



UNIVERSIDAD CENFOTEC

Tarea #1

Cuestionario Java Script

Estudiante: Daniel Cordero Guadamuz

Profesores: Francisco Jose Jimenez Bonilla

Fecha de Entrega: Semana 3

I Ciclo, 2025

1. ¿Escriba la historia del lenguaje Java Script?

JavaScript nació en 1995 de la mano de Brendan Eich, un ingeniero de Netscape, con el nombre inicial de Mocha. Su objetivo era dotar a los navegadores web de mayor interactividad, algo que no era posible con HTML puro. Más tarde, el lenguaje fue renombrado a LiveScript y finalmente a JavaScript, nombre que conserva hasta la actualidad.

Estandarización y adopción

En 1997, Ecma International, una organización de normalización, tomó las riendas del lenguaje y creó el estándar ECMAScript, con el fin de garantizar su interoperabilidad entre diferentes navegadores. Esta estandarización impulsó la adopción de JavaScript, convirtiéndolo en un lenguaje esencial para el desarrollo web.

Crecimiento y evolución

A lo largo de los años, JavaScript ha experimentado un crecimiento exponencial, tanto en funcionalidades como en popularidad. La introducción de nuevas tecnologías como AJAX y la aparición de frameworks como jQuery y React han ampliado aún más sus posibilidades, permitiendo la creación de aplicaciones web dinámicas y complejas.

JavaScript hoy en día

En la actualidad, JavaScript es omnipresente en el desarrollo web. Es utilizado para crear interfaces de usuario interactivas, animaciones, juegos web, aplicaciones web de una sola página (SPA) e incluso aplicaciones del lado del servidor con Node.js. Su versatilidad y facilidad de uso lo han convertido en el lenguaje de programación web más popular del mundo.

2. ¿Por qué se debe aprender Java Script?



Lenguaje dominante en el desarrollo web

JavaScript es el lenguaje fundamental para crear interfaces web interactivas y dinámicas. Es utilizado por la mayoría de los sitios web y aplicaciones web del mundo, desde gigantes como Google y Facebook hasta pequeñas empresas y emprendedores. Dominar JavaScript permitirá crear desde páginas web simples hasta complejas aplicaciones web de una sola página y backends de servidores.

Alta demanda de profesionales

La demanda de desarrolladores de JavaScript es enorme y sigue creciendo. Las empresas de todo tipo buscan profesionales con experiencia en este lenguaje para crear y mantener sus sitios web y aplicaciones web.

Versatilidad y aplicabilidad

JavaScript no se limita al desarrollo web. Con Node.js, se pueden crear aplicaciones del lado del servidor, herramientas de línea de comandos e incluso aplicaciones de escritorio. Esta versatilidad lo convierte en un lenguaje altamente adaptable a diferentes necesidades y proyectos.

3. ¿Cuál es la relación entre HTML y Javascript?

HTML y JavaScript son dos lenguajes fundamentales para el desarrollo web, pero cada uno tiene un propósito distinto y trabajan en conjunto para crear páginas web dinámicas e interactivas.

Su relación

- HTML proporciona la estructura y el contenido: Define la base sobre la que JavaScript puede actuar.
- JavaScript aporta la interactividad y la funcionalidad: Da vida a la página web, permitiendo que los usuarios interactúen con ella y realicen acciones.

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



Bootstrap es un framework CSS de código abierto que facilita la creación de sitios web y aplicaciones web responsive y amigables con dispositivos móviles. Ofrece una amplia gama de componentes prediseñados, como botones, formularios, menús de navegación, modales y rejillas, que puedes utilizar para construir rápidamente interfaces de usuario atractivas y consistentes.

Ventajas de usar Bootstrap con JavaScript

- Desarrollo más rápido: Los componentes prediseñados y los complementos de JavaScript permiten crear interfaces de usuario rápidamente, sin necesidad de escribir código desde cero.

- Código más limpio y organizado: Bootstrap ayuda a escribir código CSS más limpio y organizado, lo que facilita su mantenimiento y comprensión.
- Mayor consistencia: Bootstrap garantiza que las interfaces de usuario tengan un aspecto y una funcionalidad consistentes en todos los navegadores y dispositivos.
- Facilidad de uso: Bootstrap es relativamente fácil de aprender, incluso para principiantes en el desarrollo web.
- Amplia comunidad: Existe una gran comunidad de desarrolladores de Bootstrap que pueden ayudar a resolver problemas y encontrar soluciones.

