



Tarea #1

Cuestionario JavaScript

Cesia Montenegro

Universidad CENFOTEC

Fundamentos de Programación Web

Francisco José Jiménez Bonilla

08-09-2023

Cuestionario Java Script

1. ¿Escriba la historia del lenguaje Java Script?

Introducción

JavaScript fue creado en Netscape Communications por Brendan Eich en 1995. Netscape y Eich diseñaron JavaScript como un lenguaje de programación para usar con el navegador web insignia de la empresa, Netscape Navigator.



Figura 1

Brendan Eich

Born 4 July 1961 (age 62) Pittsburgh, Pennsylvania, US.

Inicialmente conocido como LiveScript, Netscape cambió el nombre a JavaScript para poder posicionarlo como un complemento del lenguaje **Java**, un producto de su socio, Sun Microsystems. Sin embargo, aparte de algunas similitudes sintácticas superficiales, JavaScript no tiene ninguna relación con el lenguaje de programación Java.



Figura 2

Imagen ilustrativa de algunas diferencias entre ambos lenguajes.

Después de su lanzamiento, cada vez más navegadores comenzaron a agregar soporte para JavaScript. Aun así, durante gran parte de su historia, JavaScript no fue considerado un lenguaje de programación serio. Sus primeras versiones sufrieron notables problemas de rendimiento y seguridad, pero los desarrolladores no tenían alternativas. Si querían ejecutar programas en el navegador, tenían que usar JavaScript.

En 2008, la creación del Chrome V8 de código abierto de Google, un motor JavaScript de alto rendimiento, supuso un punto de inflexión crucial para JavaScript. La posterior proliferación de motores JavaScript rápidos hizo posible que los desarrolladores crearan aplicaciones sofisticadas basadas en navegador con un rendimiento que competía con las aplicaciones móviles y de escritorio.

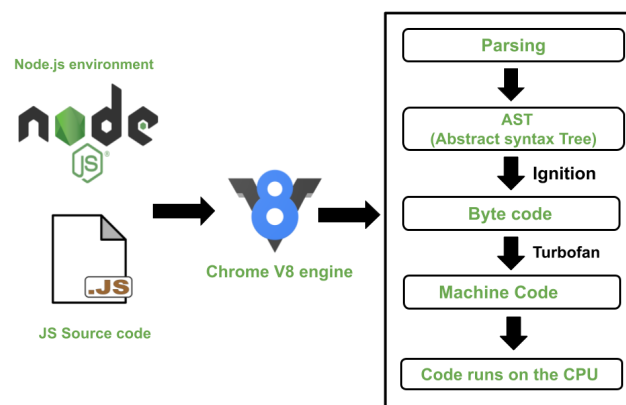


Figura 3

Explicación visual sobre los complementos entre las tecnologías node.js, JS y Chrome V8.

Poco después, Ryan Dahl lanzó un entorno multiplataforma de código abierto llamado Node.js. Proporcionó una forma de ejecutar código JavaScript desde fuera de un navegador. Liberó a JavaScript de los límites del navegador y condujo directamente a la popularidad actual de JavaScript. Hoy en día, puede utilizar JavaScript para escribir todo tipo de aplicaciones, incluidas aplicaciones de navegador, servidor, móviles y de escritorio. La mayoría de las principales empresas en línea de la actualidad, incluidas Facebook, Twitter, Netflix y Google, utilizan JavaScript en sus productos.



Figura 4

Imagen representativa sobre aplicaciones creada con JavaScript.

2. ¿Porqué se debe aprender JavaScript?

Más del 80% de los sitios web utilizan una biblioteca de JavaScript o un marco web de terceros para sus secuencias de comandos del lado del cliente, lo que hace a JavaScript uno de los lenguajes más populares del momento. Aparte de tener otras características como; fácil de aprender, versátil, demandado, cuenta con una gran cantidad y basta información en la web.

