

Actividades Unidad 1

* Se valorará positivamente el uso de hoja de estilos y en general la presentación de la página web en cada ejercicio.

1. Define la siguiente lista de términos. Para cada uno de ellos debes indicar su propósito, si pertenecen al lado servidor o cliente y al menos dos características diferenciadoras:

1. HTML5

Propósito: Lenguaje de marcado estándar para la estructura de páginas web.

Lado: Cliente.

Características: Introduce nuevas etiquetas semánticas como <article>, <section>, <header>, y <footer>. Soporte nativo para multimedia (video, audio) sin necesidad de plugins externos.

2. PHP

Propósito: Lenguaje de programación de servidor utilizado para el desarrollo de aplicaciones web dinámicas.

Lado: Servidor.

Características:

Integración sencilla con bases de datos como MySQL.

Es un lenguaje interpretado y de código abierto, popular en sistemas de gestión de contenidos como WordPress.

3. CSS

Propósito: Lenguaje de estilo utilizado para describir la presentación de documentos HTML.

Lado: Cliente.

Características: Facilita el diseño responsivo mediante media queries. Permite el uso de animaciones y transiciones para mejorar la interacción del usuario.

4. jQuery

Propósito: Biblioteca de JavaScript diseñada para simplificar la manipulación del DOM, manejo de eventos, y animaciones.

Lado: Cliente.

Características: Proporciona una sintaxis simplificada para la manipulación del DOM y la interacción con el backend. Gran compatibilidad con navegadores antiguos, aunque hoy en día su uso ha disminuido por el avance de JavaScript nativo.

5. Node.js

Propósito: Entorno de ejecución de JavaScript en el lado del servidor.

Lado: Servidor.

Características: Permite crear aplicaciones web en tiempo real gracias a su arquitectura basada en eventos y no bloqueante. Utiliza el motor V8 de Google Chrome para ejecutar código JavaScript del lado del servidor.

6. React.js

Propósito: Biblioteca de JavaScript para construir interfaces de usuario interactivas.

Lado: Cliente (aunque puede utilizarse en el servidor con SSR - Server-Side Rendering).

Características: Basado en un enfoque de componentes reutilizables que permiten un desarrollo más modular. Utiliza el Virtual DOM para mejorar el rendimiento de la actualización de interfaces.

7. AJAX

Propósito: Técnica para realizar solicitudes asíncronas al servidor sin recargar la página.

Lado: Cliente.

Características: Permite la actualización parcial de contenidos en una página web. Utiliza XMLHttpRequest o Fetch API para realizar solicitudes HTTP.

8. Fetch API

Propósito: Interfaz moderna de JavaScript para realizar solicitudes HTTP asíncronas.

Lado: Cliente.

Características:

Promesas en lugar de callbacks, lo que simplifica el manejo de respuestas y errores.

Es más limpio y moderno que XMLHttpRequest para interactuar con APIs.

9. WebSocket

Propósito: Protocolo de comunicación para conexiones bidireccionales y persistentes entre cliente y servidor.

Lado: Cliente y Servidor.

Características:

Comunicación en tiempo real sin la necesidad de realizar solicitudes repetitivas (polling).

Menor latencia y sobrecarga en comparación con las conexiones HTTP tradicionales.

10. Web Worker

Propósito: API que permite ejecutar scripts en segundo plano en hilos separados del hilo principal de la aplicación.

Lado: Cliente.

Características:

Mejora el rendimiento al permitir la ejecución de tareas intensivas sin bloquear la interfaz de usuario.

No tiene acceso directo al DOM, lo que garantiza mayor seguridad y eficiencia.

11. ECMAScript

Propósito: Especificación estándar para el lenguaje de programación JavaScript.

Lado: Cliente y Servidor.

Características:

Introduce nuevas funcionalidades con cada versión, como promesas, async/await, clases, módulos, etc.

Estándar mantenido por ECMA International que establece las bases para los motores de JavaScript en navegadores y servidores (Node.js).

2. Responde y justifica tu respuesta:

- ¿Podemos ocultar el código escrito en JavaScript de una página web?

No, no se puede ocultar completamente el código JavaScript en una página web. La razón es que JavaScript se ejecuta en el lado del cliente (navegador) y cualquier usuario puede suficiente conocimiento técnico puede verlo y analizarlo.

Justificación: El código JavaScript se descarga en el navegador del usuario junto con el HTML y CSS, por lo tanto, siempre estará disponible para quien sepa cómo acceder a él. Herramientas de ofuscación o minificación solo lo hacen menos legible, pero no invisible.

- Ventajas de incluir el código JavaScript en archivos en lugar del propio documento.

Mejor rendimiento: Al colocar el código JavaScript en archivos externos, estos pueden ser almacenados en la caché del navegador. Esto significa que si los usuarios visitan varias páginas de un sitio que utilizan el mismo archivo JavaScript, no tendrán que descargarlo repetidamente.

Mantenimiento más sencillo: Mantener el JavaScript en archivos separados facilita la organización del código y su reutilización en diferentes páginas. Además, se facilita la depuración y las actualizaciones, ya que los cambios realizados en un archivo externo se reflejarán en todas las páginas que lo usen.

- ¿Para qué sirve y cómo funciona la función `console.log()`?

Propósito: La función `console.log()` se utiliza para imprimir mensajes o valores en la consola del navegador. Es una herramienta fundamental para depurar y verificar el comportamiento del código JavaScript mientras se desarrolla.

Funcionamiento: Cuando se llama a `console.log(valor)`, el valor especificado (puede ser una variable, un mensaje de texto o un objeto) se muestra en la consola del navegador. Esto no afecta al comportamiento del sitio para el usuario final, pero ayuda al desarrollador a monitorear lo que ocurre en el código.

3. Realiza un programa en JavaScript que realice la tabla de sumar del 5, la tabla de multiplicar del 7 y la tabla de dividir del 9.

4. Utilizando la estructura for, crear un script que dado un número entero:

- Diga si es par o impar.
- Calcule el factorial.

5. Realiza un programa en JavaScript que, dada una nota entre 0 y 10, determine su calificación:

- 0-5: Suspenso
- 6: Bien
- 7-8: Notable
- 9-10: Sobresaliente

Debe tener en cuenta cuando el valor introducido no sea un número o no se introduzca ningún dato.

6. Un palíndromo es una palabra, frase, número, u otra secuencia de caracteres que se lee de la misma forma hacia adelante que hacia atrás (por ejemplo "racecar" "luz azul" o "1234123"). Realiza un programa en JavaScript que compruebe si los caracteres introducidos son un palíndromo.

7. Realiza un programa en JavaScript que emule a una calculadora. Se deben introducir dos operandos y una operación (suma, resta, multiplicación o división) y se debe mostrar el resultado.

8. Crea una estructura de decisión que permita identificar la talla de una prenda de ropa a partir de las tallas europeas. Los valores posibles de las tallas europeas serían XL, XXL, XL, L, M, XS, S y la talla esperada sería Grande, Mediana, Pequeña.

Grande={XXL, XL, L} Mediana={M} Pequeña={XS, S}