`  
                                  `}`  
                  `function miNombre(nombre){`  
                                  `alert( 'mi nombre es ${nombre}' );`  
                                  `}`  
                  `coche=new miCoche('Ford','azul');`



# IES MARQUES DE COMARES

## Ámbito de Variables javascript

A partir de javascript ECS6, podemos definir variables de ámbito con la instrucción “let”, frente a la definición de variables globales con “var”.

```
var a=5;  
{ a=7; }  
console.log(a);  
//el valor de a será 7
```

```
var a=5;  
{ let a=7; console.log(a); //el valor de a es 7 }  
console.log (a);  
//el valor de a será 5
```



# IES MARQUES DE COMARES

## Variables javascript (Expresiones)

Colección de variables, operadores y otras expresiones que se evalúan a un solo valor

- Asignación

  - asigna un valor a una variable

- Aritméticas

  - evalúan un número

- Cadenas

  - evalúan una cadena

- Lógicas

  - evalúan un valor booleano





# IES MARQUES DE COMARES

## 4.- Tipos de datos javascript

La última definición del estándar ECMAScript define 6 tipos de datos primitivos y el tipo Object.

- Tipos básicos:

Boolean

Null

Undefined

Number

String

Symbol (nuevo en ECMAScript 6)

- Tipo Object



# IES MARQUES DE COMARES

## Tipos de datos javascript

### Tipo Boolean

Solo puede tomar los valores true o false.

### Tipo Null

El tipo Null tiene exactamente el valor null.

### Tipo Undefined

No tiene ningún valor asignado

### Tipo Number

Tiene un valor numérico entre  $-(2^{53} - 1)$  y  $(2^{53} - 1)$ . No existe un tipo específico para los números enteros.

Con el tipo Number es común usar el valor simbólico NaN (Not A Number o No Es Un Número) y las constantes Number.MAX\_VALUE, Number.MIN\_VALUE.



# IES MARQUES DE COMARES

## Tipos de datos javascript

### Tipo String

Almacena caracteres o cadenas de caracteres. Es un conjunto de caracteres donde cada carácter ocupa una posición en el String. El primer carácter está en el índice 0, el siguiente en el índice 1, y así sucesivamente. La longitud de un String es el número de caracteres de la cadena.

Los strings en JavaScript son inmutables, es decir, una vez que una cadena de caracteres es creada no es posible modificarla. Sin embargo es posible crear otra basandose en una operación de la cadena original.

### Tipo Symbol

El Symbol es un nuevo tipo primitivo que puede ser usado como la clave de una propiedad de un Object. Similares a los tipos Enum



# IES MARQUES DE COMARES

## Tipos de datos javascript

### Objetos

Un objeto es un valor en memoria al cual es posible referirse mediante un identificador

### Propiedades de un objeto

En JavaScript los objetos pueden ser vistos como una colección de propiedades que pueden ser de cualquier tipo, incluyendo otros objetos lo cual permite construir estructuras de datos complejas.

Un objeto puede tener 2 tipos de propiedades: de datos y de acceso.



# IES MARQUES DE COMARES

## Tipos de datos javascript

- **Propiedad de datos:** asocia una clave con un valor.

### Atributos de una propiedad de datos

Atributo	Tipo	Descripción	Valor por defecto
[[Value]]	Cualquier tipo JavaScript	El valor obtenido mediante un acceso get a la propiedad.	undefined
[[Writable]]	Boolean	Si es <code>false</code> , el [[Value]] de la propiedad no puede ser cambiado.	false
[[Enumerable]]	Boolean	Si es <code>true</code> , la propiedad será enumerada en ciclos <code>for...in</code> . Consultar también <a href="#">Enumerabilidad y pertenencia de las propiedades</a>	false
[[Configurable]]	Boolean	Si es <code>false</code> , la propiedad no puede ser eliminada y otros atributos que no sean [[Value]] y [[Writable]] no pueden ser cambiados.	false



# IES MARQUES DE COMARES

## Tipos de datos javascript

- **Propiedad de acceso:** Asocia una clave con una o más funciones de acceso (get y set) para obtener o almacenar un valor.

