







# UNIDAD 2

# Introducción a Javascript



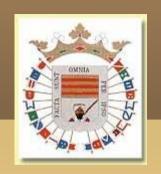






#### **ÍNDICE:**

- 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
- 9.- Orden de escritura de código en Javascript



#### 1.- Introducción

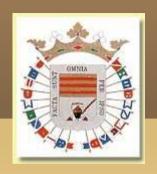
Javascript es un lenguaje de programación del lado del cliente cuyo propósito es el de ejecutarse en un navegador, lo que implica rapidez y respuesta rápida. Fue creado por Brendan Eich en 1995.

Mediante Javascript, aunque podemos crear aplicaciones que se ejecuten en un servidor, lo normal es realizar 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.



### 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="archvivo js"> </script>



### 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

#### 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={

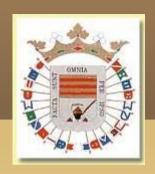
constructor (marca,color) {

this.marca=marca

this color=color

}
```

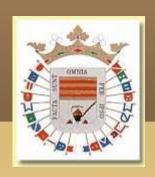
Usar notación lowerCamelCase en datos primitivos, objetos, funciones e instancias.



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



#### 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



### 4.- Tipos de datos javascript

Primitivos: Son aquellos en que se accede por valor:

string

number

boolean

null

undefined

NaN

Compuestos: Son aquellos en que se accede por referencia:

object

array

Class, etc



#### Tipos de datos javascript

#### **Tipo Boolean**

Solo puede tomar los valores true o false.

#### **Tipo Null**

El tipo Null tiene exactamente el valor null.

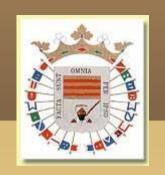
#### **Tipo Undefined**

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 N<u>aN</u> (Not A Number o No Es Un Número) y las constantes Number.MAX\_VALUE, <u>Number.MIN\_VALUE</u>.



#### 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 immutables, 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



### 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.



### 5.- Operadores javascript

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



### **Operadores aritméticos**

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



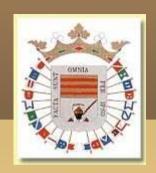
### **Operadores lógicos**

&&

"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



### Operadores de asignación



### 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
- Control de la companya del companya de la companya del companya de la companya del companya de la companya de la companya de la companya del companya de la companya del companya de la companya del companya de la companya del companya del companya del companya de la companya de la company
- >= Devuelve verdadero si el primer operando es mayor o igual que el segundo
- Compare de la compare de la
- === Devuelve verdadero si ambos operandos son iguales y coinciden sus tipos
- !== Devuelve verdadero si los operando son distintos o de distinto tipo

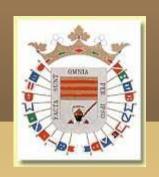


### Operadores de cadena

Permiten la unión de cadenas

Ej

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



### **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";



### **Operador intansceof**

Se aplica solo a objetos.

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

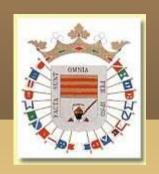
Ejemplo var fecha=new Object(); var respuesta=fecha instanceof Object

Por tanto, respuesta estará a true.

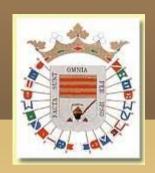


#### 6.- Estructuras de control

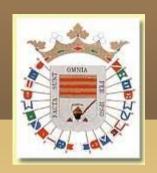
- Condicionales
- Repetitivas



#### Estructuras de control condicionales

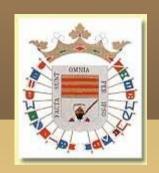


#### Estructuras de control condicionales



### Estructuras de control repetitivas

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



### Estructuras de control repetitivas

```
do ... while
   do
      declaraciones;
      while (condición);
while
   while (condición)
      declaraciones;
```



#### 7.- Comentarios

```
• De una sola línea //
```

De varias líneas /\*



#### 8.- Funciones

```
Sintaxis

function nombre(arg1,arg2,...)

{

sentencias;
}
```

Es recomendable situar las funciones entre las etiquetas <a href="head">head</a>, ya que se cargarían antes que el resto de la página.



### 9.- Orden de escritura de código en javascript

El orden de escritura correcto (o adecuado) en javascript sería el siguiente:

1º Importación de módulos

2º Declaración de variables

3º Declaración de funciones

4º Ejecución de código