



UNIVERSIDAD CENFOTEC

TAREA #1

Fundamentos de Programacion Web

Nombre de estudiante:

María Fernanda Aguilar Hernández

Profesor:

Francisco Jiménez Bonilla

I cuatrimestre 2025

TAREA #1

CUESTIONARIO JAVA SCRIPT

1. ¿Escriba la historia del lenguaje Java Script?

JavaScript fue creado en 1995 por Brendan Eich, un programador de Netscape, con el propósito de hacer las páginas web más interactivas. Inicialmente llamado Mocha, luego LiveScript y finalmente JavaScript, el lenguaje se popularizó rápidamente.

En 1997, para evitar incompatibilidades entre navegadores, se estableció el estándar ECMAScript. Durante los años 2000, con la llegada de AJAX, JavaScript permitió actualizar páginas sin recargarlas, impulsando aplicaciones dinámicas como Google Maps.

En 2009, con la aparición de Node.js, JavaScript se expandió al desarrollo backend. Más tarde, frameworks como React, Angular y Vue consolidaron su uso en el desarrollo web. Con ECMAScript 6 (2015), el lenguaje introdujo mejoras clave, afianzándose como uno de los más utilizados en el mundo. Hoy, JavaScript sigue evolucionando y es esencial en la tecnología moderna.

2. ¿Por qué se debe aprender Java Script?

Se debería aprender JavaScript porque es un lenguaje accesible, versátil y ampliamente utilizado en el desarrollo web. También según lo visto en clase, su sintaxis es sencilla y permite obtener resultados rápidamente, lo que facilita el proceso de aprendizaje. Además, cuenta con una gran comunidad de desarrolladores, lo que significa que existen numerosos recursos, tutoriales y foros donde se puede encontrar ayuda en caso de dudas o problemas.

Otra ventaja de JavaScript es su versatilidad, ya que no solo se usa en el desarrollo del lado del cliente, es decir, en la interfaz de usuario de los sitios web, sino también en el backend, gracias a entornos como Node.js. Esto permite a los programadores trabajar tanto en el diseño visual de las aplicaciones como en su funcionamiento interno.

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

HTML y JavaScript están estrechamente relacionados porque juntos forman la base de la web moderna. HTML es el lenguaje que se usa para estructurar el contenido de una página web, mientras que JavaScript permite agregarle interactividad y dinamismo.

En términos prácticos, HTML se encarga de elementos como títulos, párrafos, imágenes y enlaces. JavaScript, en cambio, permite que los botones reaccionen cuando los presionamos, que aparezcan notificaciones o que una página se actualice sin necesidad de recargarla por completo.

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

Facilita el diseño y la estructura de las páginas sin necesidad de escribir demasiado código desde cero. Bootstrap es un framework de CSS y JavaScript que proporciona estilos predefinidos y componentes listos para usar, lo que permite construir interfaces modernas y funcionales de manera rápida y eficiente.

Uno de sus mayores beneficios es que hace que los sitios web sean responsivos, además, Bootstrap incluye una amplia colección de botones, formularios, menús de navegación, tarjetas y otros elementos visuales que pueden integrarse fácilmente en un proyecto mediante clases en HTML, sin necesidad de diseñarlos manualmente.

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

PHP y JavaScript son lenguajes esenciales en el desarrollo web, pero se diferencian principalmente en dónde y cómo se ejecutan. Mientras que JavaScript se ejecuta en el navegador y se usa para mejorar la interactividad y el dinamismo de una página web, PHP se ejecuta en el servidor y se encarga de procesar datos y generar contenido antes de enviarlo al usuario. JavaScript permite modificar elementos en tiempo real sin necesidad de recargar la página, mientras que PHP maneja bases de datos, sesiones de usuario y lógica del servidor.

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

La primera forma es insertar el código directamente dentro de una etiqueta `<script>` en el mismo archivo HTML. Este método es útil para agregar pequeñas funciones o scripts rápidos sin necesidad de crear archivos adicionales. El código se escribe dentro de la etiqueta `<script>`, ya sea en el `<head>` o antes de cerrar el `<body>`, dependiendo de cuándo se necesite ejecutar.

Otra manera es utilizar un archivo JavaScript externo y enlazarlo con la página HTML a través de la etiqueta `<script>` con el atributo `src`. En este caso, el código se guarda en un archivo con extensión `.js`, que luego se referencia dentro del HTML. Este método es ideal para mantener el código más organizado, facilitar su reutilización y mejorar la legibilidad del documento.

