



Fundamentos de Programación WEB

Tarea 1

Estudiante:
Harvin Carrillo Angulo

Cuestionario

Profesor:
Msc. Francisco José Jimenez Bonilla

2/2/2024

Desarrollo

Contestar las siguientes preguntas planteadas acerca del Lenguaje de Programación Web Java Script (JS).

1. ¿Escriba la historia del lenguaje Java Script?

R/ JavaScript nació en 1995, con el objetivo de permitir interactividad en el navegador. Aunque inicialmente se llamaba "Mocha" y luego "LiveScript", finalmente se adoptó el nombre de JavaScript para aprovechar el auge de Java en ese momento.

Durante sus primeros años, JavaScript no era universalmente aceptado. Sin embargo, su capacidad para mejorar la experiencia del usuario en el navegador lo hizo indispensable. Microsoft también lanzó su propia versión, JScript, pero la competencia se calmó con la estandarización de JavaScript bajo el nombre de ECMAScript. Con el tiempo, JavaScript se consolidó como el lenguaje de programación para el desarrollo web, gracias a la creación de bibliotecas y frameworks notables como jQuery, Angular, y React. La aparición de Node.js permitió a los desarrolladores utilizar JavaScript fuera del navegador, en el lado del servidor.

Su continua evolución, con nuevas versiones de ECMAScript y el desarrollo de tecnologías como TypeScript, hace que se mantenga como un elemento esencial en la construcción del mundo digital.



2. ¿Por qué se debe aprender Java Script?

R/ Aprender JavaScript es fundamental para cualquier persona que quiera dedicarse al mundo digital, es uno de los pilares de la programación y podemos enumerar n cantidad de razones por las que es importante aprender JavaScript, entra las que encontramos las siguientes:

- **Fundamental en el Desarrollo Frontend:** JavaScript es el cimiento del desarrollo frontend. Con él, puedes controlar la manipulación del DOM (Modelo de Objetos del Documento), gestionar eventos del usuario y crear interfaces interactivas. Es esencial para cualquier diseñador y desarrollador web que aspire a crear experiencias envolventes.
- **Complementa HTML y CSS:** Aprender JavaScript complementa tus habilidades en HTML y CSS. Juntos, estos tres lenguajes forman la tríada esencial para la construcción de sitios web modernos, permitiéndote no solo estructurar y diseñar, sino también agregar interactividad y dinamismo.

- **Desarrollo Asíncrono:** JavaScript es esencial para la programación asíncrona, lo que significa que puede manejar tareas en segundo plano sin bloquear la ejecución del código. Esto es crucial para cargar contenido de manera eficiente, realizar solicitudes a servidores y crear aplicaciones más rápidas y receptivas.
- **Dominio de Frameworks y Bibliotecas Populares:** Muchos frameworks y bibliotecas ampliamente utilizados, como React, Angular y Vue.js, están basados en JavaScript. Aprender el lenguaje principal facilita la adopción de estas herramientas poderosas y demandadas en la industria.
- **Estándar de la Industria:** JavaScript es un estándar de facto en el desarrollo web. Casi todos los navegadores modernos admiten este lenguaje, lo que garantiza que las aplicaciones web escritas en JavaScript sean accesibles para una audiencia global.
- **Preparación para Tecnologías Emergentes:** Al entender los principios fundamentales de JavaScript, estás mejor preparado para adaptarte a las tecnologías emergentes en el desarrollo web. Las nuevas tendencias, como Progressive Web Apps (PWAs) y Single Page Applications (SPAs), a menudo se basan en JavaScript para su implementación.

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

R/ HTML y JavaScript son dos componentes esenciales en la creación de experiencias web dinámicas e interactivas. Su relación es más que complementaria; es una colaboración necesaria para construir sitios web modernos. Mientras que HTML se centra en la presentación y estructura, JavaScript entra en escena para dotar a la página de interactividad y comportamiento dinámico. Puedes utilizar JavaScript para responder a eventos del usuario, modificar el contenido en tiempo real, y realizar operaciones complejas sin recargar la página; por ejemplo: El DOM (Document Object Model) actúa como un puente entre HTML y JavaScript. JavaScript puede acceder y manipular los elementos HTML a través del DOM. Esto permite cambiar el contenido, los estilos y la estructura de la página en respuesta a eventos o acciones específicas.

Además, JavaScript se integra fácilmente en el código HTML. Puedes incluir scripts directamente en la página mediante la etiqueta `<script>`. Esto facilita la ejecución de código JavaScript en el contexto de la página, permitiendo la interacción dinámica con los elementos HTML. HTML y JavaScript son compañeros inseparables en el desarrollo frontend. HTML proporciona la estructura y el contenido, mientras que JavaScript aporta la funcionalidad y la dinámica. Juntos, forman la base de la creación de interfaces web modernas y efectivas.



