

# ***TAREA #1***

## **Cuestionario JAVA SCRIPT**

---

Fundamentos de Programación Web

Nombre: Nancy Quesada Flores

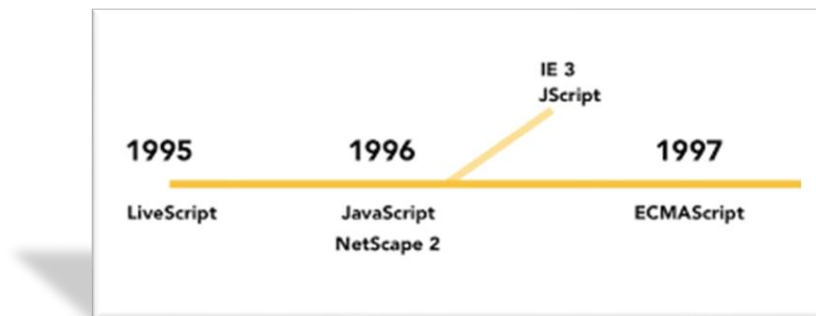
Profesor: Francisco José Jimenez Bonilla

Universidad CENFOTEC

## 1. ¿Escriba la historia del lenguaje Java Script?

El lenguaje de programación JavaScript surgió a mediados de la década de 1990 y ha evolucionado significativamente desde entonces. Fue creado por Brendan Eich en 1995 mientras trabajaba en Netscape Communications Corporation. Originalmente, se llamaba "Mocha" y luego se renombró a "LiveScript". Finalmente, en un esfuerzo de colaboración con Sun Microsystems (los creadores de Java), se le dio el nombre que conocemos hoy en día: JavaScript. Aunque comparte un nombre similar a Java, son dos lenguajes de programación distintos y no están relacionados.

JavaScript se introdujo en el navegador Netscape Navigator 2.0 en diciembre de 1995. Inmediatamente después, Microsoft creó su propia versión llamada JScript para Internet Explorer. Esta rivalidad condujo a una falta de estándares, lo que dificultó el desarrollo web compatible con múltiples navegadores durante un tiempo.



Sin embargo, en 1997, se formó el grupo ECMA International para estandarizar JavaScript. El resultado fue ECMAScript, la especificación estándar que define el lenguaje subyacente de JavaScript. A lo largo de los años, diferentes versiones de ECMAScript han sido liberadas, mejorando y expandiendo las capacidades del lenguaje.

Uno de los momentos más significativos en la historia de JavaScript fue el lanzamiento de jQuery en 2006. jQuery se convirtió en una biblioteca muy popular que simplificó la manipulación del DOM y la interacción con JavaScript. Ayudó a unificar la forma en que los desarrolladores trabajaban con JavaScript en diferentes navegadores.

En los últimos años, JavaScript ha visto un rápido crecimiento y ha evolucionado de manera espectacular con el auge de los marcos de trabajo (frameworks) como React, Angular y Vue.js, que facilitan el desarrollo de aplicaciones web más complejas y rápidas.

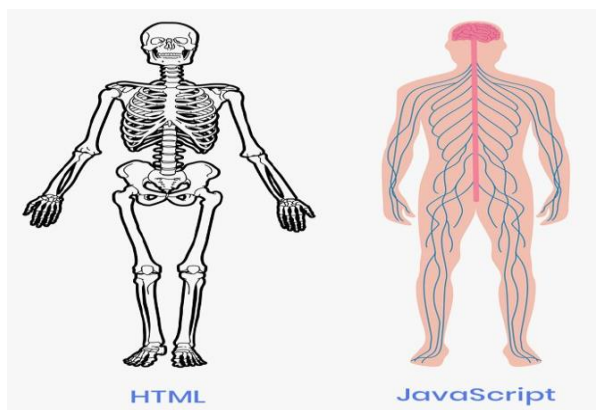
## 2. ¿Por qué se debe aprender Java Script?

JavaScript es esencial en el mundo del desarrollo web. Es la clave si se está interesado en crear sitios web interactivos y dinámicos ya que permite mejorar la experiencia del usuario; además es útil mencionar que JavaScript es compatible con todos los principales navegadores web, lo que significa que tiene una amplia aplicabilidad en el mercado laboral. JavaScript puede ser muy gratificante desde el punto de vista creativo ya que brinda la capacidad de dar vida a ideas y proyectos web de manera única.

Además, JavaScript está en constante evolución. Con la introducción de ES6 (ECMAScript 2015) y las actualizaciones posteriores, el lenguaje se ha vuelto más poderoso y moderno. Aprender JavaScript permite mantenernos al día con las últimas tendencias en desarrollo web y estar preparado para el futuro.



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



HTML proporciona la estructura y el contenido básico de una página web, que por sí solo es estático mientras que JavaScript agrega interactividad y dinamismo al permitir que los elementos HTML se modifiquen y respondan a las acciones del usuario. Puede interactuar con el DOM (Document Object Model) de la página, que es una representación en memoria de la estructura HTML. La combinación de HTML y JavaScript es esencial para crear experiencias web modernas y atractivas.



