

Tarea 1

Steven Cordero Artavia

Universidad Cenfotec

Fundamentos de Programación Web

Prof. Francisco Jimenez



Desarrollo

1. ¿Escriba la historia del lenguaje Java Script?

JavaScript es un lenguaje de programación web nacido en 1995. Fue creado por Netscape para dar vida a páginas web interactivas. A lo largo de los años, se convirtió en un lenguaje esencial para la programación web, permitiendo funciones como validación de formularios, animaciones y comunicación con servidores. Hoy, es fundamental en aplicaciones web modernas y se usa en servidores (Node.js). Su evolución constante lo mantiene relevante en el mundo digital actual.



2. ¿Porqué se debe aprender JavaScript?

Aprender JavaScript es valioso porque es el lenguaje de programación principal para crear sitios web interactivos y aplicaciones web modernas. Permite a las personas construir experiencias en línea dinámicas, como formularios interactivos, juegos, redes sociales y aplicaciones empresariales. Con JavaScript, se puede mejorar la usabilidad y la funcionalidad de proyectos web, lo que lo convierte en una habilidad esencial para desarrolladores web, diseñadores y profesionales digitales que desean crear experiencias en línea atractivas y efectivas.

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

HTML y JavaScript son dos tecnologías primordiales para la web, pero tienen roles diferentes y trabajan juntas para crear sitios web interactivos. HTML (Hypertext Markup Language) se utiliza para estructurar y presentar contenido en la web, como texto, imágenes y enlaces. JavaScript, por otro lado, es un lenguaje de programación que

agrega interactividad y dinamismo a las páginas HTML. Puedes usar JavaScript para manipular elementos HTML, responder a eventos del usuario y realizar operaciones más avanzadas, como validación de formularios o comunicación con servidores. En resumen, HTML proporciona la estructura y el contenido, mientras que JavaScript añade funcionalidad y comportamiento a las páginas web. Juntos, crean experiencias web completas y ricas.

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

Bootstrap es un marco de diseño web que beneficia a los desarrolladores de sitios y aplicaciones web en JavaScript al simplificar el diseño, asegurar la responsividad, proporcionar componentes interactivos, facilitar la compatibilidad con JavaScript y ofrecer una comunidad activa. Esto ahorra tiempo en el desarrollo y garantiza un diseño coherente, lo que resulta en experiencias de usuario más atractivas y eficientes en una variedad de dispositivos y navegadores.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min"
  </head>
  <body>
    <h1>Hello, world!</h1>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bun
  </body>
</html>
```

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

Semejanzas: Tanto PHP como JavaScript son lenguajes ampliamente utilizados en el desarrollo web, con comunidades activas y recursos en línea disponibles. Ambos son lenguajes interpretados, lo que facilita su uso y aprendizaje.

Diferencias: La principal diferencia radica en su ubicación de ejecución y propósito. PHP se ejecuta en el lado del servidor para tareas como procesamiento de formularios y acceso a bases de datos, mientras que JavaScript se ejecuta en el lado del cliente,

mejorando la interactividad en el navegador con funciones como validación de formularios y animaciones. Además, PHP es un lenguaje tipado débil, mientras que JavaScript es tipado débil o dinámico, lo que afecta la forma en que manejan los tipos de datos.

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

- **Incrustado en HTML:** Se puede incluir código JavaScript directamente en el archivo HTML utilizando la etiqueta `<script>`.

