



Universidad  
**CENFOTEC**  
SOMOS LO QUE SABEMOS

# 1 Tarea

---



## Cuestionario JavaScript

{Alumno: Juan Carlos Caamaño}

{Profesor: Francisco Jiménez}

{Curso: Fundamentos Programación Web}



## Tabla de Contenido

1. Historia Del Lenguaje Javascript .....	3
2. ¿Por Qué Se Debe Aprender Javascript? .....	4
3. Relación Entre HTML Y Javascript .....	5
4. Beneficios De Usar Bootstrap Para Sitios Y Aplicaciones Web En JS.....	5
5. Semejanza Y Diferencia Entre PHP Y Javascript .....	5
6. Formas De Agregar Código JS En Una Página Web.....	6
7. Función Principal De La Consola En JS.....	6
8. Diferencia Entre Var, Let Y Const En JS .....	7
Var:.....	7
Const:.....	7
9. Tipos De Comentarios En JS.....	8
Comentarios De Una Sola Línea:.....	8
Comentarios De Múltiples Líneas:.....	8
10.¿Qué Es EcmaScript6?.....	9
1. Declaraciones Let Y Const: .....	9
2. Funciones Flecha:.....	9
3. Plantillas De Literales: .....	9
4. Parámetros Por Defecto: .....	9
5. Desestructuración: .....	10
6. Clases: .....	10
7. Módulos:.....	10
8. Promesas: .....	10
11. Conclusiones Finales. ....	11
12. Referencias Bibliográficas:.....	12

## 1. Historia del lenguaje JavaScript

JavaScript, un lenguaje de programación esencial en el desarrollo web moderno, tiene una historia rica y fascinante que se remonta a mediados de la década de 1990. En 1995, Brendan Eich, un ingeniero de Netscape, creó JavaScript en tan solo **diez** días. Inicialmente llamado Mocha, luego LiveScript, y finalmente JavaScript, el lenguaje fue diseñado para ser un complemento ligero de Java, aprovechando su popularidad en ese momento (Eich, 1995).



En 1996, Microsoft lanzó JScript, una versión de JavaScript para su navegador Internet Explorer, lo que marcó el inicio de una competencia entre navegadores (Microsoft, 1996). Ese mismo año, Netscape y Sun Microsystems colaboraron para enviar JavaScript a ECMA International, resultando en la primera edición de ECMAScript en 1997 (ECMA International, 1997).

El crecimiento de JavaScript continuó con la publicación de ECMAScript 2 en 1998, que se alineó con el estándar ISO/IEC 16262 (ECMA International, 1998). En 1999, ECMAScript 3 introdujo mejoras significativas, incluyendo expresiones regulares y un mejor manejo de excepciones (ECMA International, 1999).

El año 2005 fue un punto de inflexión para JavaScript con el auge de AJAX (Asynchronous JavaScript and XML), que permitió la creación de aplicaciones web más interactivas y dinámicas, consolidando a JavaScript como una herramienta esencial para el desarrollo web (Garrett, 2005).

En 2009, ECMAScript 5 trajo nuevas características como el modo estricto y métodos adicionales para arrays, mejorando la sintaxis y funcionalidad del lenguaje (ECMA International, 2009). Sin embargo, fue en 2015 cuando JavaScript experimentó una de sus actualizaciones más importantes con ECMAScript 6 (ES6), que introdujo clases, módulos, promesas y funciones flecha, entre otras mejoras (ECMA International, 2015).

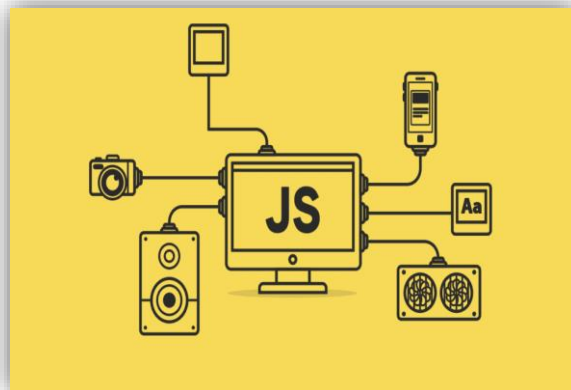
Desde entonces, se han lanzado actualizaciones anuales que incluyen características como `async/await` y mejoras continuas en la sintaxis y funcionalidad, manteniendo a JavaScript actualizado y relevante en el panorama tecnológico actual (ECMA International, 2016-2023).