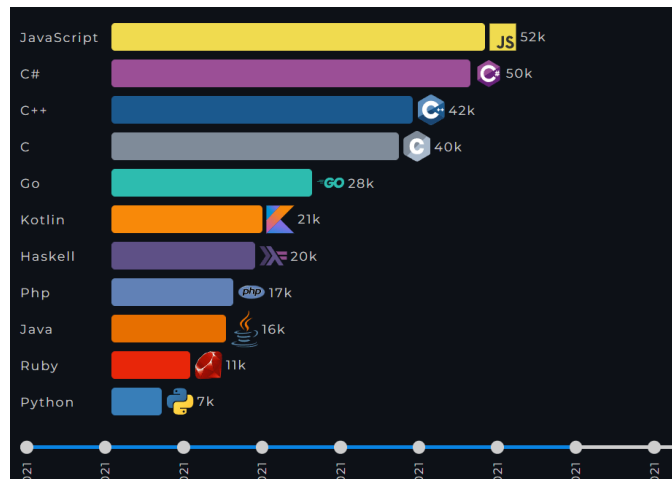


Figura 3

Demostración grafica del ano 2022 comparando la popularidad de lenguajes de programación.

Aparte de contar con una basta cantidad de frameworks, como Angular, React, Vue.

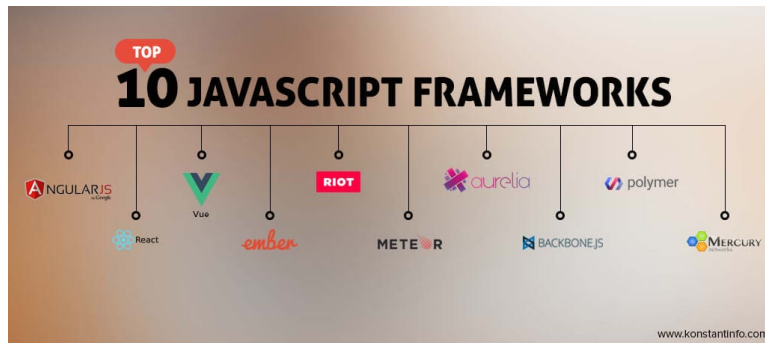


Figura 4

Listado de FrameWorks existentes de JavaScript.

3. ¿Cuál es la relación entre HTML y Java Script?

HTML es el lenguaje para estructurar y presentar contenido en una página web. JavaScript es un lenguaje de programación que permite hacer la página web interactiva y dinámica al agregar funciones y comportamientos. JavaScript se incorpora en páginas HTML para lograr esta interacción. En resumen, HTML define la estructura y el contenido, mientras que JavaScript agrega interactividad y funcionalidad.

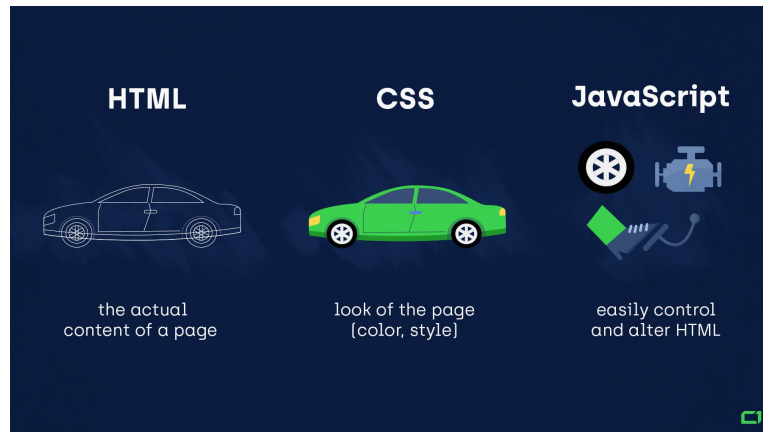


Figura 5

Explicación visual de la funcionalidad de HTML, JavaScript y CSS (CSS no se abarca en esta investigación).

4. ¿En qué beneficia usar Bootstrap para sitios y aplicaciones web en JS?

Usar Bootstrap beneficia a los sitios y aplicaciones web en JavaScript al proporcionar un conjunto de herramientas y componentes preestablecidos que facilitan el diseño, la responsividad y la estilización de la interfaz de usuario. Esto acelera el desarrollo y asegura una apariencia consistente en diferentes dispositivos y navegadores. Aquí listamos algunos de los tantos beneficios:

1. Diseño Responsivo: Bootstrap está diseñado para ser adaptable a diferentes tamaños de pantalla, lo que facilita la creación de sitios web que se ven bien en dispositivos móviles y de escritorio.