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

R/ La integración de Bootstrap en sitios y aplicaciones web en JavaScript ofrece una serie de beneficios que potencian la eficiencia y la estética del desarrollo, Bootstrap proporciona una cuadrícula flexible y componentes preestilizados que se adaptan fácilmente a diferentes tamaños de pantalla. Al combinarlo con JavaScript, se pueden crear experiencias web que se ajustan de manera dinámica, respondiendo a los cambios de tamaño y orientación del dispositivo.

Bootstrap se integra sin problemas con bibliotecas y plugins de JavaScript. Esto significa que puedes incorporar funcionalidades adicionales, como carruseles, modalidades y efectos de desplazamiento, de manera sencilla y coherente en tu proyecto, aprovechando las opciones prediseñadas de Bootstrap.

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

R/ Dentro de las semejanzas encontramos las siguientes:

- Ambos PHP y JavaScript son lenguajes interpretados. Esto significa que el código fuente no se compila antes de la ejecución, permitiendo una flexibilidad y facilidad de desarrollo.
- Tanto PHP como JavaScript son fundamentales en el desarrollo web. PHP es comúnmente utilizado en el lado del servidor para procesar datos y generar contenido dinámico, mientras que JavaScript es esencial en el lado del cliente para mejorar la interactividad en el navegador.
- Ambos lenguajes están diseñados para trabajar en conjunto con HTML. PHP se integra fácilmente en el código HTML mediante scripts embebidos, mientras que JavaScript se utiliza directamente en el código HTML o se referencia desde archivos externos.

Dentro de las diferencias encontramos las siguientes:

- La diferencia clave radica en sus roles principales. PHP se ejecuta en el lado del servidor, procesando la lógica del servidor y generando contenido dinámico antes de enviarlo al navegador. JavaScript, en cambio, se ejecuta en el lado del cliente, permitiendo la manipulación dinámica de la interfaz de usuario en el navegador.
- PHP es un lenguaje fuertemente tipado, lo que significa que las variables deben declararse con un tipo específico y su tipo no puede cambiarse fácilmente. JavaScript, en contraste, es un lenguaje débilmente tipado, permitiendo mayor flexibilidad en la asignación de tipos a las variables.
- Aunque comparten algunas similitudes sintácticas debido a su herencia del estilo C, PHP y JavaScript tienen diferencias en la estructura y la forma de escribir código. Por ejemplo, PHP utiliza signos de dólar (\$) para denotar variables, mientras que JavaScript utiliza la palabra clave "var" o "let".



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

R/

- Se puede agregar código JavaScript directamente en el cuerpo del HTML utilizando la etiqueta `<script>`

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Ejemplo de Script en Cuerpo HTML</title>
</head>
<body>
  <h1>Hola, mundo!</h1>

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

- Para proyectos más extensos y organizados, es común almacenar el código JavaScript en archivos externos. Puedes enlazar estos archivos utilizando la etiqueta `<script>` con el atributo `src` para especificar la ruta del archivo.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Ejemplo de Enlace a Archivo JavaScript</title>
  <script src="ruta/del/archivo.js"></script>
</head>
<body>
  <h1>Hola, mundo!</h1>
</body>
</html>
```

- También puedes agregar código JavaScript directamente en eventos HTML, como clics, cambios o cargas de la página. Esto se logra mediante atributos de eventos, como `onclick`, `onload`, etc.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Ejemplo de Manejo de Eventos</title>
</head>
<body>
  <button onclick="saludar()">Haz clic</button>

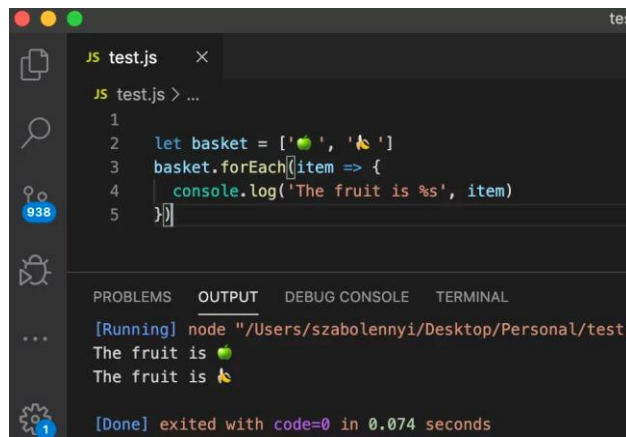
  <script>
    function saludar() {
      alert("¡Hola desde el evento onclick!");
    }
  </script>
</body>
</html>
```

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

R/ Su función principal es facilitar la depuración y el registro de información durante el desarrollo de aplicaciones web, permite a los desarrolladores imprimir mensajes, valores de variables y trazar el flujo de ejecución del programa. Los mensajes de consola proporcionan una visión detallada del estado de la aplicación en diferentes puntos, ayudando a identificar errores y entender el comportamiento del código.

Cuando ocurren errores durante la ejecución del código, la consola muestra mensajes de error específicos que indican la naturaleza y la ubicación del problema. Esto facilita la corrección rápida y eficiente de errores en el código.