Atributos de una propiedad de acceso

Atributo	Tipo	Descripción	Valor por defecto
[[Get]]	Función, objeto o undefined	La función es llamada con una lista de argumentos vacía y devuelve el valor de la propiedad cada vez que un acceso get al valor es realizado. Consultar también <code>get</code> .	undefined
[[Set]]	Function object or undefined	La función es llamada con un argumento que contiene la variable asignada y es ejecutada siempre que una propiedad específica se intenta cambiar. Consultar también <code>set</code> .	undefined
[[Enumerable]]	Boolean	Si el valor es <code>true</code> , la propiedad será enumerada en bucles <code>for...in</code> .	false
[[Configurable]]	Boolean	Si el valor es <code>false</code> , la propiedad no podrá ser eliminada y tampoco ser modificada a una propiedad de datos.	false



# IES MARQUES DE COMARES

## 5.- Operadores javascript

- Operadores aritméticos
- Operadores lógicos
- Operadores de comparación
- Operadores de asignación
- Operadores de cadena
- Operador condicional



# IES MARQUES DE COMARES

## Operadores aritméticos

- + Adición
- Sustracción
- \* Multiplicación
- % Módulo
- ++ Incremento unario
- Decremento unario
- Negación unaria: devuelve la negación del operando.





# IES MARQUES DE COMARES

## Operadores lógicos

&&

"y" lógico, devuelve true solo cuando ambos operandos son verdaderos y false en caso contrario

||

"o" lógico, devuelve true cuando alguno de los operandos es verdadero y false en caso contrario

!

"no" lógico, devuelve true si el operando es falso, y false si el operando es verdadero



# IES MARQUES DE COMARES

## Operadores de asignación

=

+=, -=, \*=, /=

&=, |=



# IES MARQUES DE COMARES

## Operadores de comparación

- ==** Devuelve verdadero si los operandos son iguales
- !=** Devuelve verdadero si ambos operandos son distintos
- >** Devuelve verdadero si el primer operando es mayor que el segundo
- <** Devuelve verdadero si el primer operando es menor que el segundo
- >=** Devuelve verdadero si el primer operando es mayor o igual que el segundo
- <=** Devuelve verdadero si el primer operando es menor o igual que el segundo
- ===** Devuelve verdadero si ambos operandos son iguales y coinciden sus tipos
- !==** Devuelve verdadero si los operandos son distintos o de distinto tipo



# IES MARQUES DE COMARES

## Operadores de cadena

Permiten la unión de cadenas

Ej

"Hola, " + "mudo" nos devolvería la cadena "Hola, mundo"



# IES MARQUES DE COMARES

## Operador condicional

Permite evaluar una expresión lógica, y devolver un valor si la expresión es verdadera y otro si es falsa

(condición) ? valor1 : valor2

### Ejemplo

respuesta=( edad<18 ) ? "Eres menor de edad" : "Eres mayor de edad";



# IES MARQUES DE COMARES

## Operador instanceof

Se aplica solo a objetos.

Devuelve verdadero si la variable de tipo objeto a evaluar coincide con el tipo objeto especificado.

Ejemplo

```
var fecha=new Object();  
var respuesta=fecha instanceof Object
```

Por tanto, respuesta estará a true.



# IES MARQUES DE COMARES

## 6.- Estructuras de control

- Condicionales
- Repetitivas



# IES MARQUES DE COMARES

## Estructuras de control condicionales

if ... else

if (condición)

{

declaraciones1;

}

else

{

declaraciones2;

}





# IES MARQUES DE COMARES

## Estructuras de control condicionales

switch

```
switch (expresión) {  
    case valor :  
        declaración;  
        break;  
    case valor :  
        declaración;  
        break;  
    ...  
    default : declaración;  
}
```



# IES MARQUES DE COMARES

## Estructuras de control repetitivas

```
for  
  for (declaraciónInicial; prueba; incremento)  
  {  
    declaraciones;  
  }
```



# IES MARQUES DE COMARES

## Estructuras de control repetitivas

do ... while

do

{

declaraciones;

}

while (condición);

while

while (condición)

{

declaraciones;

}



# IES MARQUES DE COMARES

## 7.- Comentarios

- De una sola línea

//

- De varias líneas

/\*

\*/



# IES MARQUES DE COMARES

## 8.- Funciones

### Sintaxis

```
function nombre(arg1,arg2,...)
{
    sentencias;
}
```

Es recomendable situar las funciones entre las etiquetas `<head>` y `</head>`, ya que se cargarían antes que el resto de la página.