2. Ahorro de Tiempo: Bootstrap proporciona un conjunto de componentes y estilos predefinidos que ahorran tiempo en el desarrollo, ya que no tienes que escribir todo desde cero.

3. Consistencia Visual: Con Bootstrap, puedes mantener una apariencia coherente en todo tu sitio, lo que mejora la experiencia del usuario.

4. Facilidad de Personalización: Aunque Bootstrap ofrece estilos predeterminados, es altamente personalizable, lo que te permite adaptar el diseño a tus necesidades específicas.

5. Compatibilidad entre Navegadores: Bootstrap se encarga de las diferencias entre navegadores, lo que garantiza que tu sitio funcione correctamente en varios exploradores web.

The Main Benefits of Bootstrap



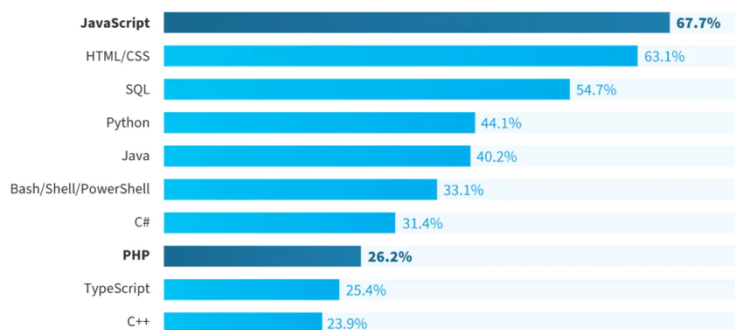
Figura 6

Imagen explicativa de los beneficios de Bootstrap

5. ¿Qué semejanza y diferencia tienen los lenguajes web PHP y Java Script?

A pesar de las similitudes que puedes tener ambos lenguajes, hay una serie de diferencias que separan el uno del otro, esas son:

1. PHP es un lenguaje de secuencias de comandos del lado del servidor, mientras que Javascript es un lenguaje de secuencias de comandos del lado del cliente.
2. PHP no se ejecuta dentro del navegador, mientras que Javascript se ejecuta dentro del navegador.
3. PHP admite bases de datos, mientras que Javascript no admite bases de datos.
4. PHP acepta variables en mayúsculas y minúsculas, mientras que Javascript no.
5. Cuando comparamos PHP y JavaScript, PHP no admite el intercambio de objetos y matrices, mientras que JavaScript admite el intercambio de objetos y matrices.



Source: StackOverflow

Figura 7

Imagen demostrativa del 2020 donde se muestra el rating de los lenguajes más utilizados en ese año.

6. ¿Cite 3 formas en que se puede agregar código JS en una pagina web?

Forma 1 >> Incrustado en el HTML: Puedes agregar código JavaScript directamente en el cuerpo de tu documento HTML utilizando la etiqueta `<script>`.

Por ejemplo:

```
<html>
<head>
  <title>Mi Página Web</title>
</head>
<body>
  <h1>Mi Página Web</h1>
  <p>¡Hola, mundo!</p>

  <script>
    // Tu código JavaScript aquí
    alert("¡Hola desde JavaScript!");
  </script>
</body>
</html>
```

Forma 2 >> En un archivo externo: Puedes crear un archivo separado con extensión ".js" que contenga tu código JavaScript y luego enlazarlo desde tu página HTML.

Por ejemplo:

```
<html>
<head>
  <title>Mi Página Web</title>
  <script src="mi_script.js"></script>
</head>
<body>
  <h1>Mi Página Web</h1>
  <p>¡Hola, mundo!</p>
</body>
</html>
```

Forma 3 >> Eventos interactivos: Es una forma común de mejorar la interactividad de una página web. Esto se logra utilizando atributos de eventos HTML para llamar a funciones JavaScript cuando ocurren ciertos eventos, como hacer clic en un botón, mover el mouse sobre un elemento, enviar un formulario,

Por ejemplo:

3.1. Agrega eventos al HTML

```
<button id="miBoton" onclick="miFuncion()">Haz clic</button>
```

3.2. Define la función JS

```
<script>
function miFuncion() {
    alert("¡Se hizo clic en el botón!");
}
</script>
```

7. ¿Cuál es la función principal de la consola en JS?