```
<script>
  // Tu código JavaScript aquí
</script>
```

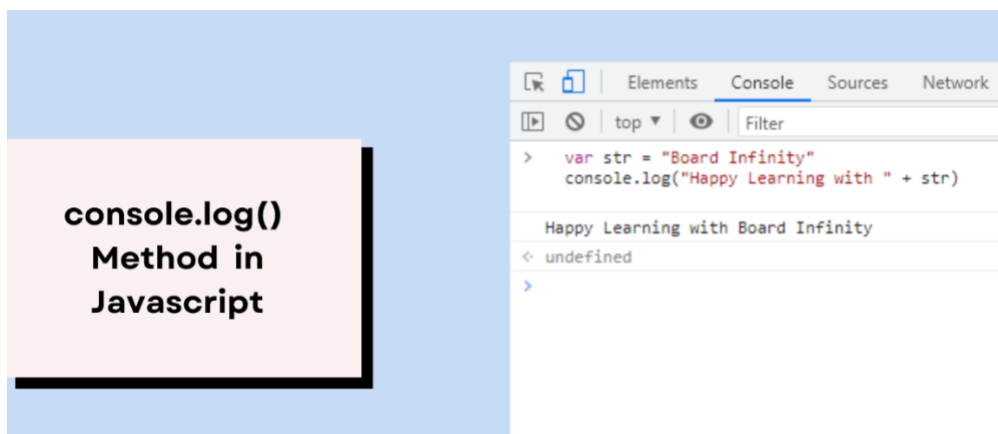
- **Enlazado desde un archivo externo:** Es posible almacenar el código JavaScript en un archivo separado con extensión `.js` y luego enlazarlo en la página HTML utilizando la etiqueta `<script>` con el atributo `src` que apunte al archivo externo.

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

- **Gestionado por un framework o biblioteca:** Si se utiliza un framework o biblioteca como React, Angular o Vue.js, se puede integrar JavaScript en la página web siguiendo las convenciones y estructuras específicas de ese framework. Esto implica generalmente el uso de componentes y archivos específicos para el código JavaScript de la aplicación.

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

La función principal de la consola en JavaScript es facilitar el desarrollo y la depuración de aplicaciones web. Sirve como una herramienta esencial para los programadores al mostrar mensajes de error detallados que ayudan a identificar y solucionar problemas en el código. Además, permite visualizar valores de variables y resultados de funciones en tiempo real, lo que mejora la comprensión del comportamiento del programa. La consola también es útil para registrar información personalizada y para interactuar directamente con el código mediante comandos JavaScript, lo que facilita las pruebas y la manipulación de datos durante el desarrollo web.



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

En JavaScript, var, let y const son palabras clave para declarar variables con diferencias clave en alcance y reasignación. var tiene un alcance global o de función, permite la reasignación y se iza (hoisting) al principio de su ámbito. let tiene un alcance de bloque, permite la reasignación, pero no se iza(hoisting). const también tiene un alcance de bloque, pero no permite la reasignación de la variable en sí, aunque permite la modificación de propiedades en objetos o elementos en matrices. La elección entre ellos depende de tus necesidades y la claridad de alcance en tu código.

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

En JavaScript, existen dos tipos principales de comentarios que los desarrolladores pueden utilizar para documentar y explicar su código. El primero es el comentario de una sola línea, que se crea con `//` y permite agregar notas breves o aclaraciones en una sola línea de código. El segundo es el comentario de varias líneas, que se delimita utilizando `/*` para iniciar y `*/` para finalizar, lo que permite incluir comentarios extensos y detallados que pueden abarcar varias líneas de código. Estos comentarios son esenciales para mejorar la legibilidad del código, facilitar la colaboración entre desarrolladores y proporcionar explicaciones sobre el funcionamiento de un programa sin afectar su ejecución, ya que el intérprete de JavaScript los ignora por completo.

```
15  /// Objeto com os campos que deverão ser devolvidos e demais opções
16  // Exemplo de passagem do "Where"
17  // [['ban_ativo', '=', '1', 'AND'],
18  // ['ban_id', '>', '0']];
19  bancoModel.dados = function (aCmp, aWhere, sOrd) {
20      // Chamar a montagem do Select do Superior
21      var sSql = model.getSql('banco_tb', aCmp, aWhere, sOrd, '');
22
23      /**
24      Chamar o model superior para realizar a consulta ao banco
25      console.Log(sSql);
26      return ['1-Banco Legal', '2-Banco de verdade'];
27      **/
28
29      return model.executarSql(sSql);
30  };
```

10. ¿Qué es ECMAScript6? Explique claramente.

ECMAScript 6 (ES6) es una versión fundamental de la especificación de JavaScript que introdujo mejoras clave en el lenguaje. Entre estas mejoras se incluyen declaraciones de variables más robustas con `let` y `const`, funciones de flecha para una sintaxis más concisa, plantillas literales para cadenas de texto dinámicas, destructuración de objetos y matrices, parámetros por defecto en funciones, clases para una programación orientada a objetos más clara, módulos para una organización eficiente del código y promesas para la gestión simplificada de tareas asíncronas. ES6 transformó JavaScript en un lenguaje más moderno, legible y poderoso, sentando las bases para el desarrollo

web avanzado y eficiente, y marcando el inicio de una serie de actualizaciones continuas en el estándar ECMAScript.

Conclusión/Enseñanza

Como conclusión, JavaScript es un lenguaje de programación fundamental en la web. Su importancia radica en su capacidad para hacer que las páginas web sean interactivas y dinámicas. Con JavaScript, podemos crear desde simples validaciones de formularios hasta aplicaciones web avanzadas. Aprenderlo es valioso porque te permite mejorar la experiencia del usuario en tus sitios web y te brinda habilidades de codificación versátiles y demandadas en el mercado laboral. Es un lenguaje accesible que te da el poder de hacer que las cosas sucedan en la web, y eso lo convierte en una gran inversión de tiempo y esfuerzo.