



**TWEB-07 Fundamentos de Programación Web SCV2**

**Alumno: Sebastián Portugal Barrantes**

**Profesor: Francisco José Jiménez Bonilla**

**Tarea #1**

**-2025-**

### **Cuestionario acerca de la programación web Java Script (JS).**

En el siguiente documento se mostrarán un máximo de diez preguntas, las cuales serán contestadas inmediatamente tras plantear cada una. Una vez terminada la reflexión al responder cada pregunta, se dará una conclusión con el aprendizaje desarrollado luego de la investigación y el contenido revisado en cada clase del curso.

#### **1. Escriba la historia de Java Script.**

Nacido originalmente como *Mocha*, JavaScript fue creado por Brendan Eich de Netscape Communications Corporations, durante los años 90 el cual fue evolucionando *LiveScript* antes de convertirse en *JavaScript*, cuando Netscape logró agregar la tecnología Java en su propio navegador en busca de agregar elementos más dinámicos. Esto coincidió con las etapas más tempranas de los navegadores en Internet, como *Mosaic* y donde los sitios web solo contaban con HTML como su única herramienta de creación.

Utilizando la solución de errores y respondiendo a la solicitud de funciones dentro del lenguaje, Eich trabajó en diseñar API con las que Mosaic pudiera interactuar con Netscape; estas API pronto se convertirían en las bases para desarrollar el Document Object Model (DOM), una clave para la interacción con HTML y XML.

A pesar de haber sido creada por Eich en Netscape, JAVASCRIPT surgió como marca registrada precisamente para evitar la apropiación de parte de otros servicios, como Microsoft, el cual tuvo que nombrar a su dialecto JavaScript: *JScript*, para evitar problemas con la marca.

## **2. ¿Qué es JavaScript?**

JavaScript es un lenguaje de programación del lado del cliente que, junto a HTML y CSS, conforma uno de los tres bloques fundamentales para la creación de sitios web modernos. Este ofrece el control sobre el comportamiento de diferentes elementos de la página y permite interactividad para las personas usuarias, desarrollando objetos, alertas y señales dinámicas que dan contexto de la interacción con la página, como errores, aciertos, animaciones y advertencias, por ejemplo, que no requieren refrescar la página de manera manual y que aun así ofrezcan diferentes rutas para que el usuario pueda tomar en su navegación.

## **3. ¿Cuál es la relación de HTML y JavaScript?**

Tal y como se mencionó previamente, HTML es uno de los bloques fundamentales para la creación de sitios web, este permite la creación de la estructura base del sitio, la cual después puede ser mejorada con la introducción de la tecnología de JavaScript, que agrega dinamismo y movimiento a elementos dentro del sitio.

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

Bootstrap es un framework donde se pueden crear sitios responsivos, principalmente destinados a móviles, con plantillas disponibles que utilizan JavaScript

para funcionar, requiriendo de su propios plugins en JavaScript para ser ejecutados.

Utilizando cuadrículas prediseñadas, quienes utilizan Bootstrap solo deben agregar el código para poder utilizar las herramientas que este ofrece.

## **5. ¿Qué semejanza y diferencia tienen los lenguajes web PHP y JavaScript?**

Personal Home Page Tools (PHP) es un lenguaje de scripting similar a JavaScript en cuanto a su función, ya que permite el desarrollo de sitios web y blogs de interacción dinámica con el usuario. Ambos trabajan con edición a tiempo real y esto permite que puedan ser modificados con un editor de texto, y con el tiempo permitieron la orientación a objetos dentro de sus comandos.

Las mayores diferencias entre ambos lenguajes de scripting residen en que PHP está pensado para desarrollarse en el backend de un sitio web, mientras que JavaScript originalmente fue creado para el frontend, aunque más adelante en su historia también dispone de tiempo de ejecución de backend. Otra de sus mayores diferencias es que PHP es un lenguaje del lado del servidor, lo que permite enviar contenido dinámico de una base de datos local o externa al usuario; mientras que JavaScript es un lenguaje pensado del lado del cliente, lo que quiere decir que los programas son ejecutados por el navegador que está procesando la información, con el riesgo de ver su indexación afectada por los motores de búsqueda y que no todos los navegadores puedan soportarlo de manera eficiente. Por último, una de sus diferencias también yace en su complementación con HTML y XML, donde JavaScript puede complementarse con ambos, mientras que PHP solo puede hacerlo con HTML.

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

Así como lo hemos hecho en clases, podemos utilizar distintos métodos para agregar código JS a una página web. La primera de ellas, por ejemplo, puede ser envolviendo el código encerrado dentro de la etiqueta `<script></script>`, la cual permitirá escribir directamente en el archivo HTML. La segunda es utilizando un archivo `.js` externo a la página HTML y se puede enlazar utilizando la etiqueta `<script></script>`, con el atributo `src`. Por último, otra forma de agregar el código JS a una página web es creando un evento directamente en HTML como `onclick`, `onmouseover`, entre otros, donde le indicamos al navegador el realizar una acción específica cuando alguno de estos ocurre.

