

Portada:

-Nombre del estudiante: Kevin Bermudez Esquivel.

-Nombre del documento: Entrega semana 1.

-Nombre del curso: Fundamentos de programación web.

-Nombre del profesor del curso: Francisco Jose Jimenez Bonilla.

-Fecha de entrega: Semana 2.

-Periodo lectivo: Cuatrimestre 3 del 2023.

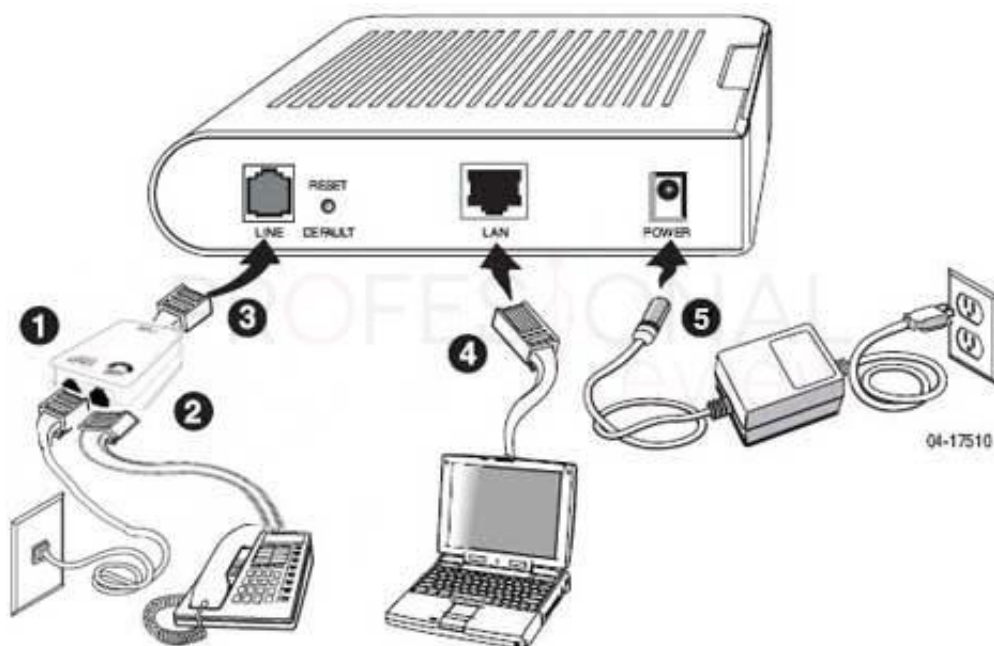


Desarrollo.

1. Historia del Lenguaje JavaScript.

La historia del lenguaje JavaScript aparece por allá de la década de los 90's, cuando conectarse a internet era toda una aventura que podías acceder con un módem de marcación telefónica que corría a 28.8k, no existía JS para su momento.

Luego de algunos años llega a nuestros hogares el módem de 56k, que parecía supersónico en comparación a su antecedente, pero que de igual manera, no se asemejaba a nuestros estándares modernos; de igual forma seguía siendo navegación sin JavaScript.



¿Cuándo aparece JavaScript?

En 1995, Brenda Eich, quien trabajaba para Netscape Communications (ahora Mozilla Corporation) ideó como solucionar las velocidades de navegación tan lentas donde aplicaciones que avanzaban sin parar, hacían ver que html se quedaba atrás. Dicha respuesta llegó en forma de un lenguaje que podía ejecutarse directamente en el navegador del lado del usuario.



Inicialmente al lenguaje se le llamó "LiveScript" y solo se ejecutaba en el navegador netscape, pero luego fue renombrado como JavaScript para capitalizar el renombre de Java. ¡Cabe destacar que JavaScript y Java son lenguajes distintos!

JavaScript se creó para poder realizar acciones de diversos grados de complejidad en sitios web sin necesidad de compilación, ósea, la necesidad de transformar un programa escrito en un lenguaje a otro formato de lenguaje.

JS permite interactuar con un mapa, incorporar animaciones y otros elementos gráficos dinámicos, JavaScript forma, junto con HTML y CSS, la trífeca del desarrollo web.

Algo que ha hecho de JS un lenguaje muy apreciado por equipos de desarrollo es el hecho de que los mismos navegadores lean y asimilen el código para efectuar las instrucciones indicadas por éste.

Básicamente, si lo que se busca es desarrollar una página web con un mayor nivel de dificultad de la que suele plantear a menudo, entonces JavaScript es el lenguaje en el que ésta se desarrollaría.