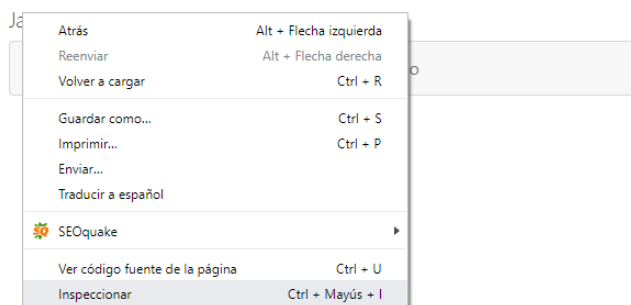
La consola del navegador es una herramienta imprescindible para los programadores web ya que te ayuda a encontrar posibles errores que pueda haber en el código de una página o aplicación.

Directamente, en la consola del navegador también puedes hacer operaciones, crear variables, arreglos, objetos o llamar a funciones definidas previamente que estén permitidas por el lenguaje de programación **JavaScript**.

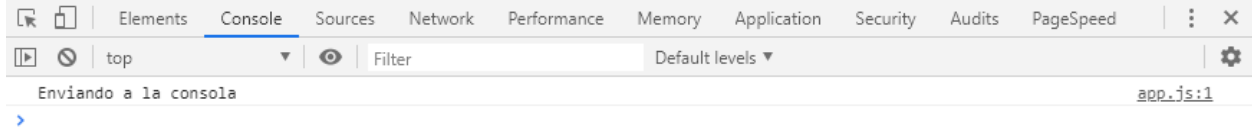
Normalmente, puedes acceder a la consola del navegador haciendo clic en el botón derecho del ratón en la ventana del navegador y en el menú que aparece vas a encontrar una opción que dice “**Inspeccionar**”.

Paso 1 >> Click en “Inspeccionar”:

La consola del navegador



Resultado:



Por defecto se muestra en la parte de abajo de la página, pero tienes la posibilidad de moverlo fácilmente a la izquierda, derecha o posición libre de la pantalla.

La consola del navegador es un excelente lugar para encontrar errores en el código y también es ideal para probar tu código **JavaScript**

8. ¿Cuál es la diferencia que existe en las declaraciones var, let y const en JS?

En JavaScript, **var**, **let**, y **const** son tres formas de declarar variables, y tienen diferencias importantes en cuanto a su alcance (scope) y su capacidad de reasignación. Breve descripción de las diferencias clave entre estas declaraciones:

1. **var**: *Alcance de función*: Las variables declaradas con **var** tienen alcance de función, lo que significa que están limitadas al bloque de código de la función en la que se declaran.

Ejemplo:

Esto es como si dijeras "Tengo 8 años, pero mañana tendré 9".

```
var edad = 8;
edad = 9; // Puedes cambiar la edad
console.log(edad); // Resultado: 9
```

2. **let**: *Es la opción más segura y moderna*. Las variables **let** tienen un alcance más ilimitado, lo que evita problemas inesperados en tu código. También pueden ser reasignadas si es necesario.

Ejemplo:

Piensa en esto como si tuvieras una caja con galletas y otra caja con galletas, pero cuando miras fuera de la caja, aún tienes 5 galletas.

```
let galletas = 5;
if (true) {
    let galletas = 10; // Dentro de esta caja, tienes 10 galletas
}
console.log(galletas); // Resultado: 5
```

3. **const**: Similar a **let**, pero con una diferencia clave: una vez que asignas un valor a una variable **const**, no puedes cambiarlo. Es útil cuando necesitas asegurarte de que una variable no cambie accidentalmente su valor.

Ejemplo:

Imagina que te llamas Ana y eso nunca cambia, como tu nombre.

```
const nombre = "Ana";
console.log(nombre); // Resultado: Ana

// Esto no funcionaría:
// nombre = "Carlos"; // ¡No puedes cambiar tu nombre después de nacer!
```

9. ¿Explique los 2 tipos de comentarios que se pueden aplicar en JS?