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

Semejanzas

- Lenguajes de scripting: Tanto PHP como JavaScript son lenguajes de scripting, lo que significa que se ejecutan línea por línea y no requieren compilación previa. Esto los hace ideales para el desarrollo web dinámico, ya que permiten modificar el contenido de las páginas web sin necesidad de recargarlas.
- Versatilidad: Ambos lenguajes son versátiles y pueden ser utilizados para una amplia gama de tareas, como crear formularios web, procesar datos, comunicarse con bases de datos y generar contenido dinámico.
- Integración con HTML: Tanto PHP como JavaScript se integran perfectamente con HTML, el lenguaje base para la estructura y el contenido de las páginas web. Esto permite combinar la potencia de los lenguajes de scripting con la estructura y el contenido de las páginas web.

Diferencias

- Ubicación de ejecución: La principal diferencia entre PHP y JavaScript es su ubicación de ejecución. PHP se ejecuta principalmente en el servidor, lo que significa que el código PHP se procesa en el servidor antes de enviarse al navegador del usuario. En cambio, JavaScript se ejecuta principalmente en el navegador del usuario, lo que significa que el código JavaScript se descarga en el navegador y se ejecuta en el equipo del usuario.

- **Propósito:** PHP se utiliza principalmente para tareas del lado del servidor, como acceder a bases de datos, procesar formularios y generar contenido dinámico. JavaScript se utiliza principalmente para tareas del lado del cliente, como crear interfaces de usuario interactivas, manipular el DOM (Document Object Model) y responder a eventos de usuario.
- **Sintaxis:** La sintaxis de PHP y JavaScript es diferente. PHP tiene una sintaxis similar a C, mientras que JavaScript tiene una sintaxis similar a C++.

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

- Etiquetas <script>

Este es el método más básico y directo para agregar código JavaScript. Se utiliza una etiqueta <script> dentro del body del documento HTML. El código JavaScript se escribe directamente entre las etiquetas de apertura y cierre de la etiqueta <script>.

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
</head>
<body>
  <script>
    // Código JavaScript aquí
    alert("¡Hola, mundo!");
  </script>
</body>
</html>
```

- Archivos JavaScript externos

En lugar de escribir el código JavaScript directamente en la página HTML, se crea un archivo JavaScript externo con la extensión .js y luego se incluye en la página HTML utilizando la etiqueta <script>. Esto permite mantener el código JavaScript organizado y reutilizable.

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
</head>
<body>
  <script src="mi_script.js"></script>
</body>
</html>
```

- Integración con frameworks JavaScript:

Si se utiliza un framework JavaScript como React o Angular, estos suelen proporcionar sus propias formas de incluir y organizar el código JavaScript.

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

Ejecutar código JavaScript

- Escribir y ejecutar fragmentos de código JavaScript directamente en la consola. Esto permite probar código, experimentar con nuevas ideas y depurar problemas.
- Ver los resultados del código ejecutado en la consola, lo que ayuda a comprender cómo funciona el código y cómo interactúa con la página web.

Ver y modificar variables

- La consola permite inspeccionar y modificar el valor de las variables en el código JavaScript. Esto es útil para depurar problemas y comprender cómo fluyen los datos a través del programa.
- Ver el valor de las variables en diferentes puntos de ejecución del código, lo que ayuda a identificar cuándo y por qué cambian las variables.

Registrar mensajes

- Se puede usar la consola para registrar mensajes de información, advertencias o errores durante la ejecución de tu código JavaScript. Esto ayuda a rastrear el flujo del programa y detectar posibles problemas.
- Los mensajes registrados se muestran en la consola, lo que permite revisarlos fácilmente y obtener información sobre el comportamiento del código.

Depurar código

- La consola proporciona herramientas para depurar paso a paso el código. Se pueden establecer puntos de interrupción en el código, examinar el valor de las variables en cada punto de interrupción y ejecutar el código línea por línea.
- Esto ayuda a identificar la causa de los errores y corregirlos de manera eficiente.

Obtener información del navegador

La consola permite acceder a información sobre el navegador, como la versión del navegador, el sistema operativo y los recursos del sistema. Esto puede ser útil para depurar problemas relacionados con la compatibilidad del navegador o el rendimiento del código.