¿Qué empresa desarrolló JavaScript?

Increíblemente esta creación fue traída a realidad por dos competidores que formaron una alianza, Netscape y Sun Microsystems, todo para hacerle competencia al gigante del momento: Microsoft.

Netscape aportó el conocimiento y la experiencia en navegadores web, mientras que Sun Microsystems contribuyó con su experiencia en lenguajes de programación, especialmente con el lenguaje Java.

Luego de la corta rivalidad Microsoft lanza Internet Explorer 3.0, que incluía soporte para JavaScript, lo que impulsó aún más la accesibilidad de JS y por ende, su popularidad.



¿Por qué aprender JavaScript?

Es idóneo para aprender programación por varios motivos muy poderosos, siendo el primero de ellos su sencillez y facilidad de aprendizaje tanto para aquellas personas que ya saben programación como para quienes aterrizan por primera vez en el mundo del desarrollo web, 100% apto para principiantes.

Es un lenguaje que se utiliza en todas partes, es decir, se reconoce y se utiliza a nivel internacional. También se encuentra formando parte en muchos dispositivos (sino todos), como lo son apps móviles, navegadores, etc.

Cuenta con una alta demanda laboral en el mundo empresarial; por su practicidad, curva de aprendizaje, fama, por ser el más utilizado, JavaScript se convierte también en el lenguaje de programación que más buscan en las ofertas de trabajo.

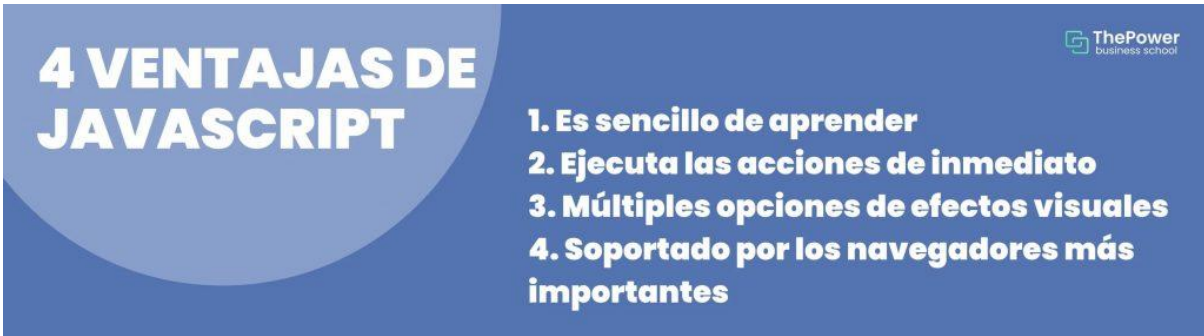
También te permite crear tus propios sitios web con gran nivel de funcionalidad e interacción con un diseño de interfaces dinámicas y atractivas para los usuarios, lo que es un factor clave para la interacción del contenido.



Gran soporte y comunidad. Entre las comunidades más reconocidas se encuentra Stack Overflow, con más de 100 millones de visitantes mensuales. Esta inmensa comunidad ofrece soporte, soluciones para codificar problemas, ayuda para nuevas herramientas o complementos, además de una gran variedad de mentores, blogs y artículos sobre JavaScript.

Como ya hemos comentado antes, JavaScript lleva 10 años consecutivos siendo el lenguaje de programación más utilizado en el mundo y en gran medida se debe a su constante evolución. Siempre hay cosas nuevas que hacer y qué aprender.

El hecho de que se diga que JavaScript es un lenguaje fácil de aprender no significa que sea simple o que tenga poco que aportar. Todo lo contrario, es un lenguaje muy versátil con el que se pueden desarrollar desde aplicaciones web (frontend y backend) a aplicaciones móviles o aplicaciones de escritorio, hasta videojuegos, pasando por herramientas que exijan la gestión de bases de datos, e incluso automatizaciones de flujos de trabajo.



4 VENTAJAS DE JAVASCRIPT

- 1. Es sencillo de aprender**
- 2. Ejecuta las acciones de inmediato**
- 3. Múltiples opciones de efectos visuales**
- 4. Soportado por los navegadores más importantes**

ThePower
business school

Relación entre HTML y JavaScript.

