

cipsa.net



Curso de programación web

Javascript

¿Qué es JavaScript?



JavaScript es un lenguaje de programación que se utiliza para crear páginas dinámicas.

Una página web dinámica es aquella que incorpora efectos como texto que aparece o desaparece, animaciones, acciones que se activan al pulsar botones o mensajes de aviso al usuario.

Características

- Técnicamente, JavaScript es un **lenguaje de programación interpretado**, por lo que no es necesario compilar los programas para ejecutarlos.
- En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

Origen de JavaScript



JavaScript fue desarrollado originalmente por **Brendan Eich** de Netscape con el nombre de **Mocha**, y fue renombrado posteriormente a **LiveScript**, para finalmente quedar como **JavaScript**.

Origen de JavaScript (2)

El cambio de nombre coincidió aproximadamente con el momento en que Netscape agregó soporte para la tecnología Java en su navegador web Netscape Navigator en la versión 2.002 en diciembre de 1995.

La denominación produjo confusión, dando la impresión de que el lenguaje es una prolongación de Java, y se ha caracterizado por muchos como una estrategia de mercadotecnia de Netscape para obtener prestigio e innovar en lo que eran los nuevos lenguajes de programación web.

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java.

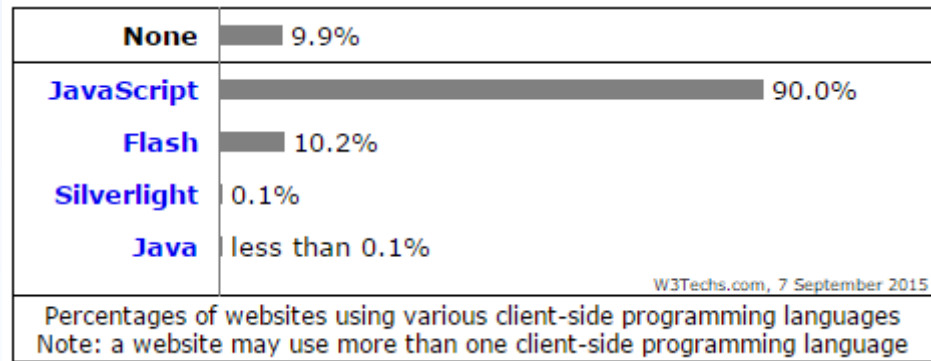
Origen de JavaScript (3)



“La creación del contenido no debería ser una tarea recóndita. No debería ser esto algo inescrutablemente arcano que sólo los expertos y dorados gurus de las ciencias de la computación pueden hacer.”

Brendan Eich Innovadores de la red.
(Innovators of the Net)

Estadísticas de uso



Client-side Programming Languages

Most popular client-side programming languages

	usage	change since 1 August 2015
1. JavaScript	90.0%	+0.2%
2. Flash	10.2%	-0.2%
3. Silverlight	0.1%	

percentages of sites

Cómo incluir JavaScript en documentos HTML

La integración de JavaScript y HTML es muy flexible, ya que existen al menos **tres** formas para incluir código JavaScript en las páginas web:

1. Incluir JavaScript en los elementos HTML
2. Incluir JavaScript en el mismo documento.
3. Definir JavaScript en un archivo externo.

Incluir JavaScript en los elementos HTML

Este último método es el menos utilizado (y NO RECOMENDADO), ya que consiste en incluir trozos de JavaScript dentro del código HTML5 de la página:

```
<!DOCTYPE html >
```

```
<html >
```

```
  <head>
```

```
    <title>Ejemplo de código JavaScript en el propio documento</title>
```

```
  </head>
```

```
  <body>
```

```
    <p onclick="alert('Un mensaje de prueba');">Un párrafo de texto.</p>
```

```
  </body>
```

```
</html>
```

Incluir JavaScript en el mismo documento.

El código JavaScript se encierra entre etiquetas `<script>` y se incluye en cualquier parte del documento.

