

# LENGUAJES DE SCRIPT DE CLIENTE I

Un lenguaje de script es un pequeño lenguaje de programación cuyo código se inserta dentro del documento HTML. Este código se ejecuta en el navegador del usuario al cargar la página, o cuando sucede algo especial como pudiera ser, pulsar sobre un link. Estos lenguajes permiten variar dinámicamente el contenido del documento, modificar el comportamiento normal del navegador, validar formularios, realizar efectos visuales, etc. Sin embargo, hay que recordar que se ejecutan en el navegador del usuario, y no en el servidor, por lo que no permite, por ejemplo, gestionar bases de datos remotas. Los lenguajes de script de cliente más conocidos son: Javascript, VB Script, applets de Java o CSS.

## ***Páginas Sencillas con Javascript.***

**Javascript** fue el primer lenguaje de script, y lo diseñó Netscape. Está basado en la sintaxis de Java y por eso se popularizó tanto.

Este lenguaje es:

- Relativamente moderno.
- Es sencillo.
- Es útil.
- Es potente.
- Es gratuito.

HTML dispone de unas etiquetas para incluir código de script en una página. Esas etiquetas pueden situarse en la sección HEAD o en el BODY de la página (en función de lo que vaya a hacer el script).

Las sentencias escritas en Javascript se meten entre las etiquetas

```
<SCRIPT LANGUAGE="JavaScript" TYPE="text/JavaScript">
```

Y

```
</SCRIPT>
```

O simplemente entre <script> .... </script>

Un ejemplo de la primera forma:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript" TYPE="text/JavaScript">
alert ('Esto es un ejemplo con JavaScript');
</SCRIPT>
</BODY>
</HTML>
```

Y de la segunda:

```
<html><body>
<script>
alert("Bienvenido a mi sitio web. Gracias...");
</SCRIPT>
</BODY>
</HTML>
```

La orden ALERT, imprime en pantalla, en una ventana aparte, el mensaje que vaya entre paréntesis y entre comillas.

**Práctica 1:** Realizar una página web que incluya un mensaje por pantalla, que muestre tu nombre y apellidos. A continuación que imprima un agradecimiento por visitar tu página web.

### Pedir datos al usuario

Para comenzar a hacer programas un poco más serios, necesitamos datos que manipular. Una primera alternativa sencilla para obtener esos datos es pidiéndoselos al usuario. Igual que tenemos un "alert" que permite mostrar una ventana de aviso, también disponemos de un "prompt" que nos servirá para pedir datos.

Por ejemplo, podemos preguntar su nombre al usuario haciendo:

```
<html>

<body>
  Pedir datos...
<script>
  prompt("Dime tu nombre");
</script>
</body>
</html>
```

Pero claro, normalmente vamos a querer guardar eso que el usuario ha tecleado, para poder utilizarlo. Para eso usaremos "**variables**": una variable es un espacio de memoria del ordenador al que damos un nombre y en el que podemos guardar datos. Por ejemplo, podríamos pedir al usuario su nombre y guardarlo en una variable llamada "nombre". Para definir una variable, le ponemos delante la palabra reservada **var**. También se puede poner la palabra **let**. La primera opción es mas anticuada, pero igualmente funcional.

```
<html>
<body>
  Tu nombre es...
<script language="JavaScript">
  var nombre = prompt("Dime tu nombre");
  document.write( nombre );
</script>
</body>
</html>
```

Con la orden **document.write**, imprimimos en pantalla el dato almacenado en la variable *nombre*.

### Operaciones con números

Ahora que sabemos pedir datos, podemos solicitarle que introduzca dos o más, y a continuación hacer operaciones con ellos. Por ejemplo, podemos (intentar) pedirle dos números y mostrar su suma, así:

```
<html>
<body>
Sumador de numeros...
<script>
  var n1 = prompt("Dime un numero: ");
  var n2 = prompt("Dime otro numero: ");
  document.write("La suma de "+ n1 + " y "+ n2 + " es: ");
  document.write(n1+n2);
</script>
</body>
</html>
```

Pero esto no funciona bien. Vamos a ver por qué está fallando:

- Si escribimos `document.write("hola")`, se mostrará "hola" en la página (pero sin las comillas).
- Si escribimos `document.write("hola"+"tu")`, se mostrará "holatu", todo junto (también sin las comillas), porque el operador de la suma (+) se puede usar para textos, y lo que hace es crear un nuevo texto formado por esos dos, es decir, concatena los textos.
- Si escribimos `document.write(2+3)`, como vimos en el apartado anterior, aparecerá un 5 en la página, porque, al no haber comillas, se toman como números y su suma tendrá un valor numérico.
- En cambio, en este programa, si introducimos los números 2 y 3, nos dirán que su suma es 23...

El motivo es que **los datos que introducimos se consideran textos**, y aunque 2+3 sea 5, si pedimos el resultado de `n1+n2` se considerará que queremos unir esos dos textos en uno, de modo que el resultado no será un número, sino un nuevo texto, "23".

Se puede solucionar: basta con decirle "considera esto como un número" antes de sumar. Eso lo conseguimos con la función `parseInt` (el dato que hay que tomar como número se indicará entre paréntesis). En javascript hay que respetar las mayúsculas y minúsculas.

A continuación, se ve la aplicación práctica de lo explicado:

```
<html>
<body>
  Sumador de numeros...
  <script>
    var n1 = prompt("Dime un numero: ");
    var n2 = prompt("Dime otro numero: ");
    document.write("La suma de "+ n1 + " y "+ n2 + " es: ");
    document.write(parseInt(n1)+parseInt(n2));
  </script>
</body>
</html>
```

Aunque también lo podemos hacer así, que resulta más elegante:

```
<html>
<body>
  Sumador de numeros...
  <br>
  <script>
    let n1 = parseInt(prompt("Dime un numero: "));
    let n2 = parseInt(prompt("Dime otro numero: "));
    let suma;
    document.write("La suma de "+ n1 + " y "+ n2 + " es: ");
    suma=n1+n2;
    document.write(suma);
  </script>
</body>
</html>
```

En este caso utilizamos `parseInt` en el mismo momento de pedir el número por teclado, de manera que ese dato ya entra en las variables `n1` y `n2` con formato de número y no de texto. Luego utilizamos una variable intermedia llamada `suma`, donde almacenamos la suma de `n1` y `n2`. A continuación, imprimimos la variable `suma`. Por cierto, aquí las variables han sido definidas con **let**, como comentábamos anteriormente.

Las operaciones más básicas que podemos hacer son:

- + Suma (ejemplo: 2+3 es 5)
- - Resta (ejemplo: 5-3 es 2)
- \* Multiplicación (ejemplo: 2\*3 es 6)
- / División (ejemplo: 3/2 es 1.5)

**Práctica 2 :** Realizar una página web que pida dos números y muestre en pantalla su suma, su resta y su producto.

**Práctica 3:** Realizar una web que pida un número e imprima su tabla de multiplicar con el formato: número x 1 = resultado. La tabla de multiplicar deberá estar dentro de una tabla.