Poner comentarios entre las líneas de código, JavaScript o de cualquier otro lenguaje, es muy útil. Indispensable para documentar el código. Por ejemplo, nos puede servir para recordar por qué hemos hecho algo de un modo y no de otro o que tipos de datos acepta una determinada función.

Comentarios JavaScript de una línea

Pon dos barras // al comienzo de la línea o justo delante del código a comentar.

Ejemplo:

```
<script>
  // Este un comentario de una única línea
  alert("Escribiendo comentarios en javascript!"); //Aquí puedo poner una nota de lo
  // alert("Esto no se ejecuta");
</script>
```

Comentarios multi-línea

Los comentarios de una línea pueden quedarse cortos en muchas situaciones. Puedes ir colocando // en varias líneas pero hay una opción mucho más sencilla: pon /* al principio y */ al final del comentario, puedes abarcar cuantas líneas quieras.

Ejemplo:

```
<script>
  alert("Escribiendo comentarios multi-línea en javascript");
  /*
  alert("Esto no se ejecuta");
  alert("Esto no se ejecuta");
  alert("Esto no se ejecuta");
  alert("Esto no se ejecuta");
  alert("Y esto tampoco");
  Y este texto puede decir lo que yo quiera
  para acordarme de algo
  */
</script>
```

10. ¿Qué es ECMAScript6? Explique claramente.

ECMAScript 6, también conocido como ES6 o ECMAScript 2015, es una importante actualización del estándar de JavaScript. Se trata de un conjunto de nuevas características y mejoras que fueron introducidas en 2015 para hacer que JavaScript sea más potente, eficiente y fácil de usar en el desarrollo de aplicaciones web y otras aplicaciones de software. ECMAScript es el nombre del estándar en el que se basa JavaScript, y cada nueva versión agrega nuevas funcionalidades al lenguaje.

Las características clave de ECMAScript 6 incluyen:

1. **Declaración de variables mejorada:** ES6 introdujo `let` y `const` como alternativas a `var` para declarar variables. `let` permite declarar variables con alcance de bloque, lo que ayuda a evitar problemas de alcance (scope) que a menudo ocurrían con `var`. `const` se utiliza para declarar variables cuyo valor no cambiará una vez asignado, proporcionando una mayor seguridad.

Ejemplo:

```
let edad = 25; // Declaración de variable con let (puede cambiar)
const pi = 3.1416; // Declaración de variable constante (no puede cambiar)
```

2. **Funciones arrow (flecha):** Las funciones de flecha (`=>`) simplifican la sintaxis de las funciones en JavaScript y permiten escribir código más conciso. Son especialmente útiles en el manejo de funciones de alto orden.

Ejemplo:

```
// Función tradicional
function suma(a, b) {
  return a + b;
}

// Función de flecha
const sumaArrow = (a, b) => a + b;
```

3. **Nuevas estructuras de datos:** ES6 introdujo conjuntos (Set) y mapas (Map), que son estructuras de datos más versátiles y eficientes para el manejo de colecciones de datos en comparación con los arrays y objetos tradicionales.

Ejemplo:

```
// Conjunto (Set)
const frutas = new Set(['manzana', 'banana', 'manzana']);
frutas.add('kiwi');
console.log(frutas); // Resultado: Set { 'manzana', 'banana', 'kiwi' }

// Mapa (Map)
const persona = new Map();
persona.set('nombre', 'Juan');
persona.set('edad', 30);
console.log(persona.get('nombre')); // Resultado: Juan
```

4. **Desestructuración:** Permite extraer valores de arrays o objetos de manera más conveniente y legible, simplificando el acceso a los datos.

Ejemplo:

```
const persona = { nombre: 'Ana', edad: 25 };  
const { nombre, edad } = persona;  
console.log(nombre, edad); // Resultado: Ana 25
```

5. **Clases:** ES6 introdujo la sintaxis de clase para crear objetos y definir la herencia de una manera más orientada a objetos y familiar para los desarrolladores que vienen de otros lenguajes de programación.