JavaScript ha evolucionado de ser un simple lenguaje de scripting a una herramienta poderosa utilizada en una amplia variedad de aplicaciones, desde desarrollo web hasta aplicaciones móviles y servidores con Node.js, demostrando su versatilidad y adaptabilidad en el mundo de la programación (Node.js Foundation, 2023).

## 2. ¿Por qué se debe aprender JavaScript?

JavaScript es esencial para el desarrollo web moderno.

Permite crear sitios web interactivos y dinámicos. Es fácil de aprender, tiene una gran demanda en el mercado laboral y es versátil, ya que se puede usar tanto en el frontend como en el backend.



### **3. Relación entre HTML y JavaScript**

HTML (Lenguaje de Marcado de Hipertexto) y JavaScript son fundamentales en el desarrollo web, cada uno desempeñando roles específicos que se complementan para crear páginas web dinámicas e interactivas. HTML establece la estructura y el contenido de una página web mediante etiquetas que definen elementos como títulos, párrafos, imágenes y enlaces. En cambio, JavaScript es un lenguaje de programación que añade interactividad y dinamismo a las páginas web.

JavaScript se incorpora en las páginas HTML mediante la etiqueta `<script>`, que puede contener código JavaScript directamente en el documento HTML o enlazar a archivos JavaScript externos. Esta integración permite a JavaScript manipular el Modelo de Objeto del Documento (DOM) generado por HTML, lo que significa que puede agregar, eliminar o modificar elementos HTML y sus estilos de manera dinámica.

Además, HTML permite definir eventos como clics de ratón y pulsaciones de teclas, a los cuales JavaScript puede responder para realizar acciones específicas, como mostrar alertas, validar formularios o cambiar el contenido de la página. Con JavaScript, es posible crear efectos visuales, animaciones y formularios interactivos, mejorando notablemente la experiencia del usuario en la página web.

### **4. Beneficios de usar Bootstrap para sitios y aplicaciones web en JS**

Bootstrap facilita el desarrollo rápido con plantillas preconstruidas, es fácil de usar, asegura la compatibilidad entre navegadores y proporciona un diseño responsivo. Esto ahorra tiempo y esfuerzo en el desarrollo de interfaces de usuario.

### **5. Semejanza y diferencia entre PHP y JavaScript**

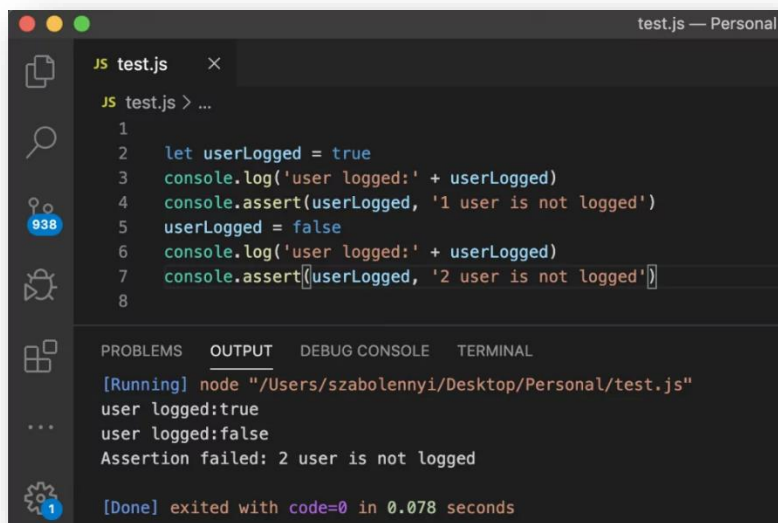
Ambos son lenguajes de scripting, pero PHP se ejecuta en el servidor y JavaScript en el cliente. PHP es sincrónico y JavaScript es asincrónico. Ambos son débilmente tipados y dinámicos, pero PHP es más adecuado para la gestión de bases de datos .

