



Nombre de Estudiante:

Emmanuel de Jesús Barquero Araya

Asignatura:

Fundamentos de Programación Web

Profesor:

Francisco José Jiménez Bonilla

Trabajo:

Tarea 1

Fecha de Entrega:

Semana 3

Año:

2025

Desarrollo

1. ¿Escriba la historia del lenguaje Java Script?

JavaScript tiene una historia fascinante que empezó en los años 90, en un momento en que internet apenas estaba tomando forma. Todo comenzó en 1995, cuando Netscape, una de las empresas pioneras de los navegadores web, se dio cuenta de que necesitaba algo más para mejorar la experiencia de los usuarios en la web. En ese entonces, las páginas web eran estáticas y aburridas: solo texto, imágenes y enlaces. No había forma de interactuar con ellas como lo hacemos ahora.

Aquí entra un programador llamado Brendan Eich. Netscape lo contrató para crear un lenguaje de programación que permitiera hacer que las páginas fueran más dinámicas e interactivas. El desafío era hacerlo rápido porque la competencia en el mercado de los navegadores era feroz, especialmente contra Microsoft. Así que Brendan desarrolló este lenguaje en tan solo 10 días, el lenguaje tenía sus defectos al principio, pero cumplía su propósito: agregar interactividad a las páginas web.

Al principio, lo llamaron Mocha, pero pronto cambió a LiveScript. Sin embargo, en ese momento, Java estaba muy de moda, y Netscape quiso aprovechar esa popularidad. Así que rebautizaron el lenguaje como JavaScript para hacer que sonara como si estuviera relacionado con Java, aunque en realidad los dos lenguajes son muy diferentes. Fue más una estrategia de marketing que otra cosa.

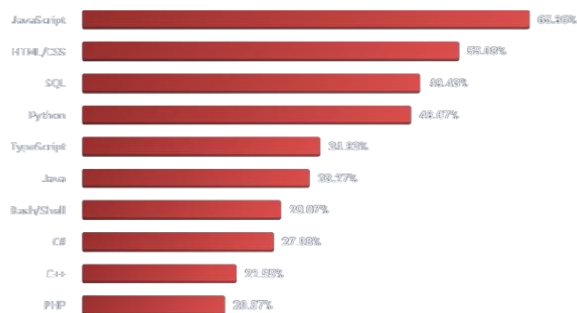
En 1996, Microsoft, que no quería quedarse atrás, creó su propia versión de JavaScript para su navegador Internet Explorer y la llamó JScript. Esto generó problemas porque no había un estándar, y los desarrolladores tenían que lidiar con las diferencias entre los navegadores. Por suerte, en 1997, JavaScript fue enviado a una organización llamada ECMA International para estandarizarlo. El resultado fue una especificación llamada ECMAScript, que es la base sobre la que JavaScript ha evolucionado desde entonces.

A lo largo de los años, JavaScript fue ganando fuerza. Al principio, se usaba solo para cosas simples, como validar formularios o mostrar alertas. Pero con el tiempo, y gracias a avances como AJAX (que permitía actualizar partes de una página web sin recargarla) y la llegada de bibliotecas como jQuery, JavaScript se convirtió en una herramienta mucho más poderosa.

El gran salto llegó con la creación de Node.js en 2009, que permitió usar JavaScript fuera del navegador, en el servidor. Esto transformó el lenguaje en una herramienta versátil y lo catapultó a ser uno de los más populares del mundo. Hoy en día, es el corazón de muchas tecnologías modernas y aplicaciones web.

En resumen, JavaScript nació de la necesidad de hacer la web más interactiva, fue creado a toda velocidad, pasó por altibajos, y ahora es esencial en el desarrollo web. Su historia no solo es un ejemplo de innovación, sino también de cómo algo que empieza pequeño puede crecer y cambiar el mundo.

2. ¿Por qué se debe aprender Java Script?