Ejemplo:

```
class Animal {  
  constructor(nombre) {  
    this.nombre = nombre;  
  }  
  saludar() {  
    console.log('Hola, soy ${this.nombre}');  
  }  
}  
  
const perro = new Animal('Fido');  
perro.saludar(); // Resultado: Hola, soy Fido
```

6. **Módulos:** ES6 introdujo un sistema de módulos estándar para organizar el código, lo que facilita la reutilización y el mantenimiento de proyectos grandes.

Ejemplo:

```
// En archivo "miModulo.js"  
export const suma = (a, b) => a + b;  
  
// En otro archivo  
import { suma } from './miModulo.js';  
console.log(suma(3, 5)); // Resultado: 8
```

7. **Promesas:** Las promesas ofrecen una forma más limpia y manejable de trabajar con operaciones asíncronas, reemplazando en gran medida el uso de callbacks.

Ejemplo:

```
const promesa = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve('Éxito');
  }, 2000);
});

promesa.then(resultado => {
  console.log(resultado); // Resultado después de 2 segundos: Éxito
});
```

8. Template literals: Permite la creación de cadenas de texto de manera más legible y con interpolación de variables directamente en la cadena.

Ejemplo:

```
const nombre = 'Ana';
const edad = 30;
console.log(`Hola, soy ${nombre} y tengo ${edad} años.`);
// Resultado: Hola, soy Ana y tengo 30 años.
```

9. Operadores mejorados: ES6 añade operadores de propagación (spread) y de descanso (rest), que hacen que la manipulación de datos sea más eficiente y flexible.

Ejemplo:

```
// Spread
const numeros = [1, 2, 3];
const numerosDuplicados = [...numeros, ...numeros];
console.log(numerosDuplicados); // Resultado: [1, 2, 3, 1, 2, 3]

// Rest
const suma = (...numeros) => numeros.reduce((total, num) => total + num, 0);
console.log(suma(1, 2, 3, 4)); // Resultado: 10
```

En resumen, ECMAScript 6 es una versión importante de JavaScript que trajo consigo muchas características nuevas y mejoras para hacer que el lenguaje sea más poderoso y fácil de usar en el desarrollo web y de aplicaciones. Estas características han contribuido significativamente a la evolución de JavaScript y han mejorado la productividad de los desarrolladores. Desde su introducción, ES6 se ha convertido en la base estándar para el desarrollo de JavaScript en la mayoría de los proyectos modernos.

¿Qué enseñanza le brindo la tarea?

La tarea me brindó un conocimiento más profundo sobre JavaScript, incluyendo su historia, su relación con HTML, las diferencias entre var, let y const, y cómo agregar código JavaScript a una página web. Lo que me ayudo en el entendimiento general del lenguaje y la lógica de su sintaxis.

También aprendí sobre los tipos de comentarios en JavaScript y la importancia de ECMAScript 6 como una actualización clave del lenguaje. En general, la tarea me proporcionó una comprensión más completa de JavaScript y su papel esencial en el desarrollo web.

Referencias:

<https://en.wikipedia.org/wiki/JavaScript>

<https://iddigitalschool.com/bootcamps/por-que-aprender-javascript/#:~:text=Se%20trata%20de%20uno%20de,empresas%20a%20d%C3%ADa%20de%20hoy.>

https://www.w3schools.com/js/js_history.asp#:~:text=JavaScript%20was%20invented%20by%20Brenda n,JavaScript%20for%20the%20Firefox%20browser.

<https://launchschool.com/books/javascript/read/introduction>
<https://www.onely.com/blog/ultimate-guide-javascript-seo/>

<https://immune.institute/blog/razones-usar-bootstrap-web/#:~:text=Esta%20herramienta%20permite%20crear%20una,tanto%2C%20la%20experiencia%20de %20usuario.>

<https://www.hostinger.es/tutoriales/insertar-javascript-en-html/>

<https://codigonaranja.com/javascript-uso-de-la-consola-del-navegador>

<https://www.jose-aguilar.com/blog/consola-del-navegador-para-javascript/>

<https://cybmeta.com/como-poner-comentarios-en-javascript>

<https://aulab.es/noticia/17/que-es-ecmascript-6.>

<https://rootstack.com/es/blog/php-vs-javascript-cual-lenguaje-debes-elegir-este-2022#:~:text=A%20pesar%20de%20las%20similitudes,comandos%20del%20lado%20del%20cliente>