¿Por qué estudiar JavaScript?

JavaScript es uno de los **tres lenguajes que** todos los desarrolladores web **deben** aprender:

1. **HTML** para definir el contenido de las páginas web
2. **CSS** para especificar el diseño de las páginas web
3. **JavaScript** para programar el comportamiento de las páginas web

Preguntas frecuentes

- ¿Cómo obtengo JavaScript?
- ¿Dónde puedo descargar JavaScript?
- ¿Es JavaScript gratuito?

No es necesario obtener ni descargar JavaScript.

JavaScript ya se está ejecutando en su navegador, en su computadora, en su tableta y en su teléfono inteligente.

JavaScript es de uso gratuito para todos.

¿Qué es JavaScript?

JavaScript es el lenguaje de programación de la web.

Puede actualizar y cambiar tanto HTML como CSS.

Puede calcular, manipular y validar datos.

¿Por qué estudiar JavaScript?

JavaScript es uno de los tres lenguajes que todos los desarrolladores web deben aprender:

1. HTML para definir el contenido de las páginas web
2. CSS para especificar el diseño de las páginas web
3. JavaScript para programar el comportamiento de las páginas web

¿Sabías?

JavaScript y Java son lenguajes completamente diferentes, tanto en concepto como en diseño.

JavaScript fue inventado por Brendan Eich en 1995 y se convirtió en un estándar ECMA en 1997.

ECMA-262 es el nombre oficial del estándar. ECMAScript es el nombre oficial del lenguaje.

La etiqueta <script>

En HTML, el código JavaScript se inserta entre las etiquetas <script>y </script>.

JavaScript en <head> o <body>

Puede colocar cualquier cantidad de scripts en un documento HTML.

Los scripts se pueden colocar en la sección <body>, en la <head>sección de una página HTML o en ambas.

JavaScript externo

Los scripts también se pueden colocar en archivos externos:

Los scripts externos son prácticos cuando se utiliza el mismo código en muchas páginas web diferentes.

Los archivos JavaScript tienen la extensión de archivo **.js** .

Para utilizar un script externo, coloque el nombre del archivo de script en el srcatributo (fuente) de una <script>etiqueta:

Puedes colocar una referencia de script externa en <head>o <body>como desees.

El script se comportará como si estuviera ubicado exactamente donde <script>se encuentra la etiqueta.

Los scripts externos no pueden contener <script>etiquetas.

Ventajas de JavaScript externo

Colocar scripts en archivos externos tiene algunas ventajas:

- Separa HTML y código
- Hace que HTML y JavaScript sean más fáciles de leer y mantener.

- Los archivos JavaScript almacenados en caché pueden acelerar la carga de las páginas

Para agregar varios archivos de script a una página, utilice varias etiquetas de script:

Ejemplo

```
<script src="myScript1.js"></script>
<script src="myScript2.js"></script>
```

Referencias externas

Se puede hacer referencia a un script externo de tres maneras diferentes:

- Con una URL completa (una dirección web completa)
- Con una ruta de archivo (como /js/)
- Sin ningún camino

Este ejemplo utiliza una **ruta de archivo** para vincular a myScript.js:

Ejemplo

```
<script src="/js/myScript.js"></script>
```

Este ejemplo no utiliza ninguna ruta para vincularse a myScript.js:

Ejemplo

```
<script src="myScript.js"></script>
```

Posibilidades de visualización de JavaScript

JavaScript puede "mostrar" datos de diferentes maneras:

- Al escribir en un elemento HTML, utilice innerHTMLo innerText.
- Escribiendo en la salida HTML usando document.write().
- Escribiendo en un cuadro de alerta, utilizando window.alert().
- Escribiendo en la consola del navegador, usando console.log().

Uso de innerHTML

Para acceder a un elemento HTML, puede utilizar el document.getElementById(id)método.

Utilice el idatributo para identificar el elemento HTML.

Luego use la innerHTMLpropiedad para cambiar el contenido HTML del elemento HTML:

Ejemplo

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My First Paragraph</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "<h2>Hello World</h2>";
</script>

</body>
</html>
```

Usando texto interno

Para acceder a un elemento HTML, utilice el document.getElementById(id)método.

Luego use la innerTextpropiedad para cambiar el texto interno del elemento HTML:

Ejemplo

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My First Paragraph</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerText = "Hello World";
</script>

</body>
</html>
```

Usando document.write()

Para fines de prueba, es conveniente utilizar document.write():

Ejemplo

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
document.write(5 + 6);
</script>

</body>
</html>
```

Usando window.alert()

Puede utilizar un cuadro de alerta para mostrar datos:

Ejemplo

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
window.alert(5 + 6);
</script>

</body>
</html>
```

Puedes omitir la windowpalabra clave.

En JavaScript, el objeto ventana es el objeto de ámbito global. Esto significa que las variables, propiedades y métodos pertenecen al objeto ventana por defecto. Esto también significa que especificar la windowpalabra clave es opcional:

Ejemplo

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
alert(5 + 6);
</script>

</body>
</html>
```

Usando console.log()

Para fines de depuración, puede llamar al console.log()método en el navegador para mostrar datos.

Ejemplo

```
<!DOCTYPE html>
<html>
<body>

<script>
console.log(5 + 6);
</script>

</body>
</html>
```

Declaraciones de JavaScript

Ejemplo

```
let x, y, z;     // Statement 1
x = 5;           // Statement 2
y = 6;           // Statement 3
z = x + y;       // Statement 4
```

Programas JavaScript

Un **programa de computadora** es una lista de "instrucciones" que deben ser "ejecutadas" por una computadora.

En un lenguaje de programación, estas instrucciones de programación se denominan **declaraciones** .

Un **programa JavaScript** es una lista de **declaraciones** de programación .

En HTML, los programas JavaScript son ejecutados por el navegador web.

Punto y coma ;

Los puntos y coma separan las declaraciones de JavaScript.

Agregue un punto y coma al final de cada declaración ejecutable:

Ejemplos