Finalmente, JavaScript también puede ser inyectado dinámicamente dentro del código HTML mediante eventos o la manipulación del DOM. Esto se logra mediante métodos como `document.write()`, la creación de elementos `<script>` desde JavaScript o incluso a través de atributos como `onclick`, que permiten ejecutar código JavaScript directamente desde elementos HTML al detectar una acción del usuario. Esta técnica es útil para agregar funcionalidades de forma dinámica sin que el código esté escrito de antemano en el HTML.

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

La consola en JavaScript cumple una función fundamental en el desarrollo y depuración de código, ya que permite a los programadores interactuar con el lenguaje en tiempo real, analizar el comportamiento del programa y detectar posibles errores. A través de la consola, es posible ejecutar comandos, probar fragmentos de código y visualizar mensajes informativos sin necesidad de modificar la interfaz de la página web.

Una de sus principales utilidades es mostrar información mediante la función `console.log()`, que permite imprimir valores, mensajes o resultados de cálculos directamente en la consola del navegador. Esto es especialmente útil para seguir el flujo de ejecución de un programa y entender cómo cambian los datos a lo largo del código.

Además, la consola facilita la detección de errores, ya que muestra advertencias y mensajes de fallo cuando el código contiene problemas de sintaxis o ejecución. Los navegadores modernos ofrecen herramientas avanzadas dentro de la consola que permiten inspeccionar variables, analizar el rendimiento del código y depurar errores de manera eficiente.

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

En JavaScript, `var`, `let` y `const` se usan para declarar variables, pero difieren en alcance y mutabilidad. `var` tiene alcance de función y permite redeclaración, lo que puede causar errores. `let` tiene alcance de bloque y permite reasignación, pero no redeclaración. `const` también tiene alcance de bloque, pero su valor no puede cambiar tras su asignación. En el desarrollo moderno, se recomienda usar `let` y `const` en lugar de `var` para un código más seguro y predecible.

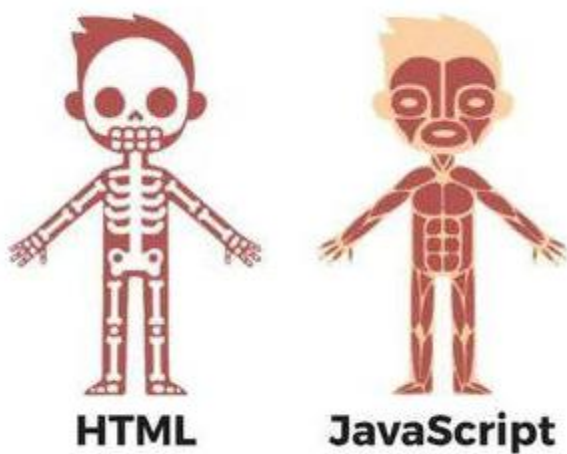
9. ¿Explique concisa y resumido los 2 tipos de comentarios que se pueden aplicar en JS?

En JavaScript, los comentarios sirven para agregar notas dentro del código sin que afecten su ejecución. Existen dos tipos principales: el comentario de una sola línea, que se indica con dos barras inclinadas (`//`) y se usa para escribir breves explicaciones

o aclaraciones sobre una línea específica de código. Por otro lado, está el comentario de múltiples líneas, que se encierra entre `/* */` y permite escribir textos más largos o desactivar temporalmente fragmentos de código sin necesidad de eliminarlos. Ambos tipos de comentarios son herramientas útiles para mejorar la legibilidad y el mantenimiento del código.

10. ¿Qué es ECMAScript6? Explique claramente.

ECMAScript6 es una versión de JavaScript que introdujo mejoras y nuevas características para hacer el lenguaje más eficiente y moderno. Entre sus principales aportes destacan las declaraciones `let` y `const` para un mejor manejo de variables, las funciones de flecha que simplifican la sintaxis, las plantillas de texto con comillas invertidas, los módulos para organizar mejor el código, y las promesas para manejar operaciones asíncronas. ES6 marcó un antes y un después en JavaScript, facilitando su uso y mejorando el rendimiento en el desarrollo web.



CONCLUSION

Después de investigar sobre estos conceptos queda claro que JavaScript es un lenguaje vital en el desarrollo web, que está lleno de herramientas y características que lo hacen útil y poderoso. Mediante este trabajo se analizó cómo JavaScript interactúa con HTML, la importancia de la consola para depurar código, las diferencias entre var, let y const, y cómo ES6 mejoró la forma en que se escribe JavaScript. También cómo los comentarios ayudan a organizar mejor el código y cómo Bootstrap facilita el diseño web. En general, entender estos fundamentos nos da una base sólida para seguir conociendo y mejorando en la programación.