Tarea 1

Fundamentos de Programación Web
CUESTIONARIO – JavaScript

Estudiante: Ricardo Botero Zaccour

Profesor: Francisco Jimenez Bonilla
Curso: Fundamentos de Programación Web
Descripción: Documento que responde 10 preguntas sobre JavaScript y su ecosistema.
Fecha: 18 de Septiembre del 2025
Institución: CENFOTEC

Desarrollo

1) ¿Escriba la historia de JavaScript?

JavaScript fue creado en 1995 por Brendan Eich en Netscape (nombre inicial Mocha, luego LiveScript). En 1997, Ecma International publicó la primera especificación ECMAScript (ES1). En 2009 apareció ES5 (gran compatibilidad). En 2015 llegó ES6/ES2015, que incorporó let/const, funciones flecha, clases y módulos; desde entonces el estándar se actualiza cada año manteniendo compatibilidad hacia atrás.

Edition	Date published	Name	Changes from prior edition	Editor
1	June 1997		First edition based on JavaScript 1.1 as implemented in Netscape Navigator 3.0.[1]	Guy L. Steele Jr.
2	June 1998		Editorial changes to keep the specification fully aligned with ISO/IEC 16262:1998.	Mike Cowlishaw
3	December 1999		Based on JavaScript 1.2 as implemented in Netscape Navigator 4.0.[2] Added regular expressions, better string handling, new control statements, try/catch exception handling, tighter definition of errors, formatting for numeric output, and other enhancements	Mike Cowlishaw
4	*Abandoned* (last draft 30 June 2003)	ECMAScript 4 (ES4)	The 4th edition was abandoned due to political differences concerning language complexity. Many features proposed for the 4th edition have been completely dropped; some were incorporated into the sixth edition.	
5	December 2009		Adds "strict mode", a subset intended to provide more thorough error checking and avoid error-prone constructs. Clarifies many ambiguities in the 3rd edition specification, and accommodates behavior of real-world implementations that differed consistently from that specification. Adds some new features, such as getters and setters, library support for JSON, and more complete reflection on object properties.[3]	Pratap Lakshman, Allen Wirfs-Brock
5.1	June 2011		Changes to keep the specification fully aligned with ISO/IEC 16262:2011.	Pratap Lakshman, Allen Wirfs-Brock
6	June 2015[4]	ECMAScript 2015 (ES2015)	See #6th edition – ECMAScript 2015	Allen Wirfs-Brock
7	June 2016[5]	ECMAScript 2016 (ES2016)	See #7th edition – ECMAScript 2016	Brian Terlson
8	June 2017[6]	ECMAScript 2017 (ES2017)	See #8th edition – ECMAScript 2017	Brian Terlson
9	June 2018[7]	ECMAScript 2018 (ES2018)	See #9th edition – ECMAScript 2018	Brian Terlson
10	June 2019[8]	ECMAScript 2019 (ES2019)	See #10th edition – ECMAScript 2019	Brian Terlson, Bradley Farias, Jordan Harband
11	June 2020[9]	ECMAScript 2020 (ES2020)	See #11th edition – ECMAScript 2020	Jordan Harband, Kevin Smith
12	June 2021[10]	ECMAScript 2021 (ES2021)	See #12th edition – ECMAScript 2021	Jordan Harband, Shu-yu Guo, Michael Ficarra, Kevin Gibbons
13	June 2022[11]	ECMAScript 2022 (ES2022)	See #13th edition – ECMAScript 2022	Shu-yu Guo, Michael Ficarra, Kevin Gibbons
14	June 2023[12]	ECMAScript 2023 (ES2023)	See #14th edition – ECMAScript 2023	Shu-yu Guo, Michael Ficarra, Kevin Gibbons
15	June 2024[13]	ECMAScript 2024 (ES2024)	See #15th edition – ECMAScript 2024	Shu-yu Guo, Michael Ficarra, Kevin Gibbons
16	June 2025[14]	ECMAScript 2025 (ES2025)	See #16th edition – ECMAScript 2025	Shu-yu Guo, Michael Ficarra, Kevin Gibbons
17	(pending)	ECMAScript 2026 (ES2026)	Pending, see features being considered: #ES.Next	(pending)