```
let a, b, c;
a = 5;
b = 6;
c = a + b;
```

Palabras clave de JavaScript

Las declaraciones de JavaScript a menudo comienzan con una **palabra clave** para identificar la acción de JavaScript que se debe realizar.

Nuestra referencia de palabras reservadas enumera todas las palabras clave de JavaScript.

Aquí hay una lista de algunas de las palabras clave que aprenderá en este tutorial:

Palabra clave	Descripción
var	Declara una variable
let	Declara una variable de bloque
const	Declara una constante de bloque
if	Marca un bloque de sentencias que se ejecutarán según una condición
switch	Marca un bloque de sentencias que se ejecutarán en diferentes casos
for	Marca un bloque de sentencias que se ejecutarán en un bucle
function	Declara una función
return	Sale de una función
try	Implementa el manejo de errores en un bloque de declaraciones

Las palabras clave de JavaScript son palabras reservadas. No pueden usarse como nombres de variables.

Sintaxis de JavaScript

La sintaxis de JavaScript es el conjunto de reglas sobre cómo se construyen los programas JavaScript:

```
var x;
let y;

x = 5;
y = 6;
let z = x + y;
```

Valores de JavaScript

La sintaxis de JavaScript define dos tipos de valores:

- Valores fijos
- Valores variables

Los valores fijos se denominan **literales** .

Los valores de las variables se denominan **Variables** .

Literales de JavaScript

Las dos reglas de sintaxis más importantes para valores fijos son:

1. **Los números** se escriben con o sin decimales:

```
10.50

1001
```

2. Las cadenas son texto, escrito entre comillas dobles o simples:

```
let x;
x = 6;
```

Operadores de JavaScript

JavaScript utiliza **operadores aritméticos** (+ - * /) para **calcular** valores:

```
(5 + 6) * 10
```

JavaScript utiliza un **operador de asignación** (=) para **asignar** valores a las variables:

```
let x, y;
x = 5;
y = 6;
```

Expresiones de JavaScript

Una expresión es una combinación de valores, variables y operadores, que se calcula como un valor.

El cálculo se llama evaluación.

Por ejemplo, 5 * 10 evalúa a 50:

```
5 * 10
```

Las expresiones también pueden contener valores variables:

```
x * 10
```

Los valores pueden ser de varios tipos, como números y cadenas.

Por ejemplo, "John" + " " + "Doe", se evalúa como "John Doe":

```
"John" + " " + "Doe"
```

Identificadores/nombres de JavaScript

Los identificadores son nombres de JavaScript.

Los identificadores se utilizan para nombrar variables, palabras clave y funciones.

Las reglas para los nombres legales son las mismas en la mayoría de los lenguajes de programación.

Un nombre de JavaScript debe comenzar con:

- Una letra (AZ o az)
- Un signo de dólar ($)
- O un guión bajo (_)

Los caracteres posteriores pueden ser letras, dígitos, guiones bajos o signos de dólar.

Nota:

No se permiten números como primer carácter en los nombres.

De esta manera, JavaScript puede distinguir fácilmente los identificadores de los números.

JavaScript distingue entre mayúsculas y minúsculas

Todos los identificadores de JavaScript distinguen **entre mayúsculas y minúsculas** .

Las variables lastNamey lastname, son dos variables diferentes:

```
let lastname, lastName;
lastName = "Doe";
lastname = "Peterson";
```

JavaScript no interpreta **LET** o **Let** como la palabra clave **let** .

Variables de JavaScript

Las variables son contenedores para almacenar datos

Las variables de JavaScript se pueden declarar de cuatro maneras:

- Automáticamente
- Usandovar
- Usandolet
- Usandoconst

En este primer ejemplo, x, y, y z son variables no declaradas.

Se declaran automáticamente cuando se utilizan por primera vez:

Ejemplo

```
x = 5;
y = 6;
z = x + y;
```

Nota

Se considera una buena práctica de programación declarar siempre las variables antes de usarlas.

De los ejemplos se puede deducir:

- x almacena el valor 5
- y almacena el valor 6
- z almacena el valor 11

Ejemplo usando var

```
var x = 5;
var y = 6;
var z = x + y;
```

Nota

La palabra clave var se utilizó en todo el código JavaScript desde 1995 hasta 2015.

Las palabras clave le ty const se agregaron a JavaScript en 2015.

La palabra clave var sólo debe utilizarse en código escrito para navegadores antiguos.

Ejemplo usando const

```
const x = 5;
const y = 6;
const z = x + y;
```

Identificadores de JavaScript

Todas **las variables** de JavaScript deben **identificarse** con **nombres únicos** .

Estos nombres únicos se llaman **identificadores** .

Los identificadores pueden ser nombres cortos (como x e y) o nombres más descriptivos (edad, suma, volumen total).

Las reglas generales para construir nombres para variables (identificadores únicos) son:

- Los nombres pueden contener letras, dígitos, guiones bajos y signos de dólar.
- Los nombres deben comenzar con una letra.
- Los nombres también pueden comenzar con $ y _ (pero no los usaremos en este tutorial).
- Los nombres distinguen entre mayúsculas y minúsculas (y e Y son variables diferentes).
- Las palabras reservadas (como las palabras clave de JavaScript) no se pueden utilizar como nombres.