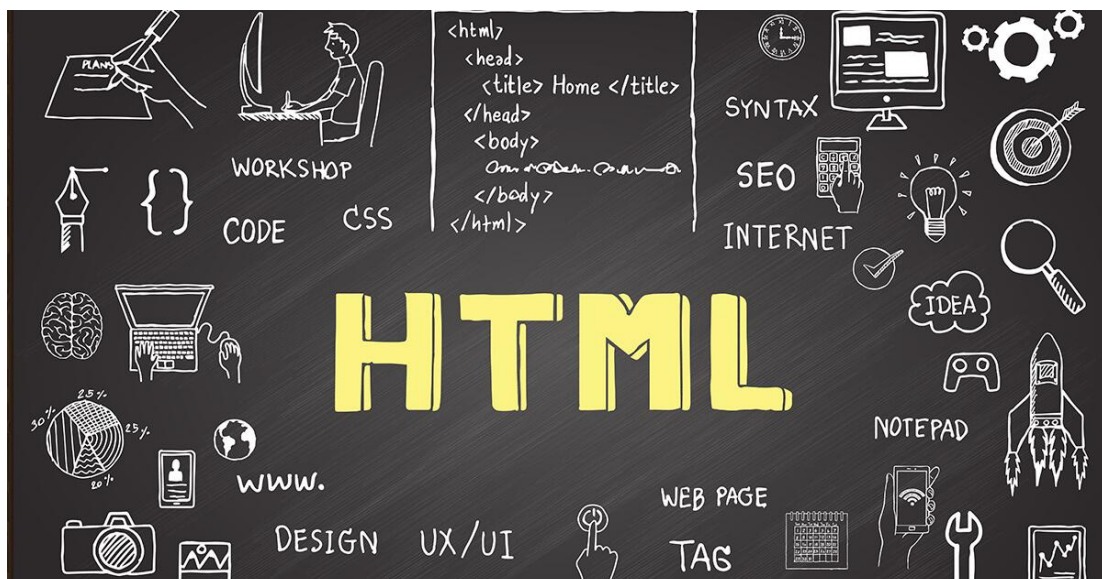
Ya anteriormente hemos aprendido qué es y cómo funciona JavaScript, así que para mayor claridad de la relación entre estas dos herramientas, vamos a dar a continuación una breve explicación sobre HTML.

¿Qué es html?

Es un lenguaje de marcado de hipertexto, utilizado normalmente en la www (world wide web). Fue creado en 1986 por el físico nuclear Tim Berners-Lee.

Este lenguaje es utilizado para estructurar y desplegar una página web y sus contenidos, por ejemplo párrafos, una lista con viñetas, imágenes o tablas de datos. Éste define la estructura de tu contenido

HTML consiste en una serie de elementos que usarás para encerrar diferentes partes del contenido para que se vean o comporten de una determinada manera.



Los dos lenguajes son parte integral de la creación de una página web funcional y visualmente atractiva. HTML proporciona el contenido y la estructura y JavaScript aporta elementos interactivos. Trabajan juntos para dar vida a un sitio web.

Sin embargo, es importante tener en cuenta que mientras HTML y CSS son necesarios para que un sitio web funcione, el JavaScript no siempre es necesario. Hay ciertos elementos que se pueden conseguir solo con HTML y CSS, como los efectos hover o el diseño responsivo. Todo depende de las necesidades y objetivos específicos del sitio web.

HTML y Javascript son lenguajes de programación que solos funcionan bien, pero juntos, juntos pueden darle a un proyecto otro nivel.



Básicamente, la relación entre estos lenguajes se resume de la siguiente manera:

HTML aporta la estructura básica y dirección de una página web, desplegar sus contenidos, provee etiquetas para describir los tipos de contenidos.

Gracias a ello, el navegador podrá comprender el contenido enviado por el servidor y representarlo en pantalla.

Mientras que *JavaScript* aporta funcionalidad y dinamismo. Además del contenido estático, con JavaScript podremos Mostrar actualizaciones de contenido, vincular eventos dinámicos a elementos HTML (clic en botones, accesos a menús, filtros en formulario), almacenar datos en variables, usar funciones complementarias como gráficos o mapas mediante APIS de terceros.

De esta forma vemos cómo ambos elementos logran funcionar en armonía para dar mayor vida a una página web en sintonía entre sí cosa que por sí solos no se lograría el objetivo que en conjunto se obtiene.



Se adjunta un corto elemento visual explicativo sobre el presente tema:

📺 DIFERENCIAS entre HTML, CSS Y JAVASCRIPT 🌐 📄 | PDC

Beneficios de Bootstrap para sitios y aplicaciones web en JS.

¿Qué es Bootstrap?