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

La consola en JS sirve como una herramienta de depuración o *debugging* que les permite a los desarrolladores el ejecutar su código. Este se realiza por medio del objeto global `console` y sus diferentes variantes que permiten ver errores [`console.error()`], advertencias [`console.warn()`], entre otros. Esta se puede ver en el navegador y permite ver valores en tiempo real, así como detectar errores en el código.

## 8. ¿Cuál es la diferencia que existe en las declaraciones `var`, `let` y `const` en JS? Brinde un ejemplo de cada una.

La declaración `var` es reasignable, permite redeclaraciones (ser nombrada más de una misma vez en el mismo ámbito sin causar errores) y hoisting o elevación, mientras que `let` permite reasignación, pero no redeclaración y sí permite elevación. Por último, `const` no permite reasignación, ni redeclaración, pero sí hoisting.

Ejemplos de cada una sería:

```
function ejemplo() {
```

```
  if (true) {
```

```
    var x = 10;
```

```
  }
```

```
  console.log(x); // Muestra 10, porque var ignora el bloque
```

```
}
```

---

```
function ejemplo() {
```

```
  if (true) {
```

```
    let y = 20;
```

```
  }
```

```
  Console.log(y);
```

```
}
```

---

```
const z = [1, 2, 3];
```

```
var fruit = "🍏 Apple";  
let animal = "🐳 Whale";  
const ball = "🏀 Basketball";
```

## 9. ¿Cuál es la función de minificar archivos JavaScript?

La minificación de los archivos en JavaScript permite que se reduzca el tamaño de estos y elimina los caracteres innecesarios como espacios en blanco, saltos de línea y comentarios o nombres de variables que son muy largos. Se realiza para reducir el tiempo de carga y para ahorrar el ancho de banda.

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

ECMAScript6 es el lenguaje base en el que se basa JavaScript. Este estándar de lenguaje es el que se utiliza para asegurar la operabilidad y estandarización de los lenguajes de scripting como JavaScript. El *Script6* de su nombre significa que se encuentra en su sexta edición y esta introduce el scope de bloque, mejorando las variables y añadiendo las funciones flecha, así como añadir las declaraciones *let* y *const*, con el objetivo de mejorar la sintaxis y el manejo asincrónico de los códigos.

## Conclusión

El desarrollo de tecnologías como JavaScript ha permitido que se creen comunidades donde desarrolladores se apoyan entre sí para crear contenido y permitirse seguir alimentando la manera en que se dinamiza y crean sitios web. Todo ello buscando el acceso para quienes se interesen en estas áreas y que las personas usuarias de la Internet puedan interactuar de manera cada vez más intuitiva y directa con el contenido que desarrollan.

Si bien ya existían formas de crear sitios web previo a su creación, JavaScript y su subsecuente desarrollo ha permitido diversificar la forma en que los usuarios crean sitios dinámicos y que pueden ser utilizados en diferentes ámbitos: personales, educativos y profesionales.



## Referencias

**Azion.** (s.f.). *JavaScript: historia y evolución*. Azion.

<https://www.azion.com/es/learning/jamstack/javascript-historia-y-evolucion/>

**Filis Antillán.** (s.f.). *Var, let y const en JavaScript*. Filis Antillán.

<https://filisantillan.com/bits/var-let-const/>

**Google Developers.** (s.f.). *Minify resources*. Google Web Fundamentals.

[https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/optimize-encoding-and-transfer#minify\\_resources](https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/optimize-encoding-and-transfer#minify_resources)

**HubSpot.** (s.f.). *Diseño web: HTML, CSS y JavaScript*. HubSpot Blog.

<https://blog.hubspot.com/marketing/web-design-html-css-javascript>

**IONOS.** (s.f.). *PHP vs JavaScript: diferencias y usos*. IONOS Digital Guide.

<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/php-vs-javascript/>

**Bootstrap.** (s.f.). *Introducción*. Documentación Bootstrap.

<https://getbootstrap.com/docs/5.0/getting-started/introduction/>

**Mozilla.** (s.f.). *Características de ECMAScript 6*. MDN Web Docs.

[https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Statements#caracter%C3%ADsticas\\_de\\_ecmascript\\_6](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Statements#caracter%C3%ADsticas_de_ecmascript_6)

**Mozilla.** (s.f.). *Console - Web APIs | MDN*. MDN Web Docs.

<https://developer.mozilla.org/en-US/docs/Web/API/console>

**Mozilla.** (s.f.). *Declaraciones de variables*. MDN Web Docs.

[https://developer.mozilla.org/es/docs/Learn/JavaScript/First\\_steps/Variables](https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Variables)

**Mozilla.** (s.f.). *Introducción a JavaScript*. MDN Web Docs.

<https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Introduction>

**Mozilla.** (s.f.). *var, let y const*. MDN Web Docs.

<https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Statements/var>

**TechTarget.** (s.f.). *¿Qué es Bootstrap?*. TechTarget.

<https://www.techtarget.com/whatis/definition/bootstrap>

**W3Schools.** (s.f.). *PHP tutorial*. W3Schools. <https://www.w3schools.com/php/>