## 6. 3 Formas de agregar código JS en una página web

1. **Internamente:** Dentro de una etiqueta `<script>` en el HTML.
2. **Externamente:** Enlazando un archivo .js externo con la etiqueta `<script src="archivo.js"></script>`.
3. **Inline:** Directamente en los atributos de los elementos HTML, como `onclick="funcion()"`.

## 7. Función principal de la consola en JS

La consola en JavaScript se utiliza para depurar y probar código. Permite a los desarrolladores ver mensajes de error, advertencias y resultados de expresiones en tiempo real.



The screenshot shows a code editor window titled "test.js — Personal". The editor contains the following JavaScript code:

```
1
2 let userLogged = true
3 console.log('user logged:' + userLogged)
4 console.assert(userLogged, '1 user is not logged')
5 userLogged = false
6 console.log('user logged:' + userLogged)
7 console.assert(!userLogged, '2 user is not logged')
8
```

Below the code editor, the "OUTPUT" tab is selected, showing the following output:

```
[Running] node "/Users/szabolennyi/Desktop/Personal/test.js"
user logged:true
user logged:false
Assertion failed: 2 user is not logged

[Done] exited with code=0 in 0.078 seconds
```

## 8. Diferencia entre var, let y const en JS

**var:**

**Ámbito:** Las variables declaradas con var tienen un ámbito global o de función. Esto significa que si se declara una variable con var dentro de una función, solo estará disponible dentro de esa función.

**Reasignación:** Las variables var pueden ser reasignadas y redeclaradas dentro de su ámbito.

**Hoisting:** Las variables var son “hoisted”, lo que significa que su declaración se mueve al inicio de su ámbito durante la fase de compilación, pero su inicialización no.

**let:**

**Ámbito:** Las variables declaradas con let tienen un ámbito de bloque, lo que significa que solo están disponibles dentro del bloque donde se declararon (por ejemplo, dentro de un if, for, etc.).

**Reasignación:** Las variables let pueden ser reasignadas, pero no redeclaradas dentro del mismo ámbito.

**Hoisting:** Las variables let también son “hoisted”, pero no se pueden usar antes de su declaración debido a la “zona temporal muerta”.

**const:**

**Ámbito:** Similar a let, las variables const tienen un ámbito de bloque.

**Reasignación:** Las variables const no pueden ser reasignadas ni redeclaradas. Una vez asignado un valor, no se puede cambiar.

**Hoisting:** Al igual que let, las variables const son “hoisted” pero no se pueden usar antes de su declaración.

## 9. Tipos de comentarios en JS

En JavaScript, existen dos tipos principales de comentarios:

### Comentarios de una sola línea:

- Se inician con `//` y comentan todo el texto que sigue en esa línea.
- Son útiles para breves anotaciones o para desactivar temporalmente una línea de código.
- `//` Esto es un comentario de una sola línea

### Comentarios de múltiples líneas:

- Comienzan con `/*` y terminan con `*/`.
- Se utilizan para comentarios más largos que abarcan varias líneas.
- `/*` Esto es un comentario
- de múltiples líneas.
- Puede abarcar varias líneas. `*/`
- `let y = 20;`





## 10. ¿Qué es ECMAScript6?



ECMAScript 6, también conocido como ES6 o ECMAScript 2015, es una versión importante del estándar ECMAScript que define cómo debe funcionar JavaScript. Fue lanzado en 2015 y trajo muchas mejoras y nuevas características al lenguaje, haciendo que JavaScript sea más potente y eficiente.

Aquí tienes un resumen de algunas de las características más destacadas de ES6:

### 1. Declaraciones **let** y **const**:

- **let** permite declarar variables con un ámbito de bloque, lo que mejora la gestión del alcance de las variables.
- **const** se utiliza para declarar constantes, es decir, variables cuyo valor no puede cambiar una vez asignado.

### 2. Funciones **flecha**:

- Proporcionan una sintaxis más concisa para escribir funciones.
- No tienen su propio **this**, lo que las hace útiles en ciertos contextos, como en métodos de objetos.