En el mundo del diseño y desarrollo web, Bootstrap es un término que ha resonado en la comunidad desde su lanzamiento en 2011. Como un marco de diseño (framework) de frontend abierto y gratuito, Bootstrap ha revolucionado la manera de desarrollar sitios web.

Esta herramienta permite crear una interfaz web, a través de lenguajes de programación como HTML, CSS o JavaScript; los cuales permiten adaptar el contenido de la web a los diferentes tipos de pantallas (diseño responsive) y mejorando, por tanto, la experiencia de usuario. Son plantillas de diseño web muy intuitivas y con excelentes resultados.



Ventajas de usar Bootstrap:

Desarrollo rápido y eficiente.

Una de las ventajas más obvias de Bootstrap es la velocidad de desarrollo. Bootstrap viene con plantillas de diseño predefinidas que las personas dedicadas a maquetación y desarrollo pueden usar para crear un sitio web funcional de manera rápida.

Esto es especialmente útil para los proyectos que tienen plazos de entrega ajustados.

Diseño adaptado a diferentes dispositivos.

Otra gran ventaja de Bootstrap es su enfoque en el diseño responsivo. Con el auge de los dispositivos móviles, es esencial que un sitio web se vea bien en todas las plataformas. Bootstrap utiliza un sistema de cuadrícula fluida que se ajusta dinámicamente al tamaño de pantalla adecuado, ofreciendo una experiencia de usuario óptima en cualquier dispositivo.

Componentes reutilizables.

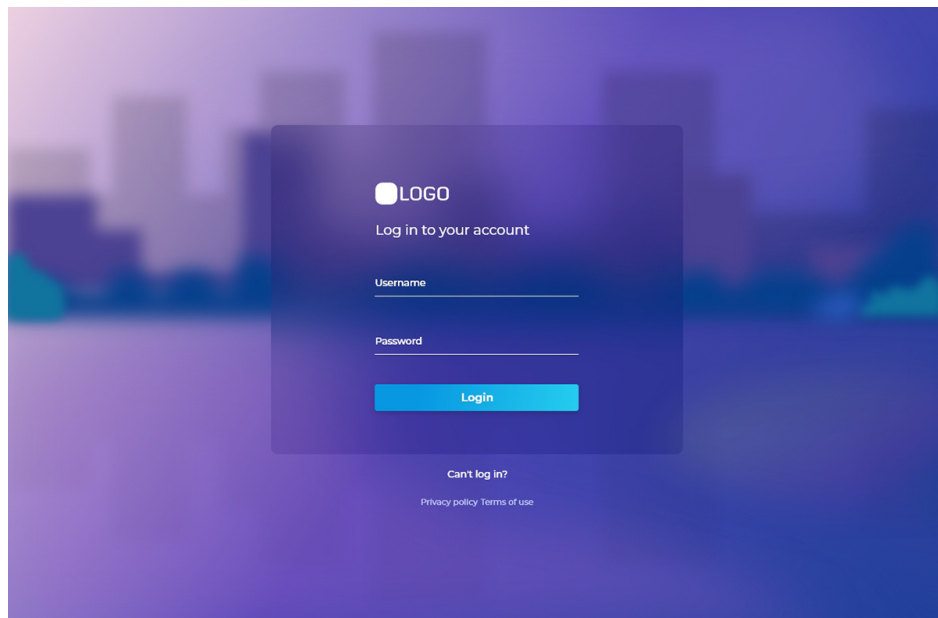
Bootstrap se distribuye con una serie de componentes predefinidos que tanto desarrolladores como diseñadores pueden usar. Estos incluyen botones, tarjetas, modales, desplegables, barras de navegación y mucho más. Estos componentes son reutilizables, lo que ahorra tiempo al diseñar y mantener la consistencia del diseño.

Compatibilidad con diversos navegadores.

Uno de los desafíos más grandes en el diseño web es hacer que los sitios web sean compatibles con todos los navegadores. Bootstrap es compatible con todos los navegadores modernos, lo que ahorra a los diseñadores mucho tiempo de prueba y depuración. Este factor es clave a la hora de pulir un sitio web.

Estilo uniforme y fácil de seguir.

Bootstrap proporciona un estilo uniforme por defecto para la mayoría de los elementos HTML. Esto puede ser una gran ventaja para los diseñadores ya que proporciona una base sólida y coherente sobre la cual trabajar. Esto no solo ahorra tiempo, sino que también asegura consistencia y coherencia en todos los elementos.