Aprender JavaScript es importante por varias razones, especialmente si se está interesado en el desarrollo web o aplicaciones interactivas. Algunas de las principales razones:

- **Lenguaje de la web:** JavaScript es el lenguaje de programación que hace posible la interactividad en páginas web. Casi todas las páginas web modernas usan JavaScript, desde los botones interactivos hasta animaciones complejas.
- **Versatilidad:** No solo se usa en el navegador, también puedes usarlo en el servidor con Node.js, lo que permite desarrollar aplicaciones completas (Front-end y Back-end) con un solo lenguaje.

- **Alta demanda laboral:** Las habilidades en JavaScript son muy demandadas en el mercado laboral. Si quieres trabajar como desarrollador web o en cualquier tipo de tecnología relacionada con la web, aprender JavaScript es esencial.
- **Gran ecosistema y comunidad:** JavaScript tiene un ecosistema enorme, con muchas librerías y Frameworks (como React, Angular y Vue) que hacen más fácil desarrollar aplicaciones modernas y escalables.
- **Facilidad de aprendizaje:** JavaScript es relativamente fácil de aprender para los principiantes. Además, puedes empezar a usarlo directamente en el navegador sin necesidad de configuraciones complejas.
- **Asincronía:** JavaScript es perfecto para trabajar con tareas que requieren operaciones asincrónicas, como peticiones a servidores, lo que lo hace fundamental para aplicaciones web dinámicas.

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

- **HTML (Hypertext Markup Language):** Es el lenguaje de marcado que se usa para estructurar el contenido de una página web. Define los elementos que aparecerán en la página, como encabezados, párrafos, imágenes, enlaces, formularios, etc. HTML le da a la página web su estructura básica.
- **JavaScript:** Es un lenguaje de programación que permite agregar interactividad y funcionalidades dinámicas a las páginas web. A diferencia de HTML, que se centra en la estructura, JavaScript permite que los elementos de una página hagan cosas como responder a eventos (clics, desplazamiento, etc.), cargar datos sin recargar la página (usando AJAX), validar formularios, y mucho más.

Relación entre HTML y JavaScript

- **Interacción con el DOM:** JavaScript interactúa con el DOM (Document Object Model), que es una representación en memoria de la estructura HTML de la página. JavaScript puede modificar, agregar o eliminar elementos HTML, cambiando dinámicamente el contenido de la página sin necesidad de recargarla.
- **Manipulación de eventos:** JavaScript puede responder a eventos generados por los usuarios, como hacer clic en un botón, enviar un formulario o cambiar el valor de un campo de texto en HTML.
- **Inserción de JavaScript en HTML:** El código JavaScript se inserta generalmente dentro de un archivo HTML mediante la etiqueta `<script>`. Puedes colocar el código JavaScript en la cabecera (`<head>`) o al final del cuerpo (`<body>`) del HTML, aunque generalmente se coloca al final para que el HTML se cargue primero.



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

Usar Bootstrap para sitios y aplicaciones web que usan JavaScript trae varios beneficios que mejoran la eficiencia, la estética y la accesibilidad del desarrollo web. Bootstrap es un Framework de diseño Front-end muy popular que incluye una serie de componentes y herramientas listas para usar.



Algunos beneficios son:

- **Diseño responsivo:** La página se adapta automáticamente a diferentes dispositivos.
- **Componentes listos para usar:** Botones, modales, carruseles, etc., con JavaScript integrado para interactividad.
- **Ahorro de tiempo:** No necesitas crear todo desde cero, lo que acelera el desarrollo.
- **Consistencia en el diseño:** Asegura una apariencia uniforme en todos los dispositivos y navegadores.
- **Flexibilidad:** Puedes personalizarlo fácilmente para que se ajuste a tus necesidades.

En resumen, Bootstrap facilita crear sitios web funcionales, atractivos y adaptables, mientras que JavaScript agrega interactividad.

