

# **HABLA** CODE

**Manual para configurar JavaScript con HTML**



**Carlos Vasquez**

# TABLA DE CONTENIDO

[Prefacio](#)

[Editores de código gratuitos](#)

[Introducción a JavaScript](#)

[Cómo usar JavaScript con HTML](#)

# ¡Gracias!

Antes de comenzar, me gustaría agradecer a las siguientes empresas por asociarse con hablacode y creer en nuestra misión de crear recursos de tecnología gratuitos.



# Prefacio

¡Bienvenidos!

Este manual es principalmente para explicar cómo configurar JavaScript con HTML, la mayor parte de la información se creará en videos, ya que JavaScript tiene mucho material para explicar.

A lo largo del tiempo, actualizaré el manual y agregaré más información relacionada con JS. Compartiré otros recursos que le ayudarán a lo largo de su viaje para aprender desarrollo front-end.

Nuestro sitio web para ver y tratar ejemplos de JS:

[https://hablacode.org/javascript\\_en\\_español/lección/introducción\\_a\\_javascript.html](https://hablacode.org/javascript_en_español/lección/introducción_a_javascript.html)

Canal de YouTube

<https://youtu.be/q1Y5ipF85tA>

## Editores de código gratuitos

Si desea seguir adelante, descargue uno de los siguientes editores (gratis)

- [Visual Studio Code](#)
- [Sublime Text](#)

# Introducción a JavaScript

A lo largo de este curso, me referiré a JavaScript como JS o JavaScript, no se confundan, significa lo mismo.

A medida que avances en tu viaje de desarrollo front-end, comienzas aprendiendo HTML y CSS. Te das cuenta de que tienes un website que se mira bien, tiene texto, botones, y enlaces que te llevan a otras páginas. Dependiendo en tu nivel de CSS, la página que acabas de crear tendrá color, cajas de texto que tendrá relleno alrededor y tal vez texto alineado en el centro para el encabezado(heading).

Puedo seguir describiendo el sitio web estático que acabas de crear. Palabra clave "**static/estático**".

## **¿Qué significa un sitio web estático?**

Un sitio web estático contiene páginas web con contenido fijo. Cada página está codificada en HTML y muestra la misma información a todos los visitantes. Un ejemplo sería si un sitio web estático tiene 2 páginas y cada página tiene contenido idéntico (footer/pie de página), ambos contienen dos versiones. Por lo tanto, si actualizas el pie de página, debes hacerlo dos veces, una en cada página.

Los sitios estáticos son el tipo de sitio web más básico y son los más fáciles de crear.

## **¿Por qué estoy hablando de sitios web estáticos?**

Porque eso es lo que has creado hasta ahora si acabas de comenzar con HTML y CSS. Eso está completamente bien y esa es la forma de comenzar tu carrera en el desarrollo de front-end. Ahora es el momento de pasar de crear sitios web estáticos a sitios web **dinámicos**.

## ¿Qué es un sitio web dinámico?

Una página web dinámica es una página web que muestra contenido diferente cada vez que se visualiza. Por ejemplo, la página puede cambiar con la hora del día, el usuario que accede a la página web o el tipo de interacción del usuario.

Aquí es donde entra **JavaScript**. **JavaScript** puede crear sitios web dinámicos con solo HTML y CSS, te permitirá presentar información y reaccionar a la entrada del usuario (como cálculos o elección de lo que presentará la página siguiente).

Lo de arriba fue solo un desglose simple de sitios web estáticos y dinámicos. Todavía queda mucho por aprender.

Basta de hablar, ¡es hora de comenzar a aprender JavaScript!

# Herramientas Necesarias

Como sugerí de uno de los dos editores de codificación gratuitos, para comenzar a escribir JavaScript no es necesario, pero tener una herramienta configurada lo hará más fácil a medida que avanza y aprendas más sobre el desarrollo front-end.

Windows: [video tutorial para descargar vscode o texto sublime](#)

Enlace para descargar vscode: <https://code.visualstudio.com/download>

También necesitarás un navegador web: Sugiero [Google Chrome](#), pero siéntase libre de elegir cualquier navegador moderno de su elección.