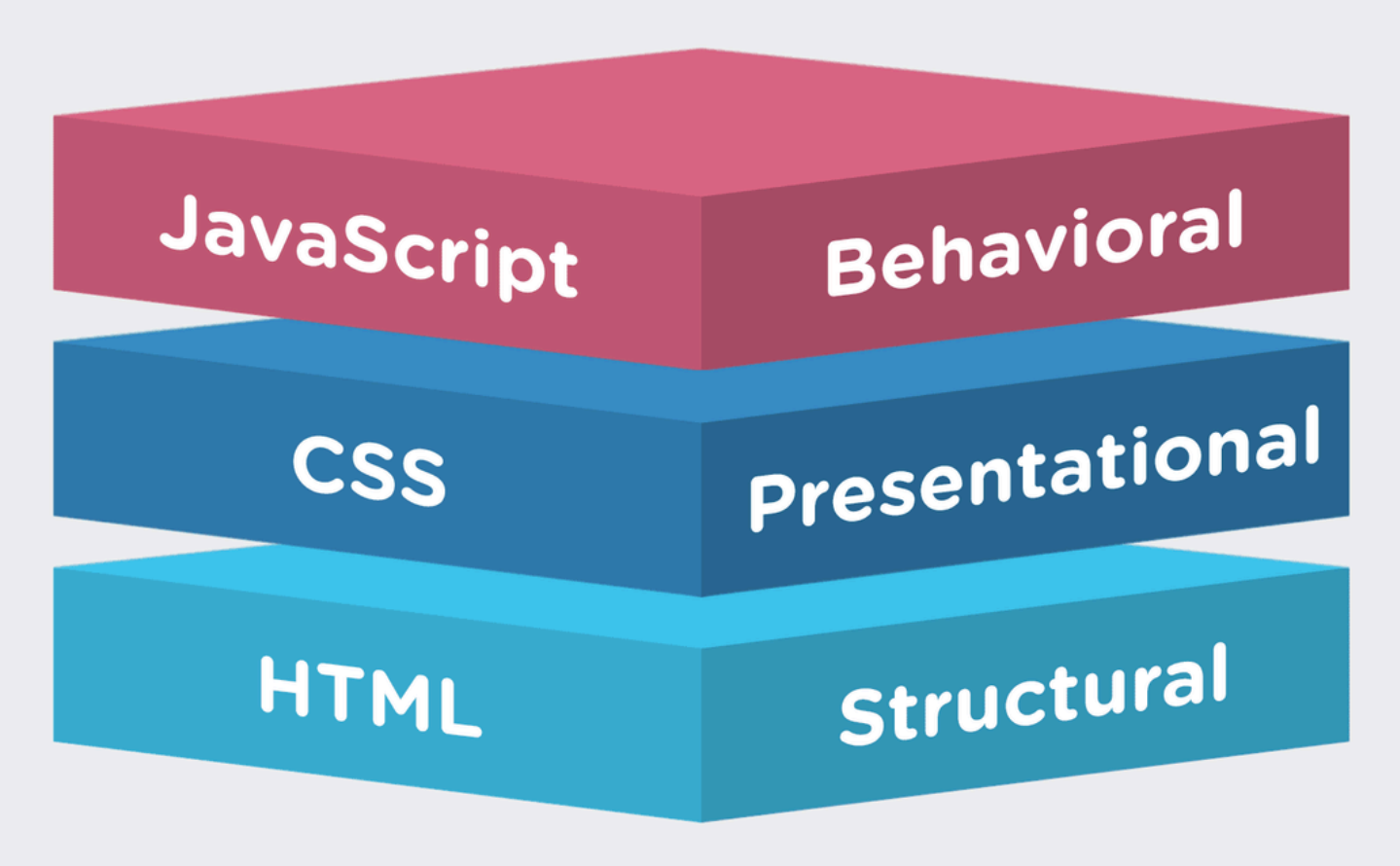
2) ¿Qué es JavaScript?