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

Semejanzas entre PHP y JavaScript:

- Ambos son lenguajes de programación utilizados en el desarrollo web.
- **Interacción con HTML:** Ambos pueden manipular y generar contenido HTML, aunque de diferentes maneras.
- **Sintaxis similar:** Comparten algunas estructuras y conceptos básicos, como variables, operadores, y condicionales.



Diferencias entre PHP y JavaScript:

- **Ejecución:**
 - **PHP:** Se ejecuta en el servidor (lado del servidor), generando contenido HTML que se envía al navegador.
 - **JavaScript:** Se ejecuta en el navegador (lado del cliente), permitiendo la interacción dinámica en la página web sin recargarla.
- **Propósito:**
 - **PHP:** Se utiliza principalmente para procesar datos en el servidor, interactuar con bases de datos y generar contenido dinámico.
 - **JavaScript:** Se usa para interactividad en la interfaz de usuario (páginas dinámicas, validación de formularios, animaciones, etc.).
- **Dependencia de entorno:**
 - **PHP:** Necesita un servidor web como Apache o Nginx para ejecutarse.
 - **JavaScript:** Funciona directamente en cualquier navegador web sin necesidad de un servidor adicional.

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

- **Dentro de una etiqueta <script> en el archivo HTML:** Puedes incluir código JavaScript directamente en el archivo HTML dentro de la etiqueta <script>.

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

- **En un archivo externo de JavaScript:** Puedes escribir tu código JavaScript en un archivo separado (por ejemplo, script.js) y vincularlo al archivo HTML usando la etiqueta `<script>` con el atributo `src`.

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

- **A través de un atributo de evento en elementos HTML:** Puedes agregar JavaScript directamente en atributos de eventos dentro de elementos HTML, como `onclick`, `onmouseover`, etc.

```
<button onclick="alert('Hola desde el botón!')">Haz clic aquí</button>
```

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

La función principal de la consola en JavaScript es proporcionar una herramienta de depuración y salida de información. Se utiliza principalmente para:

- **Mostrar mensajes:** Usar `console.log()` para imprimir información en la consola del navegador, lo que ayuda a ver valores de variables, mensajes de error o el flujo de ejecución del código.
- **Depurar código:** Permite a los desarrolladores observar el comportamiento del código, verificar si los valores son correctos y encontrar errores en tiempo real sin afectar la ejecución de la página.



```
console.log(  
  "Hello World!"  
);
```


- **Otros métodos útiles:**

- `console.error()`: Para mostrar errores de forma más destacada.
- `console.warn()`: Para mostrar advertencias.
- `console.table()`: Para mostrar datos en formato de tabla, útil para arreglos y objetos.

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

Las declaraciones `var`, `let` y `const` en JavaScript se utilizan para declarar variables, pero tienen diferencias clave en su comportamiento:

- **Var:**

- **Alcance:** Tiene un alcance **funcional** (si se declara dentro de una función) o **global** (si se declara fuera de una función). No respeta los bloques `{}` como lo hacen `let` y `const`.
- **Hoisting:** Las variables declaradas con `var` se "elevan" al inicio de su contexto de ejecución, pero su valor es `undefined` hasta que se asigna explícitamente.
- **Reasignación:** Se puede reasignar y redeclarar en el mismo ámbito.

```
var x = 10;  
var x = 20; // Esto es válido
```

- **Let:**

- **Alcance:** Tiene un alcance de bloque. Esto significa que está limitada al bloque de código en el que se encuentra, como un `if`, `for`, etc.
- **Hoisting:** También se eleva, pero no se puede acceder a ella antes de su declaración (por eso se dice que está en el "temporal dead zone").
- **Reasignación:** Se puede reasignar, pero no se puede redeclarar dentro del mismo bloque.

```
let x = 10;  
x = 20; // Válido  
// let x = 30; // Error: ya fue declarada en el mismo bloque
```

