

# JAVASCRIPT

Es un lenguaje interpretado para crear páginas web dinámicas (que acceden y modifican dinámicamente los elementos de la página según las interacciones de los usuarios).



# ¿QUÉ HACEMOS CON JAVASCRIPT?

Enriquecer el comportamiento de una página web:

- utilizar efectos de estilo dinámicos
- animaciones
- cambiar contenido HTML
- cambiar CSS

Programar robots.

Programar servidores

# ¿DÓNDE VA?

## IMPORTANTE

En HTML, el código JavaScript debe estar entre las etiquetas <script> y </script>.

## ¿DÓNDE?

Los scripts se pueden colocar:

- en el <head>
- en el <body>
- Externamente.

# SCRIPTS EXTERNOS

```
<!DOCTYPE html>
<html>

<head>
  <script src="hoja_javascript.js"></script> <!-- No buena idea -->

</head>

<body>
  <script src="hoja_javascript.js"></script> <!-- ¡Mejor idea! -->
</body>

</html>
```

# SCRIPTS EXTERNOS

## VENTAJAS DE USAR JS EXTERNOS:

- ✓ Separación del código HTML y JavaScript
- ✓ El código HTML y JavaScript son más fáciles de leer y mantener
- ✓ Los archivos en caché de JavaScript pueden acelerar la carga de la página.

Los scripts externos no contienen etiquetas <script>. Es decir, se define directamente el código JS.

# LAS VARIABLES

Se utilizan para almacenar y hacer referencia a valores.

```
var nombre;
```



En este caso hemos declarado una variable con el nombre nombre, pero no se le ha asignado ningún valor.

```
var nombre="nicolas";
```



Ahora le dimos un valor a esa variable que ya habíamos creado.

# TIPOS DE VARIABLES

- ✓ NUMÉRICAS

```
var edad = 10;
```

- ✓ STRINGS

```
var nombre = "Lucía";
```

- ✓ BOOLEANS

```
var activo = true;
```

- ✓ ARRAYS

```
var colores = ["rojo", "amarillo", "verde"];
```

```
var sisters = ["Tia", "Tamera"];
```

```
sisters[0];
```

OUTPUT

"Tia"

# TIPOS DE OBJETOS

## OBJECTS

Los objetos pueden contener diversos valores.

Los objetos poseen la siguiente sintaxis:

```
var nombre_objeto = {  
    key1: value1,  
    key2: value2,  
    ...  
}
```

```
var person = {  
    firstName:"John",  
    lastName:"Doe",  
    age:50,  
    eyeColor:"blue"  
};
```

Tenemos dos maneras de acceder a los valores de los mismos:  
`nombre_objeto.key1` o `nombre_objeto["key1"]`

# DIFERENCIA ENTRE OBJETOS Y ARRAYS

## OBJECTS

- **var arreglo**
  - ✓ `[];` // esto es un arreglo, los corchetes [] lo indican
  - ✓ `["VALUE", "VALUE"];`
- **var objeto**
  - ✓ `{};` // a diferencia del arreglo, un objeto literal que se identifica por sus llavecitas {}
  - ✓ `KEY:"VALUE";`

```
var sisters = ["Tia", "Tamera"];  
  
sisters[0];
```

OUTPUT  
"Tia"

```
var person = {  
    firstName:"John",  
    lastName:"Doe",  
    age:50,  
    eyeColor:"blue"  
};
```

# LOS OPERADORES

Los operadores nos servirán para modificar y comprobar el valor de las variables. Hay diferentes tipos de operadores:

- ✓ Matemáticos (+,-,/,\* , %)
- ✓ Lógicos (&&, ||, !)
- ✓ Relacionales (==, !=, >, <, >=, <=)

# LOS CONDICIONALES

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

Cuando se escribe programa, se suelen realizar diferentes acciones para distintas decisiones.

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

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

# EJERCICIOS

1. Crear un **objeto** Persona que va a tener atributos de tipo numérico y string.
2. Definir una variable numérica, asignarle un valor y sumarle 5.
3. Crear **variables** de datos que podría tener una persona. Debe haber un dato de tipo numérico, string, booleano y array.
4. Definir dos variables de cadenas, asignarles valores y concatenarlas.

5. Realizar un **algoritmo** que pida el nombre de una persona. Si el dato entra vacío, debe pedir nuevamente al usuario que introduzca su nombre (“No ha introducido ningún nombre”). Una vez hecho esto, que se despliegue una alerta de bienvenida con el nombre del usuario.
6. Crear algoritmo que pida la edad del usuario.
  - Sí el usuario introduce una edad entre 3 y 10 años, indicar que es niño.
  - Si el usuario introduce una edad entre 10 años y 17 años, indicar que es adolescente.
  - Si introduce una edad entre 18 y 100 años indicar que es adulto.
  - Si introduce mayor de 100 años, indicar no “ADMITIDO”.

*Nota: Imprimirlo con document.write*

7. Definir una variable numérica. Asignarle un valor entre 1 y 10 Mostrar a qué grupo pertenece:

- Grupo 1: del 1 al 3
- Grupo 2: del 4 al 6
- Grupo 3: del 7 al 10