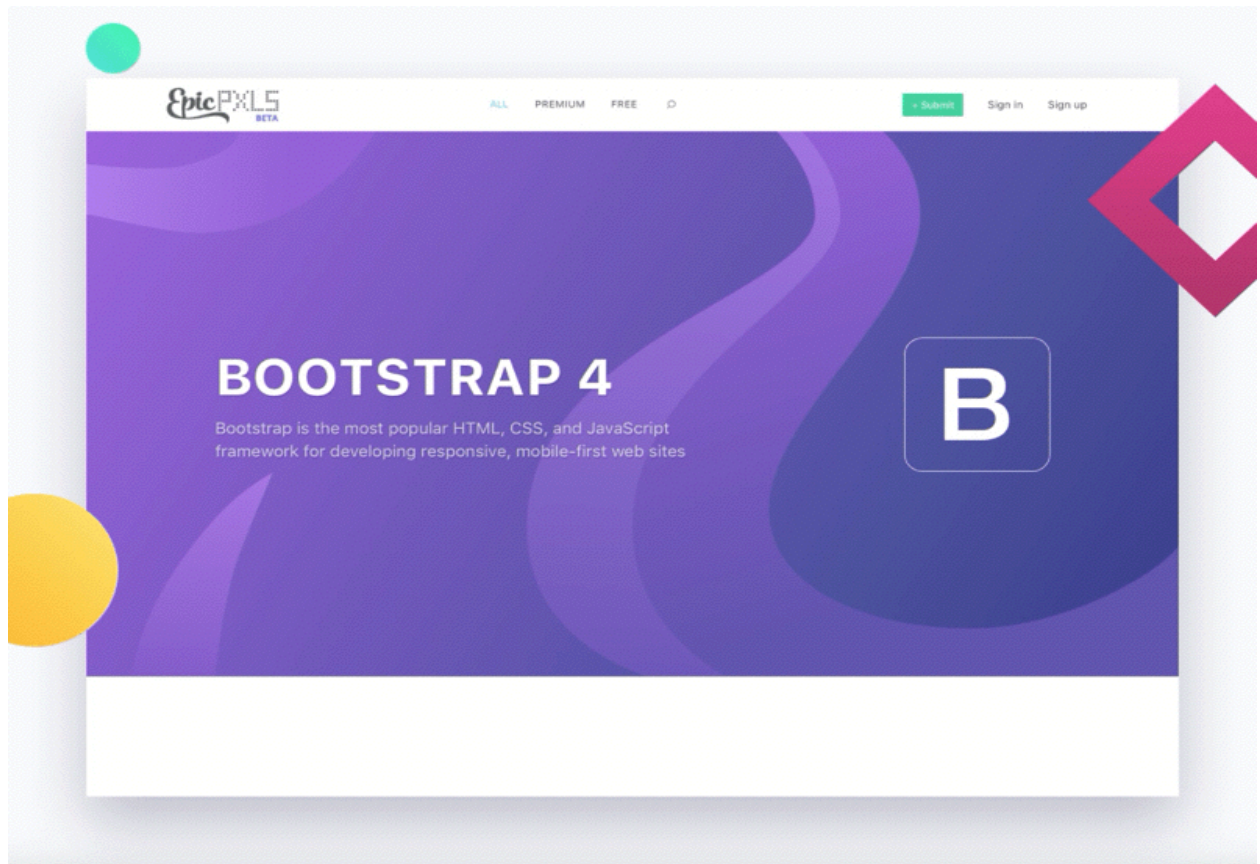


Bootstrap es sin duda una herramienta poderosa en el arsenal de maquetadores y desarrolladores Web. Sin embargo, como con cualquier herramienta, es importante entender cuándo y cómo usarla de manera efectiva.

Las ventajas de Bootstrap, como el desarrollo rápido, la consistencia, el diseño responsivo, la personalización y el amplio soporte, hacen que sea una opción atractiva para muchos desarrolladores.

Tengamos en cuenta que también está en nuestras manos usar este producto de manera creativa y eficiente para lograr sitios de gran impacto visual y muy buen rendimiento.

Al final, la decisión de usar o no Bootstrap dependerá de las necesidades y objetivos específicos de tu proyecto.



Semejanza y diferencia tienen los lenguajes web PHP y Javascript.

Primero que nada antes de empezar a comparar estos dos lenguajes, debemos preguntarnos, ¿qué es PHP?

Como anteriormente hemos hecho, vamos a dar una breve explicación de lo qué es PHP para tener un mayor entendimiento de sus semejanzas y diferencias.

