1. HTML Semántico y Accesibilidad

a) Ventaja principal del HTML semántico:

El HTML semántico le da significado real a las partes de la página. En lugar de usar solo <div>, el uso de etiquetas como <header>, <nav>, <main> explican que es cada parte en lugar de usar <div> por ejemplo. Haciendo esto las paginas que implementan correctamente html semántico tienen una ventaja competitiva de SEO por sobre las que no lo utilizan, ayuda a los buscadores a entender mejor el contenido y también ayuda a las personas con discapacidades visuales que usan lectores de pantalla.

b) Dos etiquetas HTML semánticas importantes:

<header>: Define la cabecera de la página, donde normalmente va el logo y navegación principal

<main>: indica el contenido principal de la página, lo más importante que el usuario viene a ver

Estas dos etiquetas dan semántica de la estructura de la pagina

c) Técnica de accesibilidad:

El atributo alt en las imágenes. Es importante porque describe qué hay en la imagen para personas que no pueden verla, permitiendo que los lectores de pantalla les expliquen el contenido visual.

2. CSS: Modelo de Caja, Flexbox y Responsive Design

a) Modelo de Caja (Box Model):

Cada elemento HTML es como una caja que tiene:

Contenido: El texto o imagen dentro

Padding: Espacio interno entre el contenido y el borde

Border: El borde de la caja

Margin: Espacio externo que separa la caja de otros elementos

El tamaño final se calcula sumando todas estas partes.

b) Diferencia entre Flexbox y Grid:

Flexbox: Mejor para organizar elementos en una sola dirección (fila o columna). Ideal para barras de navegación o centrar elementos.

Grid: Mejor para organizar elementos en 2 dimensiones (filas y columnas). Ideal para diseños de página completa como un periódico.

c) Media Queries:

Son reglas que se escriben en CSS(@media (max-width: 768px) etc) que aplican estilos diferentes según el tamaño de pantalla gracias al uso de estas reglas las páginas se pueden ver bien celulares, computadoras y otros dispositivos. Otra técnica es usar unidades relativas como % .

3. JavaScript: DOM y Eventos

a) El DOM:

El DOM es como un mapa que JavaScript usa para encontrar y modificar elementos de la página web. Convierte el HTML en objetos que JavaScript puede manipular., gracias a esta representación javascript puede crear nuevos elementos, eliminar elementos existentes,modificar atributos y estilos, cambiar contenido, escuchar y responder a eventos se usa con instrucciones como getElementById(), querySelector(), innerHTML, etc.

b) Código para seleccionar elementos:

i. Seleccionar elemento con id="miImagen"

const imagen = document.getElementById(‘miImagen’);

ii. Seleccionar todos los elementos con class="elementoDestacado"

const elementos = document.getElementsByClassName(‘elementoDestacado’);

c) Eventos en JavaScript:

Un evento es una acción o suceso, es algo que pasa en la página (como hacer clic por ejemplo entre otros eventos). Para manejarlo en javascript hacemos lo siguiente por ejemplo un boton:

Usamos addEventListener() para "escuchar" el evento en el botón luego escribimos Escribir una función que se ejecute cuando pase el evento(por ejemplo el click) quedando de la siguiente forma

boton.addEventListener('click',haceAlgo);

function haceAlgo() {

aquí va el código que se ejecuta al hacer clic

});