Aunque es correcto incluir cualquier bloque de código en cualquier zona de la página, se recomienda definir el código JavaScript justo antes del cierre de la etiqueta `<body>`. Anteriormente, era común su inclusión dentro de la cabecera del documento (dentro de la etiqueta `<head>`):

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ejemplo de código JavaScript en el propio documento</title>
  </head>
  <body>
    <h1>Ejemplo de Javascript</h1>
    <p>Un párrafo de texto.</p>
    <script>
      alert("Un mensaje de prueba");
    </script>
  </body>
</html>
```

Definir JavaScript en un archivo externo

Las instrucciones JavaScript se pueden incluir en un archivo externo de tipo JavaScript que los documentos HTML enlazan mediante la etiqueta `<script>`. Se pueden crear todos los archivos JavaScript que sean necesarios y cada documento HTML puede enlazar tantos archivos JavaScript como necesite.

```
<!DOCTYPE html >
<html>
  <head>
    <title>Ejemplo de código JavaScript en el propio documento</title>
  </head>
  <body>
    <h1>Ejemplo de código Javascript en archivo Externo</h1>
    <p>Un párrafo de texto.</p>
    <script src="js/codigo.js"></script>
  </body>
</html>
```

Y en un archivo separado (js/codigo.js)

```
alert("Un mensaje de prueba");
```

Glosario básico

- **Script:** cada uno de los programas, aplicaciones o trozos de código creados con el lenguaje de programación JavaScript. Unas pocas líneas de código forman un script y un archivo de miles de líneas de JavaScript también se considera un script.
- **Sentencia:** cada una de las instrucciones que forman un script.
- **Palabras reservadas:** son las palabras (en inglés) que se utilizan para construir las sentencias de JavaScript y que por tanto no pueden ser utilizadas libremente. Las palabras actualmente reservadas por JavaScript son: **break, case, catch, continue, default, delete, do, else, finally, for, function, if, in, instanceof, new, return, switch, this, throw, try, typeof, var, void, while, with.**

Sintaxis

- No se tienen en cuenta los espacios en blanco y las nuevas líneas.
- Se distinguen las mayúsculas y minúsculas.
- **No se define el tipo de las variables:** al crear una variable, no es necesario indicar el tipo de dato que almacenará. De esta forma, una misma variable puede almacenar diferentes tipos de datos durante la ejecución del script.
- **No es necesario terminar cada sentencia con el carácter de punto y coma (;)** En la práctica, aunque JavaScript no obliga a hacerlo, es conveniente seguir la tradición de terminar cada sentencia con el carácter del punto y coma y habitualmente se utiliza .
- **Se pueden incluir comentarios:** los comentarios se utilizan para añadir información en el código fuente del programa. // en una línea o /* ... */ cuando ocupa más de una línea.

Variables

Una variable es un elemento que se emplea para almacenar y hacer referencia a otro valor

En JavaScript no es obligatorio inicializar las variables, ya que se pueden declarar por una parte y asignarles un valor posteriormente.

```
var numero;  
numero = 3;
```

O directamente:

```
var numero=3;
```

Variables no declaradas

Una de las características más sorprendentes de JavaScript para los programadores habituados a otros lenguajes de programación es que tampoco es necesario declarar las variables.

Por ejemplo:

```
var numero_1 = 3;  
var numero_2 = 1;  
resultado = numero_1 + numero_2;
```

(A destacar que resultado no se ha definido previamente)

Nombre de variables

El nombre de una variable también se conoce como **identificador** y debe cumplir las siguientes normas:

- Sólo puede estar formado por letras, números y los símbolos \$ (dólar) y _ (guión bajo).
- El primer carácter no puede ser un número.

Tipos de variables - Numéricas

► Numéricas

Se utilizan para almacenar valores numéricos enteros (llamados *integer* en inglés) o decimales (llamados *float* en inglés).

En este caso, el valor se asigna indicando directamente el número entero o decimal.

Los números decimales utilizan el carácter . (punto) en vez de , (coma) para separar la parte entera y la parte decimal:

```
var iva = 16; // variable tipo entero  
var total = 234.65; // variable tipo decimal
```

Tipos de variables - cadena

► Cadenas de texto

Se utilizan para almacenar caracteres, palabras y/o frases de texto.

Para asignar el valor a la variable, se encierra el valor entre comillas dobles o simples, para delimitar su comienzo y su final:

```
var mensaje = "Bienvenido a nuestro sitio web";  
var nombreProducto = 'Producto ABC';  
var letraSeleccionada = 'c';
```


Mecanismo de escape

Si se quiere incluir...	Se debe incluir...
Una nueva línea	\n
Un tabulador	\t
Una comilla simple	\'
Una comilla doble	\"
Una barra inclinada	\\

Arrays

Un array es una colección de variables, que pueden ser todas del mismo tipo o cada una de un tipo diferente.

P.ej.