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

Alcance (Scope)

- var: Alcance de función o global. Las variables declaradas con var son accesibles desde cualquier lugar dentro de la función en la que se declaran, y también desde fuera de la función si se declaran en el ámbito global.
- let: Alcance de bloque. Las variables declaradas con let son accesibles sólo dentro del bloque de código en el que se declaran (por ejemplo, dentro de un bloque if, un bucle for o una función).
- const: Alcance de bloque. Las variables declaradas con const son accesibles sólo dentro del bloque de código en el que se declaran, al igual que let.

Redeclaración y reasignación

- **var:** Las variables declaradas con **var** pueden ser redeclaradas dentro del mismo ámbito y sus valores pueden ser reasignados en cualquier momento.
- **let:** Las variables declaradas con **let** no pueden ser redeclaradas dentro del mismo ámbito, pero sus valores sí pueden ser reasignados en cualquier momento.
- **const:** Las variables declaradas con **const** no pueden ser redeclaradas ni reasignadas. Su valor es fijo una vez que se inicializa.

Hoisting

- **var:** Las variables declaradas con **var** se elevan (hoisting) a la parte superior de su ámbito, lo que significa que pueden ser utilizadas antes de que se declaren.
- **let y const:** Las variables declaradas con **let** y **const** no se elevan, por lo que no se pueden usar antes de que se declaren.

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

Comentarios de una sola línea

Se utilizan para escribir notas o explicaciones breves sobre el código. Se escriben utilizando dos barras inclinadas seguidas de un comentario:

```
// Este es un comentario de una sola línea  
console.log("¡Hola, mundo!");
```

Comentarios de múltiples líneas

Se utilizan para escribir comentarios más extensos o que abarcan varias líneas de código. Se escriben utilizando dos barras inclinadas seguidas de un asterisco, el comentario, y otro asterisco y dos barras inclinadas:

```
/*  
Este es un comentario de múltiples líneas  
que puede abarcar varias líneas de código.  
*/  
function miFuncion() {  
    // Código de la función  
}
```

10. ¿Qué es ECMAScript 6? Explique claramente

ECMAScript 6 (ES6), también conocido como JavaScript ES6 o JavaScript 2015, es una actualización importante del lenguaje de programación JavaScript publicada en junio de 2015. ES6 introdujo una serie de nuevas características y mejoras que hicieron que JavaScript fuera más potente, flexible y fácil de usar.

Aquí algunos de los cambios más significativos introducidos en ECMAScript 6:

- Nuevas formas de declarar variables: `let` y `const` se introdujeron para reemplazar a `var`. Estas nuevas declaraciones ofrecen un mejor control sobre el alcance de las variables, lo que ayuda a prevenir errores y escribir código más limpio.
- Flechas de función: Las flechas de función proporcionan una sintaxis más concisa para definir funciones. Esto puede ser útil para definir funciones cortas y anónimas.
- Clases: ES6 introdujo soporte para clases, lo que facilita la creación de objetos orientados a objetos en JavaScript.
- Modules (Módulos): ES6 introdujo módulos, que permiten organizar el código en partes reutilizables.
- Template literals: Los template literals proporcionan una forma más flexible y legible de definir cadenas de texto.
- Destructuring: El destructuring permite extraer valores de objetos y arrays de una manera más concisa.
- Promesas: Las promesas proporcionan una forma más limpia y manejable de trabajar con operaciones asíncronas.
- Por defecto y valores extendidos (Default y Spread): Los valores por defecto y extendidos permiten definir valores predeterminados para parámetros de

función y propagar elementos de arrays y objetos de manera más sencilla.

Beneficios de usar ECMAScript 6

- Código más limpio y legible: Las nuevas características de ES6 facilitan la escritura de código JavaScript más limpio, organizado y fácil de leer.
- Mayor productividad: Las nuevas características de ES6 pueden ayudarte a escribir código más rápido y con menos errores.
- Código más moderno: Al utilizar las características de ES6, el código estará más actualizado y en línea con las prácticas modernas de desarrollo web.
- Mayor compatibilidad: La mayoría de los navegadores modernos son compatibles con ES6, lo que significa que el código funcionará en la mayoría de los dispositivos.

En conclusión, el conocimiento adquirido en este entregable brinda las bases sólidas para empezar a convertirnos en buenos y competentes desarrolladores. Es nuestro deber seguir aprendiendo, practicando y explorando nuevas tecnologías para ampliar nuestras habilidades y crecer a nivel profesional cada día mas.