Una vez que tengamos lo anterior instalado, estamos listos para continuar.

---

# ¿Cómo usar JavaScript con HTML?

Como mencionamos anteriormente, JavaScript se utiliza para controlar el comportamiento de diferentes elementos, (páginas dinámicas) o aplicaciones web interactivas.

JavaScript te permitiera escribir scripts, un script es una pieza de programa que puedes agregar interactividad a tu sitio web. Por ejemplo, una secuencia de comandos podría generar un mensaje de alerta emergente o proporcionar un menú desplegable. Este script podría escribirse usando JavaScript.

Hoy en día, la mayoría de los desarrolladores web utilizan JavaScript y los frameworks asociados como React.js, Angular.js y Vue.js para crear interfaces interactivas.

Puedes mantener el código JavaScript en un archivo separado y luego incluirlo donde sea necesario, o puede definir funciones dentro del propio documento HTML. Hay 2 formas de agregar javascript a tu documento HTML.

## JavaScript Interno

Debajo tenemos un archivo HTML que contiene código javascript, estamos escribiendo nuestro código javascript dentro de las etiquetas del `<script> </script>`.

## Index.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Tutorial de JavaScript</title>
</head>
<body>
  Se mostrará una ventana emergente, gracias a JavaScript: <br/>
  <script>
    alert("Hola Mundo - Interno JavaScript");
  </script>
</body>
</html>
```

Si ejecutas el código de arriba, recibirás una alerta con el siguiente mensaje: **"Se mostrará una ventana emergente, gracias a JavaScript:"**

Si cierra el mensaje y vuelve a cargar la página, la alerta volverá a aparecer, ya que el javascript se ejecutará al volver a cargar

## JavaScript Externo

La segunda forma de agregar código JavaScript a tu archivo HTML es utilizando también la etiqueta `<script>`, pero en lugar de escribir nuestro código javascript dentro de la etiqueta `<script>`, crearemos un archivo con la extensión **.js**, que es la extensión para crear archivos javascript. Para usar scripts de comandos externos, coloque el nombre del archivo de javascript en el atributo **src (fuente)** de la etiqueta `<script>`.



## Ejemplo:

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

Colocaremos las etiquetas del script como hicimos anteriormente al final de nuestras etiquetas `<body>`, Hacemos esto porque el navegador lee nuestros archivos html de arriba a abajo. Si tenemos todos nuestros archivos CSS y JavaScript externos dentro de nuestra etiqueta `<head>`, tardará un par de segundos o minutos en cargar nuestro sitio web.

Eso es malo, ya que el usuario verá una página en blanco esperando que se cargue el contenido. Si colocamos nuestros scripts al final de nuestras etiquetas `<body></body>`, disminuirémos el tiempo que tarda nuestra página en cargarse, además, los archivos javascript tienden a ser bastante grandes, por eso es una buena práctica cargarlos al final.

## Ejemplo:

### Index.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Tutorial de JavaScript</title>
</head>
```

```
<body>  
  Se mostrará una ventana emergente, gracias a JavaScript: <br/>  
  <script src="app.js"></script>  
</body>  
</html>
```

### **app.js**

```
alert("Hola Mundo - Interno JavaScript");
```

El código de arriba se ejecutará de la misma forma que el ejemplo interno de JavaScript.

JavaScript externo es mi opción, tener todo en un archivo es muy difícil de leer. Tener su código HTML en un archivo, estilo CSS en un archivo separado y javascript por sí solo es una buena práctica.

---

## **Donde Aprender Mas**

Como mencioné anteriormente, esto es solo una introducción a CSS. Para obtener más información, puedes visitar los siguientes sitios

- Nuestra pagina web:  
[https://hablancode.org/javascript\\_en\\_español/lección/introducción\\_a\\_javascript.html](https://hablancode.org/javascript_en_español/lección/introducción_a_javascript.html)
- Suscribete a nuestro canal de Youtube ya que subiremos más videos: [Hablancode.org](https://www.youtube.com/c/Hablancode)
- MDN Web Docs:  
<https://developer.mozilla.org/es/docs/Web/CSS>