```
var dias = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo"];
```

Para acceder a un elemento:

```
var diaSeleccionado = dias[0]; // diaSeleccionado = "Lunes"
```

Tipo de variable - booleano

Una variable de tipo *boolean* almacena un tipo especial de valor que solamente puede tomar dos valores: `true` (verdadero) o `false` (falso).

```
var clienteRegistrado = false;
```

```
var ivaIncluido = true;
```

Operadores

Los operadores permiten manipular el valor de las variables, realizar operaciones matemáticas con sus valores y comparar diferentes variables.

De esta forma, los operadores permiten a los programas realizar cálculos complejos y tomar decisiones lógicas en función de comparaciones y otros tipos de condiciones.

Operadores de asignación

El operador de asignación es el más utilizado y el más sencillo.

Este operador se utiliza para guardar un valor específico en una variable.

El símbolo utilizado es = (no confundir con el operador == que se verá más adelante):

```
var numero1 = 3;
```


Operador incremento y decremento

Estos dos operadores solamente son válidos para las variables numéricas y se utilizan para incrementar o decrementar en una unidad el valor de una variable.

Ejemplo:

```
var numero = 5;  
++numero;  
alert(numero); // numero = 6
```

(Ex: http://www.w3schools.com/js/tryit.asp?filename=tryjs_oper_incr)

Operadores lógicos - negación

Se utiliza para obtener el valor contrario al valor de la variable:

```
var cantidad = 0;  
vacio = !cantidad; // vacio = true
```

```
cantidad = 2;  
vacio = !cantidad; // vacio = false
```

```
var mensaje = "";  
mensajeVacio = !mensaje; // mensajeVacio = true
```

```
mensaje = "Bienvenido";  
mensajeVacio = !mensaje; // mensajeVacio = false
```

Operadores lógicos - AND

La operación lógica AND obtiene su resultado combinando dos valores booleanos.

El operador se indica mediante el símbolo `&&` y su resultado solamente es `true` si los dos operandos son `true`:

```
valor1 = true;
```

```
valor2 = true;
```

```
resultado = valor1 && valor2; // resultado = true
```

Operadores lógicos - OR

La operación lógica OR también combina dos valores booleanos.

El operador se indica mediante el símbolo `||` y su resultado es `true` si alguno de los dos operandos es `true`:

```
var valor1 = true;
```

```
var valor2 = false;
```

```
resultado = valor1 || valor2; // resultado = true
```

Operadores matemáticos

JavaScript permite realizar manipulaciones matemáticas sobre el valor de las variables numéricas. Los operadores definidos son: suma (+), resta (-), multiplicación (*) y división (/).

Ejemplo:

```
var numero1 = 10;
```

```
var numero2 = 5;
```

```
resultado = numero1 / numero2; // resultado = 2
```

```
resultado = 3 + numero1; // resultado = 13
```

```
resultado = numero2 - 4; // resultado = 1
```

```
resultado = numero1 * numero 2; // resultado = 50
```


Operadores relacionales

Los operadores relacionales definidos por JavaScript son idénticos a los que definen las matemáticas:

mayor que ($>$)

menor que ($<$)

mayor o igual ($>=$)

menor o igual ($<=$)

igual que ($==$)

distinto de ($!=$)

Estructuras de control de flujo - if

Se emplea para tomar decisiones en función de una condición. Su definición formal es:

```
if(condicion) { ... } else { ... }
```

Ejemplo:

```
var edad = 18;
```

```
if (edad >= 18) {  
    alert("Eres mayor de edad");  
} else {  
    alert("Todavía eres menor de edad");  
}
```