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

- **Diseño Responsivo:** Bootstrap proporciona un sistema de cuadrícula flexible y componentes diseñados para ser receptivos; facilitando la creación de sitios web que se adaptan de manera efectiva a diferentes tamaños de pantalla, desde computadoras de escritorio hasta dispositivos móviles.
- **Ahorro de Tiempo:** Bootstrap ofrece un conjunto de componentes predefinidos y estilos CSS; esto significa que no es necesario empezar desde cero cada vez que se crea un nuevo proyecto, se pueden aprovechar componentes como botones, formularios, barras de navegación etc., lo que ahorra tiempo de desarrollo.
- **Compatibilidad entre Navegadores:** Bootstrap se esfuerza por ser compatible con la mayoría de los navegadores web modernos. Esto significa que puedes desarrollar aplicaciones web que funcionen de manera confiable en diferentes entornos de navegación.
- **Facilidad de Personalización:** se pueden modificar fácilmente los estilos CSS y los componentes para que se ajusten a las necesidades de los proyectos.
- **Interacción JavaScript:** incluye componentes interactivos basados en JavaScript, como modales, ventanas emergentes, carruseles y más. Esto permite agregar características avanzadas a tu sitio web sin tener que escribir código JavaScript personalizado desde cero.



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

- La semejanza corresponde a que ambos lenguajes pueden interactuar con el navegador web. JavaScript se ejecuta en el navegador del cliente, mientras que PHP se ejecuta en el servidor y genera contenido que se envía al navegador. Pueden ser incrustados en HTML: Tanto PHP como JavaScript pueden integrarse en páginas HTML para agregar funcionalidad dinámica. PHP se incrusta en el código HTML utilizando etiquetas especiales `<?php ?>`, mientras que JavaScript se agrega en el HTML usando etiquetas `<script>`.
- La diferencia es que PHP se centra en el procesamiento del lado del servidor y la generación de contenido dinámico, mientras que JavaScript se utiliza principalmente para mejorar la interactividad y la experiencia del usuario en el navegador. La elección de cuál usar depende de las necesidades específicas de un proyecto web.

JavaScript vs PHP			
For beginner programmers		✓	✓
For high salary		✓	✗
Performance		✓	✗
Front End development		✓	✗
Back End development		✗	✓

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

1. Etiqueta `<script>` en el HTML: Se puede agregar código JavaScript directamente en el archivo HTML utilizando la etiqueta `<script>`. Esto se hace generalmente dentro del elemento `<head>` o al final del elemento `<body>` de la página web.

2. Archivo externo de JavaScript: creando un archivo de JavaScript externo con extensión .js y luego enlazarlo a la página HTML utilizando la etiqueta <script>. Esto es útil cuando se tiene un código JavaScript más extenso o se desea reutilizar en varias páginas.
3. Eventos y manipulación del DOM: Se puede agregar código JavaScript directamente en los archivos HTML utilizando eventos y manipulación del DOM. Esto implica seleccionar elementos HTML y agregar escuchadores de eventos para responder a acciones del usuario.

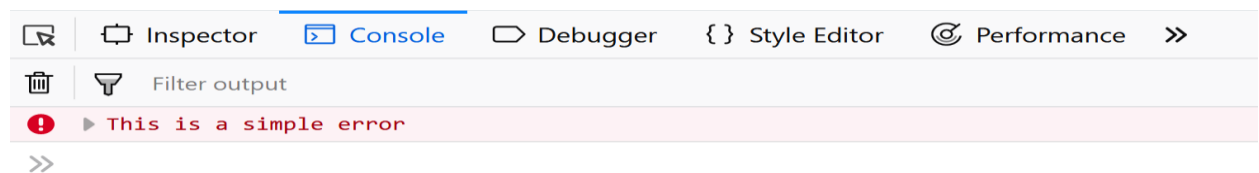
```
<HTML>
<HEAD>
<TITLE>Ejemplo01.htm</TITLE>

<SCRIPT LANGUAGE="JavaScript">
  //Visualizar un mensaje de bienvenida
  alert("¡Bienvenido a nuestra página!");
</SCRIPT>

</HEAD>
<BODY>
<a href='Ejemplo02.html'>Ir al siguiente ejemplo...</a>
</BODY>
</HTML>
```