PHP es un lenguaje de programación de código abierto del lado del servidor que se utiliza principalmente para crear páginas web dinámicas. La abreviatura nació originariamente de “Personal Home Page Tools”, aunque hoy en día se ha convertido en el acrónimo recursivo para “PHP: Hypertext Preprocessor”.

Mientras que los lenguajes del lado del cliente como HTML, CSS o JavaScript son interpretados primero por el navegador web en el momento de abrir una página, el código PHP se ejecuta en el servidor web. Allí, los scripts de PHP generan el código HTML que se envía después al navegador. Este no recibe el código real (el script de PHP), sino el resultado de la ejecución del mismo.



📺 ¿Qué es PHP? bien explicado

Comparando ambos lenguajes.

PHP y JavaScript son ambos lenguajes de scripting. Esto es opuesto a los lenguajes de programación “puros”. Los lenguajes de scripting tienden a ser interpretados más que compilados.

Esto significa que se traducen en código de máquina a través de un intermediario en lugar de hacerlo directamente. Esto tiene un impacto en su tiempo de ejecución.

Se puede pensar de esta manera: imagina que quieres traducir una página web de tu idioma nativo al Navajo. Si no conoces el idioma navajo necesitarías un intérprete para ayudarte, y la traducción tardaría más tiempo. Por eso los lenguajes compilados tienden a ser más rápidos que los interpretados.

Otro punto en común es que PHP y JavaScript están débilmente tipados. Esto significa que cuando se crea una variable en cualquiera de los dos lenguajes, no tienes que asignar su tipo de datos: se asume.

Otra cosa que PHP y JavaScript tienen en común es que los desarrolladores de ambos lenguajes tienen una gran demanda. Los desarrolladores que usan JavaScript y PHP también están bien pagados.



Diferencias de los lenguajes.

Posición de uso.

PHP es un lenguaje de scripting que se desarrolla solamente en el backend de una página web. Javascript fue creado como una solución para el frontend, aunque desde 2009 también dispone de entorno de tiempo de ejecución de backend. Es por eso por lo que JavaScript se denomina hoy en día como solución full-stack.

Lenguajes del lado del servidor (server-side) o del cliente (client-side).

PHP funciona en el servidor y se utiliza para ejecutar funciones del servidor. Por ejemplo, puede enviar contenido dinámico de una base de datos local o externa al usuario. A diferencia de PHP, Javascript es un lenguaje de scripting del lado del cliente. Esto significa que los scripts programados son ejecutados por el navegador de acceso. Esto puede llevar a accesos más rápidos

Velocidad.

En cuanto al rendimiento y a la velocidad, la comparativa de PHP vs. JavaScript se decanta por el lenguaje del lado del cliente. JavaScript muestra cierta ventaja en cuanto a la velocidad, mientras que PHP se va poniendo lentamente al día, pero de forma segura.



3 formas en que se puede agregar código JS en una página web.

Método 1:

Usar JavaScript en html desde un archivo externo.

Esta es la forma más común de ejecutar javascript en html, crea un archivo y enlazarlo a tu página utilizando la etiqueta script por ejemplo:

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

Método 2.

Incluir JavaScript en la etiqueta script

Puedes colocar esta etiqueta en cualquier parte de tu documento HTML, pero es más común colocarla al final del documento, justo antes de la etiqueta de cierre body.

Por ejemplo, si quieres insertar un mensaje de alerta en tu página web, puedes usar el siguiente código:

```
html
<script>
  alert("Hola mundo");
</script>
```

Método 3.

Incluir JavaScript en un atributo de evento

Otra forma de insertar JavaScript en HTML es incluyéndolo en un atributo de evento. Los atributos de evento son atributos especiales que se activan cuando sucede un evento determinado en la página web, como hacer clic en un botón o cargar la página.

Por ejemplo, si quieres mostrar un mensaje de alerta cuando el usuario haga clic en un botón, puedes usar el siguiente código:

html

```
<button onclick="alert('Hola mundo')">Haz clic aquí</button>
```

Función principal de la consola en JS.

La consola de JavaScript es una herramienta que se encuentra en la mayoría de los navegadores web y permite a los desarrolladores interactuar con el código JavaScript y depurar su aplicación.