Lenguaje interpretado y multiparadigma (imperativo, funcional, orientado a objetos) usado para dotar de interactividad a la web.

- En el navegador: manipula el DOM, gestiona eventos y consume datos remotos.
- Fuera del navegador: con Node.js crea utilidades, servidores HTTP y scripts de automatización.

3) ¿Cuál es la relación entre HTML y JavaScript?

HTML define la estructura; CSS la presentación; JavaScript el comportamiento (interacciones, validaciones y actualización dinámica). JS accede a los elementos mediante el DOM y responde a eventos como click/submit.



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

Bootstrap ofrece componentes listos (botones, grillas, modales) y un sistema responsivo. Beneficios:

- Prototipado rápido y consistente.
- Compatibilidad multinavegador.
- Menos CSS manual; integración sencilla con JS para inicializar componentes y escuchar eventos.

5) ¿Semejanzas y diferencias entre PHP y JavaScript?

Semejanzas: usados en web, trabajan con strings/arrays/objetos, consumen APIs.
Diferencias:

- PHP corre típicamente en el servidor y genera HTML por petición.

- JavaScript corre principalmente en el cliente (actualiza la UI sin recargar) y también puede correr en servidor con Node.js.
- PHP sigue el ciclo petición–respuesta; JS es single-threaded con event loop e integración nativa con el DOM.

6) Cite 3 formas de agregar código JS en una página

1. Archivo externo: <script src="app.js"></script> (recomendado).
2. Bloque interno: <script> ... </script> dentro del HTML.
3. Inline por evento: <button onclick="saludar()">.
   Opcional moderno: <script type="module" src="main.js"></script> para módulos ES.

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

Permite inspeccionar y depurar: ver console.log/warn/error, evaluar expresiones en tiempo real, revisar objetos/arreglos y seguir el flujo de ejecución. Es clave para detectar errores y comprender el estado de la aplicación.

8) Diferencia entre var, let y const (con ejemplo)

- var → ámbito de función o global; permite redeclarar; sufre hoisting con undefined.
- let → ámbito de bloque; reasigna pero no redeclara en el mismo bloque; TDZ.
- const → ámbito de bloque; no reasigna la referencia (los objetos referenciados sí pueden mutar); TDZ.

Ejemplos:
var x = 1; x = 2; var x = 3; (redeclaración válida).
let y = 1; y = 2; válido; let y = 3; en el mismo bloque → error.
const PI = 3.1416; no se puede reasignar.

9) ¿Cuál es la función de minificar archivos JavaScript?

La minificación reduce el tamaño eliminando espacios/saltos y, a veces, renombrando identificadores seguros. Mejora tiempos de carga y rendimiento en producción sin cambiar la lógica del script.

10) ¿Qué es ECMAScript 6 (ES6)?

ES6/ES2015 (2015) modernizó JavaScript con: let/const, arrow functions, plantillas literales, clases, módulos, Promise, desestructuración, parámetros por defecto, Map/Set, entre otros. Desde 2015 las mejoras son anuales.

Conclusión del aprendizaje obtenido

Consolidé fundamentos de JS: su historia y estandarización, el papel junto a HTML/CSS y buenas prácticas como separar scripts y minificar para producción. Comprendí el uso correcto de var, let y const, y cómo ES6 modernizó la sintaxis y el modelo de módulos. Reforcé el uso de la consola y el valor de frameworks CSS (como Bootstrap) para prototipos responsivos y consistentes.

Referencias web

- MDN Web Docs – Guías de JavaScript y DOM.
- ECMA International – Especificación ECMAScript.
- Bootstrap – Documentación oficial.