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

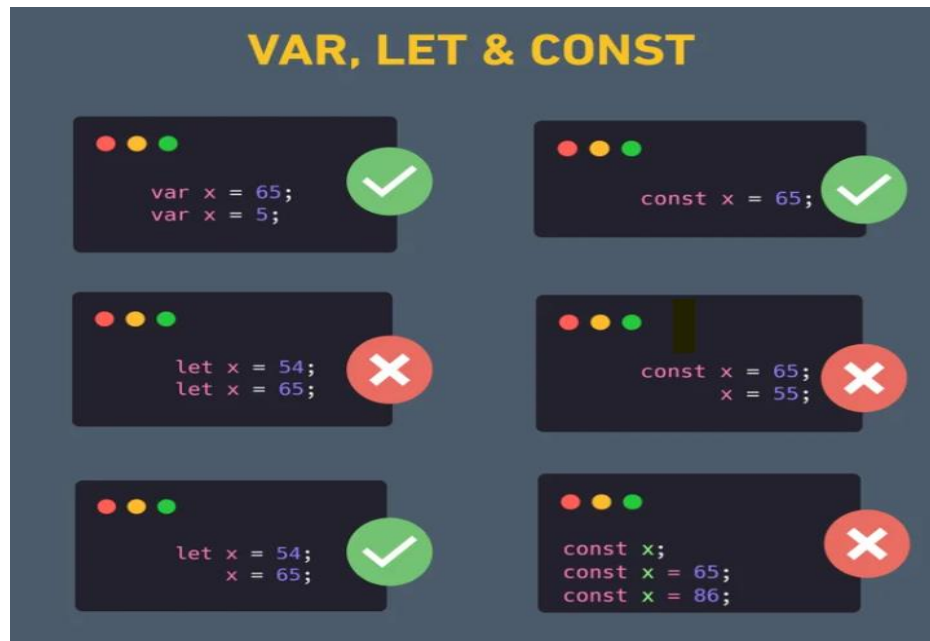
La función principal de la consola en JavaScript es proporcionar una herramienta de depuración de código y registro de información que permite a los desarrolladores ver mensajes, errores y resultados de sus scripts mientras trabajan en el desarrollo web.



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

- Declaración var: Tiene un alcance global si se declaran fuera de cualquier función o un alcance de función si se declaran dentro de una función y permite la reasignación.
- Declaración let: están limitadas al bloque de código más cercano, como dentro de una función o un bucle, permite la reasignación y un mejor control de alcance.

- Declaración const: Tiene un alcance de bloque, son constantes y no pueden ser reasignadas después de su declaración, aunque permite la modificación de propiedades de objetos o elementos de arreglos.



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

- Comentarios de una sola línea: Estos comentarios se crean utilizando // y se utilizan para agregar notas breves o explicaciones a una sola línea de código.

```
// Esto es un comentario de una sola línea
let x = 5; // Aquí declaramos una variable 'x' con un valor de 5
```

- Comentarios de varias líneas: Estos comentarios se crean utilizando /\* para iniciar el comentario y \*/ para finalizarlo. Son útiles para describir bloques de código, documentar funciones o proporcionar comentarios extensos.

```

/*
Este es un comentario de varias líneas.
Puedes usarlo para documentar funciones o
para agregar explicaciones más largas en el código.
*/

function suma(a, b) {
    return a + b;
}

```

#### 10. ¿Qué es ECMAScript6? Explique claramente.

Es la siguiente versión de JavaScript y define la sintaxis y el comportamiento del lenguaje JavaScript, introduce características que están diseñadas para facilitar el desarrollo de aplicaciones web complejas y escalables como por ejemplo el uso de `let` y `const` como alternativas a `var`, con un mejor manejo del alcance de las variables y la capacidad de declarar constantes.

Las funciones de flecha (`=>`) proporcionan una sintaxis más concisa y permiten evitar problemas de contexto (`this`) en funciones anidadas, permite extraer valores de objetos y arreglos de manera más concisa, introduce una sintaxis más orientada a objetos para la definición de clases en JavaScript, incorpora un sistema de módulos nativo para organizar y reutilizar código y facilita la gestión de operaciones asíncronas y la escritura de código más limpio.

En conclusión, ECMAScript 6 (ES6) es una versión importante del estándar ECMAScript que introdujo mejoras y nuevas características en JavaScript. Estas mejoras han contribuido en gran medida a la evolución del lenguaje y se utilizan ampliamente en el desarrollo web moderno.

```

1  // Función regular como expresión
2  const cuadrado = function(numero) {
3      return numero * numero;
4  }
5
6  // Función de flecha como expresión
7  const cuadrado = (numero) => {
8      return numero * numero;
9  }
10 console.log(cuadrado(3)); // 9
11
12 // Función con un único argumento y una sola línea de código
13 const cuadrado = numero => numero * numero;
14 console.log(cuadrado(5)); // 25

```



### **Conclusión del aprendizaje obtenido.**

Puedo concluir con el aprendizaje obtenido es que el mundo que nos ofrece el JavaScript es amplio. La historia de JavaScript, desde su creación en los años 90 hasta su evolución nos muestra cómo la tecnología web ha avanzado, Bootstrap nos facilita la vida, la relación de JavaScript y HTML es fundamental en el éxito del desarrollo web.

### **¿Qué enseñanza le brindo la tarea?**

Me llenó de conocimiento al leer todo el contenido como resultado a la investigación, a veces no nos damos cuenta de la importancia que tiene hacer trabajos investigativos ya que al ser individuales nos hacen salir de nuestra zona de confort. No tenía idea de lo que era ECMAScript6 y me impresiona que con el paso del tiempo cada vez hay algo nuevo en el mundo de la programación que viene siendo más moderno y viene a impactar el uso de estas; con sus cambios y mejoras para nuestra facilidad.