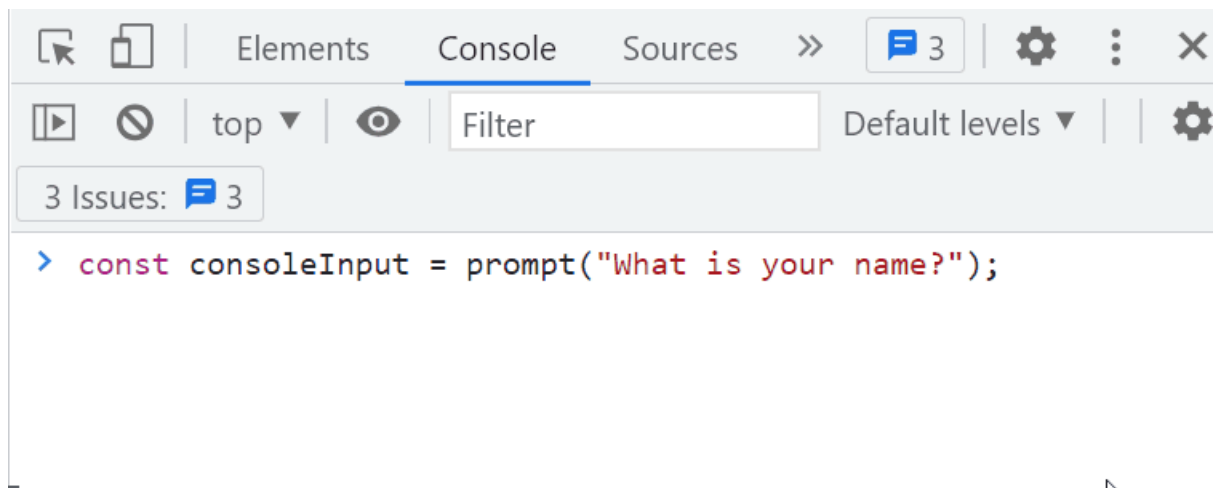
Una vez abierta, se puede escribir y ejecutar código JavaScript, así como inspeccionar elementos del DOM y ver mensajes de depuración.

La consola es esencial para los desarrolladores web y es una de las herramientas más utilizadas para el desarrollo de aplicaciones web.

Además de ejecutar código JavaScript, la consola también puede mostrar mensajes de depuración y errores que surgen durante la ejecución del código. Esto puede ser útil para identificar problemas y solucionarlos de manera efectiva. Los desarrolladores también pueden utilizar la consola para inspeccionar elementos del DOM y ver su estructura y contenido.

La consola también puede ser útil para probar pequeños fragmentos de código antes de incluirlos en una aplicación más grande. Los desarrolladores pueden experimentar con diferentes soluciones y ver de manera inmediata el resultado de sus acciones.

📺 Consola de JavaScript: `console.log()` | Curso JavaScript # 4



Diferencia entre declaraciones var, let y const en JS.

(Se subraya cada concepto para mayor entendimiento del estudiante y profesor).

En resumen, la declaración con var define una variable en el ámbito local actual y se hereda a scopes descendientes por referencia. Si la variable es declarada fuera de una función, la variable será una variable global.

Un último apunte sobre el ámbito de var, es que una variable declarada exclusivamente dentro de una función, no existe fuera de la misma, pues sería una variable local para esa función:

```
function foo() {  
    var i = "local";  
}  
console.log(i); // ReferenceError: i is not defined
```

Hoisting

Además del ámbito de aplicación visto antes, una variable declarada con var es sometida a hoisting (“izamiento” o “levantamiento”): la declaración de la variable es “levantada” hasta el inicio del ámbito de aplicación pero la asignación al valor permanece en el sitio donde se realice.

Let y const son dos formas de declarar variables en JavaScript introducidas en ES6 que reducen el ámbito de la variable a bloques (con var el ámbito era la función actual) y no admiten hoisting. Además, las variables declaradas con const no pueden ser reasignadas (aunque no significa que su valor sea inmutable, como veremos a continuación).

Let.

Un bloque en JavaScript se puede entender como «lo que queda entre dos corchetes», ya sean definiciones de funciones o bloques if, while, for y loops similares. Si una variable es declarada con let en el ámbito global o en el de una función, la variable pertenecerá al ámbito global o al ámbito de la función respectivamente, de forma similar a como ocurría con var.

```
let i = 0;
function foo() {
  i = 1;
  let j = 2;
  if(true) {
    console.log(i); // 1
    console.log(j); // 2
  }
}
foo();
```

Const.

El ámbito o scope para una variable declarada con const es, al igual que con let, el bloque, pero si la declaración con let previene la sobreescritura de variables, const directamente prohíbe la reasignación de valores (const viene de constant).


Resumen.

Var declara una variable de scope global o local para la función sin importar el ámbito de bloque. Permite hoisting.

Let declara una variable de scope global, local para la función o de bloque. Es reassignable y no permite hoisting.