### 3. Plantillas de literales:

- Permiten incrustar expresiones dentro de cadenas de texto utilizando la sintaxis `${expresión}`.
- Facilitan la creación de cadenas de texto complejas.

### 4. Parámetros por defecto:

- Permiten definir valores por defecto para los parámetros de las funciones.

## 5. **Desestructuración:**

- Facilita la extracción de valores de arrays u objetos y asignarlos a variables individuales.

## 6. **Clases:**

- Introducen una sintaxis más clara y concisa para trabajar con objetos y herencia en JavaScript.

## 7. **Módulos:**

- Permiten dividir el código en módulos reutilizables y gestionar dependencias de manera más eficiente.

## 8. **Promesas:**

- Facilitan el manejo de operaciones asíncronas, como llamadas a APIs, de una manera más legible y manejable.

## 11. Conclusiones finales.

Puedo decir que el aprendizaje obtenido durante esta tarea de investigación fue bastante amplio, dado es un lenguaje con bastante trayectoria, para mi JavaScript ha demostrado ser un lenguaje de programación fundamental en el desarrollo web moderno. Desde su creación en 1995 por Brendan Eich, ha evolucionado significativamente, adaptándose a las necesidades cambiantes de la tecnología y el desarrollo web. Su integración con HTML y su capacidad para manipular el **DOM** han permitido la creación de sitios web interactivos y dinámicos.

El lenguaje ha pasado de ser una herramienta sencilla a una plataforma poderosa utilizada en diversas aplicaciones, incluyendo desarrollo web, aplicaciones móviles y servidores con Node.js. Las actualizaciones constantes y la incorporación de nuevas características, como las introducidas en ECMAScript 6, han mantenido a JavaScript relevante y en demanda en el mercado laboral.

Aprender JavaScript no solo es esencial para cualquier desarrollador web, sino que también ofrece una gran versatilidad y oportunidades en el campo de la programación. Su facilidad de aprendizaje y su amplia aplicación en el frontend y backend lo convierten en una habilidad valiosa y buscada.

## 12. Referencias bibliográficas:

1. [JavaScript - Wikipedia](#)
2. [JavaScript History - W3Schools](#)
3. [Why Learn JavaScript? A Beginner's Guide - CareerFoundry](#)
4. [7 Critical Reasons Beginner Programmers Should Learn JavaScript in 2022](#)
5. [HTML and JavaScript Integration Basics - daily.dev](#)
6. [Understanding the Document Object Model \(DOM\) - W3docs](#)
7. [8 Reasons to Use Bootstrap for Web Development - Creative Tim](#)
8. [Ten Reasons to Use Bootstrap 5 for Better UI Development - Jade Global](#)
9. [PHP vs JavaScript: In-Depth Comparison of the Two Scripting ... - Kinsta](#)
10. [Difference between JavaScript and PHP - GeeksforGeeks](#)
11. [HTML and Javascript \(With Examples\) - Programiz](#)
12. [Learn JavaScript for Beginners – JS Basics Handbook - freeCodeCamp.org](#)
13. [An In-Depth Introduction to JavaScript: From Its Origins to Modern ...](#)
14. [Top 5 Reasons to Learn JavaScript - GeeksforGeeks](#)
15. [JavaScript History & ECMAScript](#)
16. [The relationship between HTML, CSS and JavaScript explained by building ...](#)
17. [What is the relationship between HTML, JavaScript, and CSS?](#)
18. [“Demystifying Bootstrap: Understanding Its Role in Web Development”](#)
19. [Why is Bootstrap the Perfect Framework for Your Web Application](#)
20. [The Pros and Cons of Using Bootstrap in Web Development - Fabrx Design](#)
21. [What is Bootstrap and How to Use it in Web Development? - F5 Studio](#)
22. [PHP vs JavaScript – Difference Between Them - Guru99](#)
23. [PHP vs JavaScript: Key Differences and Real-World Applications - Halfnine](#)
24. [PHP vs JavaScript: How to Choose the Best Language for Your Project](#)