- **Const:**
 - **Alcance:** También tiene un alcance de bloque, similar a let.
 - **Hoisting:** Se eleva, pero igual que let, no se puede acceder antes de la declaración.
 - **Reasignación:** No se puede reasignar después de su inicialización. Esto significa que la referencia a la variable es constante, pero si la variable es un objeto o arreglo, sus propiedades o elementos sí pueden cambiar.

```
const x = 10;
// x = 20; // Error: no se puede reasignar
```

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

- **Comentarios de una sola línea (//):**
 - Este tipo de comentario se usa para comentar una sola línea de código.
 - Todo lo que esté después de // en esa línea se considera comentario y no se ejecutará.

```
// Este es un comentario de una sola línea
let x = 10; // Este también es un comentario al final de una línea
```

- **Comentarios de varias líneas(/* */):**
 - Se usan para comentar bloques de código o varias líneas de texto.
 - Todo lo que esté entre /* y */ se considera comentario y no se ejecutará, sin importar cuántas líneas ocupe.

```
/*
Este es un comentario
que ocupa varias líneas
y puede extenderse
por tantas líneas como sea necesario.
*/
let y = 20;
```

10. ¿Qué es ECMAScript6?

ECMAScript 6 (ES6), también conocido como ES2015, es una versión del estándar de JavaScript que introdujo una serie de mejoras y nuevas características para hacer que el lenguaje fuera más potente y moderno. ECMAScript es el estándar que define cómo debe comportarse JavaScript, y ES6 fue una actualización importante que amplió las capacidades del lenguaje.

Entre estas capacidades nuevas están:

- **let y const:** Nuevas formas de declarar variables con **alcance de bloque**.
- **Funciones de flecha (=>):** Sintaxis más corta para funciones anónimas.
- **Clases:** Sintaxis más clara para la programación orientada a objetos.
- **Template literals:** Cadenas de texto con interpolación de variables.
- **Desestructuración:** Extraer valores de arrays u objetos de forma más sencilla.
- **Promesas:** Mejor manejo de operaciones asíncronas.
- **Módulos:** Sintaxis estándar para importar y exportar código.
- **Parámetros por defecto:** Asignación de valores predeterminados a parámetros de funciones.



Conclusión del Trabajo:

En conclusión, JavaScript es un lenguaje fundamental para el desarrollo web moderno. Su historia, desde sus inicios en los años 90 hasta convertirse en una herramienta versátil tanto en el navegador como en el servidor, demuestra su evolución y su impacto en la creación de experiencias interactivas en la web. Aprender JavaScript es esencial para cualquiera que quiera trabajar en el desarrollo web, ya que permite manejar tanto la parte visual como la dinámica de las páginas. Su estrecha relación con HTML y su integración con herramientas como Bootstrap facilitan el desarrollo rápido y accesible de sitios web. Además, el auge de bibliotecas y Frameworks como React, Angular y Vue ha consolidado a JavaScript como uno de los lenguajes más importantes y demandados en el mercado laboral.

A través de este estudio sobre JavaScript, he aprendido no solo su importancia en el desarrollo web, sino también cómo su evolución ha transformado la manera en que interactuamos con la web. Además, me ha quedado claro que dominar este lenguaje abre muchas puertas en el mundo del desarrollo, ya sea trabajando con el Frontend o Backend. Me siento más preparado para utilizar JavaScript en proyectos futuros.

Tabla de Evaluación

| Aspecto a Evaluar | Valor en puntos | Puntos Obtenidos | Observaciones |
|--|-----------------|------------------|---------------|
| 1. Análisis y presentación del trabajo | 5 | | |
| 2. Uso de imágenes ilustrativas | 1 | | |
| 3. Respuestas completas | 2 | | |
| 4. Conclusión de la tarea | 1 | | |
| 5. Link generado | 1 | | |
| TOTAL | | | |

Link Repositorio: https://github.com/ltsToji/Trabajos_3_Cuatrimestre