Const declara una variable de scope global, local para la función o de bloque. No es reassignable, pero es mutable. No permite hoisting.





En general, let sería todo lo que se necesita dentro de un bloque, función o ámbito global. Const sería para variables que no van a sufrir una reasignación. Var se puede relegar para cuando necesitemos hacer hoisting, vamos, casi nunca.




Activa las notificaciones

VAR VS LET VS CONST

	var	let	const
Reasignación	✓	✓	✗
Redeclaración	✓	✗	✗
Propiedad del objeto global (window)	✓	✗	✗
Function Scope	✓	✓	✓
Block Scope	✗	✓	✓
Hoisting	✓	✓	✓
TDZ (Temporal Dead Zone)	✗	✓	✓

    /lupitacode



Tipos de comentarios en Javascript.

En Javascript existen dos tipos de comentarios: los comentarios de una sola línea y los comentarios de múltiples líneas.

Una línea: Comienza con `//` y sólo comenta la línea actual desde donde se escribe.

Múltiples líneas: Comentarios extensos. Comienzan por `/*` y comentará todo el texto que escribamos hasta que cerremos el comentario con un `*/`.

Veamos un ejemplo:

```
// Comentarios cortos de una sola línea. Suelen explicar la línea siguiente.  
let a = 1;  
  
let x = 45; // También se utilizan al final de una línea.  
  
/* Por otro lado, existen los comentarios múltiples de varias líneas consecutivas.  
   Suelen utilizarse para explicaciones largas que requieren bastante  
   espacio porque se mencionan gran cantidad de cosas :-) */
```

¿Qué es ECMAScript 6?

ES6 o ECMAScript 6 es un estándar de lenguaje de scripting desarrollado de acuerdo con las indicaciones contenidas en el documento ECMA-262 proporcionado por Ecma International (European Computer Manufacturers Association). Ecma international es una asociación fundada en 1961 que se ocupa de la estandarización en el sector de las tecnologías de la información y los sistemas de comunicación.

ECMAScript es, por lo tanto, un estándar creado con el objetivo de permitir la interoperabilidad de diferentes páginas web entre distintos navegadores. El estándar de lenguaje especifica las características que debe tener un lenguaje de scripting y cómo deben implementarse.



Ejemplos de lenguajes que se han implementado a partir de ECMAScript son Javascript, ActionScript y Jscript (Microsoft), ahora muy populares como lenguajes de scripting para el lado del cliente del desarrollo web.

ES y Javascript a lo largo de los años han lanzado sucesivas versiones de su producto, en concreto al analizar ES6 vemos como la sexta versión se ha marcado los siguientes objetivos

- Desarrollo de aplicaciones complejas
- Creación y gestión de bibliotecas mejoradas
- Mejor generación de código

¿Qué enseñanza me brindó la tarea?

Al finalizar esta tarea he logrado generar un conocimiento más amplio de lo qué es el lenguaje de programación Javascript, tanto parte de su historia como también sus usos e importancias.

He logrado aprender un poco sobre la relación que existe entre HTML y JS y el objetivo de cada uno de ellos.

Una breve investigación me ayudó a comprender un poco más a fondo lo qué es Bootstrap y los usos que se le pueden dar, en los que se puede aprovechar y sus ventajas en el mundo de la programación.

También cabe recalcar el repaso de las diferentes declaraciones, qué, a pesar de aún no estar 100% claras, ya tengo un entendimiento un poco más claro que anteriormente.



Referencias.

www.hackaboss.com/blog/historia-javascript

<https://www.hackaboss.com/blog/10-razones-para-aprender-javascript#:~:text=Es%20un%20lenguaje%20tan%20pr%C3%A1ctico,espa%C3%B1ol%20significa%20Producto%20M%C3%ADnimo%20Viable.>

<https://www.google.com/imghp>

https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/HTML_basics

<https://prezi.com/vovy6wyl2m8f/relacion-entre-html-y-javascript/>

<https://www.dongee.com/tutoriales/como-se-relaciona-html-css-y-javascript/>

<https://keepcoding.io/blog/trabajar-con-html-y-javascript/#:~:text=Mientras%20que%20el%20HTML%20se,o%20frontend%20de%20un%20programa.>

<https://immune.institute/blog/razones-usar-bootstrap-web/>

<https://damiandeluca.com.ar/ventajas-de-usar-bootstrap-en-la-creacion-de-sitios-web>

<https://kinsta.com/es/blog/php-vs-javascript/>

<https://oregoom.com/javascript/consola/>