Además de la depuración, la consola también se utiliza para registrar información relevante durante la ejecución de la aplicación. Los mensajes de registro (`console.log`) permiten seguir el flujo de ejecución y entender cómo interactúan diferentes partes del código.



```
JS test.js x
JS test.js > ...
1
2 let basket = ['🍏', '🍌']
3 basket.forEach(item => {
4   console.log('The fruit is %s', item)
5 })
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

[Running] node "/Users/szabolennyi/Desktop/Personal/test.
The fruit is 🍏
The fruit is 🍌

[Done] exited with code=0 in 0.074 seconds

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

R/

- **Ámbito de Variables (var vs let y const):**
 - **var:** Las variables declaradas con var tienen un ámbito de función. Esto significa que son visibles en todo el cuerpo de la función en la que fueron declaradas, incluso antes de su declaración (hoisting).
 - **let y const:** Tanto let como const tienen un ámbito de bloque. Esto significa que están limitadas al bloque en el que fueron declaradas, ya sea dentro de una función, un bucle o un bloque condicional. No tienen hoisting en el sentido de que no se pueden utilizar antes de su declaración.
- **Reasignación de Valor (let y var vs const):**
 - **var y let:** Permiten la reasignación de valores. Puedes cambiar el valor de una variable declarada con var o let en cualquier momento.
 - **const:** Las variables declaradas con const no pueden ser reasignadas después de su inicialización. Sin embargo, ten en cuenta que si la variable es un objeto o un array, los elementos internos pueden ser modificados.

- Declaración y Hoisting:

- var: Tiene un comportamiento de hoisting, lo que significa que puede ser utilizada antes de su declaración en el código. Sin embargo, su valor inicial será undefined hasta que se le asigne un valor.
- let y const: También tienen hoisting, pero a diferencia de var, no se pueden utilizar antes de su declaración. Si intentas acceder a una variable let o const antes de su declaración, obtendrás un ReferenceError.

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

R/

- Comentarios de una sola línea (//):

- Estos comentarios se utilizan para incluir anotaciones breves o explicaciones en una sola línea de código. Todo el texto después de “//” en esa línea se considera un comentario y no afecta la ejecución del código.

```
// Este es un comentario de una sola línea
let numero = 42; // También se puede colocar al final de una línea
```

- Comentarios de varias líneas (/* ... */):

- Estos comentarios permiten documentar secciones más extensas del código y pueden abarcar varias líneas. Todo el texto entre /* y */ se considera un comentario y no se ejecuta.

```
20
21  /* Distancia de la Información
22  valores1 (arriba/abajo) valores2 (izq-der) */
23
24  .container-fluid {
25    padding: 80px 0px;
26  }
27
```

10. ¿Qué es ECMAScript6? Explique claramente.

R/ ECMAScript 6, comúnmente conocido como ES6, es la sexta edición del estándar ECMAScript. ECMAScript es la especificación que define el lenguaje de programación JavaScript y establece las reglas y normas que los motores de JavaScript deben seguir para garantizar la compatibilidad entre diferentes plataformas y navegadores.

ES6 introdujo let y const como alternativas a var para declarar variables. let permite una asignación flexible, mientras que const crea variables de solo lectura, útiles para declarar constantes.

Las funciones de flecha proporcionan una sintaxis más concisa para definir funciones. Son especialmente útiles en funciones anónimas y en situaciones donde se necesita conservar el valor de `this` del contexto circundante. Ahora es posible asignar valores predeterminados a los parámetros de una función, simplificando la escritura de funciones y manejando casos donde no se proporcionan todos los argumentos.

Los módulos permiten organizar el código en archivos separados y reutilizarlo fácilmente. Antes de ES6, se solían usar patrones como IIFE o CommonJS para lograr algo similar, pero ES6 introdujo una sintaxis nativa para definir módulos.

ECMAScript 6 es una versión del estándar ECMAScript que ha enriquecido el lenguaje JavaScript con nuevas características y mejoras significativas, brindando a los desarrolladores herramientas más eficientes y expresivas para construir aplicaciones web modernas y robusta

Conclusión General

¿Anotar la enseñanza que le brindo la tarea por medio de un párrafo de 5 renglones o más?

R/ La realización de la presente tarea, ha reforzado mi comprensión profunda del lenguaje de programación, dónde no solo entiendo el contexto y la historia del lenguaje sino también la importancia del aprendizaje de este lenguaje que subraya la versatilidad y la demanda en la industria, la tarea no solo proporcionó información teórica, sino que también fomentó la aplicación práctica de conceptos en el contexto del desarrollo web. Estas enseñanzas son esenciales para fortalecer las habilidades y conocimientos necesarios en la carrera de Diseño y Desarrollo Web.

La evaluación será de acuerdo a los siguientes puntos:

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. Enseñanza de la tarea	2		
TOTAL			

El rubro #1 es exclusivamente opinión del docente.

- Excelente: 5 puntos
- Muy bueno: 4 puntos
- Bueno: 3 puntos
- Regular: 2 puntos
- Malo: 1 punto